

The MITRATECH logo consists of the word "MITRATECH" in white, uppercase, sans-serif font, centered within a solid teal rectangular background.

MITRATECH

## **TeamConnect® 5.0**

© 2017 Mitrtech

User, Admin and Designer Guide

## **TeamConnect® 5.0 User, Admin and Designer Guide**

**Document ID: tc\_50\_uad\_1, published on 5/10/2017**

Copyright © 2017, Mitrates Holdings, Inc. All rights reserved.

### Disclaimer of Warranty

Mitrates Holdings, Inc. (Mitrates) makes no representations or warranties, either expressed or implied, by or with respect to anything in this document, and shall not be liable for any implied warranties of merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

Mitrates reserves the right to not support non-standard or non-default functionality and extended functionality available in third-party software, unless specifically documented as supported or certified in the Mitrates product documentation. For further information regarding third-party non-standard or non-default functionality, please contact Mitrates Support.

This document, along with the software that it describes, is furnished under license and may be used or copied only in accordance with the terms of such license. The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as commitment by Mitrates.

The following document is for the TeamConnect™ Collaborati Spend Management 5.0 release only. Though every effort was made to ensure that the information in this document is correct and reliable, Mitrates does not assume any liability for any errors encountered in this document.

If you need support for TeamConnect™ Collaborati Spend Management 5.0, please contact the Mitrates support team by sending an email to: [support@mitrates.com](mailto:support@mitrates.com). For more information about Mitrates, visit our web site: <http://www.mitrates.com>.

"Mitrates", TeamConnect™ Enterprise, TeamConnect™ Legal, TeamConnect™ Legal Matter Management, Collaborati®, TeamConnect™ Collaborati Spend Management®, TeamConnect™ Deadlines, TeamConnect™ AP Link, TeamConnect™ Office Suite, TeamConnect™ Legal Reports, TeamConnect™ SOP Manager, TeamConnect™ Financial Management, TeamConnect™ Screen Designer, and TeamConnect™ Upgrade Toolkit are trademarks and products of Mitrates Holdings, Inc. All other products or services mentioned in this book are the trademarks or service marks of their respective companies or organizations.

### GOVERNMENT RIGHTS LEGEND:

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the applicable Mitrates license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013 (Feb 2012) FAR 12.212(a) (1995), FAR 52.227-19 (Dec 2007), or FAR 52.227-14, as applicable.

### CONTACT US:

Mitrates Holdings, Inc.  
5001 Plaza on the Lake Suite 111, Austin, TX 78746  
Phone: (512) 382-7322

NOTE: Throughout Mitrates product publications, in addition to using full product names where necessary, we also use familiar and shorter terms to increase your ease of reading. You may find the following aliases for our product names:

TeamConnect for TeamConnect Enterprise  
Matter Management for TeamConnect Legal Matter Management  
TeamConnect Legal for TeamConnect Legal Matter Management  
CSM for TeamConnect Collaborati Spend Management  
Collaborati Spend Management for TeamConnect Collaborati Spend Management  
SOP or SOP Manager for TeamConnect SOP Manager  
Legal Hold for TeamConnect Legal Hold  
Legal Reports for TeamConnect Legal Reports  
Deadlines for TeamConnect Deadlines  
AP Link for TeamConnect AP Link  
Office Suite for TeamConnect Office Suite  
Financial Management for TeamConnect Financial Management  
Screen Designer for TeamConnect Screen Designer  
Upgrade Toolkit for TeamConnect Upgrade Toolkit

# Acknowledgements

This product includes software developed by the following organizations:

Apache Software Foundation (<http://www.apache.org/>)

OpenSymphony Group (<http://www.opensymphony.com/>).

The license agreements for these and other supplemental software packages can be found in your installation media in subfolder Supplemental\_Software\_Licenses. That subfolder also contains Open Source Components.pdf, which lists the locations, license types, and specific versions of components that are available on the web.

# Table of Contents

<b>Part I Enterprise User Help</b>	<b>36</b>
<b>1 Getting Started</b>	<b>36</b>
<b>Authenticating Users</b>	<b>36</b>
<b>Logging In</b>	<b>37</b>
<b>Logging Out</b>	<b>41</b>
<b>User Interface</b>	<b>42</b>
Getting Hints and Help	43
Web Browser Buttons and Commands	44
<b>Personalizing Your Settings</b>	<b>44</b>
Changing Preferences	45
Selecting Region, Date, and Time Formats	45
Defining Search Result Settings	48
Defining the Time Interval for the Appointment Calendar	49
Enabling Accessibility Mode	49
Changing Password	49
Setting Up Email Notifications	50
Choosing Other Setting Options	52
<b>Home Pages and Portal Panes</b>	<b>52</b>
Working with Home Pages	53
Information in Portal Panes	55
Running Business Objects Reports	56
Adding Portal Panes to Home Pages	57
Modifying Home Page Settings	58
Page Title	58
Customizing Portal Pane Layout	58
Using Search Views in Portal Panes	59
Editing Portal Pane Custom Content	60
About Synchronizing Home Pages and Portal Panes	61
<b>2 Records</b>	<b>61</b>
<b>Displaying Types of Records</b>	<b>64</b>
<b>Viewing Collections of Records</b>	<b>65</b>
Managing Collections	67
<b>Searching for Specific Records</b>	<b>68</b>
Global Search	69
Modifying a Collections Search	69
Using Custom Search	70
Specifying Search Criteria	70
Operators	72
Adding Additional Fields	75
Specifying Results Columns	76
Executing the Search	77
Saving the Search	77
Using Search Modules	78
Exporting a Search Results List	78
<b>Sorting Information</b>	<b>79</b>
Single-column Sort	80
Multi-column Sort	80



<b>Using a Record in View Mode .....</b>	<b>82</b>
<b>Creating a Record .....</b>	<b>84</b>
Wizards .....	85
Editing a Record.....	86
Editing Multiple Records.....	86
Record Locking During Editing.....	88
Working w ith Different Types of Fields.....	89
check-box .....	89
Date Field .....	89
Time Zone Independent Date Fields .....	90
List Fields .....	91
Drop-dow n List.....	91
List Box .....	92
Memo Text.....	93
Number .....	93
Option Buttons.....	94
Text .....	94
Working w ith Batch Entries.....	95
Primary Entries .....	95
Deleting an Entry .....	96
Editing Batch Entries.....	96
Setting Category Details.....	97
Canceling Your Edits.....	99
Saving an Edited Record.....	99
Handling Unsaved Records.....	100
Copying a Record.....	101
Reassigning Work.....	104
Importing Data to Create Records .....	105
Creating a Data File.....	106
Importing Data.....	108
<b>Deleting Records .....</b>	<b>114</b>
<b>Printing Records .....</b>	<b>117</b>
<b>Working with Record Security .....</b>	<b>118</b>
View ing Record Security.....	120
Editing Record Security.....	121
Public and Private Record Security.....	122
<b>Working with Record History .....</b>	<b>122</b>
Opening History Pages.....	123
View ing General History Information.....	124
Setting History Categories.....	125
Creating Record History Entries.....	125
<b>Working with Related Records .....</b>	<b>126</b>
<b>Working with Record Documents .....</b>	<b>127</b>
<b>Viewing Workflow for a Record .....</b>	<b>127</b>
<b>3 Contacts and Address Books .....</b>	<b>127</b>
<b>Contacts .....</b>	<b>127</b>
Types of Contact View s.....	129
Creating a Contact.....	130
Duplicate Contacts .....	135
Setting Contacts Primary Information.....	136
Setting Contact Details .....	136
Adding and Removing Contacts Categories.....	138
View ing the Address Books for a Specific User.....	139
Setting Contact Rates.....	139

Relating Contacts .....	143
Setting Contact Skills.....	144
Selecting Contact Territories.....	144
Viewing Company Employees.....	145
Viewing Contact Involvement.....	146
<b>Address Books .....</b>	<b>147</b>
Subscribing to Address Books .....	148
Viewing Contacts in an Address Book.....	149
Creating an Address Book.....	149
Adding Contacts to an Address Book.....	151
Removing Contacts from an Address Book.....	152
Making an Address Book Public or Private.....	152
<b>4 Projects .....</b>	<b>153</b>
<b>Creating a New Project .....</b>	<b>154</b>
<b>Creating Different Types of Projects .....</b>	<b>157</b>
Child Projects.....	157
Embedded Projects.....	157
<b>Editing a Project .....</b>	<b>158</b>
<b>Adding or Removing Project Categories .....</b>	<b>159</b>
<b>Changing Project Phases .....</b>	<b>160</b>
<b>Assigning Contacts to Projects .....</b>	<b>160</b>
Viewing Users Assigned to a Project.....	161
Adding Assignees to Projects.....	161
Changing the Primary Assignee for a Project.....	163
Unassigning or Reassigning Assignees .....	163
Removing Assignees from a Project.....	164
<b>Creating Relationships between Projects .....</b>	<b>164</b>
<b>Managing Involved Parties for Projects .....</b>	<b>165</b>
Adding or Editing Involved Parties.....	165
Changing the Status of an Involved Party.....	167
Changing an Involved Party Role.....	167
Relating Involved Parties .....	168
Adding or Editing Involved Party Accounts.....	168
<b>Managing Milestones .....</b>	<b>169</b>
Viewing Milestones.....	169
Creating or Editing Milestones .....	170
Setting Milestone Categories.....	171
Managing Appointments for a Milestone.....	171
Managing Milestone Tasks.....	172
<b>Adding Tasks to Projects .....</b>	<b>172</b>
<b>Adding Appointments to Projects .....</b>	<b>172</b>
<b>Adding Expenses to Projects .....</b>	<b>173</b>
<b>Adding Accounts to Projects .....</b>	<b>173</b>
<b>5 Reports .....</b>	<b>174</b>
<b>Viewing Reports .....</b>	<b>175</b>
Dashboards and Portal Panes.....	176
Using Report Folders.....	177
<b>Running Reports .....</b>	<b>178</b>
Running Drill-down Reports .....	179
Viewing a Cached Report.....	180
Changing the Appearance of a Report.....	180
Types of Report Formats .....	182
Exporting Reports.....	186
<b>Building Reports .....</b>	<b>186</b>

General Properties .....	186
Data Series.....	187
Entering New Data Series Information.....	190
Specifying Data for a Data Series.....	191
Filtering the Data.....	193
Determining the Data that Displays.....	196
Combining Multiple Data Series Using Cross-Object Reporting.....	198
Join Conditions.....	199
Results Display Fields.....	199
Sample for Combining Data Series.....	200
Best Practices for Combining Data Series.....	201
Chart .....	202
Scheduler.....	208
Edit Schedule Information.....	208
Viewing a Schedule.....	209
Activating or Deactivating a Schedule.....	209
Running a Scheduled Report Now .....	210
Deleting a Schedule.....	210
Changing an Existing Schedule's Run Time.....	210
Security .....	210
<b>Localizing Reports .....</b>	<b>210</b>
<b>6 Appointments Calendar and Tasks .....</b>	<b>212</b>
<b>Appointments Calendar .....</b>	<b>212</b>
Viewing the Appointments Calendar.....	213
Opening an Appointment.....	216
Appointments Created Using Outlook.....	216
Creating or Editing Appointments.....	217
Creating or Editing an Appointment Manually.....	217
Using the Appointments Scheduler.....	219
Adding and Removing Appointment Categories.....	221
Adding Appointment Attendees.....	222
Adding Appointment Resources .....	223
Viewing or Creating Appointment History.....	223
<b>Tasks .....</b>	<b>223</b>
Using the Task General Page.....	225
Viewing Tasks.....	225
Creating or Editing Tasks.....	226
Creating a Task Manually .....	227
Assigning or Reassigning Tasks to Users.....	231
Posting Tasks.....	232
Voiding Tasks.....	233
Viewing Task Transactions.....	233
Viewing or Creating Task History.....	234
Using the Task Entry Tool.....	234
Creating Multiple Tasks.....	234
Editing Multiple Tasks.....	236
Posting Multiple Tasks .....	237
Deleting Multiple Tasks.....	237
<b>7 Finance .....</b>	<b>238</b>
<b>Accounts .....</b>	<b>239</b>
Viewing Accounts.....	240
Creating or Editing an Account.....	240
Creating or Editing an Account Manually.....	241
Creating a Shadow Invoice Account.....	246

Adding or Removing Categories from an Account.....	247
Parent and Child Accounts.....	248
Assigning Parent Accounts.....	248
Viewing or Creating Child Accounts.....	249
Activating or Deactivating Accounts.....	250
Adding and Subtracting Funds.....	250
Transferring Funds Between Accounts.....	251
Setting Transaction Posting Criteria.....	253
Viewing Transactions History for an Account.....	264
Viewing or Creating Account History.....	265
<b>Expenses .....</b>	<b>265</b>
Viewing Expenses.....	267
Creating or Editing an Expense.....	267
Creating or Editing an Expense Manually.....	268
Preparing an Expense for Posting.....	270
Posting an Expense.....	274
Voiding Expenses.....	274
Viewing Expense Transactions.....	274
Viewing or Creating Expense History.....	275
Working with the Expense Entry Tool.....	275
Entering Expenses.....	275
Copying Expenses.....	277
Editing Expenses.....	278
Posting Expenses.....	278
Deleting Expenses.....	278
Updating Expense Unit Prices.....	279
<b>Invoices .....</b>	<b>281</b>
Viewing Invoices.....	284
Searching for Invoice Line Items.....	286
Appeals Invoices.....	288
Differences Among Invoices.....	290
Invoice Duplicates.....	292
Invoice Versions.....	292
Viewing Previous Invoice Versions.....	292
Viewing Invoices Submitted in Different Currencies.....	293
Invoice Procedures (without Workflow).....	294
Invoice Procedures (with Workflow).....	294
Creating an Invoice Manually.....	295
Timekeeper Rates.....	301
Adding Invoice Line Items.....	301
Copying an Invoice Line Item.....	305
Editing Invoices.....	305
Changing the Displayed Currency on an Invoice.....	306
Editing Invoice Line Items.....	307
Deleting Invoice Line Items.....	308
Preparing an Invoice for Posting.....	309
Posting an Invoice.....	313
Adjusting an Invoice Summary.....	315
Adjusting Invoice Line Items.....	322
Line Item General.....	336
Approving Invoices.....	336
Rejecting Invoices.....	337
Sending an Invoice for Review.....	338
Reviewing an Invoice.....	340

Voiding an Invoice.....	340
Using the Invoice Summary Breakdowns Page.....	341
Adjusting Net Amounts on the Summary Breakdowns Page.....	344
Viewing and Creating History for an Invoice.....	345
Viewing Invoice Summary Adjustment History.....	345
Viewing Line Item Adjustment History.....	345
Viewing Invoice Transactions History.....	346
Viewing Invoice Workflow.....	346
<b>Budget Requests.....</b>	<b>347</b>
Creating a Budget Request.....	349
Submitting a New Budget Request.....	350
Budget Requests Created by Vendors.....	350
Editing a Budget Request.....	350
Adjusting Budget Requests.....	350
Canceling a Budget Request.....	352
Deleting a Budget Request.....	352
Handling Budget Request Workflow.....	353
<b>Updating Exchange Rates.....</b>	<b>353</b>
<b>8 Documents.....</b>	<b>355</b>
<b>Searching for Existing Documents.....</b>	<b>358</b>
<b>Documents and Global Search.....</b>	<b>359</b>
<b>Viewing Documents.....</b>	<b>360</b>
Opening Documents.....	362
Sharing Documents using Email.....	363
<b>Using Documents Collections.....</b>	<b>364</b>
<b>Managing Documents.....</b>	<b>364</b>
Adding a Document to TeamConnect.....	365
Editing Documents.....	369
Checking Out a Document.....	370
Checking In a Document.....	370
Canceling a Document Check Out.....	371
Creating a Document Folder.....	372
Exporting a List of Documents in a Folder.....	372
Copying Documents.....	373
Moving Documents.....	376
Creating a Document Shortcut.....	377
Creating a Link to a Web Page.....	378
Downloading Documents.....	379
Generating Documents Using TeamConnect Data.....	380
Using Windows Explorer with Documents (WebDAV).....	382
<b>Viewing or Editing General Information of a Document.....</b>	<b>383</b>
<b>Viewing Document Versions.....</b>	<b>385</b>
Reverting to a Previous Version.....	385
<b>Adding and Removing Document Categories.....</b>	<b>386</b>
Viewing or Creating Documents History.....	386
<b>Printing Documents.....</b>	<b>386</b>
<b>Deleting Documents and Document Folders.....</b>	<b>387</b>
<b>Updating Security Rights of Document Folders.....</b>	<b>388</b>
<b>9 Workflow Processes.....</b>	<b>388</b>
<b>Using the Approval Schedule and Approval History.....</b>	<b>389</b>
<b>Setting Up Email Approvals.....</b>	<b>392</b>
<b>Using Approval Process Email Notifications.....</b>	<b>394</b>
<b>Submitting Requests for Actions that Require Approval.....</b>	<b>395</b>
Viewing Your Active Requests.....	396

Viewing Your Completed Requests.....	398
Canceling Approval Requests.....	399
<b>Approving, Rejecting, and Reviewing Requests .....</b>	<b>400</b>
Approving or Rejecting a Request.....	401
Approving or Rejecting from My Approvals.....	403
Approving or Rejecting Within Records .....	406
Approving or Rejecting through Email.....	408
Sending Requests for Review .....	409
Reviewing a Request.....	411
Reviewing from My Approvals.....	411
Reviewing Within Records.....	413
<b>Making Workflow Decisions .....</b>	<b>413</b>
Monitoring Current Requests.....	414
Reviewing Completed Requests .....	419
Returning Requests to Requestors.....	420
Reassigning Individual Requests.....	421
Reassigning All Requests for a User.....	422
Returning Requests with Failed Actions to Workflow .....	425
Returning Requests With Errors to Workflow .....	427
Moving Requests Forward in A Process.....	427
<b>Using Hold Conditions .....</b>	<b>428</b>
<b>10 Accessibility .....</b>	<b>429</b>
<b>Enabling Accessibility Mode .....</b>	<b>429</b>
<b>Using Assistive Technology .....</b>	<b>430</b>
Using JAWS® with TeamConnect.....	430
<b>Using Keyboard Shortcuts .....</b>	<b>431</b>
Commonly Used Access Keys.....	431
Using Keyboard Features .....	433
TeamConnect-Specific Access Keys.....	434
<b>11 TeamConnect Mobile .....</b>	<b>436</b>
<b>Invoices Pending for Approval .....</b>	<b>437</b>
<b>Viewing and Adjusting Invoice Details .....</b>	<b>438</b>
<b>Contacting a Vendor or Approver .....</b>	<b>440</b>
<b>12 Troubleshooting and Frequently Asked Questions .....</b>	<b>440</b>
<b>Frequently Asked Questions .....</b>	<b>441</b>
Managing Records .....	441
Contacts/Address Books.....	443
Reports .....	443
Other Questions.....	445
<b>Troubleshooting .....</b>	<b>445</b>
<b>13 Glossary .....</b>	<b>446</b>
<b>Part II Enterprise Administrator Help .....</b>	<b>457</b>
<b>1 Getting Started .....</b>	<b>457</b>
<b>Admin Area User Interface .....</b>	<b>458</b>
Group Rights Pages.....	458
Admin Settings General Page.....	459
Save Options.....	460
<b>Terminology .....</b>	<b>461</b>
<b>2 Account Administration .....</b>	<b>461</b>
<b>Planning and Best Practices .....</b>	<b>462</b>
<b>Understanding Rights .....</b>	<b>463</b>

Rights for End-Users.....	464
Rights for Administrators .....	465
Rights for Solution Developers.....	465
Rights for System Record Types.....	468
Prerequisites for Rights to System Record Types.....	470
Rights Impacts on Related Records .....	470
Rights for Categories and Custom Fields.....	471
Prerequisites for Rights to All Categories and Custom Fields.....	473
Prerequisites for Rights to Single Categories and Custom Fields .....	474
Rights for Custom Record Types.....	476
Rights for TeamConnect Features or Areas.....	477
<b>Groups .....</b>	<b>477</b>
Working with Groups .....	481
Searching for Groups .....	481
Viewing Groups.....	482
Creating or Editing Groups .....	482
Creating a Group Using a Template.....	484
Changing Group Members.....	484
Assigning or Viewing Group System Rights.....	485
Assigning or Viewing Global User Rights for a Group.....	487
Results from Assigning Global User Rights .....	489
Assigning Global Admin Rights to a Group.....	495
Results from Assigning Global Admin Rights.....	496
Assigning Global Setup Rights to a Group.....	497
Results from Assigning Global Setup Rights.....	498
Example for Assigning Rights to View Contacts.....	502
Example for Assigning Rights to Create Contacts .....	503
Example for Assigning Rights to Edit Contacts .....	503
Example for Assigning Rights to Delete Contacts.....	504
Example for Assigning All Rights to Contacts Pages and Fields.....	504
Examples for Assigning Projects Rights .....	505
Assigning All Rights to Dispute Pages and Fields.....	506
Assigning Group Category and Custom Field Rights .....	507
Example for Assigning Rights to Contact Categories and Fields .....	509
Hierarchy Overview of Group Rights .....	511
Assigning Group Tool Rights.....	512
Changing Group Default Object Views.....	514
Viewing or Creating Group History.....	515
Deleting a Group.....	515
<b>Users .....</b>	<b>516</b>
Working with Users.....	516
Searching for Users.....	517
Viewing Users .....	518
Creating or Editing Users.....	519
Creating a User Using a Template.....	523
Example for Creating a Limited-Privilege User.....	524
Activating and Inactivating a User.....	524
Prompting Users to Change Their Passwords.....	525
Editing or Viewing a User's Preferences.....	525
Unlocking User Accounts.....	525
Changing a User's Group Membership.....	526
Assigning or Viewing User System Rights.....	527
Assigning or Viewing Global User Rights for a User.....	531
Assigning or Viewing Global Admin Rights for a User .....	532

Assigning or Viewing Global Setup Rights for a User.....	533
Assigning or Viewing User Category Rights .....	534
Assigning or Viewing User Tool Rights .....	536
Viewing or Creating User History .....	537
Deleting a User.....	537
Using the User General Page.....	538
<b>3 System Administration .....</b>	<b>538</b>
<b>Working with Admin Settings .....</b>	<b>538</b>
General Admin Settings.....	539
Record Locking.....	540
BusinessObjects Web Intelligence Single Sign-on.....	540
Setting Up General Settings .....	541
User Interface Settings .....	544
Custom Logo.....	545
Color Profile Administration.....	545
Moving Action Buttons .....	551
Setting User Selection Options.....	551
Item Selection.....	552
Setting up the User Interface.....	552
Regions and Localization .....	555
Region Settings .....	555
Setting Regional Settings .....	555
Working with Locales.....	558
Localizing Data in TeamConnect.....	563
Exporting Data for Translation.....	563
Importing Translated Data.....	566
Calendar Settings.....	567
Documents Settings .....	567
Document Version Control.....	568
Network File Systems for Documents (WebDAV).....	568
Managing Documents.....	569
Search Results Settings.....	571
Maximum Search Results.....	572
Setting Up Search Results .....	573
Reporting Settings.....	576
Defining Reporting Settings.....	576
Security Settings.....	577
Defining the User Password Policy.....	577
Configuring User Authentication.....	582
Email Settings .....	584
Email Notification.....	585
TeamConnect IMAP Server .....	586
Setting Up SSL for the IMAP Server.....	587
Configuring the IMAP Server.....	588
Starting and Stopping the IMAP Server.....	589
Email Rights.....	590
Email Clients .....	590
Setting Up Email.....	590
Notification Settings.....	592
Customizing System Notifications .....	595
Invoices Settings.....	596
Setting Decimal Places for Invoice Line Items .....	598
Approval Settings.....	598
Maintenance Settings.....	600



Connections Settings .....	601
Setting Up Connections .....	602
Admin Settings History .....	603
Recording a History for Admin Settings .....	603
About TeamConnect .....	604
Viewing TeamConnect Versions and License Information .....	604
Viewing Information About Available Updates .....	605
Choosing Your Support Options .....	606
Viewing Information About Your Product License .....	606
Licensing Schemes .....	606
License Style .....	607
Uploading a License File .....	607
<b>TeamConnect Logs .....</b>	<b>608</b>
Default Logs .....	608
Working with Logs .....	609
Viewing Logs .....	609
Things to Remember .....	610
Clearing Logs .....	611
Changing System Logging Levels .....	611
Turning Audit Loggers On or Off .....	613
<b>TeamConnect Administrator Tools .....</b>	<b>614</b>
Scheduled Actions Monitor .....	615
Accessing the Scheduled Actions Monitor .....	615
Tasks Available from the Scheduled Actions Monitor .....	616
Create a New Scheduled Action .....	616
Actions Against the Table .....	617
Deactivate Scheduled Actions .....	618
Activate Scheduled Actions .....	618
Delete Scheduled Actions .....	619
Run a Scheduled Action .....	619
Things to Remember .....	619
<b>4 Authentication .....</b>	<b>619</b>
<b>Login URLs .....</b>	<b>619</b>
<b>NTLM Authentication Plug-in .....</b>	<b>620</b>
NTLM Prerequisites .....	621
Configuring Internet Information Services .....	622
Disabling Anonymous Access .....	622
Editing the NTLM Configuration File .....	623
Installing the NTLM Configuration File .....	623
Enabling NTLM Authentication for TeamConnect Users .....	624
<b>LDAP Authentication Plug-in .....</b>	<b>625</b>
LDAP Authentication Prerequisites .....	625
Creating Custom LDAP Authentication Error Pages .....	626
Defining LDAP Parameters .....	626
Installing LDAP Plug-in Files .....	630
LDAP Authentication Sample Files .....	631
Sample Custom Log-in Page .....	631
Sample Custom Error Page .....	633
Sample Restore Session Page .....	633
Sample LDAP Configuration File .....	634
<b>Custom Authentication Plug-ins .....</b>	<b>635</b>
Customizing Message Screens .....	636
<b>Enabling a Default Authentication Method .....</b>	<b>636</b>
<b>Overriding Authentication for Individual Users .....</b>	<b>636</b>

<b>5</b>	<b>Troubleshooting and Frequently Asked Questions .....</b>	<b>637</b>
	<b>Frequently Asked Questions .....</b>	<b>638</b>
	Users and Groups.....	638
	Address Books.....	640
	Budgets .....	641
	Admin Settings.....	641
	Licensed Seats.....	641
	<b>Troubleshooting .....</b>	<b>641</b>
<b>6</b>	<b>Rights for End-users .....</b>	<b>643</b>
	<b>System Record Rights .....</b>	<b>645</b>
	Account Rights.....	645
	Basic Account Rights.....	645
	Additional Account Rights.....	646
	Appointment Rights.....	648
	Basic Appointment Rights.....	648
	Additional Appointment Rights.....	650
	Contact and Address Book Rights.....	652
	Basic Contact and Address Book Rights.....	652
	Additional Contact Rights.....	654
	Document Rights.....	657
	Basic Document Rights.....	657
	Additional Document Rights.....	658
	Expense Rights.....	664
	Basic Expense Rights.....	664
	Additional Expense Rights.....	665
	History Rights.....	667
	Basic History Rights.....	667
	Additional History Rights.....	668
	Invoice Rights.....	670
	Basic Invoice Rights.....	670
	Additional Invoice Rights.....	671
	Report Rights.....	675
	Task Rights.....	676
	Basic Task Rights.....	676
	Additional Task Rights.....	677
	<b>TeamConnect Features and Areas .....</b>	<b>679</b>
	Preferences Rights.....	680
	Release Edit Locks Right.....	680
	Scheduled Actions Rights.....	680
	<b>Custom Record Rights .....</b>	<b>681</b>
	Project/Matter Rights.....	681
	Basic Project/Matter Rights.....	681
	Additional Project/Matter Rights.....	682
	Project/Matter Involved Rights.....	685
	Basic Involved Rights.....	686
	Additional Involved Rights.....	687
	Project/Matter Milestones Rights.....	689
	Basic Project/Matter Milestone Rights.....	690
	Additional Milestones Rights.....	690
<b>7</b>	<b>Rights for Administrators .....</b>	<b>692</b>
	<b>Rights for Group Records .....</b>	<b>692</b>
	Basic Group Rights.....	693
	Additional Group Rights.....	693

<b>Rights for User Records</b> .....	<b>694</b>
Basic User Rights.....	695
Additional User Rights.....	695
<b>Rights for Admin Settings and Logs</b> .....	<b>696</b>
<b>8 Rights for Solution Developers</b> .....	<b>697</b>
<b>Administer Custom Tools Rights</b> .....	<b>698</b>
<b>Global Navigation Toolbar Rights</b> .....	<b>699</b>
<b>Customize Home Pages Rights</b> .....	<b>699</b>
<b>Lookup Tables Rights</b> .....	<b>700</b>
<b>Custom Lookup Tables Rights</b> .....	<b>701</b>
<b>Multi-currency Lookup Table Rights</b> .....	<b>701</b>
<b>Miscellaneous Settings Rights</b> .....	<b>702</b>
<b>Object Definitions Rights</b> .....	<b>704</b>
<b>Customize Portal Panes Rights</b> .....	<b>712</b>
<b>Routes Rights</b> .....	<b>712</b>
<b>System Settings and Appenders Rights</b> .....	<b>714</b>
<b>9 Documents and Folders</b> .....	<b>715</b>
<b>Directory Structure</b> .....	<b>715</b>
Top Level Folder.....	716
Attachments Folder.....	717
HTTPRoot Folder.....	717
System Folder.....	718
Users Folder.....	722
<b>Automatically Created Folders</b> .....	<b>722</b>
<b>10 Glossary</b> .....	<b>724</b>

## Part III TeamConnect Setup and Development

737

<b>1 Enterprise Customization Help</b> .....	<b>737</b>
<b>Introduction to Customization</b> .....	<b>738</b>
The Meaning of Design.....	738
Preparing User and Group Accounts.....	738
TeamConnect Designer User Interface.....	739
User Interface Overview .....	739
Designer Menu Bar.....	740
Backward Compatibility Settings.....	742
Customization Sequence.....	744
Creating Custom Messages .....	747
Designing for Data Warehouse and Microsoft SQL Server.....	748
<b>Working with System Settings</b> .....	<b>749</b>
Accessing System Settings.....	749
Rule Execution Properties .....	749
Miscellaneous Settings.....	750
Setting up Invoices and Line Items.....	752
Defining Non-US Tax Categories.....	752
Setting up Invoice Duplicate Checking.....	753
Setting up Invoice Line Items.....	753
Setting up General Invoice Line Items Behavior.....	753
Setting up Invoice Line Items Adjustment Behavior .....	755
Using Duplicate Contact Manager .....	758
Using Contact Sweeper.....	759
Prerequisites.....	759

Generate Report (Step 1 of 5).....	760
Edit the Report (Step 2 of 5).....	762
Upload Report (Step 3 of 5).....	764
Fix References (Step 4 of 5).....	764
What Kinds of References are Fixed?.....	765
Logging During the Fix References Process.....	766
Delete Duplicates (Step 5 of 5).....	766
Preventing Duplicate Contacts.....	767
Activating Duplicate Contact Checking.....	767
Example 1 of Duplicate Contact Checking.....	775
Example 2 of Duplicate Contact Checking.....	777
Using XML Worksheet Tool.....	780
Creating Custom Tools.....	781
Uploading Custom Tool Files.....	782
Configuring Custom Tools.....	782
Appenders.....	783
Default System and Audit Appenders.....	784
Viewing Logs Generated by File Appenders.....	784
Clearing Logs Generated by File Appenders.....	785
Rolling Over Logs Generated by File Appenders.....	786
File Appenders.....	786
File Appender Layouts.....	789
Pattern Layout.....	789
HTML Layout.....	792
XML Layout.....	792
SMTP Appenders.....	793
Socket Appenders.....	794
Defining System Appenders.....	796
Defining Audit Appenders.....	797
Loggers.....	799
System Loggers.....	799
Audit Loggers.....	802
<b>About Objects.....</b>	<b>803</b>
System Objects.....	803
Custom Objects.....	805
Unique Custom Object Types.....	805
Related Objects.....	805
Relationship Types.....	806
Parent-Child Relations between Custom Objects.....	808
Contact-Centric Custom Objects.....	809
Sub-Objects.....	811
Embedded Custom Objects.....	812
<b>Creating and Defining Objects.....</b>	<b>813</b>
About Object Definitions.....	814
Viewing Object Definitions.....	817
Configuring System Objects.....	820
General System Object Information.....	820
Default Object Icons.....	821
Creating Custom Objects.....	822
General Custom Object Information.....	824
Creating Custom Objects.....	827
Unique Identifiers for Custom Object Records.....	829
Creating Auto Numbering Patterns.....	831
Project Link Display Format.....	833

Creating Unique Identifier Patterns from Object Attributes .....	833
Naming Patterns for Custom Object Records .....	834
Adding Object Attributes .....	837
Attribute Format Fields .....	838
Adding Literals .....	840
Creating Auto Naming Patterns .....	842
Creating Phases in Custom Objects .....	842
Adding Phases .....	844
Editing Phases .....	844
Deleting Phases .....	845
Defining Phase Transitions for Custom Objects .....	846
Creating Phase Transitions .....	849
Editing Phase Transitions .....	849
Deleting Phase Transitions .....	850
Defining Assignee Roles for Custom Objects .....	851
Adding Assignee Roles .....	851
Editing Assignee Roles .....	853
Deleting Assignee Roles .....	854
Customizing Notifications for Object Definitions .....	855
List Displays in Embedded Objects .....	855
Displaying Embedded Records .....	856
Creating Embedded Objects .....	857
Defining Display for Embedded Object Records .....	860
Converting Embedded Objects to Child Objects .....	861
Deleting a Custom Object Definition .....	863
<b>Using Categories and Lookup Tables .....</b>	<b>863</b>
Using Categories .....	864
Viewing Categories .....	865
Locations of Categories in End-User Interface .....	867
Category Tree Positions .....	868
System Object Categories .....	869
Custom Object Categories .....	870
Adding Object Categories .....	871
Editing Object Categories .....	871
Inactivating Object Categories .....	871
Activating Object Categories .....	873
Deleting Object Categories .....	874
Using Lookup Tables .....	875
Types of Lookup Tables .....	875
System Lookup Tables .....	876
Viewing System Lookup Tables .....	877
Lookup Table Structure .....	879
Contact Address Lookup Tables .....	880
Adding Lookup Table Items .....	881
Editing System Lookup Tables .....	882
Inactivating Lookup Table Items .....	882
Activating Lookup Table Items .....	884
Deleting Lookup Table Items .....	885
Custom Lookup Tables .....	885
Viewing Custom Lookup Tables .....	886
Adding Custom Lookup Tables .....	887
Deleting Custom Lookup Tables .....	888
Adding Items to Custom Lookup Tables .....	888
Editing Custom Lookup Table Items .....	889

Inactivating Custom Lookup Table Items .....	889
Activating Custom Lookup Table Items .....	891
Deleting Custom Lookup Table Items.....	892
<b>Creating Custom Pages .....</b>	<b>892</b>
Custom and System View s.....	893
Process Overview .....	894
Custom Fields.....	894
Where To Use Custom Fields.....	896
Custom Fields Tab in Object Definitions .....	897
Custom Field Name Requirements.....	899
Data Warehouse Requirements for Custom Fields .....	899
Required Fields.....	902
Field Types.....	903
List Field Type.....	906
Involved Field Type.....	906
Custom Object Field Type.....	907
Search View s for Custom Object Fields .....	910
Qualifiers for Custom Object Fields .....	911
Qualifier Syntax and Data Types.....	913
Creating Qualifiers Using Object Navigator.....	916
Creating Custom Fields.....	920
Blocks .....	922
Creating Blocks .....	923
Creating Block XML Files.....	926
Block Tags .....	927
Tags to Embed Reports in Blocks .....	932
XML File Sample.....	935
Uploading Custom Block Files.....	938
Defining Blocks.....	938
Troubleshooting Custom Blocks.....	942
Object View s .....	943
Defining Object View s .....	947
Creating Object View s.....	948
Assigning Object View s.....	950
Troubleshooting Object View s.....	950
Clearing Custom Blocks Cache.....	951
Customizing Invoice Line Item View s.....	952
<b>Creating Search Views .....</b>	<b>953</b>
Designing Search View s .....	954
Accessing the Search View Screen.....	955
Creating Search View s.....	958
Creating Search View s for Home Pages.....	959
Custom Hover Popups .....	961
Specifying Search View s Used by Global Search.....	962
Creating Searches for Approvals, Requests, and Workflow Processes.....	965
Search View Requirements for Full-Text Searching.....	966
Adding Document Content Searching to Search View s.....	966
Adding Field Content Searching to Search View s .....	967
Defining General Search View Information.....	968
Subscribing and Unsubscribing for Collections.....	972
About Auto Search.....	973
About Search View s in Search Modules .....	974
Enabling IMAP Search for Custom Object Search View s.....	975
Projects and Third-party Calendars.....	977

Custom Search Templates.....	978
Search View Results Display.....	978
Displayed Records Section Examples.....	982
Defining Object Links.....	983
Creating Object Links.....	984
Creating Results Display Columns.....	984
Search View Filter Display.....	985
Visible and Invisible Qualifiers.....	985
Qualifier Conditions.....	986
Filter Tab Example.....	989
Search Module Actions in Contact Filter Display.....	990
Creating Search View Qualifiers.....	991
Creating Section Titles.....	992
Tips for Working with Qualifiers.....	993
Troubleshooting Search Views.....	996
<b>Creating Home Pages and Portal Panes.....</b>	<b>996</b>
Home Pages and Portal Panes Basics.....	997
Portal Panes.....	999
Default Portal Panes.....	999
Opening Portal Panes.....	1000
Portal Pane Settings.....	1001
Portal Pane Contents.....	1004
Adding Action Links.....	1008
Adding Object Links.....	1009
Adding Search Views.....	1013
Adding Text.....	1014
Adding Web URLs.....	1016
Adding Custom Content.....	1017
Adding WebIntelligence URLs.....	1018
Portal Panes with Multiple Contents.....	1019
Creating Portal Panes.....	1020
Home Pages.....	1021
Home Page Settings.....	1022
Home Page Content.....	1023
Creating Home Pages.....	1024
Customization Guidelines.....	1026
About Master Home Pages and Portal Pane Templates.....	1027
Modifying Home Pages and Portal Panes.....	1028
Synchronizing Master Pages, Templates, and Copies.....	1029
About Modifying Home Pages.....	1029
About Modifying Portal Panes.....	1030
Modifying Portal Panes.....	1032
Deleting Portal Panes.....	1033
Modifying Master Home Pages.....	1035
Adding Portal Panes to Home Pages.....	1037
Creating Portal Panes from a Master Home Page.....	1038
<b>Using Global Navigation.....</b>	<b>1039</b>
Working with Tab Bar Appearance.....	1040
Working with Tab Bar Contents.....	1041
<b>Using Templates.....</b>	<b>1042</b>
Planning Templates.....	1043
Using Template Features.....	1045
General Template Information.....	1045
Defining Document Folders for Templates.....	1046

Records Tab Structure.....	1047
Creating Templates.....	1048
Defining Values in Templates.....	1050
Adding Fields to Templates.....	1050
Field Value Types.....	1053
Text or Memo-Text Fields.....	1053
Number Fields.....	1053
Defining Fields w with Literal Values.....	1054
Defining Fields w with Attribute Values.....	1055
Defining Fields w with Formula Values.....	1055
Adding Sub-objects to Templates.....	1056
Adding Categories to Templates.....	1057
Adding Assignees and Attendees to Templates.....	1058
Deleting Sub-objects from Templates.....	1060
Adding Related Objects to Templates.....	1060
<b>Creating Wizards.....</b>	<b>1063</b>
Designing Wizards.....	1063
Writing a Wizard Specification Document.....	1065
Whether to Use Templates.....	1067
Designing Wizard Layout.....	1068
Identifying Wizard Flow and Page Transitions.....	1069
Creating Wizards.....	1070
Opening the Wizard Screen.....	1071
Defining General Wizard Information.....	1072
Adding Pages to Wizards.....	1074
Defining Page Components in Wizards.....	1076
Defining Wizard Parameters.....	1081
Using Actions in Wizards.....	1084
Applying Automated Actions.....	1085
Specifying Initial Actions.....	1086
Defining Simple Actions.....	1087
Defining Page Transition Logic.....	1094
Creating Page Transition Rules.....	1097
Checking Page Order in Wizards.....	1101
Testing and Troubleshooting Wizards.....	1102
Execution and Population Order.....	1107
<b>Using Rules.....</b>	<b>1107</b>
Rule Components.....	1108
Rule Types.....	1108
Security Rules.....	1109
Pre-population Rules.....	1110
Validation Rules.....	1110
Approval Rules.....	1111
Custom Action Rules.....	1112
Scheduled Action Rules.....	1113
User Invoked Actions.....	1113
Audit Rules.....	1114
Updates, Triggers, and Audit Rules.....	1115
Audit Default Description.....	1116
Configuring History Object Definition for Audit Tracking.....	1117
Post Commit Rules.....	1118
Rule Triggers.....	1118
Triggering Events.....	1118
Creating Rules.....	1122



Defining General Rule Information.....	1122
Accessing the Rules Screen.....	1122
Setting General Rule Information.....	1126
Rule Qualifiers.....	1129
Qualifier Tab Information.....	1130
Qualifier Items.....	1131
Available Operators in Qualifier Items.....	1133
Right Argument Options in Qualifier Items.....	1135
Qualifier Logic.....	1137
Creating Rule Qualifiers.....	1139
Qualifier Item Examples.....	1140
Limitations for User Interface Qualifiers.....	1143
Rule Actions.....	1144
Defining Messages for Security and Validation Rules.....	1145
Defining Actions for Pre-population Rules.....	1145
Defining Actions for Approval Rules.....	1146
Specifying Actions for Custom Action Rules.....	1154
Defining Settings for Scheduled Action Rules.....	1157
Defining Actions for Audit Rules.....	1160
Rule Administration.....	1165
Defining User Interface Rules.....	1165
Deactivating Rules.....	1166
Modifying Rules.....	1167
Deleting Rules.....	1167
Troubleshooting Rules.....	1168
Creating Routes.....	1169
Accessing Routes Screen.....	1170
Defining General Route Information.....	1171
Route Stops.....	1171
Adding Stop Members.....	1175
Adding Groups as Stop Members.....	1176
Adding User Paths as Stop Members.....	1177
Adding Users With Roles as Stop Members.....	1178
Customizing Route Stops.....	1181
Adding Email Notification Recipients.....	1183
Requirements for Rule Specifications.....	1185
<b>Conditional Expressions.....</b>	<b>1187</b>
Creating a Condition.....	1188
Defining Expressions.....	1190
Qualifier Logic.....	1196
Finding Routes that Use a Condition.....	1196
<b>Adding Custom Code to TeamConnect.....</b>	<b>1197</b>
Uploading Rule Component Files.....	1197
Defining Start Up Classes.....	1199
Defining Rules that Use Custom Code in the User Interface.....	1200
Defining Wizard Page Transition Rules.....	1202
Defining Wizard Page Actions.....	1203
<b>Localizing TeamConnect.....</b>	<b>1203</b>
How are i18n Keys Generated?.....	1204
Features that Require Your Attention.....	1205
Reports.....	1205
Custom Messages.....	1206
Custom Blocks.....	1206
Portal Panes.....	1206

Rules .....	1206
Wizards.....	1207
Saved Searches .....	1207
Upgrade Considerations for Localizing.....	1207
Custom Blocks.....	1207
Parameters.....	1208
Date Formatting.....	1208
Best Practices.....	1208
<b>Migrating Custom Designs .....</b>	<b>1209</b>
Requirements .....	1209
Working with the Design Snapshot Tool.....	1209
Creating a Design Snapshot.....	1210
Creating a Design Upgrade File.....	1211
Reverting to a Previous Snapshot.....	1211
Working with the Design Import Tool.....	1211
Performing a Design Upgrade.....	1212
Working with Configuration Transfer Utility.....	1213
Export Design Changes.....	1214
View Older Packages.....	1216
Custom Search.....	1218
System Settings Excluded From Export.....	1219
Import Design Changes .....	1220
Imported Change History.....	1222
Troubleshooting and Best Practices .....	1223
Artifacts Captured for Custom Designs.....	1225
Troubleshooting.....	1228
Warnings.....	1228
Design Import Restrictions .....	1229
Source Design With No Design Component.....	1229
General Troubleshooting Steps.....	1230
Logging.....	1230
Sample Error Messages and Tips .....	1230
Error Messages and Performance Issues .....	1232
BUILD FAILED message.....	1232
DMT script errors.....	1232
<b>About Document Generator .....</b>	<b>1232</b>
Document Generator Basics.....	1233
Creating Templates: Process Outline.....	1237
<b>Using Document Template Headers .....</b>	<b>1238</b>
Defining Filter Pages.....	1238
Filters for Related Objects.....	1241
Filters for Sub-objects and Single Records .....	1242
Filters for Non-Related Objects.....	1245
Main Filters and Sub-Filters.....	1246
Defining Pages for User Input.....	1250
Creating User Input Fields .....	1250
Displaying Multiple Entry Fields in One Input Page.....	1252
<b>Using Document Generator Templates .....</b>	<b>1253</b>
Writing Document Template Text.....	1254
Document Generator Tags.....	1256
Document Generator Tags Summary.....	1256
tc:data .....	1259
tc:detail .....	1264
tc:loop .....	1266

Retrieving Data from All Instances of Sub-object.....	1267
Retrieving Data with tc:loop Using Qualifiers.....	1268
Qualifier Tag Attribute Syntax.....	1269
tc:loop Qualifier Examples.....	1271
tc:date .....	1272
tc:user .....	1272
tc:file .....	1273
tc:conditional.....	1274
Using tc:conditional to Compare Custom Field Values.....	1277
Using tc:conditional to Compare Sub-objects .....	1278
Using tc:conditional to Test for Object Attribute Values.....	1278
Testing for a Category.....	1279
tc:block .....	1279
tc:compute.....	1280
tc:operand.....	1281
tc:search.....	1282
tc:filter .....	1284
tc:input .....	1287
Nesting to Navigate .....	1288
Locating Object Attribute Names.....	1291
Inserting Tags and Preparing RTF Files.....	1292
<b>Completing the Document Template .....</b>	<b>1296</b>
Creating Document Tags.....	1296
Creating Header and Converting Template to XML.....	1297
Uploading Document Templates .....	1299
Testing Document Generator Templates.....	1299
<b>Using EasyDocs .....</b>	<b>1300</b>
EasyDocs vs. Document Generator: Functional Scope.....	1301
Creating a Document Template: 5-Step Process Overview .....	1302
Required TeamConnect User Group Rights .....	1304
Step 1: Create RTF with Merge Fields.....	1304
Basic Merge Fields.....	1309
Automatically Mapped System Fields.....	1311
Predefined Merge Field Codes.....	1311
Naming Merge Fields for Auto-mapping System Fields.....	1313
Automatically Mapped Custom Fields.....	1314
loop@ Merge Fields for Selecting Related or Sub-objects.....	1315
When NOT to Use loop@.....	1317
Inserting loop@ Merge Fields.....	1317
filter@ Merge Fields for User-selected Related or Sub-objects .....	1318
When NOT to Use filter@.....	1318
Inserting filter@ Merge Fields .....	1318
if@ Merge Fields for Conditional Text.....	1319
Opening the Document Templates Folder.....	1320
Step 2: Upload RTF File to TeamConnect.....	1324
Step 3: Map RTF Merge Fields to TeamConnect Data.....	1325
Opening an RTF in the Data Mapping Tool.....	1325
System Field Mapping.....	1326
Custom Field Mapping.....	1329
loop@ Merge Field Mapping.....	1330
Mapping loop@ Merge Field to Related Object or Sub-object.....	1330
Mapping Merge Fields Nested within loop@.....	1331
filter@ Merge Field Mapping.....	1333
Mapping filter@ Merge Field to a Related or Sub-object.....	1334

Mapping Merge Fields Nested within filter@.....	1335
if@ Merge Field Mapping.....	1337
Mapping to Current Date.....	1340
Mapping to Current User.....	1340
Mapping to a File.....	1341
Converting RTF File to XML File.....	1342
Mapping to XML File.....	1342
Defining a User Input Screen.....	1343
Step 4: Convert RTF File to a Document Generator Template.....	1343
Step 5: Test Your Template.....	1344
Modifying and Deleting Template Files.....	1345
Modifying an Existing RTF File.....	1345
Modifying Existing Data Mapping.....	1346
Deleting Template Files.....	1347
<b>Troubleshooting Document Templates .....</b>	<b>1347</b>
<b>Document Template Samples .....</b>	<b>1353</b>
Nesting to Navigate.....	1353
Retrieving Data through Contact-Centric Fields.....	1355
Automatically Retrieving Data from Related Objects.....	1357
Retrieving Data through Custom Fields of Type Custom Object.....	1358
Retrieving Data through Custom Fields of Type Involved.....	1359
Retrieving Data from Child Objects.....	1359
Using Qualifiers to Filter Sub-objects.....	1362
Using tc:conditional.....	1363
Comparing Custom Fields.....	1363
Comparing Sub-objects.....	1364
Testing for Existing Data.....	1364
Including Information When Certain Conditions Are Met.....	1365
<b>Using Tag Attributes .....</b>	<b>1366</b>
Field Label Tag Attributes.....	1367
Tag Attributes for All Field Types.....	1368
Number Field Tag Attributes.....	1370
Text Field Tag Attributes.....	1372
Memo Text Field Tag Attributes.....	1373
Date Field Tag Attributes.....	1374
List Field Tag Attributes.....	1374
Custom Object Field Tag Attributes.....	1376
Involved Field Tag Attributes.....	1376
<b>Using Object Navigator .....</b>	<b>1377</b>
Attributes.....	1378
Paths .....	1380
Object Navigator Window .....	1380
Starting Tables.....	1383
Filtering Sub-objects and Related Objects.....	1385
Creating Paths.....	1388
Tips for Navigating Objects.....	1390
<b>Object Model: Read This First .....</b>	<b>1393</b>
Objects and Object Model.....	1394
Legacy Names and New er Names.....	1394
Intended Audience.....	1394
Object Model Conventions.....	1395
Enumerations.....	1395
Object Model Terms.....	1396
Descriptions of Columns in Tables.....	1397

Object Model Properties .....	1399
Object Table Prefixes .....	1399
Unique Codes of System Objects and Lookup Tables .....	1401
Viewing XCT Files .....	1403
Object Names vs. Database Table Names .....	1403
List Attributes for Sub-Objects and Related Objects .....	1403
<b>Object Model: Projects .....</b>	<b>1404</b>
TProject .....	1404
JProjAssignee .....	1416
LProjAssigneeType .....	1418
JProjPhase .....	1420
WObjdPhaseType .....	1421
WObjdPhaseTransition .....	1424
JProjRelation .....	1425
LProjRelationType .....	1426
EProjDetail .....	1428
EProjUserAccess .....	1429
EProjGroupAccess .....	1432
TInvolved .....	1434
JInvlRelation .....	1441
EnvlDetail .....	1442
EnvlUserAccess .....	1444
EnvlGroupAccess .....	1446
TMilestone .....	1449
EMileDetail .....	1457
EMileUserAccess .....	1458
EMileGroupAccess .....	1461
<b>Object Model: Contacts .....</b>	<b>1464</b>
TContact .....	1464
JContAddress .....	1482
LCountryItem .....	1484
LContAddressType .....	1485
JContPhone .....	1486
LContPhoneType .....	1488
JContFax .....	1489
LContFaxType .....	1490
JContEmail .....	1491
LContEmailType .....	1492
JContInetAddress .....	1493
LContInetAddressType .....	1494
JContSkill .....	1495
LContSkillType .....	1496
JContDefaultRate .....	1498
JContRate .....	1499
JContRelation .....	1501
LContRelationType .....	1503
JContTerritory .....	1504
LContTerritoryType .....	1505
Y_Contact_Group .....	1506
JDistMember .....	1508
EContDetail .....	1510
EContUserAccess .....	1511
EContGroupAccess .....	1514
<b>Object Model: Appointments .....</b>	<b>1516</b>

TAppointment.....	1516
LApptArealtem.....	1523
JApptAttendee.....	1524
JApptResource.....	1526
LApptResourceType.....	1527
EApptDetail.....	1528
EApptUserAccess.....	1530
EApptGroupAccess.....	1532
<b>Object Model: Accounts .....</b>	<b>1535</b>
TAccount.....	1535
JTranDetail.....	1551
RTransaction.....	1554
EAcctDetail.....	1556
EAcctUserAccess.....	1557
EAcctGroupAccess.....	1560
<b>Object Model: Expenses .....</b>	<b>1562</b>
TExpense.....	1563
EExpeDetail.....	1568
EExpeUserAccess.....	1570
EExpeGroupAccess.....	1572
<b>Object Model: Tasks .....</b>	<b>1574</b>
TTask .....	1574
LTaskActivityItem.....	1582
JTaskAssignee.....	1584
ETaskDetail.....	1585
ETaskUserAccess.....	1586
ETaskGroupAccess.....	1589
<b>Object Model: Invoices .....</b>	<b>1591</b>
TInvoice.....	1591
JInvcCategoryAdjustment.....	1605
JInvcHeaderAdjustment.....	1607
JInvcLineItem.....	1610
JInvcNonUSTax.....	1618
LInvcNonUSTaxType.....	1619
JInvcTimekeeperAdjustment.....	1621
JLitmAdjustment.....	1622
EInvcDetail.....	1625
ELitmDetail.....	1626
ELitmGroupAccess.....	1628
ELitmUserAccess.....	1630
EInvcUserAccess.....	1632
EInvcGroupAccess.....	1635
YCurrencyInfo.....	1636
<b>Object Model: Histories .....</b>	<b>1637</b>
THistory.....	1638
EHistDetail.....	1642
EHistUserAccess.....	1645
EHistGroupAccess.....	1647
<b>Object Model: Documents .....</b>	<b>1649</b>
TDocument.....	1649
JDocuContent.....	1657
YDocuContentType.....	1657
EDocuDetail.....	1659
EDocuUserAccess.....	1660

EDocuGroupAccess .....	1663
<b>Object Model: Common .....</b>	<b>1665</b>
JNote .....	1665
UGrupMember .....	1666
WObjdDetailField.....	1667
WObjdProjectInfo.....	1671
YDetailLookupItem.....	1676
YDetailLookupTable.....	1678
YGroup.....	1679
YRecentlyView edRecord .....	1682
YUser .....	1683
YUserSubscribedCollection.....	1689
<b>Glossary .....</b>	<b>1690</b>
<b>2 Legal Customization Help .....</b>	<b>1708</b>
<b>Legal Customization Guidelines .....</b>	<b>1708</b>
<b>File Naming Conventions .....</b>	<b>1709</b>
Filename Parts.....	1709
Utility Class Files.....	1711
Base Classes.....	1712
Batch Display Files for Custom Java Blocks.....	1712
<b>Object Definitions .....</b>	<b>1713</b>
<b>Embedded Objects .....</b>	<b>1714</b>
<b>Categories .....</b>	<b>1715</b>
<b>Phases and Phase Transitions .....</b>	<b>1716</b>
<b>Assignee Roles .....</b>	<b>1717</b>
<b>Custom Fields .....</b>	<b>1718</b>
<b>Lookup Tables .....</b>	<b>1719</b>
<b>Custom Screens .....</b>	<b>1720</b>
<b>Rules .....</b>	<b>1721</b>
<b>Reports .....</b>	<b>1723</b>
<b>Wizards .....</b>	<b>1724</b>
<b>Custom Tools .....</b>	<b>1729</b>
Time Entry Tool.....	1730
Editing Time Entry Tool Settings.....	1730
Editing Default Time Entry Tool Fields.....	1732
Creating Billing Time Periods.....	1734
Managing Timekeepers.....	1735
Using the Time Entry Tool as Administrator.....	1737
Tool Rights.....	1739
<b>CSM Configuration for Matter Management .....</b>	<b>1740</b>
CSM e-Billing Roles Tab Configuration.....	1741
CSM Vendors Tab Configuration.....	1741
CSM Payment Tab Configuration.....	1742
<b>Groups and Users .....</b>	<b>1742</b>
<b>Service of Process Manager for Matter Management .....</b>	<b>1746</b>
Adding the CT Mapping File to Matter Management.....	1747
CT Mapping Files .....	1747
Adding the CSC Mapping File to Matter Management.....	1748
CSC Mapping Files.....	1748
<b>3 Collaborati Spend Management Customization Help .....</b>	<b>1750</b>
<b>Setup Overview .....</b>	<b>1750</b>
Prerequisites .....	1750
Overview of the Initial Setup Process.....	1752
What Happens After Setup is Complete.....	1753

Administer CSM Setup.....	1754
Viewing Invoice Validation Rule Settings.....	1754
Accessing the Invoice Validation Rule Setting Screen.....	1755
Activating the Rate Request Approval Rule.....	1755
Activating Alternative Fee Arrangements and Rules.....	1756
Updating Additional Rules for Alternative Fee Arrangements.....	1760
Creating Alternative Fee Arrangement Pages.....	1760
<b>Uninstalling CSM .....</b>	<b>1762</b>
Restoring CSM Default Data.....	1764
<b>Adjusting the Invoice Validation Rule Settings .....</b>	<b>1766</b>
Fee Charges Exceed Agreed Rates .....	1766
Expense Charges Exceed Acceptable Unit Costs.....	1767
Do Not Allow Multi-Matter Invoices.....	1768
Timekeeper Billed Too Much Time.....	1769
Invoice Service/Expense Charges are Too Old.....	1771
Invoice is Too Old.....	1772
Invoice Contains Duplicate Timekeeper Charge.....	1773
Invoice Contains Duplicate Expense.....	1775
Invoice Line Item Description Contains Unauthorized Keyw ords.....	1777
Invoice currency doesn't match vendor currency.....	1778
<b>Frequently Asked Questions and Troubleshooting .....</b>	<b>1779</b>
Frequently Asked Questions.....	1779
Troubleshooting.....	1779
<b>4 Developer's Help .....</b>	<b>1780</b>
<b>TeamConnect API Policy .....</b>	<b>1781</b>
<b>The TeamConnect API .....</b>	<b>1782</b>
Setting up Your Development Environment.....	1783
API Package Classes .....	1784
Model Classes.....	1784
Enumerations.....	1786
Service Classes.....	1786
Resource Services.....	1787
Services in Qualifiers, Actions, and Screens.....	1789
Deleting Dependency Records .....	1790
Other Service Classes.....	1790
Getting Started w ith Custom Classes .....	1790
Automated Qualifiers and Actions .....	1794
Automated Qualifiers .....	1795
Automated Actions .....	1796
Qualifier and Action Code.....	1797
Parameters in Qualifiers and Actions .....	1799
Automated Qualifier Example.....	1801
Custom Pages .....	1803
Custom Screens .....	1804
Custom Screen Sample.....	1806
Custom Tools .....	1810
Custom Tool Sample.....	1811
Scheduled Actions.....	1813
Adding Java Classes for Scheduled Actions.....	1813
Using the API w ith Scheduled Actions .....	1814
Creating New Records.....	1815
Object Definitions .....	1815
System Objects.....	1817
Accounts.....	1817



AccountService Samples .....	1817
Appointments .....	1819
AppointmentService Samples .....	1819
Contacts .....	1821
ContactService Samples .....	1822
Documents .....	1824
DocumentService Samples .....	1824
Expenses .....	1827
ExpenseService Samples .....	1827
Groups .....	1829
GroupService Samples .....	1829
History Entries .....	1831
HistoryService Samples .....	1831
Invoices .....	1832
InvoiceService Samples .....	1833
Tasks .....	1835
TaskService Samples .....	1835
Users .....	1838
UserService Samples .....	1838
Custom Objects .....	1840
Projects .....	1840
ProjectService Samples .....	1841
Involved Parties .....	1842
InvolvedService Samples .....	1842
Related Records .....	1843
Getting Related Records of Projects .....	1844
Using Custom Fields and Primary Keys to Relate Records .....	1844
Sub-Objects .....	1845
Project and Task Assignees .....	1846
Contact Sub-Objects .....	1847
Categories .....	1847
Adding or Deleting Categories in Records .....	1848
Returning Categories for a Record .....	1849
Getting and Setting Primary Categories .....	1850
Checking Records for Specific Categories .....	1850
Custom Fields .....	1850
Requirements for Getting or Setting Custom Field Values .....	1851
Getting Custom Field Values .....	1852
Setting Custom Field Values .....	1852
Lookup Table Fields .....	1852
System Lookup Table Tree Positions .....	1854
Custom Lookup Table Tree Positions .....	1856
Getting or Setting Values in List Custom Fields .....	1857
Common API Functions .....	1857
Running Code as the System User .....	1859
Getting Current User Information .....	1861
Security Rights .....	1862
Record Security .....	1862
Functional Security .....	1863
System and User Settings .....	1864
Adding Custom Settings .....	1865
Internationalization .....	1866
Searching for Records .....	1867
Creating an API Search View .....	1869

Search Criteria.....	1869
Searching on Fields .....	1873
Field Path Searching.....	1874
Search Criteria Operators.....	1876
Search Parameters.....	1876
Executing a Search.....	1877
Logging.....	1878
Specifying Logging Messages.....	1879
Checking Logger Levels.....	1880
<b>Java Samples .....</b>	<b>1881</b>
Partial Samples to Demonstrate Concepts .....	1881
Automated Qualifiers.....	1883
Automated Actions.....	1891
Rule Actions.....	1891
Parameterized Actions.....	1899
Wizard Page Actions.....	1904
Searching Samples .....	1909
<b>5 Web Services Help .....</b>	<b>1912</b>
<b>Getting Started .....</b>	<b>1912</b>
Overview .....	1912
Supported Operations.....	1914
Key Concepts.....	1918
Repository Classes.....	1918
Selective Record Updating and Reading.....	1919
Data Types.....	1919
Request and Response Formats.....	1919
Error Handling.....	1920
Record Unique Keys .....	1920
Requirements .....	1920
Generating Client Source-code and API.....	1921
Creating an Application.....	1925
Client Proxy .....	1925
SOAP Message Security Header.....	1926
Putting Components Together in a Client Application.....	1927
Common Functions.....	1927
Creating Records.....	1928
Updating Records .....	1928
Reading Records .....	1928
Searching Records.....	1928
Working with Categories.....	1930
Working with Custom Fields.....	1932
Deleting Records.....	1933
<b>Web Service API Reference .....</b>	<b>1934</b>
API Reference Summary.....	1934
AccountRepository .....	1935
AppointmentRepository.....	1937
ContactRepository.....	1939
DocumentRepository.....	1945
ExpenseRepository.....	1948
GroupAccountRepository .....	1949
HistoryRepository.....	1950
InvoiceRepository.....	1951
LookupTableSource.....	1955
ProjectRepository.....	1955

InvolvedRepository.....	1958
TaskRepository.....	1959
UserAccountRepository.....	1961
Repository Method Details.....	1962
AccountRepository Method Details.....	1962
insertAccount.....	1962
updateAccount.....	1963
readAccount.....	1963
readAccountsByCriteria.....	1964
readChildAccounts.....	1965
activateAccount.....	1965
deactivateAccount.....	1966
allocateMoney.....	1966
transferMoney.....	1966
withdraw Money.....	1967
deleteAccount.....	1967
readRecentlyView edAccounts.....	1968
AppointmentRepository Method Details.....	1968
insertAppointment.....	1968
updateAppointment.....	1970
readAppointment.....	1970
readAppointmentsByCriteria.....	1971
deleteAppointment.....	1971
readRecentlyView edAppointments.....	1972
ContactRepository Method Details.....	1972
insertContact.....	1972
updateContact.....	1974
readContact.....	1974
readContactsByCriteria.....	1975
deleteContact.....	1976
ReadRecentlyView edContacts.....	1976
DocumentRepository Method Details.....	1976
insertDocument.....	1977
createShortcut.....	1979
createSubFolder.....	1979
createHyperlinkDefaultCategory.....	1980
createHyperlink.....	1980
updateDocument.....	1981
readDocument.....	1982
readDocumentsByCriteria.....	1982
readDocumentByPath.....	1983
readChildDocuments.....	1984
readChildDocumentForName.....	1984
readDocumentFolderForDocumentOw ner.....	1985
checkIn.....	1986
checkOut.....	1986
undoCheckOut.....	1987
revert.....	1987
moveDocument.....	1988
copyDocument.....	1988
deleteDocument.....	1989
ReadRecentlyView edDocuments.....	1989
setDocumentAsRecentlyView ed.....	1989
readFolderHierarchy.....	1990

ExpenseRepository Method Details.....	1990
insertExpense.....	1991
updateExpense.....	1991
readExpense.....	1992
readExpensesByCriteria.....	1993
postExpense.....	1993
voidExpense.....	1994
deleteExpense.....	1994
readRecentlyView edExpenses.....	1994
GroupAccountRepository Method Details.....	1995
insertGroupAccount.....	1995
updateGroupAccount.....	1996
readGroupAccount.....	1996
readGroupAccountsByCriteria.....	1997
deleteGroupAccount.....	1998
readRecentlyView edGroupAccounts.....	1998
HistoryRepository Method Details.....	1999
insertHistory.....	1999
updateHistory.....	2000
readHistory.....	2000
readHistoriesByCriteria.....	2001
deleteHistory.....	2002
readRecentlyView edHistories.....	2002
InvoiceRepository Method Details.....	2002
insertInvoice.....	2003
updateInvoice.....	2005
readInvoice.....	2005
readInvoicesByCriteria.....	2006
postInvoice.....	2007
voidInvoice.....	2007
deleteInvoice.....	2008
readRecentlyView edInvoices.....	2008
adjustInvoiceHeader.....	2008
readActiveApprovals.....	2009
readCompletedApprovals.....	2010
readInvoiceApprovalsPendingOnPost.....	2010
LookupTableSource Method Details.....	2011
readSystemLookupTable.....	2011
ProjectRepository Method Details.....	2012
insertProject.....	2012
updateProject.....	2013
readProject.....	2014
readProjectsByCriteria.....	2014
readChildProjectsForEntityType.....	2015
changePhase.....	2016
deleteProject.....	2017
readRecentlyView edProjects.....	2017
readProjectEntityTypes.....	2017
readProjectIntegrationSearches.....	2018
readProjectsUsingSearch.....	2018
InvolvedRepository Method Details.....	2019
insertInvolved.....	2019
updateInvolved.....	2020
readInvolved.....	2021

readInvolvedsForProject.....	2021
readInvolvedsByCriteria.....	2022
deleteInvolved.....	2023
TaskRepository Method Details.....	2023
insertTask.....	2023
updateTask.....	2024
readTask.....	2025
readTasksByCriteria.....	2025
postTask.....	2026
voidTask.....	2026
reassign.....	2027
deleteTask.....	2027
readRecentlyView edTasks.....	2027
UserAccountRepository Method Details.....	2028
insertUserAccount.....	2028
updateUserAccount.....	2030
readUserAccount.....	2030
readUserAccountsByCriteria.....	2031
deleteUserAccount.....	2032
readRecentlyView edUserAccounts.....	2032
REST API HTTP Methods.....	2033
Abstract Classes.....	2036
Category and Custom Field Data Objects.....	2038
Data Objects Used in Search Criteria.....	2039
<b>Frequently Asked Questions.....</b>	<b>2041</b>
Batch Operations.....	2041
Operations Not Supported In Web Services.....	2041
Cosmetic Data Objects.....	2042
<b>Client Application Components (Java/Apache CXF).....</b>	<b>2042</b>
Client Proxy Sample Code.....	2043
Security Headers Sample Code.....	2044
Client Application Sample Code.....	2045
<b>Code Samples (Java/Apache CXF).....</b>	<b>2047</b>
Contacts.....	2048
Creating Contacts.....	2048
Updating Contacts.....	2050
Reading Contacts.....	2052
Searching for Contacts.....	2052
Deleting Contacts.....	2054
Accounts.....	2055
Creating accounts.....	2055
Updating accounts.....	2056
Reading accounts.....	2057
Reading child accounts.....	2058
Searching accounts.....	2059
Activating accounts.....	2060
Deactivating accounts.....	2061
Allocating money to accounts.....	2061
Transferring money betw een accounts.....	2061
Withdraw ing money from accounts.....	2062
Deleting accounts.....	2062
Appointments.....	2062
Creating appointments.....	2062
Updating appointments.....	2063

Reading appointments.....	2064
Searching appointments.....	2065
Deleting appointments.....	2066
Document Records.....	2066
Creating document records.....	2066
Updating documents.....	2067
Reading documents.....	2068
Reading child documents for a document.....	2069
Reading documents by path.....	2070
Reading the child document of a parent folder.....	2071
Reading the Document Folder for a User.....	2072
Searching documents.....	2073
Creating document shortcuts.....	2075
Creating subfolders.....	2075
Creating hyperlinks with the default Document category.....	2075
Creating hyperlinks.....	2076
Copying documents.....	2076
Moving documents.....	2076
Checking out documents.....	2077
Undoing a document checkout.....	2077
Checking in documents.....	2077
Reverting a document to a previous version.....	2077
Deleting documents.....	2078
Expenses.....	2078
Creating expenses.....	2078
Updating expenses.....	2078
Reading expenses.....	2079
Searching expenses.....	2080
Posting expenses.....	2081
Voiding expenses.....	2081
Deleting expenses.....	2081
Group Accounts.....	2081
Creating group accounts.....	2082
Updating group accounts.....	2082
Reading group accounts.....	2083
Searching group accounts.....	2084
Deleting group accounts.....	2084
History Records.....	2085
Creating history records.....	2085
Updating history records.....	2085
Reading history records.....	2086
Searching history records.....	2087
Deleting history records.....	2088
Invoices and Line Items.....	2088
Creating an Invoice with Line Items.....	2088
Updating and Adjusting an Invoice.....	2090
Adjusting Invoice Line Items.....	2091
Reading an Invoice.....	2092
Searching for Invoices.....	2093
Posting an Invoice.....	2095
Voiding an Invoice.....	2095
Searching for Line Items.....	2095
Deleting an Invoice.....	2097
Projects (Matters).....	2097











Creating a Project With Assignees and a Custom Field.....	2097
Updating a Project by Adding an Embedded Object Record.....	2099
Changing a Project's Phase.....	2100
Reading Projects.....	2100
Reading Child Projects for a Record.....	2101
Searching for Projects.....	2101
Deleting a Project.....	2102
Involved Records.....	2102
Inserting Involved records.....	2102
Updating Involved records.....	2103
Reading Involved records.....	2104
Reading Involved Parties for a Project Record.....	2104
Searching Involved Party Records.....	2105
Deleting an Involved.....	2106
Tasks.....	2106
Creating tasks.....	2106
Updating tasks.....	2107
Reading tasks.....	2107
Searching tasks.....	2108
Posting tasks.....	2109
Voiding tasks.....	2109
Reassigning tasks.....	2110
Deleting tasks.....	2110
User Accounts.....	2110
Creating user accounts.....	2110
Updating user accounts.....	2111
Reading user accounts.....	2111
Searching user accounts.....	2112
Deleting user accounts.....	2113
<b>Glossary.....</b>	<b>2113</b>

## Index

**2115**

# 1 Enterprise User Help

Welcome to the *TeamConnect® Enterprise User Help*.

 <a href="#">Contacts</a>	 <a href="#">Address Books</a>
 <a href="#">Projects</a>	 <a href="#">Reports</a>
 <a href="#">Appointments</a>	 <a href="#">Accounts</a>
 <a href="#">Invoices</a>	 <a href="#">Expenses</a>
 <a href="#">Documents</a>	 <a href="#">Approvals</a>

## 1.1 Getting Started

You get started with TeamConnect by following these steps:

1. Be able to access TeamConnect using your login information. See [Logging In](#) and [Authenticating Users](#).
2. Familiarize yourself with the user interface. See [User Interface](#) for more information about the interface.
3. Personalize your preferences and other settings. See [Personalizing Your Settings](#) for more information.
4. Modify your home page and portal panes if necessary. See [Home Pages and Portal Panes](#) for more information.

### 1.1.1 Authenticating Users

Authentication is the process of verifying that you are an authorized user by comparing your identification with information stored in the system. Login procedures vary based on the authentication mechanism used by your organization.

There are two main methods:

- Standard Login
- Single Sign-on Authentication

#### Standard Login

If you access the login URL and see a screen with username and password fields, your organization uses the standard login. This type of authentication compares your username and password with stored information and grants access if a match is made. For further details as well as notes on the Forgot Password functionality, see [Logging In](#).



The screenshot shows a login form with a light gray background. At the top left, it says 'Login' in a small font, followed by the 'TeamConnect' logo where 'Team' is orange and 'Connect' is blue. Below that, it says 'Welcome to TeamConnect'. There are two input fields: the first is labeled 'User Name' and the second is labeled 'Password'. At the bottom left is a blue 'Login' button, and at the bottom right is a blue link that says 'Forgot Password'.

Version 5.0 UIUX Build 106 | Copyright © 1999-2016 Mitrattech Holdings, Inc. All Rights Reserved.

## Login Screen

### Single Sign-on Authentication

Single sign-on authentication lets you log in without providing a username and password—assuming you are identified through some local network authentication process. If you are successfully authenticated, a new window opens and displays your home page. For more details on logging in using single sign-on, contact your administrator.

If you close the window or your session expires, you cannot reuse the original window even if you are still logged in to your local network. To use single sign-on again, you must open a new browser window and go to the login URL.

Pages are often customized for an organization's specific needs. The look of your pages, and the data shown on those pages, may be quite different from the "out of the box" version. In this documentation, screen images and other descriptions refer to the "out of the box" appearance. However, even if your organization's performed customizations, the ways of navigating within and between pages will be very similar to those described here.

### 1.1.2 Logging In

Whenever you log in, you open a working session. A working session starts when you log in to the system and ends when you properly log out. At any time, you can have only one open session per user.

If your organization has a single sign-on authentication mechanism in place, see [Single Sign-on Authentication](#).

## Localized Language Support

The language of the text on the Login screen is either the default language that is set in your browser, or the one set as the system default language. If you specified a different language in your personal Preferences page, once you log in, your selected language is used.

### Using Standard Login

If your organization is using a standard login screen, obtain the following information from your administrator:

- TeamConnect URL
- Your username
- Your default password

**(You change your personal password after you log in for the very first time. See [Changing Password](#).)**

### To log in

1. Open your browser.
2. In the **Address** field at the top of the browser window, type the URL and click Go. The Login screen is displayed.
3. Enter your username.  
Usernames are not case-sensitive. No spaces are permitted in username.
4. Enter your password.  
Passwords are always case-sensitive and no spaces are allowed in passwords. Passwords may contain special characters.
5. Click **Log In**.  
Your home page with your name displayed at the top appears in your browser window.

### Forced Password Changes

Depending on the security policies of your organization, you may occasionally see a page requiring you to change your password immediately after logging in. Type your new password twice for verification before continuing. Do not use a password that you have used in the past.

### Locked Account

After several unsuccessful attempts to enter your password, your account may lock and an error message appears. Please contact your system administrator. Your administrator can unlock the

account and assign a new password.

**Note:** *When your account is locked, you are still an active user and can be included when users schedule appointments, or assign tasks.*

#### **Forgot Password Link**

Beginning with TeamConnect 5.0, Admins can allow for password reset from the login screen.

To reset your password from login:

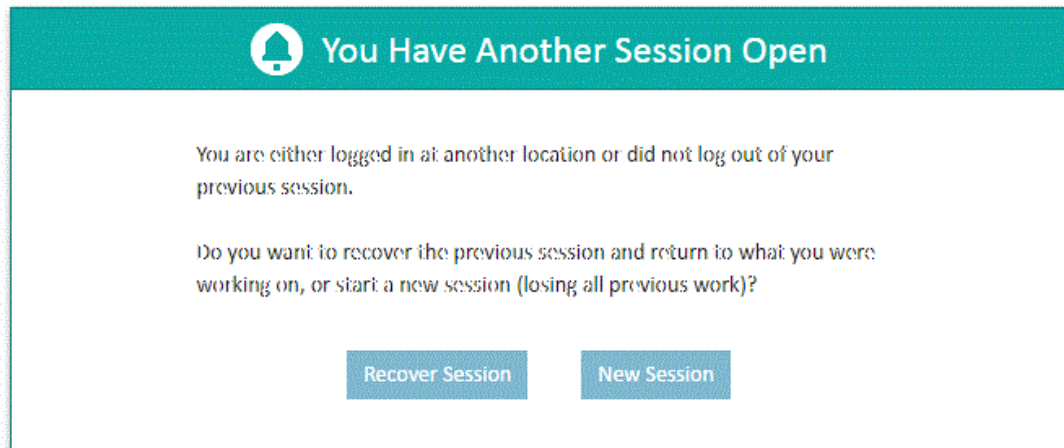
1. Click the **Forgot Password** link to reset your password.
2. Enter and confirm your User Name and your primary email address and click **Send**.
3. Once the information is confirmed, your primary email address will receive a reset password link. This single-use link will expire in 24 hours after sending.
4. Click the link in your email to reset the password. Login as usual after confirming the new password.

**Note:** *Passwords must be adhere to the password parameters set by your administrator. Passwords over 50 characters will result in the user being locked out.*

#### **Previous Session Found**

If you previously shut down by closing the browser window without logging out properly, you must choose an option before continuing.

## TeamConnect®



Version 5.0.01 Build 93 | Copyright © 1999-2016 Mitratech Holdings, Inc. All Rights Reserved.

### Previous Session Found Screen

#### Previous Session Found Screen Options

Option	Description
<b>Recover Session</b>	When you select this option, your home page is displayed on your screen:  Records that had been edited, but not saved, in the previous session will still appear in the "Unsaved Items" link.
<b>New Session</b>	When you select this option, you log in to a new session with no Unsaved Items shown.

### About Home Pages

The home page is the first page you see after logging in. Home pages function as your starting point and contain specialized sections called portal panes. Portal panes provide links to create and access various records and information that is of importance to you, such as:

- Your appointments
- Your projects
- Your documents
- Stocks and weather
- Your pending approvals

- News headlines
- Your tasks
- Business Objects® WebIntelligence reports
- Saved searches

Depending on how many user groups you belong to, and how many home pages are available, you may have access to more than one home page.

### Currently Logged in User

Every time you log in you become the currently logged in user. That means that the system recognizes who you are and displays your personal home pages, your font and color preferences on the screen, your pending tasks, approvals, upcoming appointments, and other information that is relevant to you. Your name is automatically recorded in the **Created by** and **Modified by** fields of all records that you create or edit. You are also automatically added as an attendee for all appointment records that you create.

### Session Time-out

Sometimes when you are logged in but do not use it for a certain period of time, you may be notified that your session has expired. This is also known as a session time-out. The time-out period is set by your administrator.

When you attempt to log back in, you may be notified that you already have an open session. For details on your options, see [Previous Session Found](#).

*Tip: If you must step away from your desk or are interrupted, save your work to ensure that you can continue working later.*

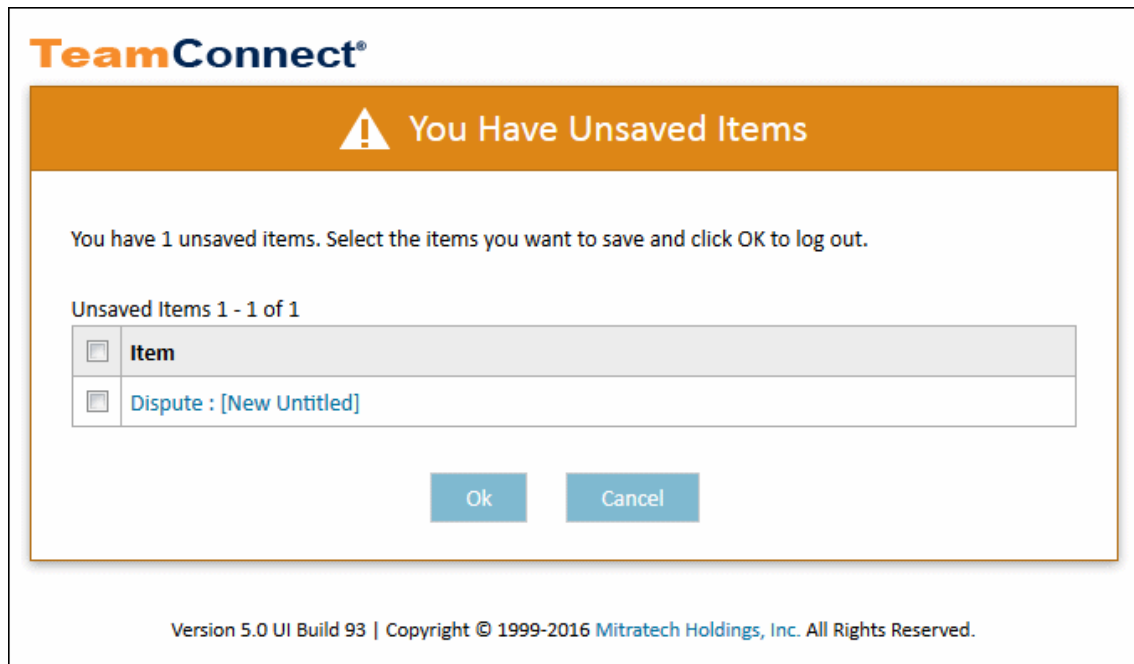
## 1.1.3 Logging Out

When you are finished working, you need to log out from it to make sure that the application is shut down properly, and no data that you have worked with is lost.

### To log out from

On the global toolbar, click the Log Out link.

- If all records that you worked on are saved, you are immediately logged out.
- If any records that you have worked with are not saved, the **Unsaved Items** screen appears.



Unsaved Items Screen

Perform one of the following options:

- If there are any records in the displayed list that you want to save, select the check-box next to them and click **OK** to go back and save the records.
- If you do not want to save any records, click **OK**.
- If you do not want to log out, and want to continue working with the records, click **Cancel**.

### 1.1.4 User Interface

A typical page displays the areas that you have rights to view. **Note:** *Browser width should be 1024px or larger for best performance.*

The fields on your page might vary, but the following components are the same:

- **Browser Title Bar**—Displays the application's name.
- **Global Toolbar**—Contains links that show your pending work and let you do other common tasks. The icon at the furthest right spot (with your login name) can bring you to your preferences page. The **All** folder is a valuable method for navigating the application.
- **Bookmarks icon**—Contains bookmarks which will be preserved after log-out and will not be visible to any other users. For more information on bookmarking, see TeamConnect User Experience Updates.
- **Global Search**—Allows you to search for records by their name or number, across one record type or all record types.
- **Tab Bar**—Contains a set of buttons that provide quick access to record types and functions. In some cases, clicking a tab immediately takes you to a new page. In other cases, clicking a tab

button presents a short group of sub-tab links in the line just below the tabs. These tabs can be shown/hidden by clicking the arrow button.

**Important:** While tabs may be configured and change between systems, using the **All** button is always an alternative for navigating between objects/items in TeamConnect.

- **Record Screen Title Bar**—Shows the record type that you are working with. For a record list page, the name of the current collection is shown. For an individual record page, the name of the record is shown.
- **Left Pane**—Provides links that take you to pages related to your current page. If you are viewing a list of records, the links in the left pane primarily point to other collections. For more information about collections, see [Viewing Collections of Records](#). If you open a record, clicking on a link in the left pane opens pages related to the record.
- **Body**—Contains the main section of the screen, where you work with records, set preferences, use tools, and view home page information. When scrolling on a page, an icon will appear allowing users to go *Back to Top* in a single click.

Edit	Contact	Email	City	State	Phone
1	System Admin				
	10th Circuit Court of Florida [Civil]		Tallahassee	Florida	(850) 488-0125
	10th Circuit Court of Illinois [Civil]		Peoria	Illinois	(309) 672-6047
	10th District Court of Appeals [Civil]		Waco	Texas	(254) 757-5200
	10th Judicial District of Kansas [Civil]		Olathe	Kansas	(913) 715-5000
	11th Circuit Business Court of Florida (Complex Business Litigation Court Procedures) [Civil]	janna.hill@mitratech.com	Miami	Florida	(305) 548-5632
	11th Circuit Court of Florida [Civil]		Miami	Florida	(305) 548-5632
	11th Circuit Court of Illinois [Civil]		Bloomington	Illinois	(309) 888-5254
	11th Circuit Court of Missouri [Civil]		St. Charles	Missouri	(636) 949-3043
	11th Circuit Court of Tennessee - Hamilton County [Civil]		Chattanooga	Tennessee	(423) 209-6700
	12th Circuit Court of Florida [Civil]		Sarasota	Florida	(941) 861-7800
	12th Circuit Court of Illinois [Civil]		Joliet	Illinois	(815) 727-8540
	12th Judicial Circuit Court of Kentucky: Civil [Civil]		Louisville	Kentucky	(502) 595-3007
	12th Judicial District Court of Nebraska [Civil]		Gering	Nebraska	(308) 436-6660
	13th Circuit Business Court of Florida (Complex Business Litigation Division Procedures) [Civil]		Tampa	Florida	(813) 276-8100
	13th Circuit Court of Florida [Civil]		Tampa	Florida	(813) 276-8100
	13th Circuit Court of Illinois [Civil]		Ottawa	Illinois	(815) 434-0786
	13th Circuit Court of Missouri [Civil]		Columbia	Missouri	(573) 886-4060
	13th Circuit Court of Virginia [Civil]		Richmond	Virginia	(804) 646-6505
	13th District Court of Appeals [Civil]		Corpus Christi	Texas	(361) 888-0416
	14th Circuit Court - Henry County [Civil]		Rock Island	Illinois	(309)558-3289
	14th Circuit Court - Mercer County [Civil]		Rock Island	Illinois	(309)558-3289

Typical Page

#### 1.1.4.1 Getting Hints and Help

### Hover Feature

In many areas of the user interface, a small informational window appears if you hover over an item. If you are hovering over a link to a record, the snapshot will contain some basic facts about that record. For example, hovering over a link to a Project record will show the Project's name, main assignee, current phase, and creation date. Almost all record links behave this way. This feature does not apply to record links in calendar views, or lists of embedded records. This feature can be customized by administrators to display select information. For further detail on this functionality, see [Custom Hover Popups](#).

## Online Help

On every page, online help is available by clicking the **Help** button in the record toolbar. In many cases the online help that appears will be specific to the page you are on; in some cases a more general help screen will appear.

### 1.1.4.2 Web Browser Buttons and Commands

You can use common web browser commands (Back, Forward, Stop, and Refresh) while working. There are some things to keep in mind regarding these buttons and commands:

- There is a limit of 10 uses of the Back button or command. You cannot see pages that are more than 10 pages older than your current page.
- If you click Back or Refresh, you lose any edits that you had done on the current page. If you are editing something, then click Back twice, then click Forward twice, you will be on the page you had been editing, but the data you typed will be gone.
- The Refresh button redisplay a page, but it does not rerun any actions that you may have taken on that page. However, if you are on a page that displays search results, clicking Refresh will re-execute the search and display the newest results.
- If you click the Stop button while a page is loading, you will leave the page in an inconsistent state. You cannot rely on items on the page working correctly unless you click Refresh to force the page to finish loading.

### 1.1.5 Personalizing Your Settings

You can customize the look of the user interface to meet your needs or personal preferences. For example, you can change the color of your screen and menus. You can also define your own settings regarding email notifications, home pages, and password. All of these settings can be defined and modified through the **Preferences** in the user menu.



Personal Settings Icon

**Note:** Some preferences, such as font size, can only be changed using the settings in your browser. For more information, see your browser's online help or documentation.

#### To access your personal settings

Click your icon in the upper-right corner of the application and select **Preferences**.

The General preferences page appears. The left pane shows links for all the preferences pages:

- **General**—Define date and time formats, search result settings, calendar settings, button defaults for saving and canceling and accessibility settings. For details, see [Changing Preferences](#).



The User Interface selection block will allow for custom color palette choices. For more detail, see [Color Profile Administration](#).

- **Reset Password**—Change your password. For details, see [Changing Password](#).
- **Email Notifications**—Request email notifications when you are assigned to new tasks and projects, or when new appointments are scheduled for you. For details, see [Setting Up Email Notifications](#).
- **Other Settings**—Enter specific information required for system integrations. For details, see [Choosing Other Setting Options](#).

### 1.1.5.1 Changing Preferences

You can customize the look of the Edit and View modes according to your personal preferences in terms of screen colors, font size, and color. You can also define email notification settings, password, and other options.

***Note:** The settings that you define and modify only affect your own screen. The changes you make to your settings have no effect on other users.*

- Region, Date, and Time formats, and the time zone. See [Selecting Region, Date, and Time Formats](#).
- Search Results appearance. See [Defining Search Result Settings](#).
- Miscellaneous settings. See [Defining the Time Interval for the Appointment Calendar](#).

#### 1.1.5.1.1 Selecting Region, Date, and Time Formats

You can specify the locale that you want to use for your sessions. You can customize the display of the date and time formats and specify your time zone. All dates and times in your records are displayed in the selected time zone, regardless of the time zone in which they were originally set.

***Important:** Be sure that the time zone that you specify matches the equivalent settings in your computer's operating system. Your computer's settings are used by your web browser when displaying the Calendar page and in other areas.*

#### To select a region, date format, time format or a time zone

1. Click your user icon and select **Preferences**.
2. In the **Date Formats** and **Time Formats** sections, fill in the appropriate fields as described in [User Preferences Field Descriptions for Regions and Dates table](#).
3. Click **Update**.

**Locale**

Locale:

---

**Date Formats**

Edit mode date format:  undefined

Read-only mode date format:  undefined

---

**Time Formats**

Edit mode time format:  undefined

Read-only mode time format:  undefined

Time Zone:

**Fields in the Region Section, General Page of Preferences**

**User Preferences Field Descriptions for Region and Dates**

Label	Description
<b>Locale</b>	If you have never specified a locale before, the existing system default locale is shown. You may choose a locale from the drop-down list. Locale controls the language of text in the application, and the format settings for date, time, and numbers. When you choose a different locale the date and time format fields, described below, change to reflect that locale.
<b>Edit mode date format</b>	To change the way dates display when a record is editable click this drop-down list and select one of the following: <ul style="list-style-type: none"> <li>• <b>Short</b>—Displays the date in a mm/dd/yyyy format. For example, 12/25/0505 to represent December 25, 2005.</li> <li>• <b>International</b>—Displays the date in a yyyy-mm-dd format. For example, 2005-12-25 to represent December 25, 2005.</li> </ul>
<b>Read-only mode date format</b>	To edit the way dates display when a record is not editable, click this drop-down list and select one of the following (where mm is month, dd is day, and yyyy is year): <ul style="list-style-type: none"> <li>• <b>Short</b>—Displays the date in a mm/dd/yy format. For example, 12/25/05 to represent December 25, 2005.</li> <li>• <b>Medium</b>—Displays the date in a Month dd, yyyy format when the month is abbreviated. For example, Dec 25, 2005.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Long</b>—Displays the date in a Month dd, yyyy format. For example, December 25, 2005.</li> <li>• <b>Full</b>—Displays the date in a day, Month dd, yyyy format. For example, Saturday, December 25, 2005.</li> <li>• <b>International</b>—Displays the date in a yyyy-mm-dd format. For example, 2005-12-25 to represent December 25, 2005.</li> </ul>
<b>Edit mode time format</b>	<p>To change the way times display when a record is editable, click this drop-down list and select one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Short HH:MM AM/PM</b>—Displays the time in a HH:MM AM/PM format. For example, 5:10 PM.</li> <li>• <b>International</b>—Displays the time in a HH:MM:SS format where HH can be from 1 to 24. For example, 17:10:52.</li> </ul>
<b>Read-only mode time format</b>	<p>To edit the way times display when a record is not editable, click this drop-down list and select one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Short HH:MM AM/PM</b>—Displays the time in a HH:MM AM/PM format. For example, 5:10 PM.</li> <li>• <b>Medium</b>—Displays the time in a HH:MM:SS AM/PM format. For example, 5:10:52 PM.</li> <li>• <b>Long</b>—Displays the time in a HH:MM:SS AM/PM Time Zone format. For example, 5:10:52 PM CST.</li> <li>• <b>International</b>—Displays the time in a HH:MM:SS format where HH can be from 1 to 24. For example, 17:10:52.</li> </ul>
<b>Time Zone</b>	<p>Click the drop-down list to change the system Time Zone.</p> <p>To ensure that the date and time fields are accurate, the time zone that you select here must match the time zone of the computer on which the browser is running. Note that when you change the time zone in your operating system, your web browser will not recognize the change until you close and reopen the web browser. Be sure to do this.</p> <p>Daylight savings time adjustments, if applicable, will be derived from the time zone that has been chosen.</p> <p>If you do not select a time zone in user preferences, the system uses the admin time zone settings. In this case, the admin time zone settings must match the time zone of the computer on which the browser is running. See the "Setting Up Region Settings" section in the <i>Administrator's Guide</i>.</p>

## 1.1.5.1.2 Defining Search Result Settings

The search result settings allow you to customize search results, such as the number of records displayed per page in search results.

**Search Results**

Number of records per result page:

Number of card columns in contact card view:

Do not update Global Search drop down with corresponding tab

Automatically populate Contact field if search returns one result

Automatically populate Project field if search returns one result

**Search Result Settings**

The search result settings are described as follows:

**User Preferences General Page Field Descriptions**

Label	Description
<b>Number of Records per result page</b>	Enter the number of records to display per page of search results. The default is 10.
<b>Number of card columns in contact card view</b>	(Only for contacts search views where the Results Display Type is set to Card) For contacts search results, enter the number of columns to display. The default is 3.
<b>Do not update Global Search drop down with corresponding tab</b>	By default, the category dropdown of the global search bar will change to reflect which tab is currently being viewed. Check this box to disable that functionality and leave the tab at a static selection.
<b>Automatically populate Contact field if search returns one result</b>	Do one of the following: <ul style="list-style-type: none"> <li>To automatically populate a Contact field when the related Contact Search Module returns one record result, select the <b>Automatically populate Contact field if search returns one result</b> check-box.</li> <li>To display a link to a record if a search returns one record result, clear the <b>Automatically populate Contact field if search returns one result</b> check-box.</li> </ul>
<b>Automatically populate Project field if search returns one result</b>	Do one of the following: <ul style="list-style-type: none"> <li>To automatically populate a Project field when the related Project Search Module returns one record result, select the <b>Automatically populate Project field if search returns one result</b> check-box.</li> </ul>

- To display a link to a record if a search returns one record result, clear the **Automatically populate Project field if search returns one result** checkbox.

#### 1.1.5.1.3 Defining the Time Interval for the Appointment Calendar

Use the **General** page of user preferences to define or change the time interval that you can view and select when creating or changing an appointment.

For example, if you select 30 minute intervals, when you create or edit an appointment, your appointment time choices are listed in 30 minute increments, such as 8:00 am, 8:30 am, 9:00 am, and 9:30 am. The following minute options are available: 5, 10, 15, 30, 60.

**Note:** This setting does not affect the way that your calendar listings appear when you click the **Calendar** tab.

#### To change the time interval

Select an option from the **Time interval** drop-down list. The default is 30 minutes.

A screenshot of a user interface element. At the top, there is a blue header bar with the word "Calendar" in white. Below this, on a light blue background, the text "Time interval:" is followed by a white input field containing the number "30" and a small downward-pointing arrow icon to its right, indicating a drop-down menu.

General Page, Calendar Time Interval Setting

#### 1.1.5.1.4 Enabling Accessibility Mode

Use the **General** page of user preferences to provide additional accessibility options for those with vision impairment or difficulty using the mouse. For more information, see [Accessibility](#).

A screenshot of a user interface element. At the top, there is a blue header bar with the word "Accessibility" in white. Below this, on a light blue background, there is a checkbox with a mouse cursor icon pointing at it, followed by the text "Enable accessibility mode".

Preferences Access

#### 1.1.5.2 Changing Password

Use the **Reset Password** page in user preferences to change your log in password.

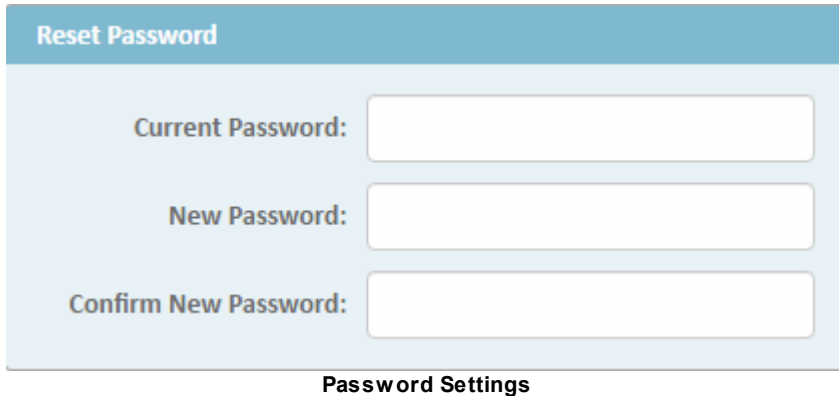
#### To change your password

1. Click the **User Icon** in the upper right corner and select **Preferences**.

The preferences **General** page opens.

- 
2. In the left pane, click the **Reset Password** link.

The **Reset Password** page opens.



Reset Password

Current Password:

New Password:

Confirm New Password:

Password Settings

If enabled, the details of your company's password policy may display on this screen.

- 
- 
3. In the **Current Password** field, type your current password.
4. In the **New Password** field, type your new password.

**Important:** If the details of your company's password policy are not displayed, contact your administrator for password requirements.

- 
- 
- 
- 
5. In the **Confirm New Password** field, retype your new password.
6. Click the **Update** option.
7. Log off and log back in with your new password.

### 1.1.5.3 Setting Up Email Notifications

You can set automatic email notifications for yourself to alert you when:

- You are scheduled to attend an appointment
- You are assigned to a project
- You are assigned to a new task
- You receive requests to approve
- You receive requests to review
- You receive back a review of an approval request that you asked for
- You have an approval request that will expire
- You have an approval request that is canceled and removed from your list

If you are a Process Manager for any approval processes, you have additional email notification options to alert you when:

- An error occurs in a process

- A route or stop expires
- A request is rejected

Rules for what actions require approval are activated in the system by your administrator. For example, email notifications can be set for approval or rejection of invoice posting.

**Note:** For more details about approval processes, see [Workflow Processes](#). Also, see [Using Approval Process Email Notifications](#).

To receive email notifications, your default email address must be correct in your personal contact record (see [Creating a Contact](#)). This email address is also used when you trigger a message to be sent to other users. For example, when you assign other users to tasks, the default email address in your contact record is used as the **From** address of the message.

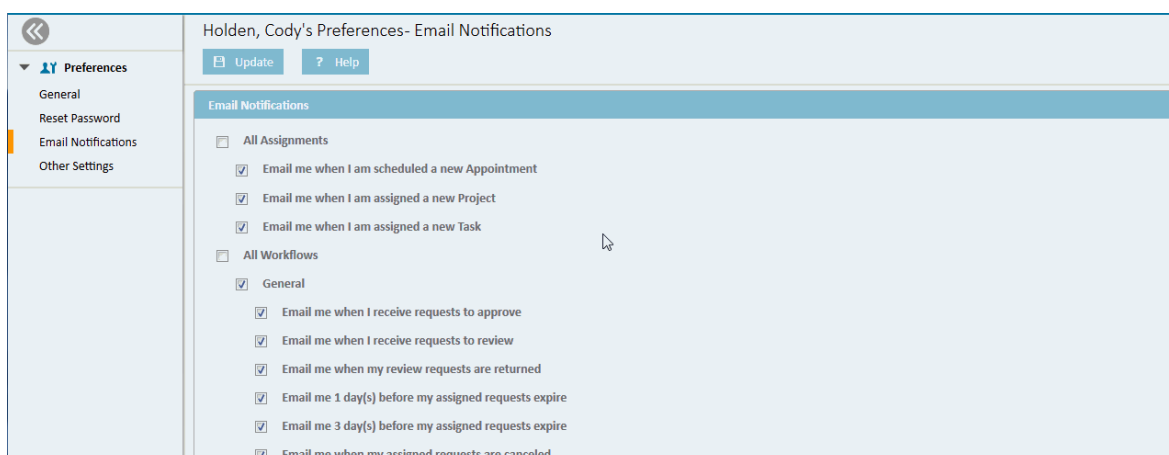
**Note:** If your system is set to a language other than default English, email notification text appears in your preferred language if you are the only recipient. When there are multiple recipients on an email, the email notification is sent in the selected system language.

### To request email notifications

1. Make sure that your email address is correct.
2. On the global toolbar, choose the **Preferences** link, then the **Email Notifications** link.
3. Select the appropriate check-boxes next to the notifications or categories of notifications you want to receive.
4. Click the **Update** option.

The next time an event occurs for which you have requested email notifications, you will receive an email.

Email notification check-boxes are categorized by "All Assignments", "All Workflows", and by project under "All Workflows". Selecting one of the parent check-boxes will automatically select all of the check-boxes in that category. If active approval rules exist, they will appear under the appropriate project heading under the "All Workflows" check-box.



Email Notifications Screen

#### 1.1.5.4 Choosing Other Setting Options

Depending on specific information entered by your administrator, the options in your **Other Settings** page vary. Unless you use another system such as Microsoft Exchange, Lotus Notes, Business Objects, or other products, you do not need to enter data into the **Other Settings** page. For more information on miscellaneous settings, contact your administrator.

### Business Objects WebIntelligence Report Settings

If your organization uses Business Objects WebIntelligence to generate reports based on the data you enter, and your TeamConnect and WebIntelligence username and password combinations are different, you should make the appropriate entries on the **Miscellaneous Settings** tab of your **Preferences** screen. This is used to allow a single sign-on to WebIntelligence.

#### To set up your access to WebIntelligence

1. On the global toolbar, click **Preferences**, then click the **Other Settings** link.
2. In the **Business Objects** settings section, type your WebIntelligence username and password. Retype the password to confirm it.



The screenshot shows a form titled "Business Objects" with a light blue header. Below the header, there are three input fields with labels: "Business Objects User Name:", "New Business Objects Password:", and "Confirm Business Objects Password:". The form is set against a light blue background. Below the form, the text "Business Object Settings" is centered.

**Important:** If you use Business Objects but you do not see this section, contact your administrator.

3. Click the **Update** option.

For details, see [Running Business Objects Reports](#).

#### 1.1.6 Home Pages and Portal Panes

The *home page* is the first page you see after you log in. Home pages contain specialized sections called *portal panes*. From portal panes, you can create and access various records and information that is important to you, such as:

- Your appointments
- Your projects
- Your documents
- News headlines, stocks, and weather



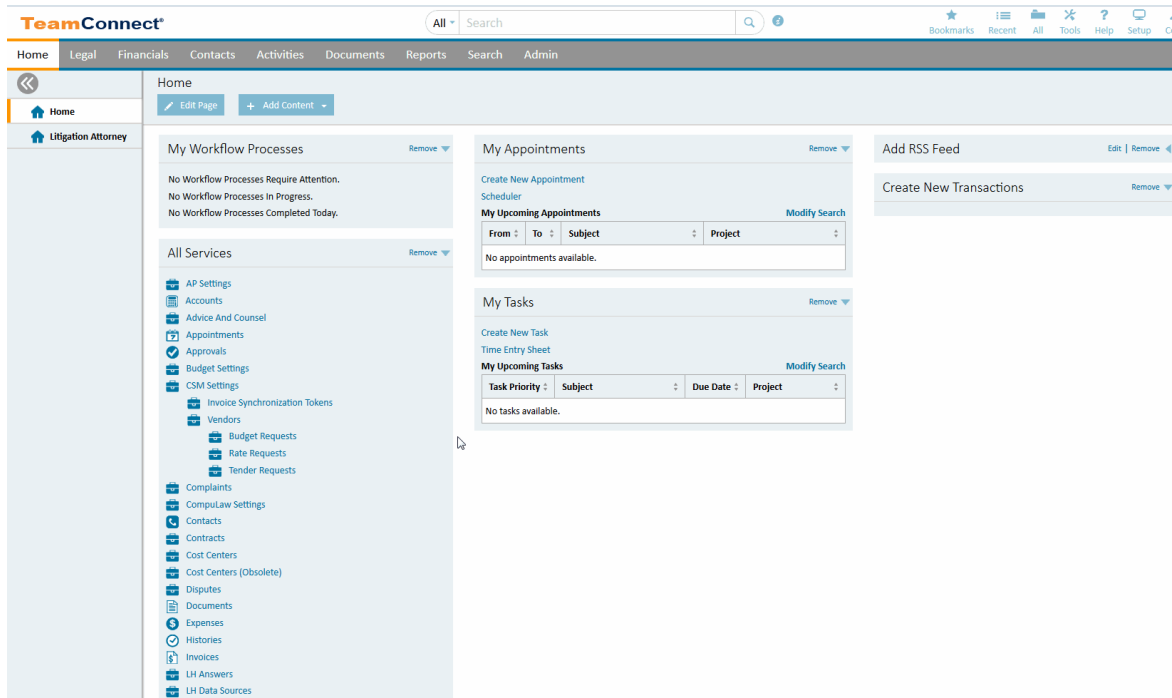
- Your pending approvals
- Your tasks
- Business Objects WebIntelligence reports
- Saved searches

The following table provides information about how to use home pages and portal panes:

	...Home Pages	...Portal Panes
Viewing...	<a href="#">Working with Home Pages</a>	<a href="#">Information in Portal Panes</a>
Adding...	N/A	<a href="#">Adding Portal Panes to Home Pages</a>
		<a href="#">Running Business Objects Reports</a>
Editing...	<a href="#">Modifying Home Page Settings</a>	<a href="#">Editing Portal Pane Custom Content</a>
	<a href="#">Customizing Portal Pane Layout</a>	<a href="#">Using Search Views in Portal Panes</a>

#### 1.1.6.1 Working with Home Pages

Your solution developer defines and creates portal panes and assigns them to the home pages of different user groups or to the home page of the entire organization. Depending on how many groups you belong to and how many home pages your solution developer created, you may have access to more than one home page. If such is the case, one of your home pages is set as the default and is the one that will always appear when you log in.



Home Page Example

**Note:** Your home page may look different based on your company's preferences.

#### Home Page Example

Component	Description
<b>Home Pages Menu</b>	<p>Click the title of the home page you would like to view. The home page you are currently viewing is displayed in bold in the <b>Home Pages</b> menu.</p> <p><b>Note:</b> You may have one home page, or multiple home pages to view.</p>
<b>Personalize Links</b>	<p>Clicking a link brings up a page where you can make changes to the home page title or you can add more portal panes to the page.</p> <p>For more details, see <a href="#">Home Pages and Portal Panes</a>.</p>
<b>Personalize Icons</b>	<p>Clicking an icon brings up an <b>Edit Settings</b> page where you can make limited changes to the title or the layout of portal panes.</p> <p>For more details, see <a href="#">Home Pages and Portal Panes</a>.</p>
<b>Portal Panes</b>	<p>Specialized menus from which you can access information and perform actions. Each portal pane is displayed with its own title</p>

	bar and contents.
<b>Portal Pane Content</b>	View the displayed content or click the displayed hyperlinks to access records, web sites, tools, and other options.
<b>Title Bars</b>	Click the available buttons to edit and change the properties of the portal pane. For details, see <a href="#">Customizing Portal Pane Layout</a> .
<b>Edit Page and Add Content</b>	These buttons allow for customization of your home page. The drop-down of the <b>Add Content</b> button will allow users to quickly and efficiently select which views are key for their home page. See <a href="#">Adding Portal Panes to Home Pages</a> for more information.

### 1.1.6.2 Information in Portal Panes

Most information and records that you can access from any of your home pages can also be accessed from the tab bar. However, home pages and portal panes provide a quicker means of doing so.

The [Home Page Example image](#) shows the different components that you may see on your home page. Some portal panes, like **Employee Messages**, contain important textual information, such as messages to employees or important announcements.

Other portal panes are more interactive. You can click hyperlinks to access saved searches, web sites and records, such as existing appointments, contacts, claims. There are also hyperlinks that open tools or record screens with blank fields so that you can create new appointments, contacts, tasks.

Certain portal panes that appear on your home page may contain hyperlinks to stocks, weather, or headline news.

Certain portal panes are designed to display search results for a specified type of record. For example, in the **My Appointments** portal pane, shown in the [Home Page Example image](#), the records displayed are filtered according to the specified search criteria. See [To use the portal pane title bar buttons](#) for details on how to change your search criteria.

The number of displayed search results is determined by your system administrator. If the number of search results exceeds the maximum number entered, you can always click the **More** hyperlink. All of the search results are displayed on a standard search screen.

**Note:** *In order to use the links that do searches, you must be subscribed to the object collections that are associated with the searches. (More information about subscribing to collections is in [Managing Collections](#).) Ordinarily this subscription is done for you automatically by the solution developers for your organization. However, if you manually unsubscribe from an object collection, then later click a search link that is associated with that collection, you will see an error message that says you must first subscribe to the collection before attempting to use the search link.*

Portal panes that contain graphic output from reports cannot "wrap" the graphic output as they can wrap text output. If the pane is too small, the graphic output is truncated. For this reason, consider the size of the graphic output when deciding how to size and place the portal pane in a home page.

### 1.1.6.3 Running Business Objects Reports

If your organization uses Business Objects WebIntelligence to generate reports based on the data you enter and generating reports is part of your job responsibilities, you can access WebIntelligence directly. This capability is known as single sign-on.

## Requirements

Make sure the following requirements are met before you can run Business Objects reports:

- Your TeamConnect or WebIntelligence administrator has configured the single sign-on capability and has informed you that it is ready to be used.
- If your username and password combination is different for TeamConnect and WebIntelligence, make sure you enter your WebIntelligence username and password in the appropriate fields on the **Miscellaneous Settings** tab of the **Preferences** screen. For details, see [Choosing Other Setting Options](#).
- Make sure your home page has a portal pane with hyperlinks to WebIntelligence reports.

This portal pane can be created by your solution developer.

## To access a Business Objects report

1. On your home page, find the portal pane with WebIntelligence hyperlinks to the reports that you want to generate.
2. Click the hyperlink to the report you need to access. Alternatively, click the hyperlink to the WebIntelligence home page.

Now your WebIntelligence password is verified. You should be automatically logged in to WebIntelligence. Otherwise, one of the following may take place:

- A dialog box appears, which requests you to enter your WebIntelligence username and password.
- An error is displayed that informs you that the single sign-on has failed. If this happens, note the information displayed on the error page.

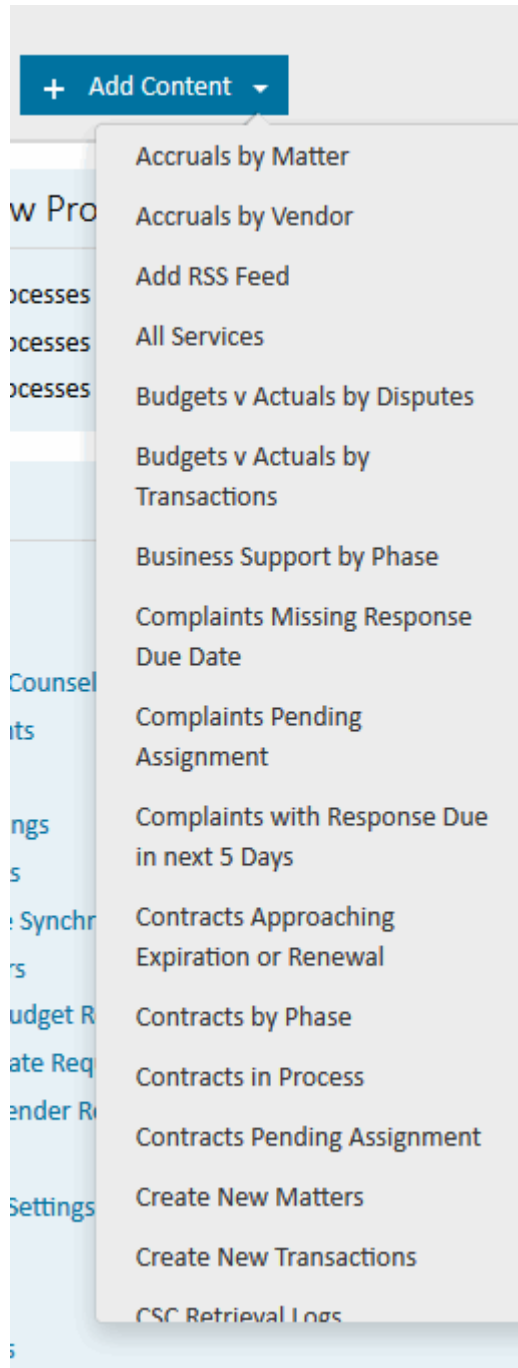
**Important:** *If you are not automatically logged in to WebIntelligence, single sign-on has failed. Check the [Requirements](#) listed above or contact your administrator.*

3. If presented with the dialog box, type your WebIntelligence username and password in the displayed dialog box and click **OK**.
4. If presented with an error, check the settings on the **Preferences** screen and try accessing the report again or click the hyperlink for the manual login. If the problem persists, contact your administrator.

#### 1.1.6.4 Adding Portal Panes to Home Pages

You can add content to your home pages by selecting portal panes from a pre-defined list. Click the **Add Content** link in the home page, which displays the **Add Content** dropdown.

The **Add Content** window displays all of the portal panes that are available to you to add to your home pages. You cannot create new portal panes yourself. Click the name of a portal pane, and it is added to your home page.



**Add Content Window**

### 1.1.6.5 Modifying Home Page Settings

Changes that you make to your home page will only affect your view of that home page, not the view of other users who may have access to the same home page. To better understand the relationship between your changes and the changes made by your solution developers, see [About Synchronizing Home Pages and Portal Panes](#).

#### 1.1.6.5.1 Page Title

You can modify your home page's title by clicking the **Edit Page** link in the home page. The current title appears in a small window and you can change it to a title of your choice. This does not change that home page's title for any other users; only for you.

#### 1.1.6.5.2 Customizing Portal Pane Layout

You can modify the existing portal panes that appear on your home pages. The title bar of a portal pane typically provides a set of buttons that enables you to customize your portal pane. Panes can be collapsed/expanded by clicking the carat arrow icon in the right corner of each respective pane.

#### To move a portal pane

While the cursor is inside the title bar of the portal pane, click and hold the left mouse button. Then drag the pane to another location on the home page and release the mouse button. The portal pane will appear in the new location, and other portal panes will move aside to make room for it in that location.

#### To use the portal pane title bar buttons

The following table describes the functionality of each of the buttons that you can select from the title bar of a portal pane.

Button	Description
<b>Edit</b>	If the portal pane contains custom content, and you have authority to modify that content, clicking Edit presents a page that lets you alter the parameters of that custom content. See <a href="#">Editing Portal Pane Custom Content</a> .
<b>Minimize</b>	Click to minimize the selected portal pane on the home page. The portal pane shrinks until only its title bar, not its content, is visible.
<b>Maximize</b>	Click to maximize the selected portal pane. After clicking this button, the portal pane will occupy its default size.
<b>Remove</b>	Click to remove the selected portal pane from the home page. A confirmation window appears. If you confirm the removal, the portal pane disappears.  This button will always be available in the title bar.

If you do not see the title bar buttons, or only a few buttons, then you are not allowed to change that portal pane.

Any customization made will affect only that portal pane in the home page from which you are currently working. If the same portal pane appears on more than one of your home pages and you would like to customize it across all of your home pages, you must open each home page and make the changes.

#### 1.1.6.5.3 Using Search Views in Portal Panes

A search view in a Portal Pane lets you view information that meets selected search criteria. For example, you may have a search defined that searches for, and then displays your upcoming appointments for the next 30 days, including the time of the appointment and the name of the associated project.

You can modify the search criteria and, if necessary, reset the criteria back to the default search view.

**Note:** Typically, the default search view criteria is set by an administrator. Any changes that you make are temporary and can be lost if the Home Page or Portal Pane is synchronized. See [About Synchronizing Home Pages and Portal Panes](#) for specific information.

### Modifying the Search View

1. In the appropriate portal pane, click the **Modify Search** link.

The search criteria screen opens in the Portal Pane and displays the current search criteria. If no criteria is defined, a "No visible criteria available" message is displayed.

Field	Operator	Value
Subject	Contains	
Appointment Date	In the Next	90 days
Attendee	Is	Hill, Rebecca Hill, Shane Hochstetler, Josh Hoffman, Christina Holden, Cody
Project	Equal To	All Projects
Category	Is	(Any) Appointment Appeal Arbitration

Search Criteria Screen in Portal Pane

2. Make the appropriate changes and then click **Search**. You are returned to the original screen in the Portal Pane.

**Reminder:** Any changes that you make are temporary and may disappear when Home Pages and Portal Panes are synchronized.

### Resetting to the Default Search View

Click **Reset to Default**, and then click **Search**.

### Canceling Changes

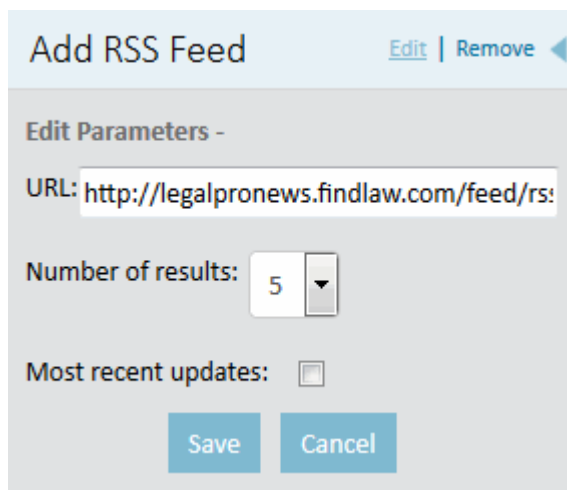
Click **Cancel** to cancel changes before saving.

#### 1.1.6.6 Editing Portal Pane Custom Content

Custom content in portal panes sometimes has parameters that determine what data is shown. You may be able to edit those parameters. For example, for a custom content portal pane that displays weather conditions, you may be able to change parameters such as the zip code of the area for which you want weather information.

To know whether there are editable parameters in a given portal pane, look for the **Edit** button in the title bar of the portal pane. If the **Edit** button is not visible, either there are no editable parameters or you do not have permission to edit parameters.

When you click the **Edit** button, the **Edit Parameters** window displays along with varying parameter fields, as shown in the following image. The original appearance of the portal pane is shown in the lower part of the window.



**Edit Custom Content Parameters Window**

The parameter fields in the **Edit Parameters** window vary depending on the type of custom content portal pane your administrator created.

1. Fill in the fields that are presented.
2. Click **Save**.
3. Changes that you make to the custom content parameters take effect immediately within the portal pane.



### 1.1.6.7 About Synchronizing Home Pages and Portal Panes

Because you may have the ability to customize your own home pages and portal panes, a distinction must be made between the changes that you make and those made by the administrator.

#### Home Page Personalization

If you have the right to personalize home pages, changes that you make to the **Page Settings** take precedence over solution developer changes.

However, if the solution developer adds portal panes to a home page, you will always see these changes. Even if you add a portal pane to a home page by clicking the **Add Content** link from the page, and then the solution developer adds the same portal pane to the same home page, you will see two of the same portal panes on the home page. If desired, you can then remove the duplicated portal pane yourself.

#### Portal Pane Customization

If you have the right to customize a portal pane (see [Customizing Portal Pane Layout](#)), changes you make take precedence over administrative changes.

For example, if you click the **edit** icon on a portal pane and customize that portal pane, the changes you made appear on your home page. Changes made by the solution developer to that same portal pane do not appear on your home page. In short, the moment you customize a portal pane, those changes take precedence over solution developer changes to that same portal pane. But if you do not edit a portal pane, then solution developer changes take precedence.

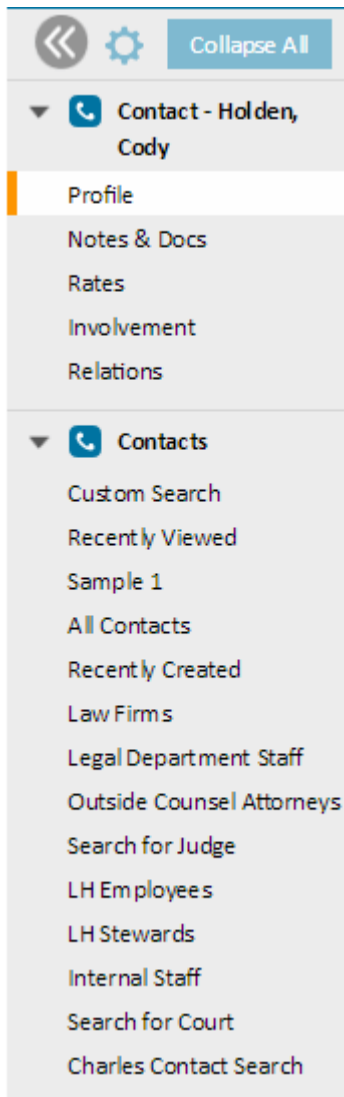
#### Home Page and Portal Pane Synchronization

Although you might have the ability to personalize your home pages and customize your portal panes, your administrator can overwrite those changes at anytime. Solution developers have the ability to synchronize home pages and portal panes contents when necessary. Solution developers are advised to notify users before synchronizing home pages or portal panes.

## 1.2 Records

Large organizations may have millions of records in their databases to store information.

A single record contains information that displays on several separate pages, shown by the links in the left pane.



**Links for Each Page in the Left Pane of a Contact Record**

**Note:** You must have the appropriate access rights to perform certain tasks. If you don't have rights, you will not have access to the buttons, icons, or links that allow you to access part of a record. For more details about how rights work, see [Working with Record Security](#).

The following table shows how you can use most records.

...Records	How...
Finding	<a href="#">Displaying Types of Records</a>
	<a href="#">Viewing Collections of Records</a>

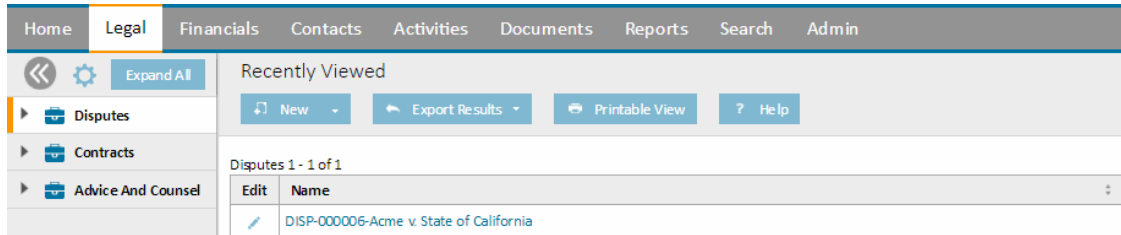
		<a href="#">Searching for Specific Records</a>
		<a href="#">Sorting Information</a>
Viewing		<a href="#">Using a Record in View Mode</a>
Creating		<a href="#">Creating a Record</a>
		Using <a href="#">Wizards</a>
		<a href="#">Importing Data to Create Records</a>
		<a href="#">Working with Batch Entries</a>
Editing		<a href="#">Editing a Record</a>
		<a href="#">Saving an Edited Record</a>
		<a href="#">Copying a Record</a>
		<a href="#">Reassigning Work</a>
Adding	Categories	<a href="#">Setting Category Details</a>
	Security	<a href="#">Working with Record Security</a>
	History	<a href="#">Working with Record History</a>
	Relations	<a href="#">Working with Related Records</a>
	Documents	<a href="#">Working with Record Documents</a>
	Workflow	<a href="#">Viewing Workflow for a Record</a>
Deleting		<a href="#">Deleting Records</a>
Printing		<a href="#">Printing Records</a>

## 1.2.1 Displaying Types of Records

Records are categorized by record types, which handle the storage of different kinds of business information. For example, Contacts, Invoices, and Appointments are all considered record types.

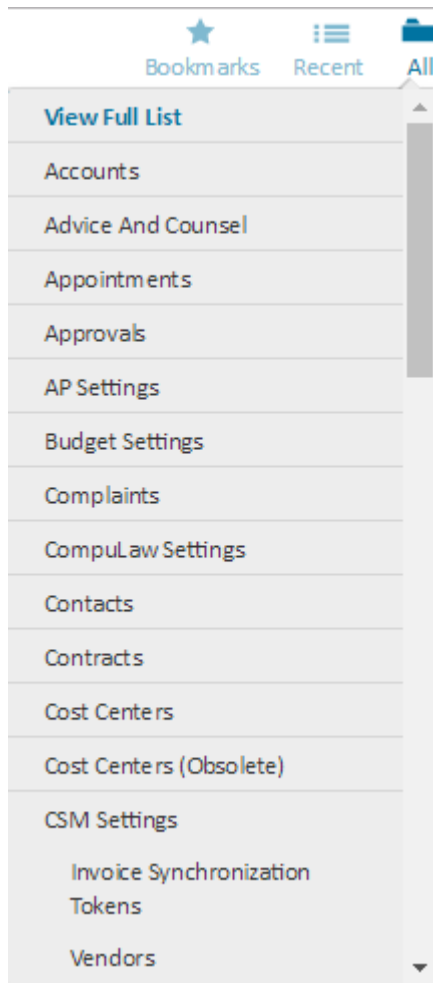
You can access record types in one of the following ways:

- Click a link on the global navigation bar—The global navigation bar shows a row of tabs that represent some of the main pages. When you click on a tab, such as **Contacts**, a page opens that lists contact records that you have the rights to view. Others, such as **Finance**, display a group of sub-tab links just below the main tabs. Clicking one of these links takes you to a page that displays a list of the selected record type. The record list is generated from the Recently Viewed search (see [Viewing Collections of Records](#)).



**Tabs and sub-tab links on the Global Navigation Bar**

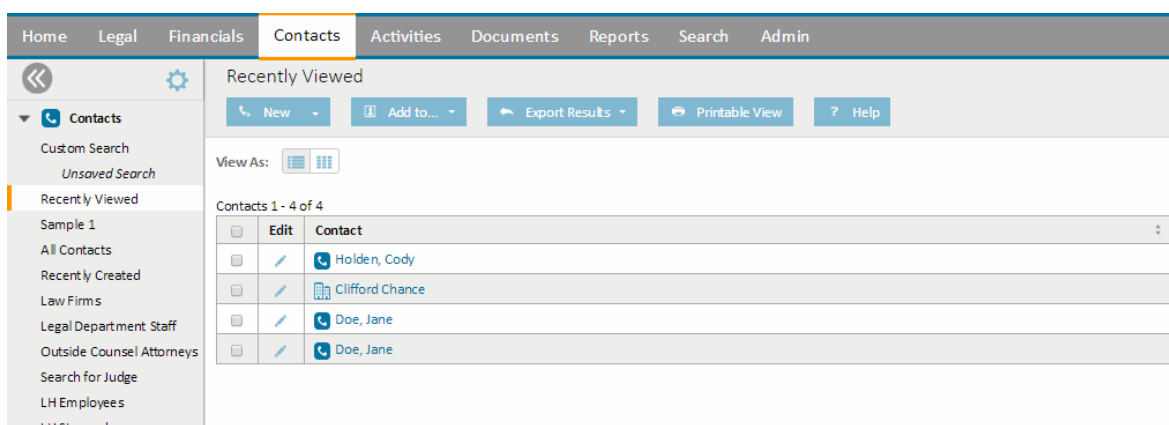
- Click a link from the **All** button—The **All** button drop-down list provides access to all the standard record types in addition to any custom record types your organization may have developed. The **All** button also includes links to any tools and home pages to which you have access.



All Drop-down List

## 1.2.2 Viewing Collections of Records

A collection is a group of related records that you can view by clicking a collection link in the left pane. Use collections to access records without having to search for them.



The Contact Record Type with Many Collections

You can define many collections for a single record type. In the graphic above, all of the links in the left pane, such as Companies, People, and Leigh's address book, are collections. Each collection can use a different combination of search criteria to present a different list of records.

By default, most record types have one or more of the following collections:

- **Recently Viewed**—Displays up to 25 of the last records that you viewed, in reverse chronological sequence, over a period of 90 days. This link is always the first collection listed in the left pane, and you cannot delete this collection or change its order.
- **All <record type>**—Displays all of the records of the selected type that you have the rights to view.
- **Recently Created**—Displays a list of records of the selected type that were added to within the last 30 days.
- **Recently Modified**—Displays a list of records of the selected type that were edited within the last 30 days.

Although **Recently Viewed** is the default collection, you can click another collection during your session and that collection will be remembered for the duration of the session. For example, if you leave a record type, do work in another area, and then later return to that record type, the collection you last used for that record type will be the one that appears now.

In addition to the collections listed above, some record types also have specific collections as follows:

#### Specific Collections by Record Type

Record Type	Specific Collection
<b>Calendar</b>	<b>Upcoming</b> —Displays a list of appointments that will occur 90 days from today's date.
<b>Invoices</b>	<ul style="list-style-type: none"> <li>• All Invoices</li> <li>• Not Posted</li> <li>• Approved</li> <li>• Pending</li> <li>• Rejected</li> <li>• Failed</li> </ul>
<b>Projects</b>	<b>My Assigned</b> —Displays a list of the projects to which you are assigned.
<b>Tasks</b>	<ul style="list-style-type: none"> <li>• My Pending Tasks</li> <li>• My Overdue Tasks</li> <li>• My Completed Tasks</li> <li>• My Billed Tasks</li> </ul>


### 1.2.2.1 Managing Collections

The collections that you see in the left navigation pane are initially determined by your system administrators, but you can also affect which ones are shown by subscribing or unsubscribing to collections.

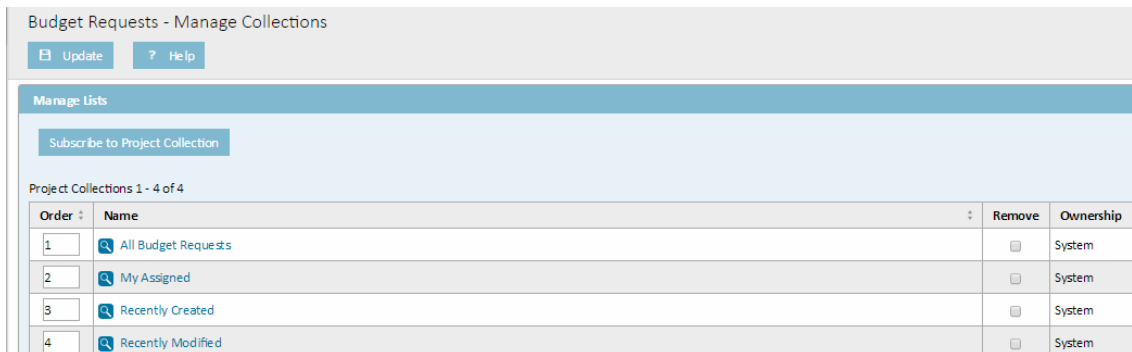
On the **Manage Collections** page, you can do the following:

- View the collections to which you currently subscribe
- Unsubscribe from a collection
- Subscribe to a collection
- Reorder the sequence of collections

#### To open the Manage Collections page

In the left navigation pane, click the gear icon . Select the desired collection from the resulting menu.

**Note:** For the Contact record type, collections may also include "Address Books". For all other record types, the term "Collections" is used. Regardless of the term, managing collections is done in the same way for all record types.



Order	Name	Remove	Ownership
1	All Budget Requests	<input type="checkbox"/>	System
2	My Assigned	<input type="checkbox"/>	System
3	Recently Created	<input type="checkbox"/>	System
4	Recently Modified	<input type="checkbox"/>	System

Example of the Manage Collections Page

#### To subscribe to a collection

Click the link **Subscribe to <record type> Collection**.

#### To unsubscribe from a collection

Click the **Remove** check-box next to the collection that you want to delete, and then click **Update**.

#### To reorder the sequence of collections

1. Type sequence numbers into the **Order** field in each row. Collections with the smallest numbers will appear first in the left navigation pane. You cannot reorder the **Recently Viewed** collection—it will always appear before any other collections.

You cannot change the order of the **Recently Viewed** collection. It is always the first collection listed.

2. Click **Update**.

#### Manage Collections Page

View	Description
<b>Subscribe to &lt;record type&gt; Collection</b>	Click this link to open the <b>Subscribe to Collections</b> page. Select a collection on that page and click <b>Subscribe</b> .
<b>Order</b>	Specifies the order, starting with 0, that you want the collections to appear on the left side. Change the numbers in the text boxes to update the order.
<b>Name</b>	Displays the name of the collection. Click a link to open a collection.
<b>Remove</b>	Select a collection to remove it from your collections list.
<b>Ownership</b>	Specifies the owner of the collection.

### 1.2.3 Searching for Specific Records

Besides the automatic Recently Viewed search that runs when you begin working with a record type, there are several other ways to search for records:

- ElasticSearch Global Search
- [Modifying a Collections Search](#)
- [Using Custom Search](#)
- [Using Search Modules](#)

All pages that list search results also allow you to specify how many records you want to see on a single result page, by choosing a number from the "per page" drop-down list at the bottom of the results. If you choose a number during one search, your choice is remembered and used in any other searches you may run for that record type.

**Note:** The actual name of this drop-down will vary depending on the record type you are viewing. For example, if you are viewing Document records, the name of the drop-down will be "Documents per page".



If you have already run a Global Search or a Custom Search, the results of that search are available from the **Search** link, found at the bottom of the list of collections in the left pane. If you click away from your search results to a different tab or a different collection, clicking the **Search** link will restore your original search results, without the need to rerun your search.

**Note:** You can export the search results into an Excel worksheet or create a .pdf document that you can open with Adobe Acrobat Reader. See [Exporting a Search Results List](#) for more information.

### To navigate in long lists

Some lists, such as those produced by searching, might need more than one page to be displayed. In this case, a series of links will appear at the bottom of the list page. You can specify a particular page to view (for example, page 4 of 9) by clicking the page number, or you can click the left-arrow link or the right-arrow link to move from page to page. The **Up** arrow icon in the right corner can be used to go "back to top" without scrolling.

<input type="checkbox"/>	<input checked="" type="checkbox"/>	Bailey, Frank			
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Bailey, Grace	Baltimore	Maryland	410 244 7400
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Bailey, Kathleen			(310)-555-0468
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Bailey, Micheal	Los Angeles	California	1 213 485 1234
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Baker, Eric			(310)-555-0978
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Baker, Irene			(310)-555-0862

1 2 3 4 5 ▶

Contacts per page: 200

Version 5.0 UI Build 117 | Copyright © 1999-2016 Mitratech Holdings, Inc. All Rights Reserved.

#### Navigation Links for Long Lists

### 1.2.3.1 Global Search

Users with TeamConnect 4.1+ are able to integrate Elasticsearch for more robust global searching (including searching on attachments). For more detail on this global search, please view [TeamConnect Search Help](#).

### 1.2.3.2 Modifying a Collections Search

The record(s) you are seeking might be present in a collection other than the one you are viewing. You can choose a new collection by clicking on a link in the left navigation pane of a page. Clicking a collection link will produce one of two possible pages:

- If the collection has its "Auto-search" flag turned on, then the search is immediately executed and a search results page is shown. The search results page has a Modify Filter link which, if clicked, will show the Modify Filter page.
- If the collection has its "Auto-search" flag turned off, then the Modify Filter page appears, showing the search criteria and results columns that are associated with the collection.

From the **Modify Filter** page, you can edit the search criteria that are associated with the collection, then run the edited search. If you wish, you can save the edited search, as described in [Saving the Search](#).

If you are viewing a list of search results, then the last link in the left navigation pane is titled **Search**. Clicking that link will re-execute your most recent search and will refresh your list of search results with the most current data.

### 1.2.3.3 Using Custom Search

Each record type has its own dedicated list page, which has a link for **Custom Search**. When you click **Custom Search** to get a new search screen, all of your previous modifications to criteria and result fields are discarded and a search page opens.

Along with the **Add Criteria** and **Add Another Column** buttons, users can add fields by clicking the green plus icon. Likewise, fields can be removed from the red minus icon.

Fields in the Results Display can be reordered by clicking the up and down carat icons on the right side of the row.

Field	Operator	Value	Action
User Name	Contains		⊖ ⊕
Account is Active		Yes	⊖ ⊕
Description	Not Equal To		⊖ ⊕

Order	Field	Action
1	Name	⊖ ⊕ ↕
2	Username	⊖ ⊕ ↕
3	Created Date	⊖ ⊕ ↕

#### Custom Search for the Contact Record Type

##### 1.2.3.3.1 Specifying Search Criteria

You specify search criteria by selecting options on the Custom Search screen. Default criteria are provided, but it is not necessary to enter values in every field. If you do not enter any values, and you clear any existing default values, then click the **Search** button, Custom Search returns all of the records in your database, which could be a very large result set. Selecting options on this screen will narrow your search results to a more manageable result set.

**Note:** For custom objects, the criterion "Main Assignee" has a default value equal to the logged-in user. If you want your search to include more records than this, be sure to alter or clear this criterion.

## Filtering Criteria

There are three options for filtering your search criteria.

When you select radio button **All of the following**, every criterion that you specify must be satisfied for a record to appear in the search results. In this case, adding more criteria makes your result set narrower (smaller).

When you select **Any of the following**, a record needs to satisfy only one of the criteria to appear in search results. In this case, adding more criteria potentially makes your result set larger, increasing the time it takes to return the results.

The **Group the following** radio button combines the functions of the two previous radio buttons. When you select **Group the following**, a new **Group** column appears in the table. Search fields with the same **Group** values will only produce results if all the group's criteria are met, similar to selecting the **All of the following** radio button. Qualifying search results from each separate group (designated by matching numerical values) will all appear in the search results, similar to the **Any of the following** radio button.

## Selecting Criteria

To select criteria, complete one or more of the following:

- **Group**—This column only appears if you select the **Group the following** radio button. Group your criteria together by giving them matching values, separating the groups with unique values. Group values can be numbers 1-9 only. Alphabet characters, special characters, and 0 are invalid values. There is no limit to the amount of criteria allowed in a single group.
- **Field**—Select the name of the field that you want to search. This drop-down list contains all of the field names associated with this record type.
- **Operator**—Select the option that describes the criteria that you want to use to further narrow your results, such as **Contains** or **Has Value**. See [Operators](#) for more information.
- **Value**—Enter the specific value that you want to use to return specific results, based on the Field and Operator that you selected, if available. For example, if you want to find all of the users with the last name Smith, in the Field list, select Contact Name, in the Operator list, select Contains, and in the Value field, enter Smith.

## Selecting Multiple Values

For text fields, user fields, number fields, and projects, you can search using multiple values by clicking the plus sign in the **Value** area to add additional value fields. For example, if you want to find all of the projects modified by Deby Howard and Kevin Ranch, in the **Field** list, select **Modified By**, in the **Operator** list, select **Equal To**, and in the **Value** drop-down list, select a project type. Enter Howard, Deby in the first box, click the plus sign and enter Ranch, Kevin in the second box. The search results will contain all of the projects that were modified by these users.

For specific information about which Operators you can use when using multiple-value options, see [Operators](#).

**Note:** For custom objects, there is a default criterion of "Assignee" equal to the logged-in user. If you want your search to include more records than this, be sure to alter or clear this criterion.

#### 1.2.3.3.2 Operators

Use the information below to help you select the appropriate operators.

### Text Fields

For text fields, there are several operators:

<b>Begins With</b>	The text string that you specify must be present at the beginning of the record's related field. Any additional text beyond that string is not considered part of the search.
<b>Equal To</b>	The text string that you specify must be exactly the same as the record's related field.
<b>Not Equal To</b>	The text string that you specify must not be the same as the record's related field.
<b>Contains</b>	The text string that you specify must be present somewhere in the record's related field, not necessarily at the beginning or the end.
<b>Does Not Contain</b>	The text string that you specify must not be present somewhere in the record's related field.
<b>Ends With</b>	The text string that you specify must be present at the end of the record's related field. Any additional text before that string is not considered part of the search.
<b>Has No Value</b>	The specified field does not contain a value.  You might use the <b>Has No Value</b> operator if you are searching for records where the selected field does not contain a value, such as contact records that do not have a value in the <b>Social Security</b> field.
<b>Has Value</b>	The specified field contains a value.  The <b>Has Value</b> operator is useful if you are searching for records where the selected field contains a value, such as contact records that provide a value in the <b>Job Title</b> field.
<b>Before</b>	The search will return all records that appear alphabetically before the text string that you specify.

<b>Before (Including)</b>	The search will return all records that appear alphabetically before the text string that you specify, including the text string that you specify.
<b>After (Including)</b>	The search will return all records that appear alphabetically after the text string that you specify, including the text string that you specify.
<b>After</b>	The search will return all records that appear alphabetically after the text string that you specify.

*Note: In all of the options above, case is ignored when testing for matching characters.*

If you are searching using multiple options in the **Value** area, you can use **Begins With**, **Ends With**, **Equal To**, and **Contains** only.

## Date Fields

When you place the cursor inside a date field, a popup window appears so that you can select a date from the calendar.

For date fields, you can choose from several operators to search for dates:

<b>Older Than</b>	Days on and before the date you enter.
<b>Newer Than</b>	Days on and after the date you enter.
<b>Between</b>	A range of dates, including the dates you enter. To search for one date, you can leave one of the fields empty or use both fields for the same date.
<b>Has No Value</b>	The date field does not contain a value.
<b>Has Value</b>	The date field contains a value.

## Time Fields

Time fields allow you to specify an exact time, to the minute and provides a time icon you can click to open a list of possible times of day. For convenience, you can click on an entry in the list rather than typing the time value manually.

## Number Fields

For number fields, there are several operators:

<b>Equal To</b>	The number that you specify must be exactly the same as the record's related field.
-----------------	---

<b>Greater Than</b>	The number in the record's related field must be greater than the number that you specify.
<b>Greater Than or Equal To</b>	The number in the record's related field must be greater than, or equal to, the number that you specify.
<b>Less Than</b>	The number in the record's related field must be less than the number that you specify.
<b>Less Than or Equal To</b>	The number in the record's related field must be less than, or equal to, the number that you specify.
<b>Between X and Y (Inclusive)</b>	When you choose this option, a second input field appears next to the existing input field. You must type a number in the existing input field and a second, higher number in the second input field. The record's related field must have a value that lies between the two numbers that you specify, or matches one of the two numbers that you specify.
<b>Has No Value</b>	The specified field does not contain a value.
<b>Has Value</b>	The specified field contains a value.

### User Fields

<b>Equal To</b>	The results must contain the specified users. This operator is only available if you are using multiple options in the <b>Value</b> area.
<b>Not Equal To</b>	The results must not contain the specified users. This operator is only available if you are using multiple options in the <b>Value</b> area.

### Memo Text Fields

For Memo text boxes, you can use the **Contains**, **Has No Value**, and **Has Value** operators.

*Note: Searching memo text fields with the **Contains** operator only supports alphanumeric characters.*

### List Fields

For Lists, you can use the **Is**, **Is Not**, **Has No Value**, and **Has Value** operators.

### Multi-Value List Fields

For multi-value list fields, there are several operators:

<b>Contains Any</b>	Any of the values that you specify must be contained in the saved values of the multi-value list field.
<b>Contains None</b>	None of the values that you specify must be contained in the saved values of the multi-value list field.
<b>Contains All</b>	The full set of values that you specify must be contained in the saved values of the multi-value list field.
<b>Does Not Contain All</b>	The full set of values that you specify must not be contained in the saved values of the multi-value list field.
<b>Contains Exactly</b>	All of the values that you specify must match exactly to the saved values of the multi-value list field.
<b>Does Not Contain Exactly</b>	All of the values that you specify must match exactly to the values that are not saved in the multi-value list field.
<b>Has Value</b>	The specified multi-value list field contains a value.
<b>Has No Value</b>	The specified multi-value list field does not contain a value.

## Project Fields

For Projects, you can use the **Equal To**, **Not Equal To**, **Has No Value**, and **Has Value** operators. If you search using multiple value criteria, you can use the **Equal To** operator only.

## Object Fields

For object fields, you can use the **Equal To**, **Not Equal To**, **Has No Value**, and **Has Value** operators.

### 1.2.3.3.3 Adding Additional Fields

If you want to use a criterion that isn't already visible as part of the default, click the link **Add Criteria** or click the green plus icon at the end of the row (see image below). An extra row is added to the criteria section, and a drop-down in that row lets you choose a criterion from a list of all possible criteria for this record type. The drop-down is organized with all the system fields at the top of the list, and all the custom fields at the bottom of the list. The custom fields are grouped by category, and each category displays its title above its custom fields. In the example below, the Invoice record type has a **Terms** category, with a detail **Net Terms (days)** in that category.

The screenshot shows the 'Filter Criteria' section of a software interface. At the top, there are three radio buttons for search logic: 'All of the following' (selected), 'Any of the following', and 'Group the following'. Below this is a table with columns for 'Field', 'Operator', 'Value', and 'Action'. The first row has 'User Name' in the Field column, 'Contains' in the Operator column, and an empty text box in the Value column. A second row is partially visible with '(Select)' in the Field column. A third row shows a search for 'Account is Active' with a dropdown menu open below it, listing various fields: '(Select)', 'Account is Active', 'Contact', 'Created By', 'Created Date', 'Description', 'First Name', 'Last Name', 'Modified By', and 'Modified Date'. The 'Action' column contains icons for adding, removing, and sorting criteria.

#### Custom Fields Drop-down List

If you don't want to use a criterion, you can remove it using the "minus" icon at the right of the criterion's row. Or you can simply not enter a value for that criterion, and it will be ignored when the search is run.

**Note:** If you enter invalid data in a search criterion field (such as entering text characters in a numeric field), an error message is generated after you click **Search**.

The values that you type in the Custom Search page are used to filter the search results. Search is flexible, and can do more than look for a simple equality between your values and a record's values. Listed below are the possible types of matches.

### Links to other records

You can search based on a record's link to another related record. For example, you can search for Invoice records that have a specific Timekeeper value in a Line Item record, or that do not have that specific value. "Equal To" and "Not Equal To" are the only possible choices in drop-down lists for such links.

### Other Criterion Features

You can erase all the criterion values that you have entered by clicking the link **Clear Values**.

You can restore the list of criteria to their default values by clicking the link **Reset to Default**. This will also reset any changes that you have made to results columns, which are described in the next topic.

#### 1.2.3.3.4 Specifying Results Columns

By default, Custom Search shows you a list of fields that will be present in the search results list display. The first entry shown will appear on the left of the search results list display; the last entry shown will appear on the right. You can modify this list:

- Change the names that appear in the "Field" drop-down lists. In this way you can, for example, include the same fields as the default but in a different order, left to right.
- Remove a field from the search results list display using the "minus" icon to the right of the field name.
- If you remove a field you were using to sort information, the search results continue to sort in the original order, minus the removed column. See [Multi-column Sort](#).



- Put a new field in the search results list display. Do this by clicking the link **Add Another Column** or by clicking the "plus" icon to the right of a field name. A new row appears in the Column Display section for the search definition. A drop-down list in that row lets you choose which of all possible fields, for the current record type, should be used as a search results field. System fields appear first, followed by custom fields, which are grouped by category.

#### 1.2.3.3.5 Executing the Search

Click the **Search** button to run the search using the criteria and display your results.

Edit	Name	Username	Created Date
✓	System Admin	systemadmin1	3/2/14
✓	Adames, Alex	Alex.Adames	2/19/16
✓	Adams, Sharon	tranattorney7	3/2/14
✓	Alberico, Chris	Chris.Alberico	2/19/16
✓	Alex, Akca	Akca.Alex	2/19/16
✓	Allen, Mark	bsattorney13	3/2/14
✓	Allibhoy, Zuber	Zuber.Allibhoy	2/19/16
✓	AlZubaidi, Suhair	Suhair.AZubaidi	2/19/16
✓	Amores-Villalobos, Amber	Amber.Amores-Villalobos	2/19/16
✓	Anderson, Jose	bsattorney33	3/2/14
✓	Approver, Legal Hold	tcjh_approval	3/4/14
✓	Arcangel, Hilda	Hilda.Arcangel	2/19/16

Search Results Page

**Note:** If your search produces no results, a **Modify Search** link appears. Click this link to revise your search criteria.

The **Search** link in the left pane also executes the search that you have just created and displays a list of results that meet the current criteria. This link executes the search for the duration of your session, unless you change the search criteria for the record type. Click **Custom Search** to change the search criteria.

#### 1.2.3.3.6 Saving the Search

When you save a search, you are saving the search's criteria, sorting information, and results display layout.

If you frequently search within a record type using a set of specific criteria, you can speed up your work by saving your search and giving a name to the saved search. That name will appear as a link in the left pane of the page, in the collections list.

A search can't be saved until it has been executed. After the search results are displayed, click the Save Search link.


- A suggested name is displayed; you can edit the name if you wish.
- Check-box Display Results Automatically, when checked, will cause the search to be executed immediately whenever you click on its name in the link in the left pane. If the box is not checked, the search criteria will be displayed instead, giving you an opportunity to review or modify the search before you execute it.

When you click Save Search, your custom search is saved and its name appears in the left pane. If a search already existed with that name, it is overwritten by the one you just saved.

Later you can re-run the saved search simply by clicking its name in the link in the left pane. Each time you click on the link in the left pane, the search criteria are freshly evaluated and the most current results will be displayed when the search executes.

**Note:** *When you save a search, you are not saving the list that the search results return. If you want to save the list of search results as a .pdf or an Excel worksheet, use the Export Results feature, as described in [Exporting a Search Results List](#).*

#### 1.2.3.4 Using Search Modules

You can search for a record while editing a record, if the record being edited displays a search module. For example, when editing a Contact record of type Person, a search module for Company is displayed. Clicking the **Search** icon  invokes a search window that shows Contact records of type Company.

Click the name of a record that is displayed by a search module to select that record and link it to the record that you were editing, and to close the search module window.

Some search modules have a special feature: **auto-suggest**. After you have typed a few characters of text into the name field, even before you click the Search icon, a list of up to ten records with names that match the text is displayed next to the module. You can click an entry in the list to choose that record. Record types that use auto-suggest include Contact, Account, User, and Project.

**Note:** *Auto-suggest is only a convenience feature, not a comprehensive search. It lists up to a maximum of ten records, even if there are more than ten records that match the characters that you typed. Typing additional characters will narrow your search and improve the relevance of the displayed list. Also, the sequence in which those ten records are displayed may differ from the sequence shown if you click the **Search** icon. Clicking the **Search** icon will return all results, not just ten.*

#### 1.2.3.5 Exporting a Search Results List

You can export information from a search results list to an Excel spreadsheet or create a .pdf file that you can open with Adobe Reader. The exported results include the headings, columns, and contents.

Additionally, you can export a list of the contents in a Documents folder. See [Exporting a List of Documents in a Folder](#) for more information.

##### To export a search results list

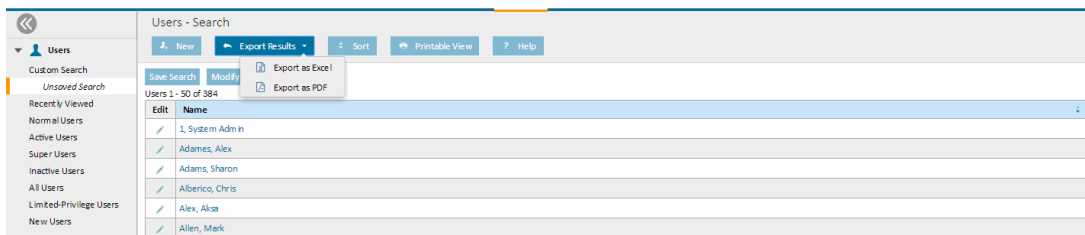
1. Click the tab for the appropriate object.
2. If necessary, click the appropriate link under the tab bar.  
The list view opens for the selected record type.
3. Select one of the following options:

- In the record list view, click the **Custom Search** link, enter your search criteria, and then click **search**.
- Click a collection link in the left pane, click **Custom Search**, enter your search criteria, and then click **search**.

The results appear in a list view.

4. Click **Export Results**, and then select one of the following options:
  - Export as Excel
  - Export as PDF

When you export search results that have fields with multiple values, the Excel spreadsheet or .pdf file displays all values, not just the first 10.



Export Results Drop-down Menu

5. When the File Download window opens, select one of the following options:
  - **Save**—Saves the file to the selected location without opening it.
  - **Open**—Immediately opens the file in the appropriate application.

If you select this option, you must save this file if you want to access it later.

If you are exporting localized text for Chinese, Japanese, or Korean, you must have the appropriate font packs installed for Adobe Reader. If you do not have the font packs, when you try to open the .pdf, a message appears prompting you to go to the Adobe site and download them. Once you download and install the appropriate font packs, you can view the .pdf in your preferred language.

## 1.2.4 Sorting Information

You can sort information in most list views, document lists, and groups of line items on an invoice. You can sort a list of records by a single column in ascending or descending order, or you can select columns by which to sort the information.

**Note:** *In a few places the list is pre-sorted and cannot be altered by clicking on column headers. Such places include the workflow displays (Active Requests, Completed Requests, etc.) for various record types.*

If you want to preserve the sorted information, export the search results to a different format. The sorted information only lasts until you close the page. Once you close a page, the information reverts to its default order. See [Exporting a Search Results List](#) for more information.

### 1.2.4.1 Single-column Sort

You can sort the list of records into ascending or descending order by clicking on the header or arrows for a column. The list is sorted by the information in that column only.

**Note:** As an alternative to clicking, you may use the **Tab** key to navigate to a column header, then press **Enter** to sort.

<input type="checkbox"/>	Edit	Contact Name ▾	City ▾	State/Province ▾	Phone Number ▾	Birthday ▾	Category ▾	Contact Type ▾	Created Date ▾	Nickname/Alias ▾
<input type="checkbox"/>		1, System Admin				7/14/16	Contact	Person	3/24/14	
<input type="checkbox"/>		10th Circuit Court of Florida [Civil]	Tallahassee	Florida	(850) 488-0125		Court	Company	11/10/14	
<input type="checkbox"/>		10th District Court of Appeals [Civil]	Waco	Texas	(254) 757-5200		Court	Company	11/10/14	

List View - Single Column Sort

### 1.2.4.2 Multi-column Sort

You can also sort by up to five columns on the same page.

For example, for a large group of line items on an invoice, you can execute a search that groups all Fees and Expenses together, and then sort the associated Codes in alphabetical order.

#### To sort by multiple columns

1. Open the record or page that contains the information that you want to sort.
2. Select one of the following options:
  - **Sorting a list view**—To sort a list view, click the **Sort** button on the toolbar.

<input type="checkbox"/>	Edit	Contact Name ▾	City ▾	State/Province ▾	Phone Number ▾	Birthday ▾	Category ▾	Contact Type ▾	Created Date ▾	Nickname/Alias ▾
<input type="checkbox"/>		1, System Admin				7/14/16	Contact	Person	3/24/14	
<input type="checkbox"/>		10th Circuit Court of Florida [Civil]	Tallahassee	Florida	(850) 488-0125		Court	Company	11/10/14	
<input type="checkbox"/>		10th District Court of Appeals [Civil]	Waco	Texas	(254) 757-5200		Court	Company	11/10/14	

List View - Sort Button

- **Sorting a documents list**—To sort a **Documents** page for a record, open a record, and then click the **Documents** link in the left pane. Click the **Sort** button on the icon bar. This icon is available in View and Edit mode.



Documents Page for a Record - Sort Button

- **Sorting invoice line items**—To sort a list of invoice line items, click the **Sort** button in the **Line Items** section.



Invoice General Page - Sort Button for Invoice Line Items

Regardless of the button or icon you select, the **Sort Columns** window opens.

Sort Columns Window

3. In the **Sort By** drop-down list, select the first column by which you want to sort information, and then click **Ascending** or **Descending** for the selected column.
4. In the **Then Sort By** drop-down list, select the second column by which you want to sort information, and then select **Ascending** or **Descending** for the selected column.
5. Repeat Step 4 for up to five columns.

You cannot sort spanned columns and they are not available as options in the sort drop-down lists. Spanned columns appear as horizontal rows in the results display.

6. Click **Sort**.

You are returned to the original page and the information is sorted based on your column selections. Unless you save the search, the sorted information lasts until you log out.

When you perform a sort, the layout of the page does not change; only the order of the information in the selected columns. If you save the search, then you remove a sorted column from the search results display, the search results continue to sort in the order you specified, minus the removed column. See [Saving the Search](#).

## 1.2.5 Using a Record in View Mode

To open a record, click on the link that shows the record name.

Most records have a toolbar, title bar, and access information.

### Using the Record Toolbar

All record screens typically have a separate toolbar with a set of buttons that allow you to edit, modify, copy, and delete the selected record.



The General Page for a Contact Record

The [Record Page Toolbar Buttons](#) table explains the functionality of each button. There are, however, a few key points that you should know about all of these buttons:

- The buttons that you see are controlled by the rights you have to the selected record type.  
For example, if you do not see the **Delete** button in an Appointment record, this means that you do not have the right to delete Appointment records in general or in this specific record.
- The current state of a record may prevent you from seeing some buttons.  
For example, you do not see the **Delete** button in an Expense record that was posted to an Account. Or you are unable to edit a record if it, or its parent project, is pending approval and the approval rule prevents anyone from modifying the record.
- Regardless of your access rights to the record, you always see the **Printable View** button on the toolbar.
- If you have the rights to create new records, you always have the **Create a Copy** button available in all records, except projects. In projects, access to this button is controlled by your administrator.
- Embedded projects have two separate buttons, **OK** and **Cancel**.

#### Record Page Toolbar Buttons

Button text	Description
<b>Save</b> <b>Save &amp; Close</b> <b>Save &amp; View</b> <b>Save &amp; New</b>	Saves the current record and may take additional action after the save. For more details, see <a href="#">Saving an Edited Record</a> . <b>Note:</b> Administrators can customize the default selection for this button as well as the cancel button, as described in <a href="#">Setting up the User Interface</a> .
<b>Cancel</b> <b>Cancel &amp; Close</b> <b>Cancel &amp; View</b>	<b>Cancel</b> —Cancels and returns to the previous page if you are creating a new record or creating a new Contact record on a search page. <b>Cancel &amp; Close</b> —In edit mode, cancels all record changes and returns the user to the previous screen. <b>Cancel &amp; View</b> —In edit mode, cancels the record changes and returns the user to view mode.

<b>Create a Copy</b>	Click this button to create a copy of the currently open record. See <a href="#">Copying a Record</a> .
<b>Share</b>	Click this button to bring up a window that provides a document URL link that you can copy, and then paste into an email message or other document.  For more information, see <a href="#">Sharing Documents using Email</a> .
<b>Printable View</b>	Opens a print preview of the current page. See <a href="#">Printing Records</a> .
<b>Add Bookmark</b>	Sets the record as a <b>Bookmark</b> for easy access on your button toolbar.
<b>Delete</b>	Removes the currently open record from the database. See <a href="#">Deleting Records</a> .
<b>Edit</b>	Opens a record in edit mode so that you can make changes. See <a href="#">Editing a Record</a> .
<b>OK</b>	Saves the open project or the changes you made to it.  This button is available only in embedded project records.  <i><b>Important:</b> You must also save the parent project to save the changes made to the embedded project.</i>

## Using the Title Bar

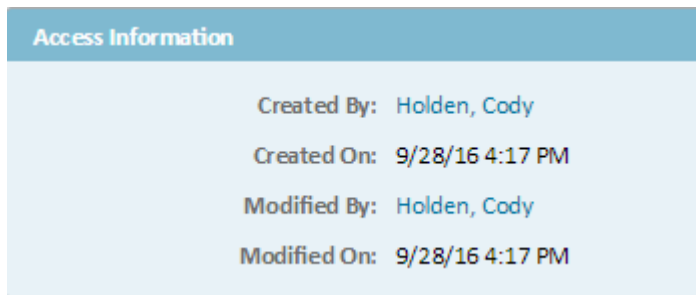
In each record page, its title bar displays the following information:

- Record type and its icon, followed by a hyphen.
- Name of the record (depending on the records type, it may be the record's subject or description).
- A "Back to" link. For example, if you are viewing a Contact record, there will be a link, **Back to Contacts**, in the title bar.

If you click the **Back to** link, the record page is replaced by a page that displays records from the most recently used list for this object type.

## Access Information

Each record displays access information, usually on its General page. This information tells you who created the record and who most recently changed the record. The information is read-only.



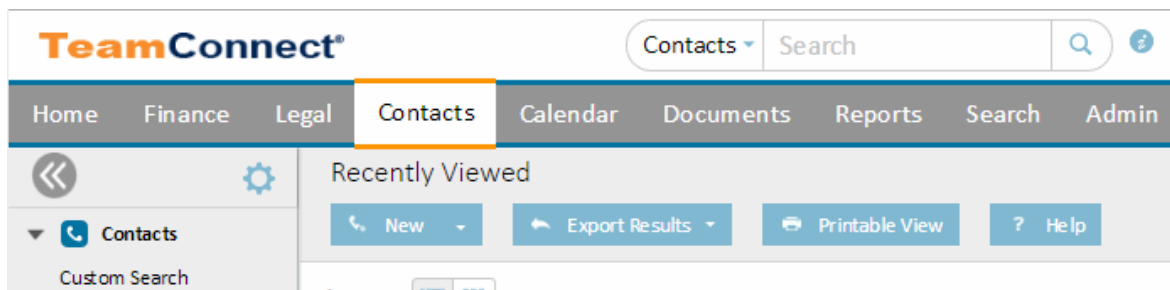
Access Information Section

## 1.2.6 Creating a Record

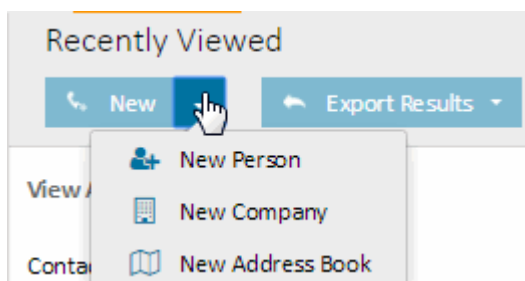
You can create a new record from a list view or from an open record that provides an option to create a new record.

### List View

To create a new record in list view, click the **New** button in the button bar.



Locating the New Button on the Toolbar



Click the New Button for a Contact Record

In many cases, clicking **New** will immediately take you to a new page. In some cases, such as the Contact record type shown above, you may need to make a further selection from a list to indicate what kind of record you wish to create, such as a wizard.

**Note:** The **New** options that you see depend on your company policy. For example, some companies only allow you to create new records using a wizard, so that will be the only available option.

### Open Record



Search for a specific record, and then click the **New** button next to the Search icon, if it is available.

When you create a new record, you are shown pages that contain fields that you can edit. The details of editing these fields are discussed in [Editing a Record](#).

For new records of any record type, the **History** and **Documents** links in the left pane are disabled until the record is saved for the first time. After that, **History** and **Documents** are accessible. Similarly, the **Address Books** link for a new **Contact** record is disabled until the **Contact** is saved for the first time.

### 1.2.6.1 Wizards

A Wizard is a utility that displays a set of pages to guide you through the process of entering the information that is necessary to create a valid record.

For example, the **Scheduler** wizard can guide you through the steps necessary to create an appointment record. On predefined pages, you can specify the time of the appointment, add attendees, and then add resources for the event. This is the only default wizard.

All other wizards are created for your organization based on your business practices and procedures. For example, a matter wizard for a legal department might be designed to help you collect and enter information regarding a legal transaction.

## Opening a Wizard

A wizard may be one of the available options to add a new record or the only option. If a wizard is your only option, it opens immediately when you click the **New** button. If several wizards are available, click the **New** button, and then select the appropriate wizard from the drop-down list.

## Using a Wizard

To advance through a wizard, complete the appropriate fields on the first page, and then click **Next** to advance to the second page. The next page that you see depends on the options that you select on the previous page. On each page, complete the appropriate fields, and then click **Next** to advance to the next page. If you make a mistake, or need to change an option on a previous page, click **Back** to go back to the appropriate page. If you advance to the final page of the wizard, click **Finish** to save the record.

If available, any wizard page may include a **Finish** button that allows you to stop and save the wizard before the final page. Once the record is saved, you can go back and add additional information at any time.

**Important:** When you are using a wizard, do not use the browser's **Back** or **Forward** buttons. Instead, use the buttons in the wizard's toolbar (**Next**, **Back**, **Cancel**, and **Finish**) to navigate through the wizard.


You can also use the links in the left pane to advance through the wizard. When you open a wizard, the left pane displays all of the wizard pages in the correct order. The wizard pages that you can access in the left pane also depend on the selections that you make on a previous page.

**Note:** If you use the left pane to advance through the wizard, it is recommended that you complete each page in order.

### 1.2.6.2 Editing a Record

Editing a record is the process of changing the current values or entering additional values into a saved record.

#### To edit a record

1. Select one of the following:
  - When viewing a record, click the **Edit** button in the button bar.
  - When viewing a record, click any **Edit** link in any section on that record's page.
  - When viewing a list of records, click the **Edit** icon  next to the record you wish to edit.
2. Edit the appropriate fields. See [Working with Different Types of Fields](#).
3. Select a save option, as described in [Saving an Edited Record](#).

### 1.2.6.3 Editing Multiple Records

In cases where you need to edit multiple records at a time, use the Mass Edit feature. This feature allows you to update the fields that appear in the results display of a search view page in multiple records at a time.

#### To edit multiple records

1. Select one of the following:
  - Navigate to an existing search view page (e.g. Recently Created, Recently Modified, a saved search) for the record type you want to mass edit.
  - Perform a new Custom Search for the record type you want to mass edit. The Results Display options you define in your Custom Search will dictate which fields are available on the Mass Edit page. See [Using Custom Search](#).
2. When viewing the search results, click the **Mass Edit** button. **Note:** *Users can only Mass Edit parent category or 'root' items.*
3. Select the check-boxes of the records you want to mass edit.
4. In the Mass Edit table, enter the following items:
  - In the **Field** column, select the field that you want to mass edit. The available fields are dictated by the results display of the originating search view page. See [Specifying Results Columns](#).
  - In the **Operator** column, select the operation to perform on the data contained within the record. If only one operator is available, this drop-down list will be grayed out and not interactive. See [Supported Field Types and Operators](#).

- In the **Value** column, enter the new data to be applied to the selected records. When using the Replace operator with Assignees, the Value column will contain two data fields. See [Supported Field Types and Operators](#).
5. Click the **Update** button. Your changes will be applied to all selected records on the Mass Edit page.  
You can also edit individual records directly in the record list.
  6. Click the **Save & Close** button.

**Important:** Your changes are *not* permanently saved in the edited records until you click the **Save & Close** button!


The table below displays the field types and operators that are supported for mass editing:

**Supported Field Types and Operators**

Field Type	Operators	Description
<b>Main Assignee</b>	<ul style="list-style-type: none"> <li>• <b>Replace</b></li> </ul>	The <b>Replace</b> operator replaces the Main Assignee in the selected records with the selected <b>User</b> and <b>Role</b> .
<b>Assignees</b>	<ul style="list-style-type: none"> <li>• <b>Replace</b></li> <li>• <b>Add/Assign</b></li> <li>• <b>Unassign</b></li> </ul>	The <b>Replace</b> operator deactivates the selected <b>Current Assignee</b> and adds the selected <b>New Assignee</b> as an Assignee on the selected records. If the selected <b>Current Assignee</b> is the Main Assignee, then the selected <b>New Assignee</b> will become the new Main Assignee.
		The <b>Add/Assign</b> operator adds or assigns the selected user as an Assignee to the selected records. Selecting the <b>Primary Assignee</b> check-box will also designate the <b>New Assignee</b> as the main assignee on the selected records.
		The <b>Unassign</b> operator unassigns the selected user as an Assignee from the selected records. This operator will skip records where the selected <b>Current Assignee</b> is the Main Assignee.
<b>Involved</b>	<ul style="list-style-type: none"> <li>• <b>Add New</b></li> <li>• <b>Set Existing</b></li> <li>• <b>Remove</b></li> </ul>	The <b>Add New</b> operator adds the selected <b>Contact</b> in the selected <b>Role</b> as an Involved Party on the selected records.
		The <b>Set Existing</b> operator changes the <b>Role</b> of the selected <b>Contact</b> on the selected records. If the selected <b>Contact</b> is not an Involved Party on the selected records, no changes will be made.
		The <b>Remove</b> operator removes the selected <b>Contact</b> in the selected <b>Role</b> as an Involved Party on the selected records.


<b>Boolean</b>	<ul style="list-style-type: none"> <li>• <b>Replace</b></li> </ul>	The <b>Replace</b> operator replaces the Boolean value in the selected records with the new designated value.
<b>Text</b>	<ul style="list-style-type: none"> <li>• <b>Replace</b></li> </ul>	The <b>Replace</b> operator replaces the text in the selected records with the new designated text.
<b>Numeric</b>	<ul style="list-style-type: none"> <li>• <b>Replace</b></li> <li>• <b>Adjust Amount</b></li> </ul>	The <b>Replace</b> operator replaces the numeric value in the selected records with the new designated value.
		The <b>Adjust Amount</b> operator adds the designated numeric value to the value in the selected records. You can subtract an amount by designating a negative numeric value.
<b>Date</b>	<ul style="list-style-type: none"> <li>• <b>Replace</b></li> </ul>	The <b>Replace</b> operator replaces the date in the selected records with the new designated date.
<b>Memo</b>	<ul style="list-style-type: none"> <li>• <b>Replace</b></li> <li>• <b>Add To End</b></li> </ul>	The <b>Replace</b> operator replaces the memo text in the selected records with the new designated text.
		The <b>Add To End</b> operator appends the new designated text to the end of the existing memo text, starting on a new line, in the selected records.
<b>Lists</b>	<ul style="list-style-type: none"> <li>• <b>Replace</b></li> </ul>	The <b>Replace</b> operator replaces the list in the selected records with the new designated list.

#### 1.2.6.4 Record Locking During Editing

The system administrator can enable or disable record locking. If record locking is enabled, when you edit a record, it is automatically locked to prevent changes by others while you are editing it. If another user who has editing rights to that record brings up a list that includes that record, they will see a lock icon  displayed next to it.

Record locks can last indefinitely, or can expire after a specified time interval. This expiration interval is determined by your system administrators.

It is possible to override a record lock, if you have been given rights to do so. Click on the record name link, and you see an warning message like the one shown below. If you wish to edit the record yourself, click the link that says Release the lock to edit the record.

 **Warning:** Sacky, Kristen is currently editing this record. Opened on 9/30/16 2:16 PM. [Release the lock to edit the record.](#)

#### Example of Overriding a Record Lock

If you override a record lock and then edit and save the record, the person who was originally editing that record will receive a warning message. If they still need to edit the record, they should examine its information carefully, since their original work may have been overwritten by your edits.

**Note:** When administrators disable record locking, multiple users can edit the same record simultaneously. However, only the user who first attempts to save the record can save it successfully. When the other users attempt to save, they will see a warning message that tells them that their save failed.

### 1.2.6.5 Working with Different Types of Fields

There are several types of fields:

- [check-box](#)
- [Date Field](#)
- [List Fields](#)
- [Memo Text](#)
- [Number](#)
- [Option Buttons](#)
- [Text](#)

When a field is **required**, you must make an entry in it before saving the record. Required fields are indicated by a bold label and an asterisk at the beginning of the label.

#### 1.2.6.5.1 check-box

Use a check to turn an option on or off. When the option is on, a check mark appears in the box. To select or clear a check-box, click inside it with the mouse, or navigate to it with the keyboard and press SPACEBAR. You can check multiple check-boxes as appropriate.

Edit mode	Read-only mode
<input checked="" type="checkbox"/>	(not applicable)

#### 1.2.6.5.2 Date Field

Use date fields to store and display dates in the format specified in your user preferences (see [Selecting Region, Date, and Time Formats](#)).

Edit mode	Read-only mode
Birthday: <input type="text" value="1/1/2001"/> 	Birthday: 1/1/01

In date fields, you can do either of the following:

- Manually type the date in the accepted format. This method may be faster when entering dates that are in the past or distant future. You can copy and paste data, if necessary. A calendar window pops up, but does not interfere with your manual data entry.
- Put the cursor in the date field to automatically open the **Calendar** pop-up window and select the necessary date in the pop-up window.



#### To select a date in a pop-up calendar:

1. Use the left or right arrow controls to set the desired month and year. The double arrows move by one year per click, and the single arrows move by one month per click.
2. Select the desired day on the screen.

After you select the day, the **Calendar** window closes.

#### 1.2.6.5.3 Time Zone Independent Date Fields

Depending on each object setting, date fields can be one of the following:

- **Time zone dependent**—Converts dates entered by other users to your time zone. If you enter a date in a time zone dependent date field and a user from a different time zone views the date, the date will be converted to that user's time zone.
- **Time zone independent**—Dates entered by another user are not converted according to time zone, regardless of your time zone setting. This means that dates should show up "as is", regardless of a user's Time Zone.

Some System Date fields will also be set to be Time Zone Independent by default.

#### 1.2.6.5.4 List Fields

You can display list items in the following ways:

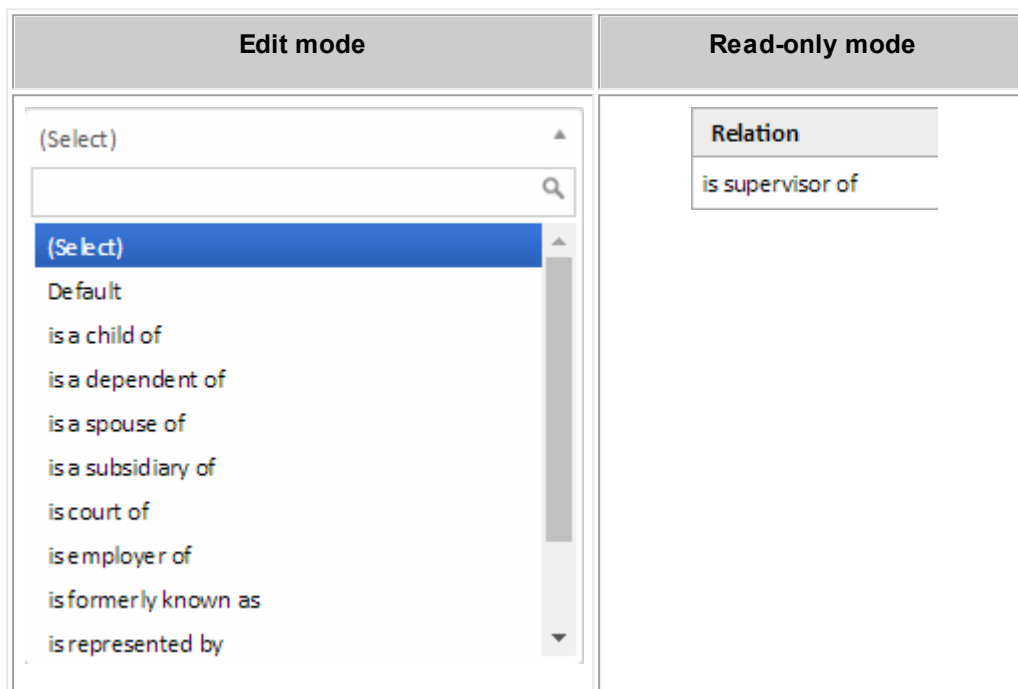
- [Drop-down List](#)
- [List Box](#)

#### 1.2.6.5.5 Drop-down List

A drop-down list is a field where you click the arrow on the right or press the DOWN ARROW key and choose an item in the resulting list. You can make only one selection in a drop-down list.

If items in a drop-down list are hierarchically related to each other, these relations are indicated by the appropriate indentation and a colon between the items.

The advantage of drop-down lists is that you can see all available items and can make your selection quickly. However, if you have a large number of items, you have to scroll up and down a long list making sure to select the correct item.



As you can see, in read-only mode, a drop-down list presents its value as a simple field or background text. There is no indication that it is actually a list.

#### To find the necessary item in a drop-down list

1. Click the drop-down list or navigate to it using the keyboard.
2. Type the first letter of the item (or the first digit for numbers).
3. To go further down in the list of words starting with the selected character, keep pressing that key.

When you reach the end of the list of the words starting with the selected character, the first word in the list is selected again.

**Important:** This technique does not work for the items displayed with spaces at the beginning of the item text (indented items).

## Points to Remember

- To open the list, click or tab to the drop-down list.
- To select the necessary option, click the item or use the arrow keys.

If there is an adjacent command button next to the drop-down list, you must click the button or navigate to it and press ENTER to complete the selection.

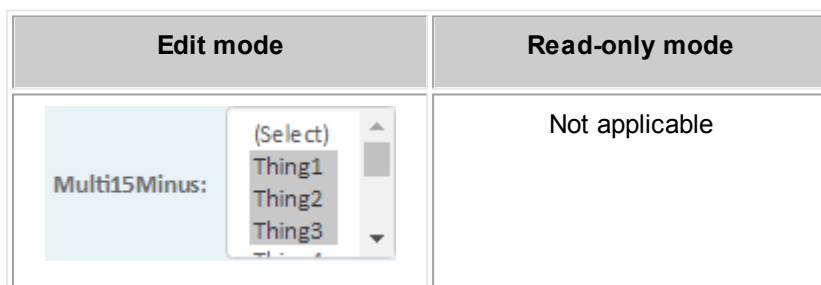
- In long lists, you can scroll up and down the list.

**Important:** For full keyboard accessibility in drop-down lists, you must enable Accessible Mode. For more details, refer to [Enabling Accessibility Mode](#).

- In Accessible mode, use the arrow keys to select an option, press the TAB key to navigate to the **go** button, then press ENTER to make your selection.
- Categories or lookup table items in a list that are displayed in gray are inactive. If accessibility settings are enabled, **(inactive)** is added to the item name and they are not displayed in gray.
- Inactivated items must not be added to records, so they are not displayed in lists that are used to add items to records. If a category or lookup table item that you are familiar with is not displayed in a list, it may have been inactivated.
- If search screens include category lists or lookup tables for the type of record, any inactive items are displayed in gray. You can select these items to search for records that had the item selected before it was inactivated.
- If an inactive item was added to the record before it was made inactive, you can save the record without changing the item's value. If you accidentally change the value and save the record, you cannot restore the inactive value to the record, but you can contact your administrator to temporarily activate the needed item.

### 1.2.6.5.6 List Box

List boxes allow you to simultaneously select one or more items. List boxes are only available in search screens and in some wizards. It is also known as a "multi-select" list.





### To make a single selection

Select the item in the list using a mouse or keyboard arrow keys.

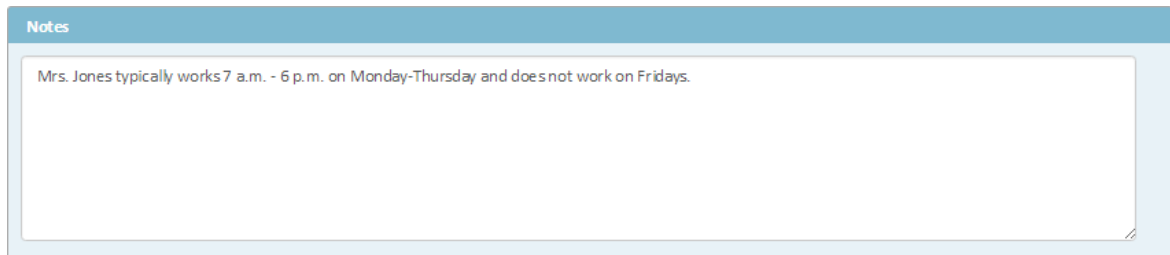
### To make multiple selections

- Select the first item, press and hold the SHIFT key, and use the keyboard arrow keys or a mouse to select several consecutive items in the list.
- Press and hold the CTRL key and click on the desired items. If you prefer to use a keyboard, press SHIFT+F8 to activate the multiple select capability, use the arrow keys to move to items, and press SPACEBAR to select several random items in the list.

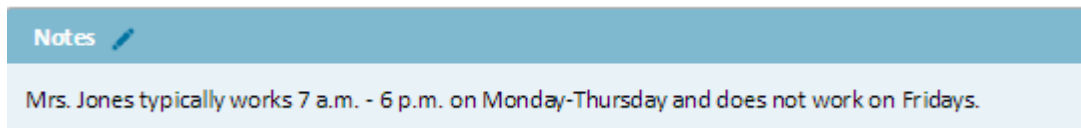
#### 1.2.6.5.7 Memo Text

Memo Text fields allow you to type large amounts of plain text.

Edit mode:



Read-only mode:



If you copy and paste the necessary text from other word processing applications, you lose the formatting, including bullets, numbers, and indentations.

**Note:** *In the unusual case that you enter 250 characters or more, without any embedded spaces, into a Memo Text field, the web browser might try to word-wrap the content by making the field much wider, causing any fields to the right of the memo to move much farther to the right. The fields are still accessible by using the horizontal scroll bar.*

#### 1.2.6.5.8 Number

Number fields can store and display decimals, percentages, and integers. In other words, Number fields accept only numeric characters. Also, each Number field may accept only appropriate values, for example, the **Number of Plaintiffs** field accepts decimal amounts, whereas a **Total Amount**

field in accounts can accept both integers (whole numbers) and decimals. Most decimal amounts are rounded to two places. Invoice expense rates can have up to five decimal places.

**Note:** In read-only mode, negative numbers are displayed in parentheses. In edit mode, negative numbers are preceded by a minus sign. You should use the minus sign, not the parentheses, when editing a negative number.

Edit mode	Read-only mode
Number: 10000000	Number: 10,000,000.00

All number fields automatically format your entries as necessary, for example, to dollar or percentage amounts, and display the appropriate signs. In other words, you do not have to type formatting characters.

#### 1.2.6.5.9 Option Buttons

Option buttons are fields that look like circles that you can use to select only one of the mutually exclusive options. These buttons are sometimes referred to as "radio buttons." To make a selection, click the desired item with your mouse or, alternately, move the focus to the option buttons with the Tab key and use the arrow keys to make your choice. In the example below, the option button serves to set the 'Primary' option.

Edit mode	Read-only mode
<p><b>Phone Number</b></p> <p>Business <input type="text" value="111-111-1111"/> <input checked="" type="radio"/> <input type="radio"/></p> <p>Home <input type="text" value="222-222-2222"/> <input type="radio"/> <input checked="" type="radio"/></p> <p>Mobile <input type="text" value="333-333-3333"/> <input type="radio"/> <input checked="" type="radio"/></p>	<p><b>Phone Numbers</b></p> <p>Home: 222-222-2222</p> <p>Business: 111-111-1111 (Primary)</p> <p>Mobile: 333-333-3333</p>

#### 1.2.6.5.10 Text

Text fields accept any characters. Unlike memo text fields that accept large amounts of text, you can only type up to 50 - 250 characters in text fields.

If you are working with the Workflow Approvals or Invoices, you can enter up to 2000 characters in any **Comments** or **Reason** field.

### 1.2.6.6 Working with Batch Entries

Many of the pages allow you to put in the same kind of information multiple times within a single record. For example, by default, the following pages have batch appearance and functionality:

- **Categories**
- **Relations and Involved** (in projects and contacts)
- **Assignees** (in projects)
- **Invoice Line Items**
- **Batch Tasks and Batch Expenses**
- **Attendees and Resources** (in appointments)
- **Address, Phone, Fax, Email, Internet Address, Skills, Rates, Territories** (in contacts)

Initially the Phone section in the Contact record has only a single entry available.

To add more entries, click the **Plus** button that is shown to the right of the input fields, one or more times.

Now the section contains multiple entries, all still blank. You can edit all of these entries.

The Phone Section with Multiple Entries after Editing

#### 1.2.6.6.1 Primary Entries

For some record types, when multiple entries are allowed, you must designate one of the entries as primary. You do this by clicking the **Option** button next to the **Plus** button. When an entry is designated as primary, it is used in reporting and in any area where only one entry will be shown, no matter how many other entries exist.

In edit mode, the primary designation appears as a check mark in an option button. In view mode, however, the word "Primary" appears next to the primary entry instead.

Viewing a Primary Entry when in Read-only Mode

#### 1.2.6.6.2 Deleting an Entry

When a section already has multiple entries, and you wish to delete one of them, click the **Minus** button next to the entry. That entry will disappear but the other entries will remain.

#### 1.2.6.6.3 Editing Batch Entries

For custom blocks, if enabled, you can add or delete one or more fields. Additionally, you can tab to specific fields in the batch display and edit them as necessary. A search feature is also provided so that you can search for information to populate a field.

**Note:** *This feature is only available if it has been enabled by the system administrator. If not enabled or if you do not have access to this feature, the following screens and functions are not available.*

##### To add a new entry

- To add a single row, click the **Add New Item** button above the batch display. A new empty row is added to the bottom of the display and the cursor moves to the **Work Item field** so that you can immediately begin adding information.
- To add multiple rows, click the **Add Item** drop-down list and select the number of rows that you want to add. The selected number of rows appear at the bottom of the batch display.

##### To edit an entry

Click the edit icon next to the line that you want to edit. The field becomes editable and you can make changes.

##### To copy an entry

Click the copy icon next to the line that you want to copy. An exact duplicate of the selected line appears at the bottom of the list.

When you are done adding, editing, or copying entries, click **Save** to save the changes.

##### To remove an entry

Click the check-box for the entries that you want to remove, and then click **Remove**.

##### To search in a batch display

Some fields provide a search icon that you can click to search for specific information to populate a field. Click the search icon to enter search criteria. For information about using the search module, see [Searching for Specific Records](#).

### 1.2.6.7 Setting Category Details

You can add many additional custom information fields to the pages that appear in the "out of the box" version. To support these custom fields, they are organized into categories. A group of custom fields that belongs to a category is called category details or just details.

Categories are a way of specifying a particular kind of record. For example, in a Contact record of an employee, you may need to enter information about their hire date, manager, and department. To enter these details, the appropriate category, such as **Employee**, should be selected for that contact record. But those same details would not make sense in a Contact record for an outside vendor. For vendors, you may need other specific details instead, such as vendor type and payment methods. This can be addressed by adding the **Vendor** category to the contact record.

Categories control which specific sections of details are displayed on the **Details** sections of a record's **General** page. By adding or deleting categories in a record, you display or hide the associated sections of details in that record.

**Note:** Contact records differ from other record types, because all of the details related to categories are displayed on a **Details** page, not in details sections in the **General** page.

You might have many other categories of contacts. Categories can be structured like a family tree. For example, the **Employee** category might be subdivided into **Manager**, **Non-Exempt**, and **Temporary** categories. When you assign the **Non-Exempt** category to a contact record, you automatically include the parent **Employee** category, so that detail sections for both categories would be visible in that record.

The categories used by your organization are entered by your system administrator into a list that is then available for you to choose from when you work on Contact records, Project records, or any custom record type.

## Categories and Record Types

Besides the **Contact** record type, categories are also used to control sections of custom fields (category details) in the **Project** record type, and in each custom record type.

In the **Account** record type, categories are used for organizing accounts, but accounts do not have custom fields associated with their categories.

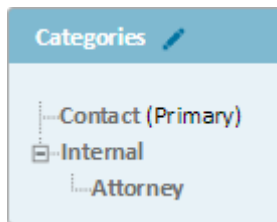
The examples in this portion of the documentation have categories related to the Contact record type.

## Assigning Categories

When viewing an existing record's Categories page, only the categories that are presently assigned to the record will be shown. There might be many other categories available for assignment, but they are not visible during viewing.

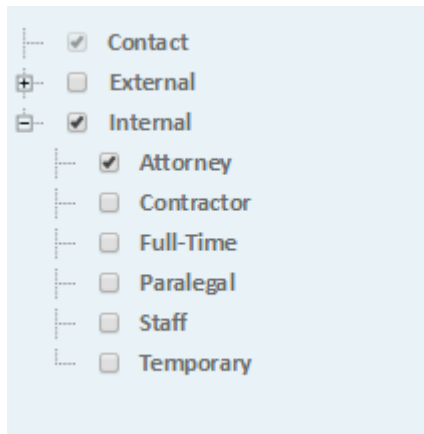
When editing the Categories page for a record, all of the categories that are available for assignment are shown. The categories that are presently assigned are indicated in check-boxes. You can assign multiple categories to a record.

The list of categories is structured like a family tree. Those branches that are expanded are marked by the minus icon, and those that are presently collapsed are marked by the plus icon. You can expand and collapse branches by clicking these icons.



**Expanding Categories**

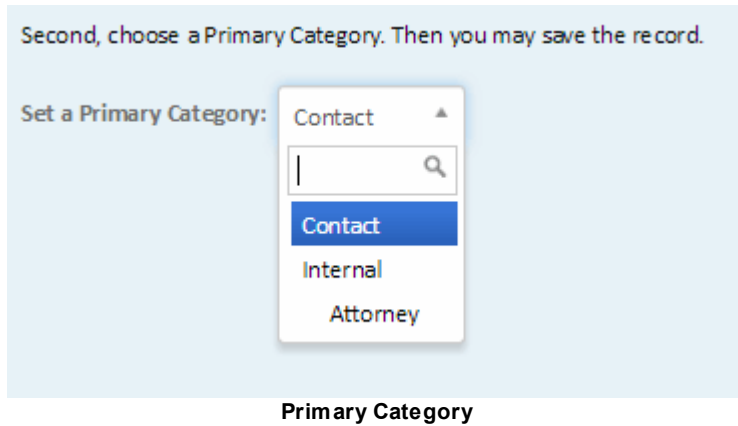
You can assign or unassign a category by clicking the check-box next to the category name to place or remove a check mark in the box.



**Placing Checkmarks Next to Categories**

Note that assigning a category automatically causes any parent categories to be assigned, too. In this example, assigning **Attorney** to the record caused **Internal** to be assigned, because **Internal** is the parent of **Attorney**. It is also possible to assign a parent category to a record without assigning any of the child categories.

Set a primary category before saving as detailed in the image below.



## Unassigning Categories

You can unassign a category by clicking the check-box next to the category name to remove the check mark in the box. Unassigning a child category does not automatically unassign the parent category.

**Important:** When you edit an existing record and unassign one of its categories, then save the record, all of the information in the category details for that category will be permanently deleted. Even if you re-assign the category the previous information cannot be recovered. Therefore you should use caution when unassigning a category.

### 1.2.6.8 Canceling Your Edits

If you choose to cancel the edits you made, click **Cancel & Close** or **Cancel & View** in the record toolbar. Then answer the resulting confirmation dialog box. The record's page disappears and you will return to the page that you were viewing before you began editing.

### 1.2.6.9 Saving an Edited Record

When you are done working on a record, use the **Save** button on the button bar to save your work. This button might have a label different than **Save**, depending upon how you have been using the button in your session. If you click or hover over the drop-down arrow, as shown in this example, you have one of the following options:

- **Save**—Saves the record but leaves the record open and in edit mode.  
Since you requested that the record remain in edit mode, it will still appear in your list of Unsaved Items. To remove the record from that list, choose a different Save option.
- **Save & Close**—Saves the record, closes the record's page, and then returns you to the previous page.
- **Save & View**—Saves the record, then returns to a read-only view of the record.
- **Save & New**—Saves the record, then opens a new blank record of the same type so that you can enter new information.

The default option out of this selection can be set by administrators. For more information, see [Setting up the User Interface](#).

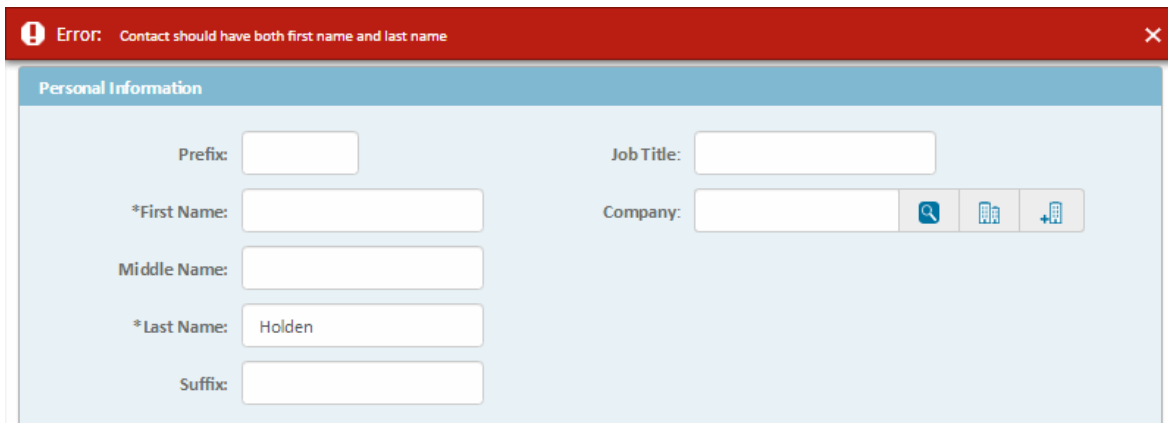


Save Options

If the label on the **Save** button is already the choice you want, simply click the button. If not, click the drop-down arrow on the right edge of the button and make your choice from the list that is shown there.

**Note:** Records contain fields to indicate when, and by whom, a record was created, last modified, and last opened. These fields are useful when searching or doing reports. All of these applicable fields are populated with your user name, and the current date and time, when you click a Save option.

If there are errors in the fields that you were editing, the save operation may fail and one or more error messages will appear at the top of the page.



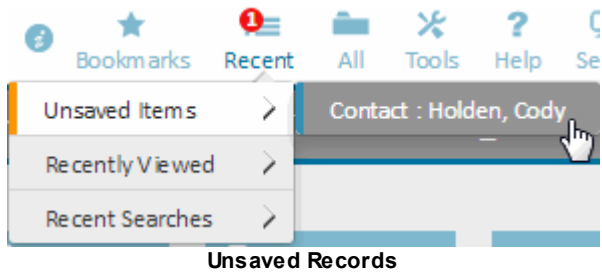
An Error Message during a Save Attempt on the Contact/General Page

The record remains in edit mode. Correct the errors that are described in the error messages, and try to save the record again.

#### 1.2.6.10 Handling Unsaved Records

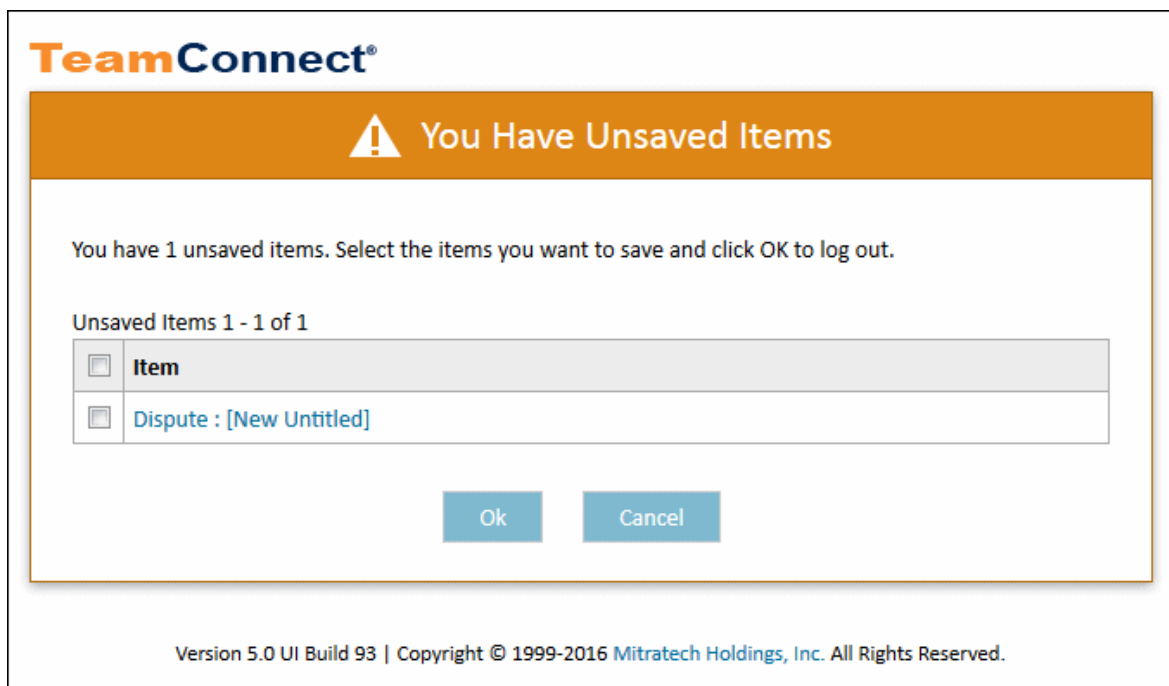
You can edit a new or existing record, then leave the page where you were editing the record and perform other tasks. When you do this, the record that you had been working on is remembered as an unsaved item. The **Recent Items** button in the upper right portion of the screen will show the presence of unsaved records, as shown in this example.





When you hover the mouse over the **Recent Items** button, a window appears that contains links to records you have worked on recently as well as unsaved records. To return to an unsaved record, simply click the link for that record. You can then edit the record further and/or save the record.

You are also reminded about unsaved records when you attempt to log out while there are still unsaved records present in your session. A window appears that lists the unsaved items.



Each item has a check-box next to it. If you wish to save the items before you log out, click the related check-boxes and then click **OK**. If you do not wish to save any items, leave the check-boxes blank and click **OK**. If you decide not to log out after all, click **Cancel**.

### 1.2.6.11 Copying a Record

Copying a record is recommended when the new record is very similar to an existing one, and you will not have to change many of the values that are automatically populated.

To copy a record, on the record toolbar, click the **Create a Copy** button. The new record opens and some of the information is identical to the record you copied. You can now edit and save the new record.

**Note:** *Not every piece of information will be copied from the old record to the new record. Each record type has its own set of excluded information. Generally, child records (such as Involved records for a Project or a custom record type) are never copied to the new record. In addition, any piece of information that displayed as "SECURED" in the old record will not be copied, and that field in the new record is blank. Likewise, in the case that multiple items are selected and one (or more) are deactivated, the inactive items will not be copied.*

The table below displays the copy details for each record type:

#### Copying a Record

Record Type	What is Copied	What is Not Copied
<b>Accounts</b>	<ul style="list-style-type: none"> <li>• <b>General</b> page</li> <li>• <b>Categories</b> page</li> <li>• <b>Details</b> page</li> <li>• <b>Posting Criteria</b> page (<b>Posting Criteria Expense</b>, <b>Posting Criteria Invoice</b>, <b>Posting Criteria Involved</b>, <b>Posting Criteria Task</b>, <b>Posting Criteria Vendor</b> sections)</li> <li>• <b>Security</b> page (<b>Group Rights</b>, <b>User Rights</b> sections)</li> </ul>	<ul style="list-style-type: none"> <li>• <b>General</b> page - <b>Amount</b> section, <b>Account Summary</b> section</li> <li>• <b>Child Accounts</b></li> <li>• <b>Documents</b></li> <li>• <b>History</b></li> <li>• <b>Transactions</b></li> <li>• <b>Workflow</b></li> </ul> <p><i>Note: If any field displays <b>SECURED</b> as the value, then the new record will not contain a value for that field.</i></p>
<b>Appointments</b>	<ul style="list-style-type: none"> <li>• All of the information on the <b>General</b> page (except the <b>Subject</b>)</li> <li>• All of the attendees on the <b>Attendees</b> page</li> <li>• All of the categories selected on the <b>Categories</b> page</li> <li>• All user and group rights selected on the <b>Security</b> page</li> <li>• All of the resources selected on the <b>Resources</b> page</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Documents</b> that are attached to the <b>Documents</b> page</li> <li>• Historical entries on the <b>History</b> page</li> <li>• Approval information on the <b>Workflow</b> page</li> </ul>

<b>Contacts</b>	<ul style="list-style-type: none"> <li>• <b>General</b> page, except for the information in the <b>Name</b> field</li> <li>• <b>Security</b> page</li> <li>• <b>Rates</b> page</li> <li>• <b>Skills</b> page</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Documents</b> page</li> <li>• <b>History</b> page</li> <li>• <b>Involvement</b> page</li> <li>• <b>Relations</b> page</li> <li>• <b>Territories</b> page</li> <li>• <b>Workflow</b> page</li> </ul> <p>Additionally, none of the address books that the original user created are copied to the new user, and none of the address books to which the original user subscribed are copied.</p>
<b>Documents</b> <b>Document Folders</b>	See <a href="#">Copying Documents</a> for more information.	
<b>Expenses</b>	<ul style="list-style-type: none"> <li>• <b>General</b> page &gt; <b>General Information, Access Info</b> sections</li> <li>• <b>Security</b> page (<b>Group Rights, User Rights</b> sections)</li> </ul>	<ul style="list-style-type: none"> <li>• <b>General</b> page &gt; <b>Billing Information</b> section</li> <li>• <b>Documents</b></li> <li>• <b>History</b></li> <li>• <b>Transactions</b></li> </ul>
<b>Invoices</b>	<ul style="list-style-type: none"> <li>• <b>Categories</b> page</li> <li>• <b>Details</b> page</li> <li>• <b>General</b> page (except <b>Line Items</b>)</li> <li>• <b>Security</b> page &gt; <b>Access Info, Group Rights, User Rights</b> sections</li> </ul>	<ul style="list-style-type: none"> <li>• <b>General</b> page &gt; <b>Line Items</b> section</li> <li>• <b>Documents</b></li> <li>• <b>History</b></li> <li>• <b>Transaction</b></li> <li>• <b>Workflow</b></li> </ul> <p><i>Note: If any field displays SECURED as the value, then the new record will not contain a value for that field.</i></p>
<b>Projects</b>	<ul style="list-style-type: none"> <li>• All of the information on the <b>Assignees</b> page</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Accounts</b> page</li> <li>• <b>Appointments</b> page</li> </ul>

	<ul style="list-style-type: none"> <li>• All of the information on the <b>Categories</b> page</li> <li>• All of the information on the <b>General</b> page, except the name and number of the original project</li> <li>• All of the user and group rights on the <b>Security</b> page</li> <li>• All of the information on the <b>Relations</b> page</li> <li>• Additionally, all of the embedded projects are copied to the new record</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Documents</b> page</li> <li>• <b>Expenses</b> page</li> <li>• <b>History</b> page</li> <li>• <b>Involved</b> page</li> <li>• <b>Phase History</b> page</li> <li>• <b>Tasks</b> page</li> <li>• <b>Workflow</b> page</li> </ul> <p>Also, none of the related child projects are copied to the new project.</p>
<b>Tasks</b>	<ul style="list-style-type: none"> <li>• All of the information in the <b>General Information</b> section on the Task <b>General</b> page</li> <li>• All of the information on the <b>Task Security</b> page</li> </ul>	<ul style="list-style-type: none"> <li>• None of the information in the <b>Billing Information</b> section on the <b>General</b> page</li> <li>• <b>Documents</b></li> <li>• <b>History</b></li> <li>• <b>Transactions</b></li> <li>• <b>Workflow</b></li> </ul>

#### 1.2.6.12 Reassigning Work

The **Reassignment Tool** is used to delegate the work of one user to another user or multiple users. With this tool you can reassign all:

- Appointments
- Tasks
- Projects

The tool lists all projects. However, you can only reassign projects that have the option to be assigned to a user.

When you reassign a project, the tool also reassigns all incomplete tasks and future appointments part of that project.



**Note:** *If you want to reassign all tasks and appointments as well as projects, the tool first reassigns projects and their related tasks and appointments, and then reassigns tasks and appointments that are not related to any specific project.*

**Important:** *For performance optimization, the tool reassigns large numbers of projects in small batches of 10 project records. If any record within a batch fails, the tool skips the entire batch and the next batch in the sequence executes. You can view the log for specific*

information about a failed batch. It is recommended that you reassign large groups of projects during off-peak hours.

### To reassign work

1. Click the **Tools** tab.
2. In the **Tools** section, click **Reassignment Tool**.  
The **Reassignment Tool** page opens.
3. Select or search for the user whose work you want to reassign.
4. Select the check-box for the **Record Type** that you want to reassign.

<input type="checkbox"/>	Record Type	Setting	New Assignee	Status
<input checked="" type="checkbox"/>	Appointments	Future ▾	McNeil, Charles 	
<input checked="" type="checkbox"/>	Tasks	Incomplete ▾	Holden, Cody 	

Record Type on the Reassignment Tool

5. For each record type that you want to reassign, select the appropriate option in the **Setting** drop-down list:
  - **Appointments**—Select **Future** for appointments that have not occurred or **Future and Past** for all appointments.
  - **Tasks**—Select **Incomplete** for tasks without **Completed** as the **Status**, or select **Un-posted** for tasks that have not be posted.
  - Projects or custom objects—Select the appropriate option. The options are most likely **Open** projects or **Open and Closed** projects.
6. In the **New Assignee** column, select or search for a user to be the new assignee of the selected record types.
7. Click **Reassign**.

The status of the reassignment appears in the Status column. If the reassignment was successful, the status looks like the following image.

If the reassignment of a related task or appointment fails, for example, a record is pending approval at the time of the reassignment, the project reassignment fails as well. When the record reassignment is complete, a message displays the number of records successfully reassigned and the number of failures. You can view the log for detailed information. If the reassignment failed, the status looks like the following image.

#### 1.2.6.13 Importing Data to Create Records

Use the Data Import Wizard to create multiple records at the same time. The wizard requires you to add all records to a spreadsheet so that you can import the data from that spreadsheet into TeamConnect.

You can import the following record types:

- Contacts
- Custom Objects
- Involved Contacts

**Note:** The Data Import Wizard does not support Embedded Objects.

To import records, you must first create a data file. See [Creating a Data File](#) and [Importing Data](#) for more information.

**Note:** If you import data for a record that already exists, TeamConnect creates a duplicate record. In the case of contact records, if a solution developer has configured your system to prevent duplicate contacts, the wizard returns an error and does not create the duplicate record.

#### 1.2.6.13.1 Creating a Data File

To import data, you must create an Excel spreadsheet that contains the record information that you want to import. The spreadsheet looks similar to the following image.

	A	B	C	D	E	F	G	H	I	J	K	L
1	First Name	Last Name	Company	Job Title	Type	Address	Address Type	City	State	Zip	Phone	Phone Type
2	Beatrice	Wilk	Hill, Inc.	Accountant	P	34009 Washington Pl	Business	San Diego	CA	90016	310-555-1215	Business
3	Willie	Nels	Hill, Inc.	Accountant	P	34009 Washington Pl	Business	San Diego	CA	90016	310-555-1216	Business
4	Annette	Perkins	Hill, Inc.	Sr. Accountant	P	34009 Washington Pl	Business	San Diego	CA	90016	310-555-1217	Business
5	Daryn	White	Hill, Inc.	Account Intern	P	34009 Washington Pl	Business	San Diego	CA	90016	310-555-1218	Business
6	Helen	Washington	Hill, Inc.	Account Supervisor	P	34009 Washington Pl	Business	San Diego	CA	90016	310-555-1219	Business
7												

**Contacts Example**

**Note:** If your Excel workbook has multiple sheets, only the first sheet uploads.

#### To create a data file

1. (optional) Open the appropriate record type and use the label names in the record as a guide to determine label names for your data file.
2. Enter the label names in the first row of the spreadsheet.

For example, for contact records, you could enter First Name, Last Name, Company, Job Title, Type, etc., as shown in the [Contacts Example](#) image. If you are creating a contact import file, see [Importing Contacts](#) for additional information.

3. Enter data for each record in a row.

**Note:** It is recommended that you import a small number of records to ensure that your data meets the guidelines.

## Data File Guidelines

The following guidelines apply for the first row:

- The first row cannot contain duplicate label names, even if the names have different cases.

For example, Address, address, and aDdRess are all considered duplicates.

- Label names cannot contain the following symbols:

< (less than)

> (greater than)

. (period)

& (ampersand)

' (single quote)

" (double quote)

[ (left bracket)

] (right bracket)

In addition, TeamConnect ignores macros and formulas and removes currency and percent symbols when importing data.

## Localized Labels

If you want to create a data file that contains localized data for system and custom lookup fields, assignee roles, and categories, enter the field's tree position instead of translating the field.

## Importing Categories

If you use a **Category** label in your data file, you can enter more than one category in the same field, separated by commas. The first category in the cell is automatically the **Primary** category.

For multiple categories, the **Category** cells may appear as follows:

A	B	C	D
First Name	Last Name	Company	Category
Beatrice	Wilk	Hill, Inc.	Employee, Account Dept.
Willie	Nels	Hill, Inc.	Employee, Attorney
Annette	Perkins	Hill, Inc.	Employee, Paralegal
Daryn	White	Hill, Inc.	Employee, Admin
Helen	Washington	Hill, Inc.	Employee, Account Dept, Manager

Categories Example

## Importing Contacts

Contact data files must contain a **Type** field for each entry. The options for the **Type** field must be one of the following:

- P** or **Person**.
- C** or **Company**.

This field is not case sensitive. See the Type field in the [Contacts Example](#) image for an example.

In addition, each entry that has a **Type** drop-down box, such as the **Address** or **Phone** label, must have a **Type** entry in the data file. For example, if you have an **Address** heading label in your data file, you must also have an **Address Type** heading label, as shown in the following image.

	A	B	C	D	E	F	G	H	I	J	K	L
1	First Name	Last Name	Company	Job Title	Type	Address	Address Type	City	State	Zip	Phone	Phone Type
2	Beatrice	Wilk	Hill, Inc.	Accountant	P	34009 Washington Pl	Business	San Diego	CA	90016	310-555-1215	Business
3	Willie	Nels	Hill, Inc.	Accountant	P	34009 Washington Pl	Business	San Diego	CA	90016	310-555-1216	Business
4	Annette	Perkins	Hill, Inc.	Sr. Accountant	P	34009 Washington Pl	Business	San Diego	CA	90016	310-555-1217	Business
5	Daryn	White	Hill, Inc.	Account Intern	P	34009 Washington Pl	Business	San Diego	CA	90016	310-555-1218	Business
6	Helen	Washington	Hill, Inc.	Account Supervisor	P	34009 Washington Pl	Business	San Diego	CA	90016	310-555-1219	Business
7												

Contacts Example

## Adding Secondary Field Labels

You can also add secondary information for the same label type, such as a secondary address for a contact.

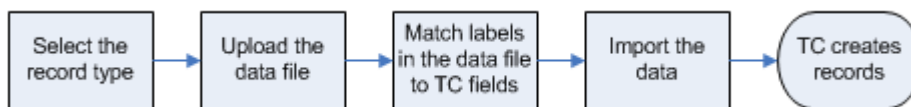
For example, for a secondary address, create and then populate labels similar to the following labels:

- address2
- city2
- state2
- zip2

Once you import the data file, you can use the **Add More Fields** option to [map these fields as secondary addresses](#).

### 1.2.6.13.2 Importing Data

When the data file is complete, use the Data Import Wizard to perform the following process.



Data Import Wizard Process

**Note:** The information that you see and the tasks that you can perform while using the import wizard depend on your access rights.

An incomplete import wizard appears in the list of **Unsaved Items** in the top-right of your browser. If you click away from the wizard at any time, click **Unsaved Items** to resume your work. The wizard no longer appears under **Unsaved Items** when you finish or cancel the wizard or log out of TeamConnect.

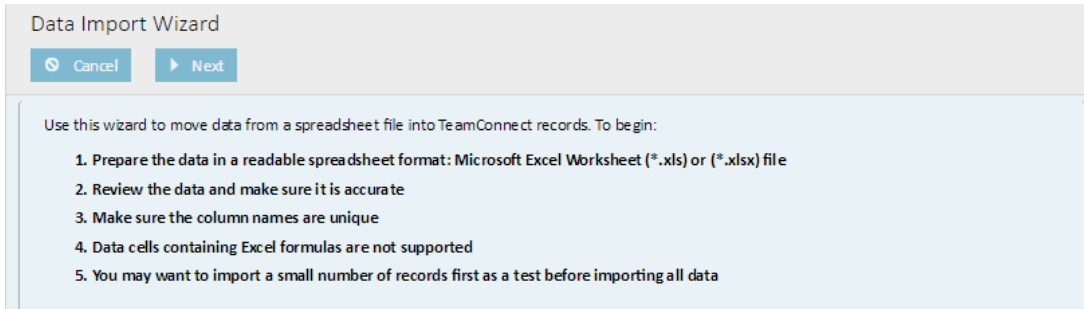
### To import records

**Note:** The following procedure assumes that you created or obtained a data file that meets the import guidelines. See [Creating a Data File](#) for more information.



1. Click the **Tools** button.
2. In the **Tools** section, click **Data Import Wizard**.

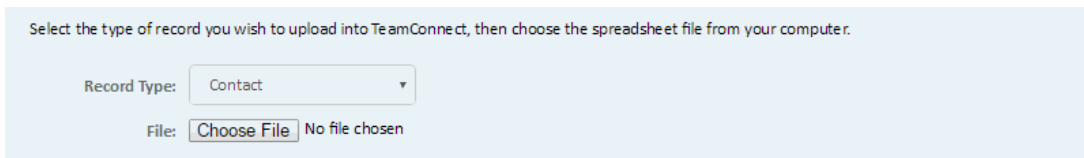
The **Data Import Wizard** opens to the first page.



**Data Import Tool, Page One**

3. Review the first page, and then click **Next** to continue.

The **Upload Spreadsheet File** page opens.



**Data Import Tool, Upload Spreadsheet File Page**

4. In the **Record Type** drop-down list, select the type of records that you want to create.
5. Click **Choose File** or **Browse** to locate the spreadsheet file that contains the data that you want to import.
6. Click **Next**.

TeamConnect searches system and custom field labels to match the imported labels to TeamConnect fields.

When the tool finishes analyzing the data, the **Map Record Fields** page opens.

The label names from your data file appear in the **Import Field** column. If TeamConnect finds a match, the suggested label names appear in the **TeamConnect Field** column.

Map the fields from your file to TeamConnect record fields. Please note that any fields left unmapped will not be imported into the system. Click Next to start importing records.

[Add More Fields](#) ▾

Import Field	TeamConnect Field
First Name	First Name ▾
Last Name	Last Name ▾
Company	Company ▾
Job Title	Job Title ▾
Type	Contact Type
Address	(Select) ▾
Address Type	(Select) ▾
City	City ▾
State	(Select) ▾
Zip	Zip/Postal Code ▾
Phone	(Select) ▾
Phone Type	(Select) ▾

Data Import Tool, Map Record Files Page

- For each incorrect or unmapped field, click the **TeamConnect Field** drop-down list, and select the correct label name.

If you added secondary fields or fields not in the drop-down list to the imported data, you can also [adding more fields](#).

- Click **Next**.

If you mapped a field twice, an error message appears. Correct the error, and click **Next** again to continue.

A **Confirmation** page opens that displays the number of records to be imported.

**Confirmation**

Your data is ready to be imported. Click Next to import.

**Number of records to be imported: 2**  
**Import Confirmation Page**

- Click **Next** to import the mapped data.

Importing may take several minutes, depending on the number of records you are importing.

During the import, TeamConnect ignores all blank lines and empty cells. Any custom rules that trigger upon record creation automatically execute.

When the import is complete, a confirmation page provides the details of the data import, including the number of records imported and the number of records that did not import because of errors. See [Data Import Wizard Errors](#) for how to troubleshoot errors.

- Click **Finish** to close the wizard. **Note:** After import, the Excel file will not be retained within the TeamConnect instance.

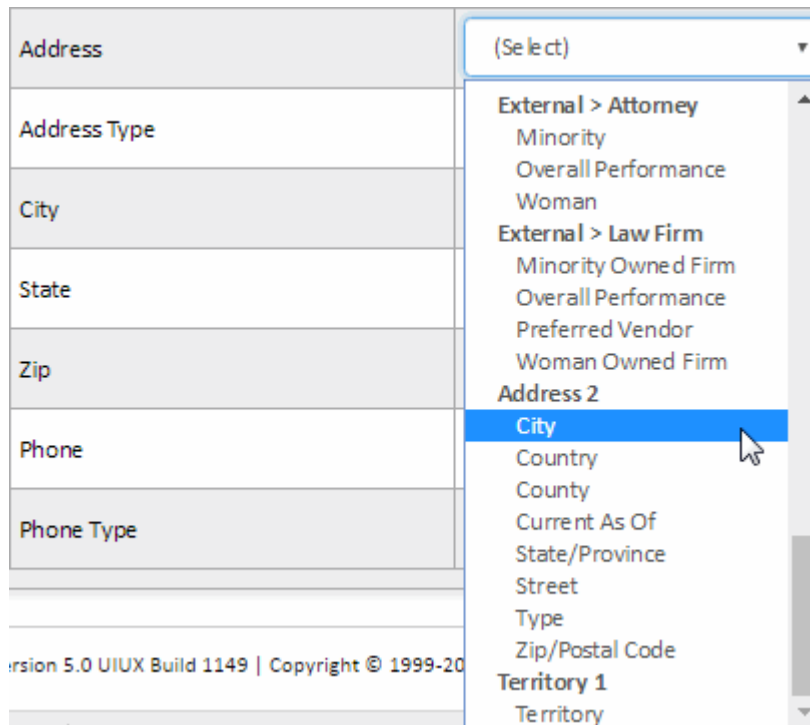
### To add more fields

To add more fields, click **Add More Fields**, and select the type of field that you want to add.

Add more fields in one of the following situations:

- You want to import secondary fields, such as a second address or phone number.

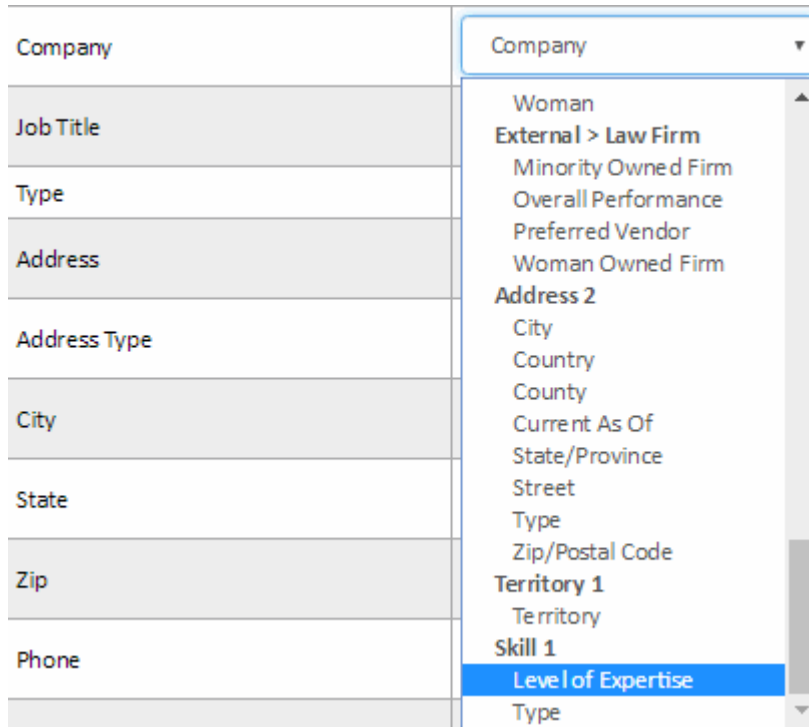
For example, if you want to add a second address, click **Add More Fields**, and then select **Add Address**. The secondary address fields appear as **Address 2** in all drop-down lists. You can now map these labels to the secondary address labels.



**Address 2 Appears After You Select Add Address**

For an additional group of fields, such as a third address, add more fields again.

- You want to import fields not currently in the drop-down, such as rates or skills. Select **Add Invoice Task Rate**, **Add Task Rate**, or **Add Skill** to add these types of fields to the drop-down list.



Skill 1 Appears After You Select Add Skill

For example, if you are importing contact records, the following table includes the mapping options for that record type.

**Adding More Fields For Contacts**

Add More Fields Selection	TeamConnect Field on Mapping Page	TeamConnect Field in Contacts Record
<b>Add Address</b> (for the <b>General</b> page)	City	City
	Country	Country
	County	County
	State/Province	State/Province
	Street	Street
	Type	Type
	Zip/Postal Code	Zip/Postal Code
<b>Add Default Rate</b>	From Date	Start Date

(for the <b>Rates</b> page)	Rate	Rate
	To Date	End Date
<b>Add Email Address</b> (for the <b>General</b> page)	Email Address	Email Address
	Email Address Type	Email Address Type
<b>Add Invoice Task Rate</b> (for the <b>Rates</b> page)	Category	Task
	From Date	Start Date
	Rate	Rate
	To Date	End Date
<b>Add Phone Number</b> (for the <b>General</b> page)	Phone Number	Phone Number
	Phone Number Type	Phone Number Type
<b>Add Relation</b> (for the <b>Relations</b> page)	Direction (Enter TO_THIS or FROM_THIS in the spreadsheet)	Switch Relation
	Related Contact	Contact
	Type	Relation
<b>Add Skill</b> (for the <b>Skills</b> page)	Level of Expertise	Level of Expertise
	Type	Type
<b>Add Task Rate</b> (for the <b>Rates</b> page)	Category	Task
	From Date	Start Date
	Rate	Rate
	To Date	End Date
<b>Add Territory</b>	Territory	Territory

(for the <b>Territories</b> page)		
<b>Add Web Address</b> (for the <b>General</b> page)	URL	Web Address
	Web Address Type	Web Address Type

The selections for **Add Address**, **Email Address**, **Fax Number**, **Phone Number**, and **Web Address** are all for adding secondary fields. The primary fields already exist in the drop-down.

## Data Import Wizard Errors

If all records do not import, you can troubleshoot the errors that appear in the spreadsheet TeamConnect returns.

For example, if TeamConnect checks for duplicate contact records before creating a new record, the system marks the duplicate record as failed and logs the error.

### To troubleshoot Data Import Wizard errors

1. From the **Import Confirmation** page, click **Download spreadsheet file with failed rows** to review them.
2. Open the spreadsheet to display explanations for the rows that did not import.
3. Use the error explanations to make corrections.
4. Repeat [the import process](#).

**Note:** To save time and to avoid duplicate records, re-import only the data that did not previously import.

## 1.2.7 Deleting Records

Pages that are used to view a single record may also contain a button labeled **Delete**. Click **Delete** to submit a request to delete that record.

**Note:** The **Delete** button is unavailable on pages that are used to edit a record.

After clicking **Delete**, you are shown a confirmation message box that asks whether you truly want to delete the record. Click **Yes** to proceed. A success message is displayed at the top of the page and the record is then deleted from the database, subject to the exceptions below.

There might be workflow rules associated with a deletion. In this case, although your deletion request has been accepted, the record would not disappear from the database immediately, but would remain while the deletion request is pending approval. If the request is approved, the record disappears from the database. If the deletion request is rejected, the record will not be deleted.

Sometimes a deletion request will be denied because the record is linked to other records. An error message will appear when this happens. That message means that the link must be removed, or the other records must be deleted, before you can delete the current record.

Deletion of a record might cause other records to be deleted or altered. To see what effects deletion can have, refer to the table below.

#### Deleting Records

Record Type	What Deletes	Conditions that Prevent Deletion
<b>Accounts</b>	<ul style="list-style-type: none"> <li>Child accounts</li> <li>Associated transactions in expense, task, and invoice records</li> <li>Associated approvals</li> <li>Associated documents and folders</li> <li>Associated history records</li> </ul>	<ul style="list-style-type: none"> <li>The account balance is other than zero.</li> <li>The account has an attached document that is currently checked out.</li> <li>The Accrual amount is other than zero.</li> </ul>
<b>Address Books</b>	<p>The address book and all of its contacts.</p> <p>Only the book is deleted; contacts remain in the system.</p>	n/a
<b>Appointments</b>	<p>Appointment is removed from the calendar along with all of the information on the additional calendar pages.</p> <p><i>Note: You can delete appointments in List view.</i></p>	If a document that is attached to an appointment is currently checked out, you cannot delete the appointment.
<b>Contacts</b>	<ul style="list-style-type: none"> <li>All associated addresses, phone numbers, and all of the information on the additional contact pages</li> <li>All relations with projects and other contacts</li> <li>Associated approvals</li> <li>Associated Involved records</li> <li>Associated documents and document folders</li> <li>Associated history records</li> </ul>	<ul style="list-style-type: none"> <li>If listed as a contact in an expense record</li> <li>If listed as a contact in a task record</li> <li>If listed as a vendor in an invoice record</li> <li>If listed as a timekeeper in an Invoice Line Item record</li> <li>If listed in an account record as one of the following: <ul style="list-style-type: none"> <li>General - vendor</li> </ul> </li> </ul>

		<ul style="list-style-type: none"> <li>○ Posting Criteria - involved</li> <li>○ Posting Criteria - vendor</li> <li>○ User Account records (on the Admin side)</li> <li>● If the contact has a document checked out</li> </ul>
<b>Documents</b> <b>Document Folders</b>	See <a href="#">Deleting Documents and Document Folders</a> .	
<b>Expenses</b>	<ul style="list-style-type: none"> <li>● Associated approvals</li> <li>● Associated documents and folders</li> <li>● Associated history records</li> </ul>	<ul style="list-style-type: none"> <li>● The expense has an attached Document that is currently checked out</li> <li>● The expense is currently posted to an account</li> </ul>
<b>Invoices</b>	<ul style="list-style-type: none"> <li>● Associated Line Items if they have not posted to an account.</li> <li>● Associated approvals</li> <li>● Associated documents and folders</li> <li>● Associated history records</li> </ul>	<ul style="list-style-type: none"> <li>● The invoice has an attached Document that is currently checked out</li> <li>● At least one of the invoice Line Items is posted to an account</li> </ul>
<b>Projects</b>	<ul style="list-style-type: none"> <li>● Assignees, phases, and other contents of the record</li> <li>● Relations with other projects</li> <li>● Associated involved records</li> <li>● Associated approvals</li> <li>● Associated documents and folders</li> <li>● Associated history records</li> <li>● Accounts that use this project as a posting criterion</li> <li>● Appointments created for this project</li> <li>● Associated Expense and Task records</li> </ul>	A document on the <b>Documents</b> page for this project is checked out. See <a href="#">Editing Documents</a> for more information.



	<ul style="list-style-type: none"> <li>• Associated Invoice line items</li> <li>• All child projects that have this project selected as their parent</li> </ul>	
<b>Reports</b>	The report is deleted from the list view, all collections, and folders.	<ul style="list-style-type: none"> <li>• The user does not have system rights to delete reports</li> <li>• The user does not have record level rights to delete a specific report.</li> <li>• The record is currently locked for editing by any user</li> <li>• The report is executing</li> </ul>
<b>Tasks</b>	<ul style="list-style-type: none"> <li>• Approvals</li> <li>• Documents and folders</li> <li>• History records</li> </ul> <p><i>Note: You can delete many tasks in one operation. See <a href="#">Using the Task Entry Tool</a> for more information.</i></p>	<ul style="list-style-type: none"> <li>• The task has an attached document checked out.</li> <li>• The task is currently posted to an account</li> </ul>

### 1.2.8 Printing Records

Clicking the **Printable View** button in the toolbar will create a separate browser window with a printer-friendly report of the information you are currently viewing. The button behaves differently depending on whether you are viewing a list of records or viewing an individual record.

#### To print a list of records

1. Click **Printable View** in the toolbar.  
The **Printable View** window appears.
2. (Optional) Enter a title for your report.
3. Indicate whether you would like filter information to print on the report.  
Filter information indicates what search restrictions were used when originally generating the list of records.
4. Click **OK**.  
The **Printable View** window closes. A new browser window appears, displaying your record list.

5. In the new browser window, click **Print This Page**.
6. Click **Close This Page**.

#### To print an individual record

1. Click **Printable View** in the toolbar.
2. The **Sections** window appears.

All of the possible information sections related to your record are shown. They are all presently selected with a check-box. Uncheck any sections that you do not want to print.

The check-box next to the word **Sections** will select, or de-select, all of the individual sections simultaneously when you click it.

Sections that are marked with an asterisk require more retrieval time. You can still print these sections, but response time will be longer than if you did not include them.

3. Click **OK**.  
A new browser window or tab appears, displaying your record's information.
4. In the new browser window or tab, click **Print This Page**.
5. Click **Close This Page**.

### 1.2.9 Working with Record Security

From the Security page, you can perform the following tasks:

- [Viewing Record Security](#)
- [Editing Record Security](#)

There are two kinds of security. The first is a comprehensive security feature, used by system administrators, that [assigns rights to user groups](#). These rights are the ones that determine whether you can view, edit, create, or delete records for a specific record type.

This section will describe the second kind of security - record security that can be assigned by an ordinary user and applies only to one single record at a time.

Examples of when you may need to set record security include:

- Public records need to be made accessible to limited-privilege users.
- Public records need to be made inaccessible to specific groups.
- Private records need to be made accessible to specific users or groups other than the user who created the record.

**Public** records are available to all users, and groups must be added to make the records inaccessible.

**Private** records must have all users or groups added to its permissions in order to make the record accessible.

More information on Public vs. Private record security can be found [here](#).

The security features described here have no effect on a user whose level of access is "superuser" rather than "normal" or "limited". A superuser can manipulate any record, even if you explicitly deny them access to a record in the record's **Security** page.

Users who create a record typically have access to it, regardless of the settings on the **Security** page—provided they have the right to read that record type. Similarly, all project and task assignees and appointment attendees are automatically granted the rights to read, update, and delete the records.

Depending on your organization's security settings, related records of projects may automatically inherit the same security settings as their parent project. For example, if a parent project's security is set so that you cannot delete it, then you may not be able to delete its related appointments, histories, and tasks, even if you are the user who created those related records.


Similarly, while a parent project is pending approval, you may not be able to modify that parent project or any of its related appointments, histories, tasks, and other options, depending on the administrative security settings.

The individual record security feature cannot be used to override the comprehensive security feature. For example, if you explicitly grant View rights for a Contact record to group "reviewers", but "reviewers" does not have View rights for the Contact record type, then that group's users would still be unable to view that Contact record.

The ability to control security for an individual record is a set of two rights (**view security** and **edit security**) that are granted to you by your system administrator. If you do not have these rights, you will not be able to see or use the **Security** page of a record.

The individual rights that can be assigned on the **Security** page are:

#### Record Security Rights

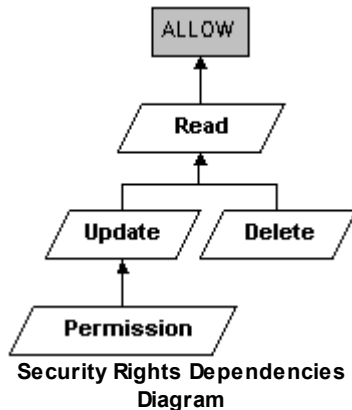
Right	Description
<b>Read</b>	<p>Allows the users in a group to view or read the record. If a user or group is denied this right, the record will be invisible to them (will not appear in lists or search results.)</p> <p><b>Important:</b> If you grant any of the other three rights to a user or group, you automatically grant them the right to read the record. See <a href="#">the security rights dependencies diagram</a>.</p>
<b>Update</b>	<p>Allows the users in a group to edit and change the record.</p> <p>If a group has Read rights to a record, but does not have Update rights, the record will be included in lists and search results, but a small image of a lock  will appear where the edit icon would normally appear. This tells the user that the record cannot be edited.</p>
<b>Delete</b>	<p>Allows the users in a group to delete the record.</p>

**Perm  
(Permission)**

Allows the users in a group to set the record security through its **Security** tab to grant or deny access rights to other users.

Rights are somewhat dependent on each other. For example, if you want to grant a Permission right, to edit the record security, you must also assign the Read and Update rights. If you want to assign the Update right, or the Delete right, you must also assign the Read right. Note that the application will automatically assign the dependent rights when you select a right that has dependencies.

The following diagram illustrates the dependency of the record rights when you want to allow any of them to a user or group. The arrow in the diagram indicates that the right it points FROM requires the right it points TO.



When denying access rights, the dependencies cause slightly different effects. By default, if you deny a group the Read right, the rest of the rights for the record are automatically denied. Also, if you deny a group the Update right, the Permission right is denied. Again, the application automatically sets the other dependent rights when you select an individual right.

### 1.2.9.1 Viewing Record Security

While viewing a record, click on the **Security** link in the left pane.

#### To view the two Security page sections

1. Privacy determines whether the record behaves as **public** or **private**. If it is public, the record will be shown or hidden based on the rights that have been assigned to the overall record type. If it is private, the record will be hidden from all users except the user who created it. However, there can be exceptions to those behaviors, and the exceptions are determined by entries in the **Group Rights** section.
2. **Group Rights** contain exceptions to the public/private behaviors, organized by user group. For example, if a record is private, you could enter an exception that would allow a specific group to edit that record, while all other groups would still be unable to see the record.

Field Name	Description
------------	-------------

<b>Option</b>	Indicates whether this row Allows or Denies specific rights to the record.
<b>Group</b>	The user group to be affected by these rights.
<b>Read, Update, Delete, Perm</b>	Four fields, each containing <b>Yes</b> or <b>No</b> , to indicate which rights are being allowed or denied. See a detailed explanation of these rights in <a href="#">Record Security Rights</a> .

See [Working with Record Security](#) for more information.

### 1.2.9.2 Editing Record Security

While editing a record, click on the **Security** link in the left pane.

See [Working with Record Security](#) for conceptual information.

#### To edit the Security page sections

1. Privacy determines whether the record behaves as public or private. Click the appropriate option button to make that determination.
2. Group Rights contain exceptions to the public/private behaviors, organized by individual users. Enter information as described below.

Field Name	Description
<b>Option</b>	Choose whether to Allow or Deny specific rights to the record.
<b>Group</b>	Select a user group to be affected by these rights.
(check-boxes)	A group of check-boxes determines what kind of rights are being allowed or denied. Click the boxes that should be affected. See a detailed explanation of these rights in <a href="#">Record Security Rights</a> .
<b>Action</b>	Click the minus icon to delete the associated row. Click the plus icon to add a new row of information.

3. Click **Save**.

### 1.2.9.3 Public and Private Record Security

Records can be set to either **Public** or **Private** security. These settings can be accessed via the **Security** link on the left side navigation pane of any record.

Public records automatically allow all AllRights users to view the record. Only limited-privilege users need to be added to view the record. Similarly, you must specifically add groups or users with AllRights to "Deny" in order to block these records from those users.

Contact - Holden, Cody



Save Cancel & View Create a Copy More Actions Share Printable View Help

Add Bookmark

**Security**



Privacy:  Public  Private

**User Rights**

<input type="checkbox"/>	Line	Option	User	Read	Update	Delete	Perm	Action
<input type="checkbox"/>	1	Allow	Holden, Cody	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	 

Remove Add New Item

**Group Rights**

<input type="checkbox"/>	Line	Option	Group	Read	Update	Delete	Perm	Action
<input type="checkbox"/>	1	(Select) ▼	(Select) ▼	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	 

Remove Add New Item

#### Public Record Security

Private records default to only let the record creator view the record. You must specifically add groups or users to Allow and select their Read, Update, and Delete permissions to make this record accessible.

### 1.2.10 Working with Record History

When working with record history, you can perform the following tasks:

- [Opening History Pages](#)
- [Viewing General History Information](#)
- [Creating Record History Entries](#)

History records reflect chronology or history of the record to which they are related. Every history record is related to a parent record - there are no standalone history records. History entries may contain such information as payments made by involved parties, and automated assignments. History records of a project may register such things as when the main assignees were changed and the dates when a lawsuit was filed. History records may be referred to as docketing or calendaring, narratives, logs, diaries, notes, or journals.

History records can be generated automatically or created manually. One of the more common uses of manual history records is to enter narratives associated with legal matters. For history entries that track changes made to a parent record, automated rules typically generate the appropriate history entries automatically when certain events occur in the parent record.

Most records have their own **History** link where you can create, search for, view, and delete all necessary history entries pertaining to the record. The **History** page behaves like a regular collection, and has a button that allows you to add new manual history entries.

From the **History** page on a record, change the **View** for the page.

#### Views on the History Page

View	Description
<b>Recent History</b>	Displays history entries associated with this record.
<b>Recent Narratives</b>	Displays recent matter narratives associated with this record.
<b>By Date Range</b>	Displays a search filter for the entries associate with this record.

#### 1.2.10.1 Opening History Pages

Most records have their own **History** list page where you can create, search for, view, and delete all necessary history entries pertaining to the record. The **History** page looks and functions like a regular collection. Each history entry also has its own set of pages for displaying and editing its information.

You can access the history page of a record in the following ways:

- Through the **Histories** link in the **All** navigation pane. You can use this option when you want to search through all history entries in the database.
- Clicking the **History** link of a record while viewing or editing that record. This is a convenient way of viewing all history information pertaining to that specific record.

#### To open the pages of a history entry

1. Click the **History** link in the left pane while viewing the record.
2. Find the desired history record by examining the list of results. Click on the record's name.
 

If the parent record has a large number of history entries, and you do not wish to examine a large list to find the specific history entry you are interested in, you can click the **Show Filter** link to request a more filtered list of entries.
3. The corresponding History entry page appears in Read-only mode with its **General** page displayed by default.
4. To change information in the entry, click **Edit** on the record toolbar to make the read-only fields editable.
5. Click the link related to the information you would like to view or change.

## History entry pages

History entries typically have the following links in the left navigation pane:

- **General**—Displays the general information of the history entry. See [Viewing General History Information](#).
- **Categories**—Displays the categories added to the selected history entry. For more background information, see [Setting Category Details](#).
- **Documents**—Displays documents associated with the selected entry. See [Viewing Documents](#) for more details.
- **Security**—Displays users and user groups who are given or denied access to the selected record. See [Viewing Record Security](#) for more details.
- **Workflow**—Displays the approvals or rejections log for the history entry.

### 1.2.10.2 Viewing General History Information

General history information is typically entered and displayed on the **General** page of the history entry. It includes the date and time when the entry was made, its default category, the name of the user who made the entry, the details of the entry, and the hyperlink to the record that created it.

The following table describes the items on the **General** tab of history entries.

**General Page of a History Entry**

Field or control	Description
<b>Date</b>	Type the date or click the <b>Calendar</b> icon to specify the date on which the history entry is created.  By default, the system displays the current date.
<b>Time</b>	Type the time or click the <b>Time</b> icon to specify the time in which the history entry is created. By default, the current time is displayed.  By default, the system displays the current time that is read from the server.
<b>Description</b>	Type the appropriate details of the history entry. Make sure they are concise and to the point.  <i><b>Note:</b> The first 250 characters are displayed as the <b>Description</b> for this entry in the search results.</i>
<b>Parent Record</b>	Automatically displays a hyperlink to the record within which the history entry is created.



### 1.2.10.3 Setting History Categories

Companies often find it useful to organize their history entries by categories. History categories are certain events that necessitate making the corresponding history entries. For example, in litigation, categories can be depositions, pleadings, amendments to petitions, notice of litigation, and notice of termination. These may include subcategories. All categories can be added, viewed and deleted on the **Categories** page of the history entry.

#### To set history categories

1. Click on the **Categories** link in the left pane of the history entry page.
2. Choose categories for this entry as described in [Setting Category Details](#).

### 1.2.10.4 Creating Record History Entries

History entries are often generated automatically. Automatically generated history entries are added to the appropriate records according to the rules set by your administrator. However, you can also create history entries manually. This section describes that procedure.

#### To create a history entry for a record

1. Open the record where you want to make a history entry (for example Appointment, Task, or Claim).
2. Click the **History** link in the left pane.  
Your **History** search results appear.
3. Click **New**.  
The **General** page of a new **History** entry appears with the blank fields displayed for you by default.
4. Enter the appropriate values for the new History entry as described for the [General page of a History entry](#).
5. (Optional) Add other appropriate information on the other pages.
6. **Save** the entry, as described more generally in [Saving an Edited Record](#).

#### Points to Remember

- Any History entry can be made only from its "parent" record to which it is related.
- The information in the **Description** field should be brief and to the point so that users can easily distinguish among different history records in a list of search results.

## 1.2.11 Working with Related Records

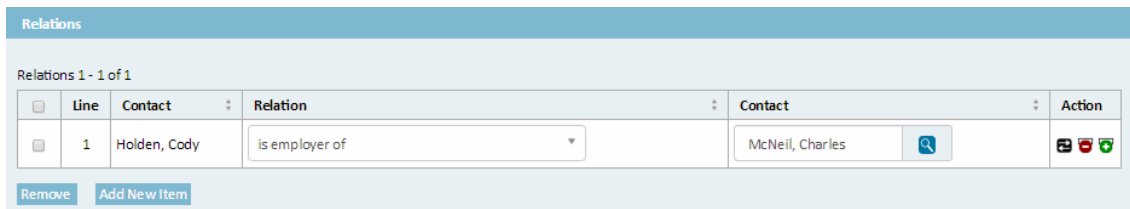
Records can be related in various ways. When two records of the same record type are associated with each other, they are related, and the relationships can be seen on the **Relations** page of each record. You can only relate records of the same type on the **Relations** page. For example, when two Contacts are associated with each other, such as one Contact is the supervisor of another Contact, that association appears on the **Relations** page.

If the record types are different, the association between them is shown on a page other than Relations. For example, when two Contact records are associated with an Appointment record, the two contacts are listed on the **Involved** page of the Appointment, not the **Relations** page, because Contact and Appointment are different record types.

A record can have none, one, or multiple relations. Contacts have relations, as do all custom record types, and the Project record type. Other record types do not have the Relations feature.

### To view relations

While a record is open, click the **Relations** link in the left pane. The **Relations** page opens and displays existing relations. Clicking the **Edit** link on this page allows you to set relations for this record.



The screenshot shows the 'Relations' page for a record. At the top, it says 'Relations 1 - 1 of 1'. Below this is a table with the following structure:

Line	Contact	Relation	Contact	Action
1	Holden, Cody	is employer of	McNeil, Charles	[Remove] [Save] [Add]

Below the table are two buttons: 'Remove' and 'Add New Item'.

A Contact Record with One Existing Relationship to Another Contact

### To set relations

To set relations for a record, go to the **Relations** page for that record. If you are in view mode, click the **Edit** link to enter edit mode.

### To add a new relation

Click the **Add New Item** button on the **Relations** page. Alternately, you can click the plus button on an existing row. In the new row that appears, [fill out the fields](#). Then click a save option.

### To delete an existing relation

On the **Relations** page, click the check-box next to a Relation, then click the **Remove** button. Alternately, you can click the minus button on a Relation row. Then click a **Save** option.

## 1.2.12 Working with Record Documents

Every record can have documents associated with it. Documents are external files, such as images or word processing files, that are uploaded into the database. There is a **Documents** tab in the main tab bar which lets you search for and view documents across the entire database. There is also a **Documents** link in the left pane of each record type. This link takes you to a **Documents** page that lists only documents that are associated specifically with the record that you are currently viewing or editing.

Details of how to use the **Documents** page to manage your record's documents can be found in [Viewing Documents](#).

## 1.2.13 Viewing Workflow for a Record

Workflow is the process of approving and rejecting requests in TeamConnect. A request originates when you perform an action that is subject to one or more rules. The rules that have been defined for your system determine the workflow processes that take place while you work in TeamConnect. A request for approval is automatically sent to one or more people according to a predefined sequence.

For more details about how to work with approval requests, see [Workflow](#).

## 1.3 Contacts and Address Books

The [Contacts](#) area is a central storage space for names, addresses, phone numbers, and other details about people and organizations.

You can also organize contacts into address books that you can use to quickly view information about all of the contacts in the group, or send messages to a group.

### 1.3.1 Contacts

Contacts within TeamConnect are individuals, companies, or organizations for which your organization needs to store detailed information. Contacts may include vendors, outside counsel, claimants, injured parties, witnesses, agencies, and internal employees.

#### Types of Contact Records

TeamConnect contains two general types of contacts records:

- **Persons**—Individuals that are internal or external to your organization.
- **Companies**—Organizations, such as external agencies, outside counsel law firms, or internal subsidiaries of your company.

#### Types of Contact Members

TeamConnect contact records contain the following types of members:

- **Contacts who are TeamConnect users**—Based on their assigned rights, these users have access to TeamConnect and can be assigned to tasks or projects, and added as attendees to appointments. These contacts are usually internal employees.
- **Contacts who are not TeamConnect users**—These users do not have access to TeamConnect and may include internal and external contacts, for example employees, clients, outside companies, or vendors.

Each contacts record may have multiple sets of contact information, and, for easy access, you can enter all of the information on the same record.

The following table shows how you can view, create, and edit contacts.

...Contacts		How...
Viewing		<ul style="list-style-type: none"> <li>• <a href="#">Viewing the Address Books for a Specific User</a></li> <li>• <a href="#">Viewing Company Employees</a></li> <li>• <a href="#">Viewing Contact Involvement</a></li> </ul>
Creating		<ul style="list-style-type: none"> <li>• <a href="#">Creating a Contact</a></li> <li>• <a href="#">Duplicate Contacts</a></li> </ul>
Updating	General Information	<ul style="list-style-type: none"> <li>• <a href="#">Setting Contacts Primary Information</a></li> <li>• <a href="#">Setting Contact Details</a></li> <li>• <a href="#">Adding and Removing Contacts Categories</a></li> </ul>
	Hourly Rates	<ul style="list-style-type: none"> <li>• <a href="#">Setting Contact Rates</a></li> </ul>
	Related Contacts	<ul style="list-style-type: none"> <li>• <a href="#">Relating Contacts</a></li> </ul>
	Skills	<ul style="list-style-type: none"> <li>• <a href="#">Setting Contact Skills</a></li> </ul>
	Territories	<ul style="list-style-type: none"> <li>• <a href="#">Selecting Contact Territories</a></li> </ul>
	Security	<ul style="list-style-type: none"> <li>• <a href="#">Editing Record Security</a></li> </ul>

### Best Practices for Contacts

- If you do not see any contacts when you click the **Contacts** tab, click the **All Contacts** link or any of the other page links in the left pane.
- To create a new contact for a company, make sure you click the **New** drop-down and select **New Company**.
- Add a contact to an address book from the address book, not the contact record.

### 1.3.1.1 Types of Contact Views

You can view contacts in a list view or card view. The default view is always the last view that you used to display contacts.

You can also view contact records in a collection or search for records. See [Modifying a Collections Search](#) and [Searching for Specific Records](#).

#### Contact List View

The contacts list view lets you view the contacts that you have the rights to view in a list format.

##### To view contacts in a list view

1. Click the **Contacts** tab.
2. If the list view does not appear by default, in the **Display** drop-down list, select **List View**.

*Tip: If you use your mouse to hover over a contact in any view, a pop-up window displays the primary contact details. This function is customizable by administrators as detailed in [Custom Hover Popups](#).*

The first time that you try to view contacts, it may appear that there are no contacts available in the list view because the list view defaults to the **Recently Viewed** collection.

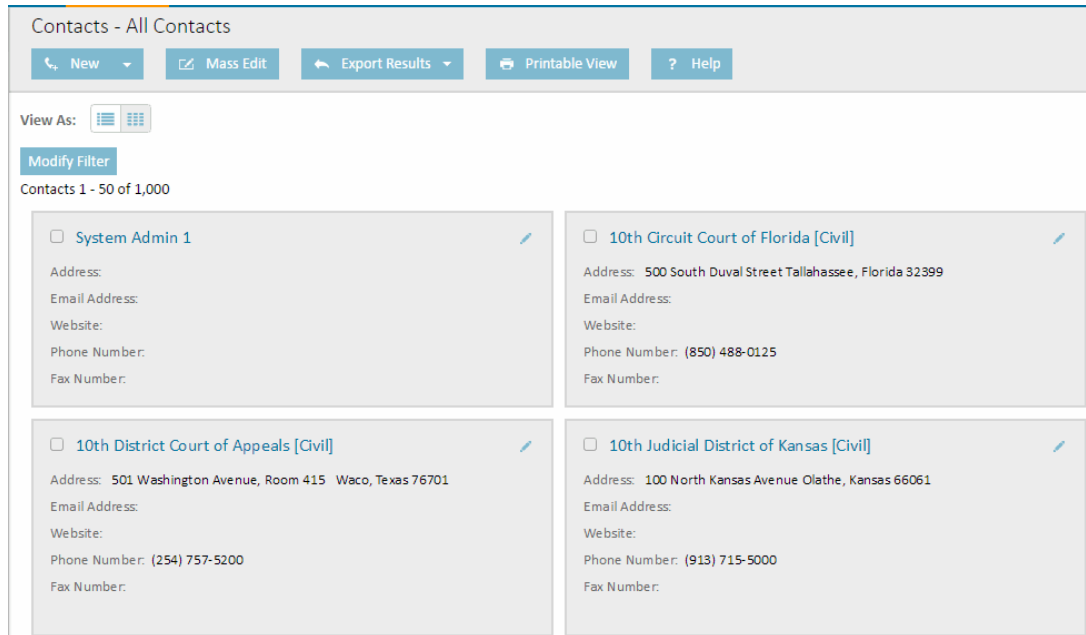
To view all of the contacts that you have the rights to view, click the **All Contacts** link in the left pane. The next time that you access projects, the list view defaults to the last collection that you opened.

#### Contacts Card View

The card view lets you view contacts in a "business card" view that displays all of the primary information that appears on the General page.

##### To view contacts in a card view

1. Click the **Contacts** tab.
2. If the card view does not appear by default, click the **Card View** button (located above the **Modify Filter** button).



**Contacts Card View**

For information about the specific areas in the **Contacts** list view, see [User Interface](#).

### To open contact records in other areas

You can open contact records in the following locations in TeamConnect:

- **General** page in the **Access Information** section—Click on the link in the **Created by** or **Modified by** fields. See [General page field descriptions](#) for more information.
- **Projects**—On the **Assignees** page for a project. See [Viewing Users Assigned to a Project](#).
- **Global Documents** page or **Documents** page for a record—Any field on the **General** page that contains a user name, such as **Record** or **Author**. See [Viewing or Editing General Information of a Document](#).
- **Tasks**—In the Tasks list view, click on a name in the **Assignee** column.

#### 1.3.1.2 Creating a Contact

You can create a contact in one of many ways:

- Using a wizard—Your system administrator may provide one or more wizards to help you create new contacts using a guided, step-by-step process. If you click **New** and a wizard opens, complete the required fields on each page, and then click **Next** to progress to the end of the wizard. When you click **Finish** at the end of the wizard, the record is available in the list view.

Your company may only allow you to create a contact using a wizard, so the option to create a contact manually may not be available.

- Copying an existing contact—TeamConnect allows you to create a contact by copying an existing contact, which is useful if you have contact records that may have different names, but similar properties. See [Copying a Record](#) and [Duplicate Contacts](#) for more information.
- Entering new contact information from the **General** page—If you click New and the General page opens, enter contact record information, as shown in the following procedure.

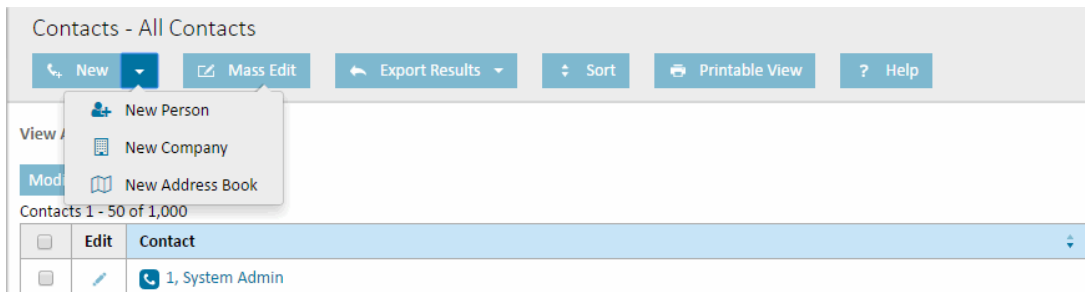
Once you create a contact, with the appropriate rights, you can edit the information in the contact record. See [Editing a Record](#) for more information.

**Note:** Before you create a new contact, it is recommended that you perform a search to avoid creating duplicate contact records. See [Searching for Specific Records](#).

### To create a contact

1. Click the **Contacts** tab.

The Contacts list view opens.



Contacts List View

2. Click the **New** drop-down, and then select **New Person** or **New Company**.

The screenshot shows the 'New Person' form with the following sections:

- Personal Information:** Fields for Prefix, \*First Name, Middle Name, \*Last Name, Suffix, Job Title, and Company.
- Address Information:** Fields for Type (Business), Street, City, State/Province, Zip/Postal Code, Country, and Current as of.
- Phone Number:** Field for Business phone number.
- Fax Number:** Field for Business fax number.
- Email Address:** Field for Business email address.
- Web Address:** Field for Business web address.
- Categories:** Instructions to select categories and a 'Set a Primary Category' dropdown menu.

### Contacts General Page

3. On the **Profile** page, enter or edit the appropriate information. See the [General page field descriptions](#) for more information.

Fields with an asterisk are required and must be completed before you can save this record.

4. To enter multiple addresses, phone, fax, web, and email addresses, select the + sign in the appropriate information block, and then enter the appropriate information.

If you are entering multiple addresses, phone, fax, web and email addresses, a **Primary** radio button appears. Click the radio button that indicates the main information for this contact.

5. Use the links in the left pane to access the contact-specific pages:

- **Details**—Enter or edit additional identification information, such as an employee ID, social security number, or Tax ID number.

Custom fields may also appear on this page based on the selections you make on the **Categories** page. See [Setting Contact Details](#) and [Setting Category Details](#).

- **Rates**—Enter or edit the fees charged by a vendor for services rendered to your company. Rates may be automatically referenced when you are working with tasks and invoices. See [Setting Contact Rates](#).
- **Skills**—Enter or edit the skills that a contact possesses and indicate the level of expertise for each skill. See [Setting Contact Skills](#).
- **Territories**—Enter or edit the geographic areas where the selected contact conducts business. See [Selecting Contact Territories](#).



- **Involvement**—Add or edit a non-TeamConnect contact's role in a project, such as respondent, claimant, or witness. See [Viewing Contact Involvement](#).
6. If necessary, use the additional contact pages to enter or edit supplementary information, such as historical details or relationships between contacts.
  7. [Select a save option](#).

**Note:** If you get a Duplicate Contact message, see [Duplicate Contacts](#) for more information.

The tables below contain the default fields for both person and company contacts. The fields that you see may vary based on your system settings and assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

#### Person and Company Information

Field (* = required)	Description
<b>Company</b>	The organization associated with this contact. Click the search icon, and then complete the search criteria to locate the appropriate company. If you cannot find the company that you want, click <b>New</b> to create it.
<b>*Company Name</b>	The name of the company for which you are creating the contact record.
<b>*First Name</b>	The contact's first name.
<b>Job Title</b>	The contact's job title. Maximum length = 250 alpha-numeric characters.
<b>*Last Name</b>	The contact's last name.
<b>Middle Name</b>	The contact's middle name.
<b>Prefix</b>	The contact's salutation, such as Mr., Mrs., or Ms.
<b>Suffix</b>	The contact's suffix, such as Jr., III., or M.D.

#### Addresses

Multiple addresses are allowed for a contact.

Field	Description
-------	-------------

<b>City</b>	The name of the contact's city.
<b>Country</b>	The name of the country that applies to the address in this block.  This order of options in this list is based on your company preferences, so the countries may not be in alphabetical order.
<b>County</b>	The name of the county that applies to the address in this block.
<b>Current as of</b>	The date on which this address information is up-to-date.
<b>State/Province</b>	The abbreviation for the state or province that applies to the address in this block. Click the icon next to this field to select the appropriate state code or province, or enter the information manually. The items in this list are based on the <b>Country</b> you select and your company's preferences.
<b>Street</b>	The street number and name.
<b>Type</b>	The type of address, such as home or business.
<b>Zip/Postal Code</b>	The zip code that applies to the address in this address block.

#### Phone and Fax Numbers

Multiple phone and fax numbers are allowed for a contact.

Field	Description
<b>Number</b>	Type the phone number.  <b>Important:</b> Use a consistent format for all contacts, such as (nnn) nnn-nnnn.
<b>Type</b>	Select the phone number type, such as home or business.

#### Email Address

Multiple email addresses are allowed for a contact.

Field	Description
<b>Email</b>	The email address for this contact, such as yourname@yourcompany.com.

	<b>Note:</b> You can enter up to 250 alphanumeric characters, but only 28 characters will be visible in the field. To see the rest of the address, you have to scroll to the right.
<b>Type</b>	Select the email address type; for example, business or personal.

**Web Addresses**

Multiple web addresses are allowed for a contact.

Field	Description
<b>Address</b>	The Internet address for this contact, such as www.yourcompanyname.com.  You can enter up to 250 alphanumeric characters, but only 28 characters will be visible in the field. To see the rest of the address, you have to scroll to the right.
<b>Type</b>	The type of the Internet address, such as, business or personal.

**Access Information**

Field Name	Description
<b>Created By</b>	The name of the user who created the contact record.
<b>Created On</b>	The date that the contact record was added to TeamConnect.
<b>Modified By</b>	The name of the user who last modified the contact record.
<b>Modified On</b>	The date on which the contact record was modified.

**1.3.1.3 Duplicate Contacts**

If your system administrator enabled a duplicate contact check, when you select a save option, you may see a warning and a list of possible duplicate contacts.

Do one of the following:

- Review the duplicates to ensure that your contact is unique. If your new record is a duplicate, click **Cancel**.
- If you still want to save the new contact record, click the **Allow contact to be saved** check-box and complete the appropriate fields. See [General page field descriptions](#) for more information.

### 1.3.1.4 Setting Contacts Primary Information

Use the Contacts **Profile** page to designate the main contact information for a person or company.

Contacts can have several mailing, email, and Internet addresses, as well as several phone and fax numbers. Typically, the item that you add first in a section is set as the primary, so if you have multiple addresses and numbers, it is important that you designate which information is the main information.

The primary items in each section are marked with a check mark in a circle, and this information is what is displayed on both the list view and card view.

You can change the designated primary entry at any time.

#### To set contact primary information

1. Search for a contact and open it.
2. In the left pane, click the **Profile** link.
3. Click the **Edit** pencil icon next to the section of information you would like to edit.
4. If necessary, click the **+** sign to add an empty text block, and then enter the new information.
5. Select the radio button next to the information that you want to designate as the primary information.

The screenshot displays the 'Address Information' section of a contact's profile. It contains two address entries. The first entry is for a business at '1234 Road Rd., New York, New York'. The second entry is for a business at '9876 Street St., New York, New York'. The second entry is designated as the primary address, indicated by a checkmark in a circle and a white arrow pointing to it. Each entry includes fields for Type, Street, City, State/Province, Zip/Postal Code, County, and Country, along with a 'Current as of' date field and a status icon.

Contacts General Page with Primary Designation

6. [Select a save option.](#)

### 1.3.1.5 Setting Contact Details

Use the Contact **Profile** page to view or edit additional contact identification information.

## How Categories Affect the Details Page

In addition to the default fields, additional fields may appear based on the options you select on the **Categories** page and your access rights. These additional fields are called custom fields.

For example, if you select a category named *Adjuster* on the **Categories** page, and then return to the **Details** page, you may see fields that allow you to select the states in which the company is licensed, or the limits on payments that can be made on a claim without a supervisor's authorization. For a vendor, these fields are not relevant, so if you select **Vendor** on the **Categories** page, a different set of custom fields appear on the **Details** page, such as vendor type and payment method.

### To set contact details

1. Search for a contact and open it.
2. In the left pane, click the **Details** page.

Contacts Details Page

3. View or edit the appropriate information.  
See [the Details page field descriptions](#) for more information.

Based on the selections that you made on the **Categories** page, additional blocks of text may appear on this page. Review this page and complete any new fields, especially required fields. You cannot save this record if any required fields are blank.

4. [Select a save option.](#)

### Details Page Field Descriptions

The tables below contain the default field information for person and company details. The fields that you see may vary based on your system settings, assigned rights, and the options you selected on the **Categories** page.

Field descriptions are listed in alphabetical order and may display in a different order in the application.

#### Details Page Field Descriptions

Field	Description
<b>Birthday</b>	The contacts birthdate. You can enter this date manually, or click the <b>Calendar</b> icon next to the field to open the calendar and select the desired date.
<b>Company ID</b>	The numbering system that your organization uses to identify a company. This may be pre-defined or you can enter it manually.
<b>Drivers License</b>	The contact's driver's license number, including the state and the number, for example CA A5306279.
<b>Employee ID</b>	The numbering system that your organization uses to identify an employee.
<b>Nickname/ Alias</b>	The name that the contact prefers other than their legal first name.
<b>Social Security</b>	The social security number for this person.  <b>Important:</b> Use the <i>nnn-nn-nnnn</i> format for the social security number.
<b>Tax ID</b>	The tax identification number that applies to this company.

### 1.3.1.6 Adding and Removing Contacts Categories

Use the Contacts **Categories** page to assign a contact to a group that may describe the type of contact or its function. Categories are usually pre-defined by your system administrator.

Categories are displayed in a hierarchical view with the main categories at the top and related subcategories listed below each main category. For example, a main category named Employees may have subcategories under it named *Accountants*, *Paralegals*, and *In-house attorneys*. A main category named Non-Employees may have subcategories under it named Outside Counsel, Vendors, or Clients.

The categories that you see apply to the specific TeamConnect area that you are viewing and your specific rights.

### How Categories Affect the Details Page

In addition to categorizing contacts, the categories that you select may cause additional information fields to appear on the **Details** page. These new fields may be required and if left blank, you are not able to save the record.

For example, if you select the *Internal Employees* category, and then select the *Attorney* category, when you return to the **Details** page, you may see custom fields that allow you to enter information about the attorney's specialty, home office, and current projects.

When you select a category, it is recommended that you return to the **Details** page to check for required fields.

See [Setting Category Details](#) for more information.

### 1.3.1.7 Viewing the Address Books for a Specific User

Use the **Address Books** page to view the address books that the contact is part of.

*Tip: In previous versions of TeamConnect, Address Books were referred to as Contact Groups.*

#### To view the address books for a specific user

1. Search for a contact and open it.
2. In the left pane, click the **Address Books** link.

The Address Books page opens and displays the names of the address books for which the selected user is a member.

### 1.3.1.8 Setting Contact Rates

Use the Contacts **Rates** page to view or edit the standard rates charged by a contact, the rate that is charged for non-standard tasks, the rate for an invoice task, and the preferred type of currency. The options that you see on this page depend on your access rights.

There are three types of rates:

- **Default**—The standard rate and currency that is used for a vendor for most of the services rendered to your company. This rate applies unless a specific task-based or invoice task rate is defined.
- **Task-based**—The rate that appears when you create a Task or add an entry to the Task Entry Tool. If defined for a contact, this rate overrides the default rate during the specified time period. See [Assigning or Reassigning Tasks to Users](#) for more information about tasks.
- **Invoice task rates**—The rate that applies to a specific task on an Invoice when the associated contact is selected as a timekeeper. If defined, this rate overrides the default rate during the specified period of time.

Each type of rate can have multiple entries with specific start and end dates for each entry.

**Important:** Information that is defined in each section is completely separate from other sections on the Rates page. For example, rates and dates defined in the Task section do not affect or apply to rates and dates defined in the Invoice Task Rates section.

#### Notes for Rates

- If you define a rate with no start or end date, you cannot add any other rates in that specific rate area. For example, if you define a task rate with no start or end date, you cannot define other task rates.

- If you enter a start date and the end date is blank, the rate starts on the date specified and the end date is indefinite.
- If the start date is blank, and you enter an end date, the rate applies to any date that is less than, or equal to the end date.
- If a contact has multiple rates in a specific area, such as invoice rates, overlapping date ranges are not allowed. For example, for the same contact, if one invoice rate has a start date of 3/16/09 and an end date of 4/16/09, and a different rate has a start date of 3/17/09 and an end date of 4/17/09, an error message appears and you must correct the overlapping dates.

## How Rates Affect Invoice and Task Records

The rates that you enter on this page also apply to other areas in TeamConnect:

- **Invoices—**
  - If an invoice is created for this contact, the type of currency that you enter on the contact Rates page is automatically used as the currency on the associated invoice. See [Viewing Invoices](#) for more information.
  - If a contact typically bills your company for services, the Invoice Task Rate that you enter on the contacts Rates page is automatically added to an invoice line item when the associated contact is selected as a timekeeper and the item type is Fee. See [Timekeeper Rates](#) for more information.
- **Tasks—**Tasks can be billable time items that are tracked by their hourly rate. The rate information on the Rates page will auto-populate the Rate field for tasks assigned to this contact. The Default Rate is used to auto-populate Rate fields if a task does not have a current rate assigned to it.

### To set a default rate

1. Search for a contact and open it.
2. In the left pane, click the **Rates** link.  
The **Rates** page opens.
3. Click **Edit**. You can do this by clicking the **Edit** button at the top of the screen, or by clicking the **pencil** icon next to the **Currency** tab.

Line	Rate	Start Date	End Date	Action
1	150.00	9/15/2016	10/14/2016	

Contacts Rates Page - Default Rate Section

4. If necessary, in the **Currency** drop-down list, select or change the type of currency that applies to this contact, for example, **United States, Dollars**. See [the Rates page field descriptions](#) for more information.



5. In the **Default Rates** section, in the **Rate** field, enter or edit the standard rate charged by this contact for most services.
6. If necessary, in the **Start Date** and **End Date** fields, enter the period during which this rate applies.  
  
If you do not enter a start and end date, the default rate is applied indefinitely to related invoices and tasks.
7. If you want to add additional task-based rates, click the **Add New Item** button and repeat steps 5 and 6.
8. [Select a save option.](#)

When you are working with billable tasks and invoices, this rate is used as the default rate unless a specific task-based or invoice based rate is defined. See [Tasks](#) for more information.

### To set a task-based rate

1. Search for a contact and open it.
2. In the left pane, click the **Rates** link.
3. Click **Edit**.

Line	Task	Rate	Start Date	End Date	Action
1	(Select)	0.00			

[Add New Item](#)

#### Contacts Rates Page - Task-based Rates Section

4. In the **Task Rates** section, on the **Task** drop-down list, select the task to which the selected rate applies.
5. In the **Rate** field, enter the rate charged by the contact for this specific task.
6. If necessary, in the **Start Date** and **End Date** fields, enter the period during which this rate applies.
7. If you want to add additional task-based rates, click the **Add New Item** button and repeat steps 4 through 6.  
  
See [the Rates page field descriptions](#) for more information.
8. [Select a save option.](#)

When you are working with a billable task, this rate overrides the **Default Rate** only during the specified time period.

### To set an invoice task rate

1. Search for a contact and open it.
2. In the left pane, click the **Rates** link.

- Click **Edit**. You can do this by clicking the **Edit** button at the top of the screen, or by clicking the **pencil** icon next to the **Task Rates** tab.

The screenshot displays the 'Invoice Task Rates' section. At the top, it says 'Invoice Task Rates 1 - 1 of 1'. Below this is a table with the following columns: Line, Task, Rate, Start Date, End Date, and Action. The first row contains: Line '1', a dropdown menu for 'Task' with '(Select)' as the current value, a 'Rate' field containing '0.00', and 'Start Date' and 'End Date' fields, each with a calendar icon. An 'Add New Item' button is located at the bottom left of the table area.

#### Contacts Rates Page - Invoice Task Rates Section

- In the **Invoice Task Rates** section, on the **Task** drop-down list, select an option.
- In the **Rate** field, enter the rate charged by the contact for this specific task.
- If necessary, in the **Start Date** and **End Date** fields, enter the period during which this rate applies.
- If you want to add additional task-based rates, click the **Add New Item** button and repeat steps 4 through 6 as necessary.

See [the Rates page field descriptions](#) for more information.

- [Select a save option.](#)

When you are working with invoices, this rate overrides the **Default Rate** during the specified time period.

#### To remove a rate

To remove a rate, see [Deleting an Entry](#).

## Rates Page Field Descriptions

The tables below contain the default fields for the **Rates** page. The fields that you see may vary based on your system settings and assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

#### Rates Page Field Descriptions

Field	Description
<b>Add New Item</b>	Adds a new, blank rate row.
<b>Currency</b>	<p>The currency used for payments to this contact.</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>The currency that you select is displayed in the <b>Currency</b> field of the Invoice screen when this contact is selected as a vendor.</li> <li>The <b>Billing Information</b> section in task and expense records cannot use multiple currencies, and contacts who are also users may be responsible for the costs associated with these records.</li> </ul>

	<ul style="list-style-type: none"> <li>The selected currency is reflected in the contact's rates and in any custom fields for financial amounts in the contact record.</li> <li>Changing the currency selected in this field for an existing contact does not change the rate amounts for the contact. When changing the currency of the contact, you must also adjust the rate amounts.</li> <li>Changing the currency of the contact does not affect any existing invoices where that contact is selected as the vendor. The currency and amounts in those invoices stays the same.</li> </ul>
<b>Rate</b>	<p>The rate that applies to the selected task or invoice.</p> <p>For Tasks and Invoice line items, this value is automatically displayed in the <b>Rate</b> field when you select a category.</p>
<b>Remove</b>	<p>Removes the selected line item from the <b>Rates</b> page.</p> <p>Click the appropriate check-box, and then click <b>Remove</b>.</p>
<b>Start Date/ End Date</b>	<p><b>Start Date</b>—The earliest date that this rate is applicable.</p> <p><b>End Date</b>—The latest date that this rate is applicable.</p> <p>Click the <b>Calendar</b> icon to select a date.</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>If you define a rate with no start or end date, you cannot add any other rates in that specific rate area. For example, if you define a task rate with no start or end date, you cannot define other task rates.</li> <li>If you enter a start date and the end date is blank, the rate starts on the date specified and the end date is indefinite.</li> <li>If the start date is blank, and you enter an end date, the rate applies to any date that is less than, or equal to the end date.</li> <li>If a contact has multiple rates in a specific area, such as invoice rates, overlapping date ranges are not allowed. For example, for the same contact, if one invoice rate has a start date of 3/16/09 and an end date of 4/16/09, and a different rate has a start date of 3/17/09 and an end date of 4/17, an error message appears and you must correct the overlapping dates.</li> </ul>

### 1.3.1.9 Relating Contacts

Use the Contacts **Relations** page to view or establish the relationships between a contact and other people or companies, for example, plaintiff-to-defendant, spouse-to-spouse, or employee-to-company. A contact may have several different relationships with other contacts and more than one relationship with the same contact.

After you create a relationship between contacts in one record, the related contact's **Relations** page is automatically updated with the same information.

With the appropriate rights, you can view, edit, and delete relations for a contact.

See [Working with Related Records](#) for more information.

### 1.3.1.10 Setting Contact Skills

Use the Contacts **Skills** page to define a contact's skill set, and indicate the level of expertise for each skill. Skills associated with contacts can help you identify the best person or company for a specific task.

#### To set contact skills

1. Search for a contact and open it.
2. In the left pane, click the **Skills** link.

Line	Type	Level of Expertise	Action
1	Skilled	99	

Remove Add New Item

#### Contacts Skills Page

3. Select one of the following options:
  - To add a new skill, in the **Type** drop-down, select the desired **Type** of skill, and then type a number to indicate the corresponding **Level of Expertise**. The level of expertise field maxes out at '99'.
  - To edit a skill, in each data entry row, change the desired **Type** of skill, and/or the number that indicates the corresponding **Level of Expertise**.
4. To add an additional skill, click **Add New Item** and repeat step 3 until you've added all of the appropriate skills for this contact.
5. [Select a save option.](#)

#### To delete contact skills

To delete skills for a contact, see [Deleting an Entry](#).

### 1.3.1.11 Selecting Contact Territories

Use the Contacts **Territories** page to select the geographic areas where the selected contact conducts business. A contact may be associated with several territories.

Based on your assigned rights, you can view, edit, and remove territories.

### To select contact territories

1. Search for a contact and open it.
2. In the left pane, click the **Territories** link.

The **Territories** page opens.

Contact - Holden, Cody

Save Cancel & View Create a Copy More Actions Share Printable View Help

Add Bookmark

**Territories**

Territories 1 - 2 of 2

Line	Territory	Action
1	Northwest Territory	
2	(Select)	

Remove Add New Item

**Contacts Territories Page**

3. In the **Territory** column, use the drop-down list to select the contact's territory.
4. To add an additional territory, click + in the **Action** column or click **Add New Item**, and then select the necessary information.
5. [Select a save option.](#)

### To delete contact territories

To delete territories for a contact, see [Deleting an Entry](#).

#### 1.3.1.12 Viewing Company Employees

Use the Contacts **Employees** page to view a list of employees for a selected company. This list is automatically generated from contact records when you select a company in the Company field on the General page for a person.

**Note:** This page is only visible for company contacts.

### To view company employees

1. Search for a company contact and open it.
2. In the left pane, click the **Employees** link.

The employees that are associated with this company are listed.

### 1.3.1.13 Viewing Contact Involvement

Use the Contact **Involvement** page to view information about people or companies who are involved in a project. These non-TeamConnect users are referred to as Involved Parties and their roles may vary, such as opposing party, outside counsel, claimant, witness, law firm, or attorney.

For example, Beth files a lawsuit against Jerry. An internal lawyer in your organization is assigned to the lawsuit and is responsible for it. Both Beth and Jerry are involved parties with roles - Beth's role is plaintiff and Jerry's role is defendant. Neither Beth nor Jerry is assigned to the project, but it is important that the details of their involvement is available to those assigned to the project.

If a contact is involved in several roles in a project, only one project entry displays on the **Involvement** page with the default role for the contact.

Note: You cannot modify involved parties on this page; you must open the related project to make any changes. See [Managing Involved Parties for Projects](#) for more information.

#### To view contact involvement

1. Search for a contact record and open it.
2. In the left pane, click the **Involvement** link.

The **Involvement** page opens.

Involvement		
Display:	All	
Involved Parties 1 - 6 of 6		
Type	Project Name	Role
Transaction	TRAN-000004-afia transaction record	Outside Counsel Firm
Dispute	DISP-000039-Matter 6235 1474386419925	Outside Counsel Firm
Dispute	DISP-000041-Matter 7043 1474388295541	Outside Counsel Firm

Contacts Involvement Page

3. If necessary, use the **Display** drop-down list to view a list of specific contact records.  
You can view the project type, project name, and the role that a contact has in a project.  
Contacts may have more than one role in the same project, or may be involved in many different projects, so you may see more than one listing on this page.

### Involvement Page Field Descriptions

The tables below contain the default fields. The fields that you see may vary based on your system settings and assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

#### Involvement Page Field Descriptions

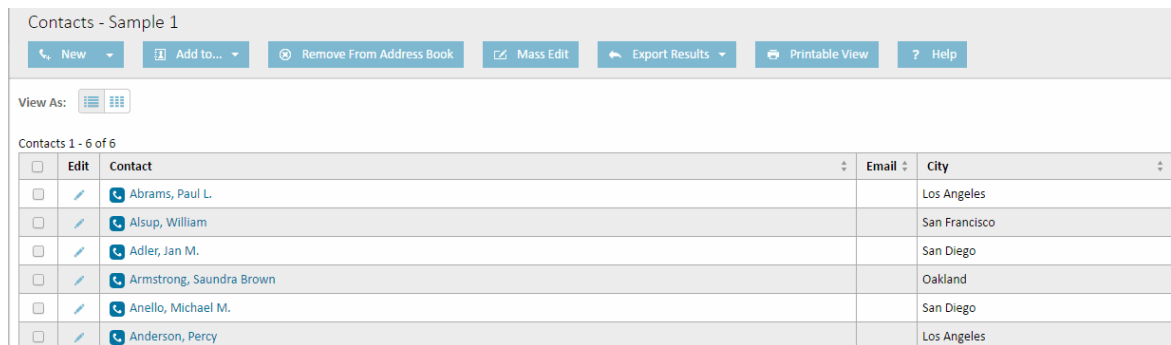
Field	Description
-------	-------------

<b>Application</b>	The applications for each project in which the selected contact is involved.
<b>Project Name</b>	All projects for the selected application.  Click the individual project name link to view the corresponding detailed information.
<b>Role</b>	The contact's roles in each project.
<b>Show Involved Projects For</b>	The application for which you want to view the contact's involvement.  <b>Note:</b> <i>If you want to see all projects the contact is involved in, select the default option <b>All</b>.</i>

### 1.3.2 Address Books

If you have a large number of contact records in your system, you may want to organize them into address books so that you can quickly communicate with everyone in the group at one time. For example, you can create a group of your colleagues and add the group as attendees to an appointment, or send a document to everyone in the address book.

You can find all address books that you created and subscribe to in the left pane under the **Contacts** tab. The address book icon indicates the address book icon.



Address Books in the Contacts Left Pane

The following table shows how you can view, create, and use address books.

...Address Books	How...
Viewing	<ul style="list-style-type: none"> <li><a href="#">Viewing Contacts in an Address Book</a></li> <li><a href="#">Printing Records</a></li> </ul>
Creating	<ul style="list-style-type: none"> <li><a href="#">Creating an Address Book</a></li> </ul>

	<ul style="list-style-type: none"> <li>• <a href="#">Making an Address Book Public or Private</a></li> </ul>
Updating	<ul style="list-style-type: none"> <li>• <a href="#">Adding Contacts to an Address Book</a></li> <li>• <a href="#">Removing Contacts from an Address Book</a></li> <li>• <a href="#">Deleting Records</a></li> </ul>
Utilizing	<ul style="list-style-type: none"> <li>• <a href="#">Subscribing to Address Books</a></li> </ul>

## Best Practices for Address Books

When you create and save an address book, make sure to specify the correct **Public** or **Private** designation. See [Making an Address Book Public or Private](#) for more information.

### 1.3.2.1 Subscribing to Address Books

Use the **Manage List** page to subscribe to the public address books.

Public address books are available for all users to use and view. Public address books may contain contacts with whom you have regular contact, such as all of the people who are involved in a project to which you are assigned. You access a public address book by subscribing to it. Once you subscribe, you can access the address book, but you cannot make changes to it unless you have been granted rights to do so by the person who created it. See [Working with Record Security](#) for more information.

The Address Books to which you subscribe appear on the **Contacts** list page in the left pane and are indicated by the address book icon. You can change the order in which the contact collections and address books are listed in the left pane, and remove those that you no longer need. See [Managing Collections](#) for more information.

#### To subscribe to public address books

1. Click the **Contacts** tab.
2. Click the Gear icon to access the **Manage** page.
3. The Manage Lists screen will display with all available address books and search views for the Contact object.
4. Click **Subscribe to Address Books**.
5. Select the address books with the check-boxes.
6. Click **Subscribe**.

The selected address books appear on the **Contacts** list page in the collections pane.



### 1.3.2.2 Viewing Contacts in an Address Book

You can view the contacts in an address books that you created, address books to which you subscribe, and all public address books.

You can also see a list of address books to which a contact belongs. See [Viewing the Address Books for a Specific User](#) for more information.

#### To view contacts in an address book

1. Click the **Contacts** tab.
2. Select one of the following options:
  - If the address book whose contacts you want to view is visible in the left pane, click its link. The contacts are displayed in the list view.
  - If the address book is not visible in the left pane, click the **Manage** link. On the **Managed List** page, click the link of the appropriate address book.

For either option, the address book opens and you can view its contacts.

The screenshot shows the 'Contacts - Sample 1' interface. At the top, there is a navigation bar with tabs: Home, Legal, Financials, **Contacts**, Activities, Documents, Reports, Search, and Admin. Below the navigation bar, there is a left sidebar with a 'Contacts' section containing links for Custom Search, Recently Viewed, Sample 1, All Contacts, Recently Created, Law Firms, Legal Department Staff, Outside Counsel Attorneys, Search for Judge, LH Employees, and LH Stewards. The main content area displays a list of contacts for 'Sample 1'. The list has a header 'Contacts 1 - 6 of 6' and a table with columns for 'Edit' and 'Contact'. The table contains six rows of contact information:

<input type="checkbox"/>	Edit	Contact	
<input type="checkbox"/>		Abrams, Paul L.	
<input type="checkbox"/>		Alsup, William	
<input type="checkbox"/>		Adler, Jan M.	
<input type="checkbox"/>		Armstrong, Sandra Brown	
<input type="checkbox"/>		Anello, Michael M.	
<input type="checkbox"/>		Anderson, Percy	

Contents of an Address Book

### 1.3.2.3 Creating an Address Book

Address books let you organize person and company contacts into groups that you use frequently. You can use an address book to send messages to a group of people, or to send a meeting invitation to a group of users.

You can create address books in two ways:

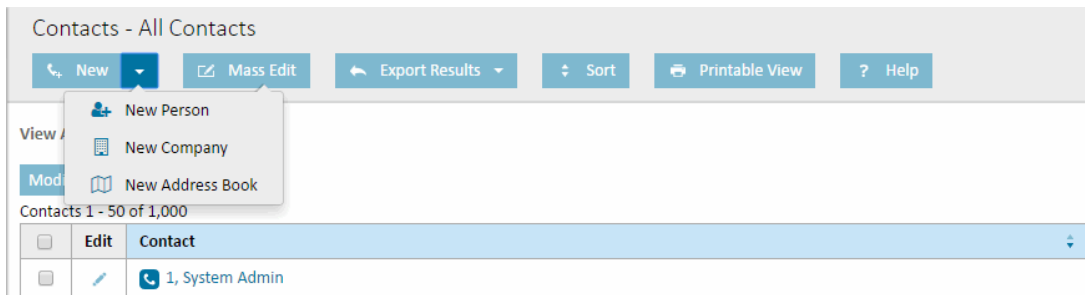
- The **New** button on the Contacts list view
- On the **Address Books Manage List** page

By default, address books that you create are private.

#### To create an address book on the contacts list view

1. Click the **Contacts** tab.

- Click **New**, and then select **New Address Book**.



**New Address Book**

The **Address Book General** page opens.

 A screenshot of the 'Address Book - New' form. The top bar contains 'Save', 'Cancel', and 'Help' buttons. The main section is titled 'Address Book Information' and contains a text input field for '\*Name:' with the value 'Sample 1'. Below the name field is a checkbox labeled 'Public'. At the bottom of the form, there is a note: '\*Required fields are noted by an asterisk'.

**Address Book General Page**

- In the **Name** field, enter the name of this address book.
- Select one of the following options:
  - Select the **Public** box if you want other users to have access to this address book.
  - Leave the **Public** check-box unchecked if you do not want others to have access to this address book.

Once you mark an address book **Public** or **Private**, you cannot change this setting. If necessary, you can delete the address book, and add it again with the appropriate designation.

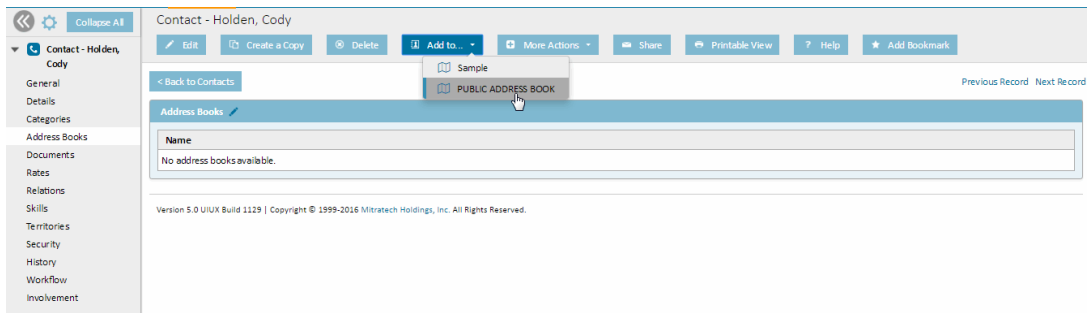
- Click **Save**.

You return to the **Manage Address Books** page and your address book appears in the list.

### To create an address book on the manage list page

- Click the **Contacts** tab.
- In the left pane, click the **Manage** link.

3. On the **Manage List** page, click **New Address Book**.



**Manage List Page - Add New Address Book**

4. In the **Name** field, enter the name of this address book.
5. Select one of the following options:
  - Select the **Public** box if you want other users to have access to this address book.
  - Leave the **Public** check-box unchecked if you do not want others to have access to this address book.
6. Click **Save**.

You are returned to the **Contacts** list view and your address book appears in the left pane.

To add users to address books, see [Adding Contacts to an Address Book](#).

#### 1.3.2.4 Adding Contacts to an Address Book

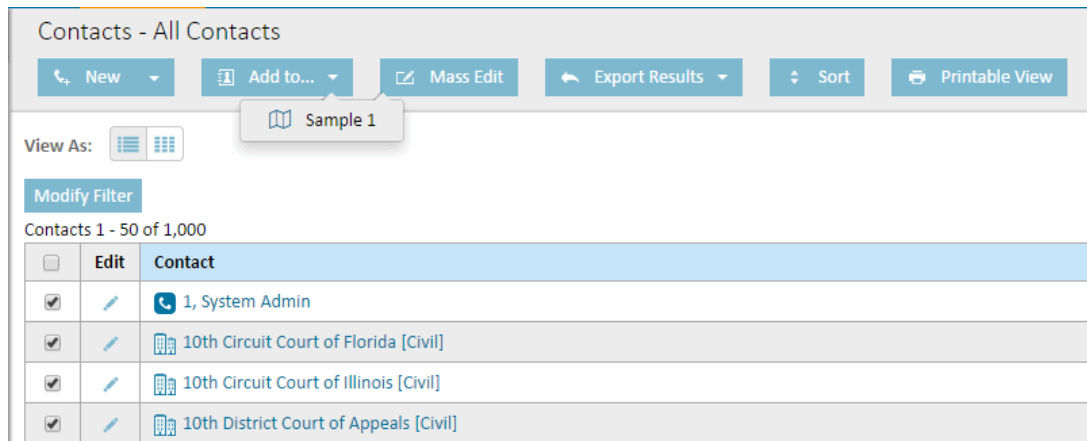
Use the Contacts list view to add individual contacts or company contacts to the address books that you created or address books to which you have the appropriate rights. You can add a single contact or multiple contacts at the same time.

##### To add contacts to an address book

1. Click the **Contacts** tab.
2. In the left pane, select the collection or address book that contains the contacts that you want to add to the selected address book.

**Tip:** Click the **All Contacts** link to view all of the contacts that you have rights to view and all of the contacts that you created.

3. Select the check-boxes of the contacts that you want to add to the address book.



#### Add Users to an Address Book

4. Click **Add to**, and then select an address book from the drop-down list.

The selected contacts are added to the selected address book.

### 1.3.2.5 Removing Contacts from an Address Book

You can remove contacts from address books that you created or address books to which you have appropriate rights. You can remove a single contact or multiple contacts at the same time.

#### To remove contacts from an address book

1. Click the **Contacts** tab.
2. In the left pane, click the appropriate address book.
3. Select the contacts that you want to remove, and then click **Remove from Address Book**.

The selected contacts are removed from the address book, but not from TeamConnect.

### 1.3.2.6 Making an Address Book Public or Private

There are two types of address books:

- **Private**—By default, when you create an address book, it is private. A private contact group is only accessible to the user who created it and to TeamConnect administrators. Private address books are useful for organizing contacts for your personal use and reference.
- **Public**—Public address books may be viewed, and subscribed to, by other users in addition to the user who created the book. When you subscribe to an address book, its link appears in the left pane along with the collections links.

You cannot change the designation of an address book once you set it. If necessary, you can delete the address book, create it again and give it the appropriate **Public** or **Private** designation. See [Creating an Address Book](#) for more information.

## 1.4 Projects

Projects, also referred to as matters, let you organize, compile, and manage the details of an incident or event from the beginning to its conclusion. Details may include information about the people involved, all relevant dates, and tasks that must be completed, users and specialists assigned to the project, and all of the related documents.

### Use Case for Projects

For example, if you want to manage an accident claim, you can perform some or all of the following actions:

1. Create a new project to capture relevant facts, such as the date and location of the accident, vehicle information, witnesses, and the law enforcement officers who were at the scene.
2. To move the project forward, you may want to assign internal staff members to perform specific tasks, such as visiting the claim site or interviewing claimants.
3. You can schedule regular status meetings using the appointment calendar to determine if the project is on schedule or if the completion date needs to be extended.
4. You can create invoice and expense records from the project to track the costs related to the project.
5. If litigation is involved, you can add all of the relevant court documents to the project.

When all of the tasks are done, the reports are filed, and the claim or court case is resolved, you can close the project so that the assigned resources can move on to other projects.

### Using Projects

The following table shows how you can view, create, and edit projects.

...Projects		How...
Creating		<ul style="list-style-type: none"> <li>• <a href="#">Creating a New Project</a></li> <li>• Creating <a href="#">Child Projects</a></li> <li>• Creating <a href="#">Embedded Projects</a></li> <li>• <a href="#">Adding or Removing Project Categories</a></li> </ul>
Updating	Phases	<ul style="list-style-type: none"> <li>• <a href="#">Changing Project Phases</a></li> </ul>
	Assignees	<ul style="list-style-type: none"> <li>• <a href="#">Adding Assignees to Projects</a></li> <li>• <a href="#">Changing the Primary Assignee for a Project</a></li> </ul>
	Relations	<ul style="list-style-type: none"> <li>• <a href="#">Creating Relationships between Projects</a></li> </ul>
	Involved Parties	<ul style="list-style-type: none"> <li>• <a href="#">Adding or Editing Involved Parties</a></li> </ul>

		<ul style="list-style-type: none"> <li>• <a href="#">Changing an Involved Party Role</a></li> </ul>
	Tasks	<ul style="list-style-type: none"> <li>• <a href="#">Adding Tasks to Projects</a></li> </ul>
	Appointments	<ul style="list-style-type: none"> <li>• <a href="#">Adding Appointments to Projects</a></li> </ul>
	Expenses	<ul style="list-style-type: none"> <li>• <a href="#">Adding Expenses to Projects</a></li> </ul>
	Accounts	<ul style="list-style-type: none"> <li>• <a href="#">Adding Accounts to Projects</a></li> </ul>

## Project-specific Pages

Most projects have the following project-specific pages where you can view and edit information:

- **General**—View, enter and edit general information about the project, such as its name, number, when it was opened, and its current phase.
- **Phases**—View and change the phase of the project to show the progress from the beginning to the end of the project.
- **Assignees**—View or edit the users who are assigned to the project and their roles in the project.
- **Involved**—View or edit the people or companies who are involved in the project with roles such as witness, plaintiff, or outside counsel.

Each project record also has several other pages where you can view, enter, and edit additional information that applies to a project, such as appointments, expenses, and categories for a project.

## Opening a Project Record

To open a project, click its link. The project opens to its **General** page. Once you open a project, with the appropriate rights, you can use the links in the left pane to manage project information.

### 1.4.1 Creating a New Project

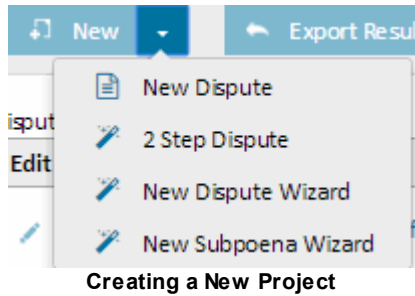
You can create a new project by entering data into a blank project record or using a wizard.


#### To create a project

1. Click the **All** tab, and then select the type of project that you want to create, such as **Disputes**.

You can access projects on the **All** tab, but your system administrator may have also provided a customized way for you to access projects, such as a portal pane on your home page.

2. On the project list page, click the **New** drop-down and select an option.



3. Complete the fields, depending on the option you selected:
  - If you select a wizard, denoted by the  icon, the wizard opens. When the wizard opens, complete the required fields on each page, and then click **Next** to progress to the end of the wizard. When you click **Finish** at the end of the wizard, the record is available in the appropriate list view and collection.
  - If you do not select a wizard, the project record opens to its **General** page. Enter the appropriate data in the available sections and select a save option. See [General page field descriptions](#) for more information.

## Project General Page Field Descriptions

The tables below contain the default fields for projects. The fields that you see may vary based on your system settings and assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

### General Information Field Descriptions

Field (* = required)	Description
<b>Closed On</b>	Date and time when the record was closed. This information only displays when the phase is changed to Close.  See <a href="#">Changing Project Phases</a> .
<b>Current Phase</b>	The current phase of the project.
<b>*Name</b>	The name of this project.  If you see <b>(Auto)</b> displayed in the <b>Name</b> field, projects of this type are named automatically.

	<p><b>Note:</b> If one of the fields used in the automatic naming sequence is changed, the name of the project may change. For example, if the <b>Contact</b> field is used for naming projects and you select a different contact, the project name will change when you save the record.</p> <p>If you do NOT see <b>(Auto)</b> displayed in the <b>Name</b> field, type a descriptive name according to your organization's standards.</p>
<b>Project ID</b>	<p>The number assigned to this project.</p> <p>If you see <b>(Auto)</b> displayed in the <b>Number</b> field, projects of this type are numbered automatically.</p> <p>If you do NOT see <b>(Auto)</b> displayed in the <b>Number</b> field, type a number according to your organization's standards.</p> <p><b>Note:</b> After you save the record, you may not be able to change this number.</p>
<b>Opened On</b>	<p>Date and time that the record is opened. These values are set automatically when the project record is created.</p>
<b>Parent Project</b>	<p>If applicable, this is the project that you want to set as a parent for this record.</p> <p>Click the search icon to locate the appropriate project.</p> <p><b>Note:</b> This field is available only for those projects that can have other projects as parents.</p> <p>In Read-only mode, this field is displayed as a hyperlink to the selected project.</p>

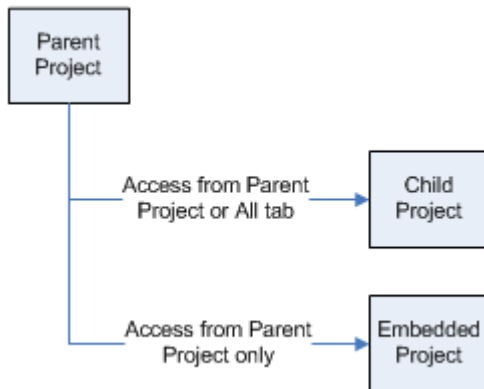
#### Access Information Field Descriptions

Field Name	Description
<b>Created By</b>	<p>The name of the user who created this project.</p> <p>To view the associated contact record, click the link in this field.</p>
<b>Created On</b>	<p>The date and time that the project was added to TeamConnect.</p>
<b>Modified By</b>	<p>The name of the user who last modified the project. To view the associated contact record, click the link in this field.</p>
<b>Modified On</b>	<p>The date on which the project was last modified.</p>



## 1.4.2 Creating Different Types of Projects

Some projects may not exist independently and always require a parent project. The following diagram illustrates the different types of projects.



**Child Projects**—You create child projects the same way you create parent projects. The only difference is that child projects require a parent project. See [Child Projects](#) for more information.

**Embedded Projects**—Embedded projects are child projects that you can only access from within their respective parent projects. In addition, you cannot search for embedded projects. See [Embedded Projects](#) for more information.

### 1.4.2.1 Child Projects

Some projects cannot exist independently, so they become child projects of parent projects. While a child project can only have one parent project, a child project can be a parent for other projects.

For example, if you file a claim in the insurance industry, your adjuster uses your original insurance policy to connect to the new claim. The policy is the parent record and the claim is the child record.

#### To create a child project

- From the **All** tab, find the child project under its parent project. Create a new project using [Creating a New Project](#).
- Open an existing project and select another existing project as a parent in the **Parent** field on the **General** page. The project in the **Parent** field automatically becomes the parent project.
- Add a new child project from an existing parent record. The original project automatically becomes the parent project.

### 1.4.2.2 Embedded Projects

Like involved records, embedded projects are child projects that you can only access or create from within their respective parent projects. Embedded projects never display in the **All** list and you cannot perform a search to locate them.

**Caution:** You cannot save embedded projects without saving their respective parent records.

Embedded projects have a number and a name to identify them and can be associated with a specific contact, such as a doctor in a medical report, or an arbitrator in an arbitration.

Unlike regular or child projects, embedded projects do not have the following:

- Involved parties
- Milestones
- Accounts
- Tasks
- Child projects
- Assignees
- Phases
- Wizards

### To create an embedded project

1. Open the parent record to which you want to add the new embedded project.
2. Open the section of the record that displays embedded projects of the appropriate type, such as **Agreements** in a Contract.
3. Click the New button to create an embedded project. See [Creating a New Project](#) for more information about creating a new project.
4. [Select a save option](#) to save the parent project.

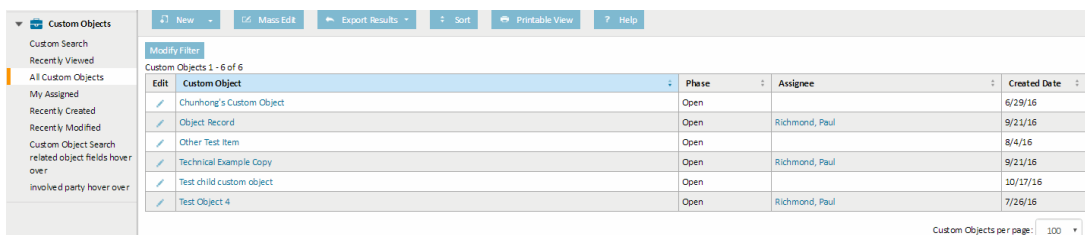
## 1.4.3 Editing a Project

With the appropriate rights, you can edit a project at any time.

### To edit a project

1. Click the **All** tab, and then click the project type that you want to edit.

The project list view opens.



Custom Object	Phase	Assignee	Created Date
Chunhong's Custom Object	Open		6/29/16
Object Record	Open	Richmond, Paul	9/21/16
Other Test Item	Open		8/4/16
Technical Example Copy	Open	Richmond, Paul	9/21/16
Test child custom object	Open		10/17/16
Test Object 4	Open	Richmond, Paul	7/26/16

Project List View

- Click the link for the project that you want to edit.

The project opens to its **General** page.

Custom Object - Technical Example Copy

Save Cancel & View Create a Copy Share Printable View Help Add Bookmark

**General Information**

Name: Technical Example Copy

\*Custom Object ID: 4

Opened On: 9/21/16 10:53 AM

Current Phase: Open

**Access Information**

Created By: Patil, Komal

Created On: 9/21/16 10:53 AM

Modified By: Holden, Cody

Modified On: 10/21/16 11:06 AM

\*Required fields are noted by an asterisk

### Projects General Page

- Click **Edit**.

The **General** page opens in edit mode.

- Make the appropriate changes to the **General** page. See the [General page field descriptions](#) for more information.
- Click a link in the left pane to make the appropriate changes to any additional project pages.
- [Select a save option](#).

## 1.4.4 Adding or Removing Project Categories

Use the **Categories** page to assign a project record to a grouping that may describe its type or purpose. Categories are usually pre-defined by your system administrator and each category may include one or more subcategories. Subcategories are listed under the main category in a drop-down list.

### How Categories Affect the General Page

Categories are also associated with groups of custom fields that appear or disappear based on the categories that you select. If defined, these custom fields appear on the **General** page, in the **Details** section.

For example, you may have a project that you assigned to a category named Medical Malpractice. Once you assign a project to this category, the **Details** section on the **General** page may display custom fields that allow you to enter details about the incident, such as the name of the doctor, the date the incident occurred, and the name and location of the hospital.

Based on your access rights, you can view, add, change, or delete categories.

See [Setting Category Details](#) for more information.

## 1.4.5 Changing Project Phases

Use the **Change Phase** button to change the phase of a project record.

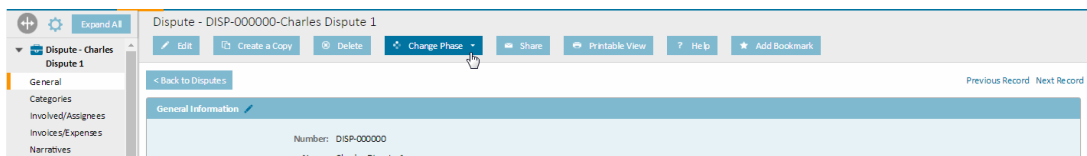
Each project goes through specific phases--from the creation of the project record to the end of the project. Together, these phases represent the project's life cycle. Each time you change the phase of a project, you move it along in the life cycle that is defined by your company policy.

For example, when you create a project, its initial phase may be Intake. As project tasks are completed, you select the phase that indicates that the current stage is completed and the project is moving to the next stage. As the project progresses, you continue to change the phase until the end of the project, which is usually the Closed phase.

You cannot skip a required phase or go back to a previous phase. For example, you cannot select **Re-open** after the **Open** phase is completed; the current phase must be **Closed** in order for you to select **Re-Open**.

### To change the phase of a project

1. Search for a project and open it.
2. Click **Change Phase**, and then select a phase from the drop-down list.



Project Change Phase Button

Use the **Phases** page of a project to view the date on which the phase was set, the name of the phase, who changed the phase, and the length of time the project was in a phase.

## 1.4.6 Assigning Contacts to Projects

**Assignees** are TeamConnect users who also have contact records. Use the **Assignees** page to select and manage the people who are working on a project. You can assign one or more users to a project and each assignee can have a different role in the project, such as attorney or paralegal.

Typically, one active assignee is set as the main assignee and is in charge of the overall progress of the project. The rest of the assigned users are responsible for tasks, milestones, or assignments that move the project forward to completion.

You can perform the following tasks to manage assignees for a project:

- [Viewing Users Assigned to a Project](#)
- [Adding Assignees to Projects](#)
- [Changing the Primary Assignee for a Project](#)
- [Unassigning or Reassigning Assignees](#)
- [Removing Assignees from a Project](#)

### 1.4.6.1 Viewing Users Assigned to a Project

Use the **Assignees** page of a project record to view a list of the users who are currently assigned to the project and the assignment history if there are any changes.

#### To view users assigned to a project

1. Click the **All** button.

You can access projects on the **All** tab, but your system administrator may have provided a customized way for you to access projects. Check with your system administrator for details.

2. Select a project.
3. On the **Project** list page, click the project whose assigned users you want to view.
4. In the left pane, click the **Involved/Assignees** link.

You can view the status, role, date of assignment, and the date on which the user was unassigned from the project, if it applies.

Status	Assignee	Role	Assigned On	Unassigned On
Active	Holden, Cody *	Attorney	10/21/16	

Projects View Assignees

### 1.4.6.2 Adding Assignees to Projects

Use the **Assignees** page to add assignees and their roles to a project. Roles are the responsibility that each assignee has for a project, such as attorney or accountant.

Projects may also have a project leader who is responsible for the progress of a project. Typically, the project leader is assigned the Primary designation.

#### To add a user to a project

1. Search for a project and open it.
2. In the left pane, click the **Involved/Assignees** link.
3. Click **Edit**.
4. Select one or both of the following options:
  - If there are no current assignees, select or search for the appropriate user name in the **Assignee** field.
  - To add an additional assignee, click **Add New Assignee**, and then select the appropriate user name in the **Assignee** field.
5. In the **Role** drop-down list, select the role that applies to the selected user.

The same user can be assigned to a project several times if the user has a different role for each assignment.

6. If there is more than one user assigned to this project, in the **Primary** column, click the radio button to select a main assignee.
7. [Select a save option.](#)

If email notification is enabled, each of the users assigned to the project will receive an automatically generated message to inform them of their assignment. If you assign yourself to a project, you will not receive an email notification. You only receive email notifications if you are assigned to a project by someone else. See [Setting Up Email Notifications](#).

## Assignees Page Field Descriptions

The table below contains the default fields for the Assignees page. The fields that you see may vary based on your system settings and assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

Assignees Page Field Descriptions

Field or Control	Description
<b>Action</b>	Add or remove an assignee for this project. <ul style="list-style-type: none"> <li>• Click + to add a new row</li> <li>• Click - to remove a user</li> </ul>
<b>Add New Assignee button</b>	Assign a user to this project.
<b>Assignee</b>	The name of the user that is assigned to this project.
<b>Primary</b>	The designated project leader.  <i>Note: Only assignees with the Active Status may be set as the main assignee.</i>
<b>Reassign button</b>	Set the selected users status to <b>Active</b> .
<b>Remove button</b>	Permanently delete a user from this project.
<b>Role</b>	The duty of a user during this project, for example, Attorney or Accountant.
<b>Status</b>	The current status of each assignee in the list.  Options are:

	<ul style="list-style-type: none"> <li>Active—Currently assigned to this project</li> <li>Inactive—Not currently assigned to the project</li> </ul>
<b>Unassign button</b>	Sets the selected users status to <b>Inactive</b> .

### 1.4.6.3 Changing the Primary Assignee for a Project

Use the **Assignee** page to assign or change the designated project leader. You can change the primary assignee at any time.

#### To change the primary assignee in a project

1. Search for a project and open it.
2. In the left pane, click the **Involved/Assignees** link.
3. In the **Primary** column, select the radio button for the user that you want to set as the main assignee.

Assignees						
Assignees 1 - 2 of 2						
<input type="checkbox"/>	Line	Primary	Status	Assignee	Role	Action
<input type="checkbox"/>	1	<input checked="" type="radio"/>	Active	Holden, Cody	Attorney	
<input type="checkbox"/>	2	<input type="radio"/>	Active	Johnson, Justin	Associate General Counsel	

Remove Add New Assignee Unassign Assign

Assignee Page

4. [Select a save option.](#)

On the **Involved/Assignees** list page for a project, the main assignee is indicated by an asterisk.

Status	Assignee	Role	Assigned On	Unassigned On
Active	Holden, Cody *	Attorney	10/21/16	
Active	Johnson, Justin	Associate General Counsel	10/21/16	

Projects Assignee Page

### 1.4.6.4 Unassigning or Reassigning Assignees

Use the **Involved/Assignees** page to temporarily remove a user from a project, or to reassign a user to a project. There are certain situations when you might need to end or temporarily end an assignee's participation in a project. For example, if the assignee's workload needs to be adjusted, you may give the assigned projects to a different user.

Unassigning assignees instead of deleting them is useful so that you have a record of the project assignment history.

#### To unassign or reassign users

1. Search for a project and open it.
2. In the left pane, click the **Involved/Assignees** link.
3. Select one of the following options:
  - To temporarily remove users from a project, select the appropriate check-boxes, and then click **Unassign**. The status changes to **Inactive**.
  - To reassign users, select the appropriate check-boxes, and then click **Reassign**. The status changes to **Active**.
4. [Select a save option](#).

#### 1.4.6.5 Removing Assignees from a Project

Use the **Involved/Assignees** page to completely remove a user from a project.

#### To remove a user from a project

1. Search for a project and open it.
2. In the left pane, click the **Involved/Assignees** link.
3. Select the check-box next to the name of the assignees that you want to remove from the project. You can select more than one check-box.
4. Click **Remove**.

The selected user is completely removed from the **Involved/Assignees** page, but not from TeamConnect.
5. [Select a save option](#).

#### 1.4.7 Creating Relationships between Projects

Use the **Relations** page to view or establish the relationships between projects, such as parent-to-child projects. A project may have several different relationships with other projects.

After you create a relationship between projects in one record, the related project's **Relations** page is automatically updated with the same information.

With the appropriate rights, you can view, edit, and delete relations for a project.

See [Working with Related Records](#).



## 1.4.8 Managing Involved Parties for Projects

Use the **Involved/Assignees** page to view, add, and edit the names of people or companies who are involved in a project. Involved parties are participants in a project, but are external to your organization and are not TeamConnect users. Involved parties may have roles such as witness, plaintiff, outside counsel, or opposing party.

Involved parties have a set of additional pages where you can add information. For example, for each involved party, you can set up an account, establish relationships between other involved parties, or manage documents that are related to an involved party. Even though these pages apply to an involved party, the procedures are the same as for any project in TeamConnect.

Involved party records can only be created and viewed from within their related projects.

You can perform the following tasks to manage involved parties for a project:

- [Adding or Editing Involved Parties](#)
- [Changing an Involved Party Role](#)
- [Adding or Editing Involved Party Accounts](#)

### 1.4.8.1 Adding or Editing Involved Parties

When you add an involved party to a record, a corresponding record is automatically created on the **Involvement** tab of the selected contact's record. See [Viewing Contact Involvement](#) for more information.

#### To add an involved party

1. Search for a project and open it.
2. In the left pane, click the **Involved/Assignees** link.
3. In the **Involved Parties** section, click **New**.

The involved party record opens to the **General** page.

4. In the **Contact** field, click the find icon  , and then enter the appropriate search criteria.
5. Click **Search**.

The **Contact Search** page opens with a list of contacts who match the search criteria that you selected.

***Tip:** If you cannot locate the appropriate contact, you can create it. Click **New**, select an option, and then follow the instructions to create a new contact. See [Creating a Contact](#) for more information. Once you save the new contact, you are returned to the project record and the new contact name appears in the **Contact** field.*

6. On the **Contact Search** page, click the name of the appropriate contact.

You are returned to the **General** page and the selected contact name appears in the **Contact** field.

Complete the additional fields. See [Involved General page field descriptions](#) for more information.

7. To activate this involved party, click the **Is Active** check-box.
8. [Select a save option.](#)

The new contact is added to the list of involved contacts for the selected project.

### To edit an involved party

1. Search for a project and open it.
2. In the left pane, click the **Involved** link.
3. Click **View Involved Party Record** next to the appropriate contact.  
If you click the involved party's name, the contact record opens.
4. Click **Edit**, and then make the appropriate changes. See [Involved General page field descriptions](#) for more information.
5. If necessary, click a link in the left pane to enter or edit other involved party information, such as historical details, or relationships between projects.
6. [Select a save option.](#)

## Involved General Page Field Descriptions

The table below contains the default fields for the involved **General** page. The fields that you see may vary based on your system settings and assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

Involved General Page Field Descriptions

Field or control	Description
<b>Contact</b>	The name of the contact that you want to add to this record. After the record is saved, the contact name is always displayed as a hyperlink to the contact's record.
<b>Default Role</b>	The role that a contact usually has in a project, for example, Attorney or General Counsel. The role that you select is automatically added to the list and set as the default on the <b>Roles</b> page when you save the record. You can add an involved party to the same project multiple times as long as each entry has a different role.
<b>Is Active</b>	The current status of the involved party. If checked, the involved party status is <b>Active</b> ; if unchecked, the status is <b>Inactive</b> .

<b>New</b>	If available, clicking this button opens a drop-down list that lists all active wizards for this related object. Clicking a wizard on the list opens it.
<b>Notes</b>	Notes that may apply to this involved party.
<b>Show Details For</b>	Select the role from the list, the details of which you want to be displayed in this screen.  This drop-down list is linked to the <b>Roles</b> tab and displays all roles added to the contact. If the <b>Default Role</b> has custom fields associated with it, they will be displayed below this drop-down list. If not, nothing will be displayed.

#### 1.4.8.2 Changing the Status of an Involved Party

Use the involved **General** page to change the status of an involved party.

##### To change the status of an involved party

1. Search for a project and open it.
2. In the left pane, click the **Involved/Assignees** link.
3. Click **View Involved Party Record** next to the appropriate contact.  
The involved party record opens to the **General** page.  
If you click the involved party's name, the contact record opens.
4. Click **Edit**.
5. In the **Is Active** field, clear the check-box if you want to inactivate an involved party; select the check-box if you want an involved party to become active.
6. [Select a save option.](#)

#### 1.4.8.3 Changing an Involved Party Role

Use the **Categories** page to change an involved party's role in a project.

Typically, companies organize their involved party contacts by categories or roles. Depending on the organization practices, these categories may include subcategories. Contacts may be involved in a project in a number of roles such as respondent, claimant, witness, or outside counsel. One of the roles is always set as the default role for the selected contact.

With the appropriate rights, you can add, view and delete involved party roles.

##### To change the role of an involved party

1. Search for a project and open it.
2. In the left pane, click the **Involved/Assignees** link.

The involved party list view opens.

3. If necessary, in the **View** drop-down list, select an option to view specific records.
4. In the Involved Party column, click **View Involved Party Record** next to the name of the user whose role you want to change.

The involved **General** page opens.

5. In the left pane, click the Roles link, and then click **Edit**.

The **Categories** page opens. This is the page that you use to change involved party roles.

6. On the **Categories** page, select the role that applies to this contact for this project.

The selected role is also added to the **Default Role** drop-down list on the Involved page.

7. Verify that the appropriate role is set as the default.
8. [Select a save option.](#)

For details on deleting involved roles, see [Deleting Records](#).

#### 1.4.8.4 Relating Involved Parties

Use the involved **Relations** page to define the relationships between contacts who are involved in the same project, such as plaintiff-to-defendant or attorney-to-client. An involved party may have several different relationships with other contacts and more than one relationship with the same contact.

After you create a relationship between contacts in one record, the related contact's **Relations** page is automatically updated with the same information. See [Working with Related Records](#) for more information.

#### 1.4.8.5 Adding or Editing Involved Party Accounts

Use the involved Party **Accounts** page if you need to create separate accounts to track specific transactions by contacts involved in the project.

You can do this in any of the following ways:

- From within the project to which the involved party is related
- From within the involved record

The procedure for adding an involved party account is the same as it is when you are adding or editing an account in the Finance area.

##### To add or edit an existing account to an involved contact

1. Search for a project and open it.
2. In the left pane, click the **Involved/Assignees** link.
3. Select the **View Involved Party Record** link next to the appropriate involved party.

If you click the involved party's name, the contact record opens.

4. In the left pane, click the **Accounts** link.
5. Select one of the following options:
  - To add a new account, click **New**.
  - To edit an existing account, click its link in the **Name** column, edit the appropriate fields, and then click the **Back to Involved Party** link to return to the involved party record.
6. To continue, see [Creating or Editing an Account](#).

The name of the account is automatically added to the list of the accounts in both the involved record and the selected project.

## 1.4.9 Managing Milestones

Use the **Milestone** page to view, create, and edit the steps or activities to complete when working on a project, such as claim investigation or a site survey. Each project may contain several milestones, and each milestone can have a number of tasks and appointments associated with it.

Milestones also have a set of additional pages where you can add additional information, such as a **Documents** page where you can add documents that relate to the milestone.

You can view, create, and edit Milestone records only after you open the associated project record.

**Note:** *You are only able to view and use milestones if you have upgraded from TeamConnect 2.x.*

You can perform the following tasks to manage milestones for a project:

- [Viewing Milestones](#)
- [Creating or Editing Milestones](#)

### 1.4.9.1 Viewing Milestones

Use the **Milestones** page to view the milestones created for a selected project. You can only view milestones after you open the associated project.

#### To view a milestone record in a project

1. Search for a project and open it.
2. In the left pane, click the **Milestones** link.

The Milestones list view opens.

You can also sort milestones columns in the list view. See [Sorting Information](#).

### 1.4.9.2 Creating or Editing Milestones

#### To create or edit a milestone

1. Search for a project and open it.
2. In the left pane, click the **Milestone** link.
3. Select one of the following options:
  - To create a new milestone, click **New Milestone**.
  - To edit a milestone, click the edit icon next to the milestone record that you want to edit.

The Milestone record opens to its **General** page.

4. On the **General** page, enter or edit the appropriate information. See [Milestone General page field descriptions](#) for field information.
5. Use the links in the left pane to add or edit information on additional milestone pages.
6. [Select a save option](#).

#### Milestone General Page Field Descriptions

Field or control	Description
<b>Subject</b>	The name of the milestone.
<b>Projected Date</b>	Click the <b>Calendar</b> icon next to the field to select the desired <b>Projected</b> or <b>Extension</b> dates.
<b>Extension</b>	<p><i>Note: If you do not click the <b>Dates Unknown</b> check-box, you must specify at least one date in each milestone record.</i></p> <p><i>Note: Once the project phase is reached, the <b>Completion Date</b> field is e</i></p>
<b>Completion Date</b>	
<b>Dates Unknown</b>	Check this box if you do not have any date information to enter in the <b>Projected Date</b> or <b>Extension Date</b> fields.
<b>Completed on phase change</b>	<p>Check this box if you want to automatically mark the milestone as complete when the project phase that is selected in the <b>Phase</b> drop-down box is reached. Once the project phase is reached, the <b>Completion Date</b> field is automatically is populated.</p> <p>Leave this check-box blank if you plan to set the milestone changes manually.</p>
<b>Transition</b>	If the <b>Completed on phase change</b> check-box is selected, and a <b>Phase</b> is selected, select whether the milestone should be

	automatically marked as complete when transitioning to the selected phase or transitioning from it.
<b>Phase</b> (drop-down list)	<p>If the <b>Completed on phase change</b> check-box is selected, select the appropriate project phase at which to automatically mark the milestone complete.</p> <p>This drop-down lists all phases defined for this project's milestones.</p> <p><i>Note: This field does not change the phase of a project record; you are defining a condition under which this milestone is automatically marked as completed.</i></p> <p>If you want to change the phase of a project, see <a href="#">Changing Project Phases</a>.</p>
<b>Notes</b>	Type any additional comments related to this milestone.

#### 1.4.9.3 Setting Milestone Categories

Use the Milestone **Category** page to assign a milestone to a group that may describe the type of milestone or its function. Categories are usually pre-defined according to your company's needs.

The categories that you see apply to the specific TeamConnect area that you are viewing and your assigned rights.

See [Setting Category Details](#) for more information.

#### 1.4.9.4 Managing Appointments for a Milestone

Use the Appointment page to schedule or edit appointments that are related to a milestone.

##### To add appointments to a milestone

1. Search for a project and open it.
2. In the left pane, click the **Milestone** link.  
The **Milestones** list view opens.
3. Open the appropriate milestone.
4. In the left pane, click the **Appointments** link.
5. To continue, go to step 4 in [Creating or Editing an Appointment Manually](#).

For more information about appointments, see [Appointments Calendar](#).

#### 1.4.9.5 Managing Milestone Tasks

Use the Milestone **Tasks** page to manage tasks that are related to milestones. Each milestone may include one or more tasks that must be completed before the milestone is closed. You can view, create and delete all tasks related to a specific milestone.

##### To manage milestone tasks

1. Search for a project and open it.
2. In the left pane, click the **Milestone** link.  
The **Milestones** list view opens.
3. In the left pane, click the **Tasks** link.
4. To continue, see step 3 in [Creating a Task Manually](#).

#### 1.4.10 Adding Tasks to Projects

Use the Projects **Tasks** page to designate the users and assignments that are needed to complete a project or milestone. You can change task assignments at any time and also assign tasks to yourself.

The procedure for creating and editing project tasks is the same as it is when you are creating or editing a task using the **Calendar** area.

##### To create a task for a project

1. Search for a project and open it.
2. In the left pane, click the **Activities** link.
3. Click **New**.  
The task record screen opens to the **General** page, and the project automatically displays in the **Project** field.
4. To continue, see step 4 in [Creating a Task Manually](#).

To edit a task from the **Activities** page, click the task and click **Edit**.

#### 1.4.11 Adding Appointments to Projects

Use the **Appointments** page to schedule appointments and meetings that are related to a specific project.

When you create an appointment, the project record name is automatically displayed in the **Project** field of the appointment record.

The procedure for adding an appointment is the same as it is when you are adding an appointment using the **Calendar** area.



### To create an appointment for a project

1. Search for a project and open it.
2. In the left pane, click the **Activities** link.
3. Click **New**.

The appointment record screen opens to the **General** page.

4. To continue, see [Creating or Editing Appointments](#).

To edit a task from the **Activities** page, click the appointment and click **Edit**.

## 1.4.12 Adding Expenses to Projects

Use the **Expenses** page for a project to manage the cost of merchandise or services that are necessary to conduct your business.

You may associate several expense items to a project. All expenses fall into one of the following categories:

- Project expenses, such as the cost of sending an investigator to a location out-of-town, or the cost of dinner for the project team when they work late.
- Global expenses, such as the cost of office supplies, or the cost of hiring contractors to paint the corporate offices.

Once you create expenses, you can post them against the appropriate accounts and track expenses by project and vendor. See [Expenses](#) for more information.

The procedure for creating expenses is the same as it is when you are working with expenses in the Finance area.

### To create an expense for a project

1. Search for a project and open it.
2. In the left pane, click the **Invoices/Expenses** link.
3. To continue, see [Creating or Editing an Expense](#).

## 1.4.13 Adding Accounts to Projects

Use the **Accounts** page to create or edit accounts for projects.

Money in an account is used for a specific purpose in your organization, such as a client escrow account or a petty cash account. Accounts can also have a parent-child relationship, such as a child account that is funded by the parent project account.

You can easily withdraw from and deposit funds to accounts, transfer funds between parent and child accounts, and post transactions to the appropriate accounts according to the specified posting criteria.

The procedure for creating accounts is the same as it is when you are working with accounts in the Finance area.

#### To create an account for a project

1. Search for a project and open it.
2. In the left pane, click the **Accounts** link.
3. To continue, see [Creating or Editing an Account](#).

## 1.5 Reports

TeamConnect provides a native analytical reporting service that is used to design, save, and execute graphical reports. This service allows you to view reports in a tabular format similar to a spreadsheet or in graphical representation formats, such as a pie chart, bar chart, or line chart.

Since these are dynamic reports and are under the control of the TeamConnect rights managements services, report administrators who have the necessary rights are able to modify the properties and appearance of these reports at any time to improve their usability.

The following table shows how you can find, create, and run reports.

...Reports	How...
Finding	<ul style="list-style-type: none"> <li>• <a href="#">Viewing Reports</a></li> <li>• Adding Reports to <a href="#">Dashboards and Portal Panes</a></li> <li>• <a href="#">Printing Records</a></li> <li>• <a href="#">Using Report Folders</a></li> </ul>
Building	<ul style="list-style-type: none"> <li>• Adding <a href="#">General Properties</a></li> <li>• Creating One or More <a href="#">Data Series</a></li> <li>• Specifying <a href="#">Chart</a> Characteristics</li> </ul>
Running	<ul style="list-style-type: none"> <li>• <a href="#">Running Reports</a></li> <li>• Using the <a href="#">Scheduler</a> to Schedule Run Time</li> <li>• <a href="#">Changing the Appearance of a Report</a></li> <li>• <a href="#">Running Drill-down Reports</a></li> </ul>

### Report-specific Pages

Each report has the following pages on which you can view and edit information:

- **General**—Contains the report name, type and description.

- **Data**—Lets you view or create data series to define the properties for a report.
- **Chart**—Lets you view or edit information about the properties of the chart, including the report type and details.
- **Security**—Lets you manage user rights to reports.
- **Schedule**—Lets you set up recurring, automatic runs of reports that are then emailed to a distribution list that you define.

### 1.5.1 Viewing Reports

You can view and manage reports using the **Reports** tab.

You can display a list of reports in the right pane as follows:

- By clicking a collection link in the left pane
- By clicking a report folder in the left pane

In the list view, you can sort reports by several default options.

Additionally, you can add graphic representations and tabular reports to a dashboard or home page. See [Dashboards and Portal Panes](#) for more information.

#### To view Reports

1. Click the **Reports** tab.  
The Reports main view opens to the **Recently Viewed** collection.
2. In the left pane, click the Collection or Folder that contains the report that you want to view. The appropriate reports appear in the right pane.  
If you do not see any reports listed in the right pane, click the **All Reports** collection link.

The Reports main view contains the following elements:

- **Collections**—Contains reports in the named groupings, such as **Recently Viewed** which contains the last 20 reports that you opened. See [Viewing Collections of Records](#) for more information.
- **Folders**—A view of the report folder hierarchy. See [Using Report Folders](#) for more information.
- **Chart icon**—Displays the type of chart type that is defined for this report. The chart type may be a graphic representation, such as a bar or line chart, or tabular which looks like a spreadsheet.
- **Report name**—The name of the report. You can click the report name to run a report. See [Running Reports](#) for more information.
- **Properties icon**—Opens the Properties page, which lets you view report information and do limited edits to report parameters.

From the **Properties** page, you may also click the **Edit** link to edit the report's properties. See [Building Reports](#) for more information.

You may click the **Delete** button to delete the report entirely.

If the report you want to delete is a "drill-down" report (is linked to other reports), a pop-up warning informs you of this condition and lists the names of the other reports. You may then choose whether to continue with the deletion. If you do continue, the drill-down report is deleted and the links between the drill-down report and the other reports are erased.

You may also run the report from the Properties page, by clicking the **Run Report** button in that view.

- **Edit properties icon**—Opens the report in edit mode. See [Building Reports](#) for more information.

Additionally, you can view when the report was last run, who last modified the report, and the date when it was modified.

Once you run a report, with the appropriate rights, you can make changes to its properties, change the way it appears, and then run it again. See [Running Reports](#), and [Changing the Appearance of a Report](#) for more information.

## Sorting Reports in the List View

By making a selection in the **Sort By** drop-down list, you can sort reports in the right pane in the following ways:

- **Recently Viewed**—Displays the reports in the **Recently Viewed** collection. This selection only appears if the **Recently Viewed** tab is clicked on the report list page.
- **Name A-Z**—View reports in ascending order.
- **Name Z-A**—View reports in descending order.
- **Last Run Date - Newest to Oldest**—View reports with the most recently run at the top of the list.
- **Last Run Date - Oldest to Newest**—View reports with the oldest date on top.

### 1.5.1.1 Dashboards and Portal Panes

The home page is the first page you see after you log in. Home pages contain specialized sections called portal panes. From portal panes, you can create and access various reports that are of importance to you. Dashboards provide a quick visual representation of the reports.

## Reports List Portal Pane

TeamConnect has a standard portal pane for listing reports that contains a drop-down list of report folders. Choose a folder name from that drop-down to see links to the reports that are contained in the folder. Click on a link to execute the report.

*Tip: You can set a portal pane to remember a folder name by clicking the Edit icon in the pane's title bar, entering a folder name, and then clicking the **Save** button. In the future, each time the portal pane is displayed, that folder is automatically selected.*

Based on your user group and the home pages defined by your organization, your home page may already contain dashboard portal panes that display specific reports.

For information about dashboards and portal panes, see [Home Pages and Portal Panes](#).

### 1.5.1.2 Using Report Folders

Use report folders to organize reports into useful groups. For example, you may want to group all financial reports into one folder, and all Dispute matter reports into another. When you select a folder in the left pane, its contents appear in the right pane. Folders can contain reports and sub-folders.

Folders may be public or private; a public folder can be changed by any user with the appropriate rights, and a private folder can only be changed by the user who created it.

You can create new folders and drag-and-drop reports from the right pane into folders in the left pane. You can also drag-and-drop a folder into a different folder so that it becomes a sub-folder.

Depending on your rights, you can delete any folder except the top-level Reports folder.

#### To create a new folder

1. Click the **Reports** tab.
2. Click the parent folder under which you want to create a new folder.
3. In the main view, click the **New Folder** button.

The folder's **General** page opens.

4. In the **Folder Name** field, enter a name.

The **Folder Path** name is determined by the option you selected in step 2. If you selected the **Reports** folder name, this field shows **Reports**; if you selected an existing folder, this field shows the same information as the folder you selected.

5. (optional) Enter a **Unique Key** that identifies this folder and a **Description** of the contents of this folder.
6. [Select a save option](#).

#### To move reports into folders

**Note:** You must have [appropriate rights to move reports](#) and you can only move reports to which you have rights.

1. Select the folder that contains the reports that you want to move.

The contents of the folder appear in the right pane.

2. In the right pane, click the appropriate report, and then drag-and-drop it into the appropriate folder in the left pane.

**Tip:** You can view all reports in the **All Reports** collection regardless of its current folder.

#### To move a folder into a different folder

In the left pane, click and drag a folder into a different folder.

The selected folder, its sub-folders, and all of its Reports are moved.

#### To delete a report folder

1. Open the properties of the report folder that you want to delete.
2. Click the **Delete** button.

The folder and all of its contents are deleted if you have the appropriate rights.

If the folder that you want to delete contains reports to which you do not have access, only the reports that you have rights to are deleted. The folder is not deleted. You cannot delete Reports that are locked for editing.

If the folder that you want to delete contains "drill-down" reports (reports that are linked to other reports), a warning pop-up informs you of this condition and lists the names of the other reports. You may then choose whether to continue with the deletion. If you do continue, the drill-down reports are deleted and the links between the drill-down reports and the other reports are erased.

#### To set report folder security

1. Follow the steps associated with security for any TeamConnect record, as described in [Editing Record Security](#).
2. The field **Folder Security** allows you to choose whether your security settings apply to this folder only, or to this folder and all subfolders (default value).

## 1.5.2 Running Reports

Running a report generates a report based on its defined data series and parameter information. Once you run the report, the data is used to create a graphic or tabular chart of this information. Each time that you change the appearance of a report, you can run it again to see the new results.

Before you can run a report, all of its properties and data series options must be defined by someone with appropriate rights. You can select the appearance of the report data at a later date if necessary. See [Building Reports](#) for more information.

**Note:** This section discusses how to request an immediate run of a report. It is also possible to schedule a report to run automatically at predetermined intervals. For details on how to schedule a report, see [Scheduler](#).

#### To run a report

1. If necessary, click the **Reports** tab.
2. Select one of the following options:
  - In the Reports list view, in the right pane, click the report that you want to run.

- Open the Collection or Folder that contains the appropriate report, and then click the report that you want to run.

If you have already run this report recently, you might see cached output instead of new output retrieved from a full report run. To learn more about cache, see [Viewing a Cached Report](#). See [Changing the Appearance of a Report](#) if you want to make cosmetic changes to a report. This will not change the appearance of the original report.

### To run a report with parameters

**Note:** *If there are no parameters associated with this report, it runs immediately and appears as a chart or tabular report based on its settings.*

If Parameters are defined (values that you enter before the report is run), the **Parameters** page opens. Review or change the parameter values, and then click **Run Report** to display the chart. See [Building Reports](#) for more information.

After a report with parameters runs, link **Clear Values** will erase any parameter values that had been entered previously. Link **Reset to Default** will erase any previous parameter values, then replace them with the default values that were defined when the report was built.

Neither of these two links will cause the report to run again. You must explicitly click the **Run Report** link to run the report again.

#### 1.5.2.1 Running Drill-down Reports

Some reports are designed to use drill-down reports. If the report that you are running supports drill-down reports, you can launch the drill-down report by clicking on an element of the graphical chart output, such as:

- Column
- Bar
- Pie Slice
- Line

When you click on one of these elements, the drill-down report opens, and its chart output replaces the chart output that you had been viewing. (To return to the chart you had been viewing, click the link **Back to Report Execution**.)

The output of the drill-down report is restricted to those records that match the element you clicked on. For example, if you had been viewing a Column chart in which each column represented an invoice type, and you clicked on the column labeled "Accrual", the drill-down report output that appears would contain only Accrual invoices.

**Note:** *Drill-down reports are not supported on the Apple iPad.*

### 1.5.2.2 Viewing a Cached Report

When you run a report, the output is automatically cached (stored in the computer's memory). If you navigate away from the report to do other work in TeamConnect, then return to the report, the output is retrieved from the cache without re-running the report.

**Note:** Reports are cached on your Home Page only.

TeamConnect continues to use the cached output unless it encounters one of the following conditions. If any of these conditions are true, the report output will be produced from a new run, not from cache:

- The Data Warehouse has been refreshed since the last time you ran the report.
- The report properties have been edited and saved (by you, or another user) since the last time you ran the report.
- The report uses the Tabular Only chart type. Output for this chart type is not cached. All other chart types are cached.
- The report has parameters, and you are executing the report from within the Parameters page. In this case, cache will not be used. (A report with parameters that is executed by simply clicking its name in the report list page will still use cache.)
- You run out of cache memory on your computer. If you cache a large number of reports and the cache uses all the memory reserved for it, the cache will remove the oldest cached report output in order to make room for new, incoming report output. The older reports will no longer be available in cache.
- The report is re-run by clicking on a report in the list view, or by clicking the **Run Report** button.

It is possible to change the appearance of report output using the chart bar, as described in [Changing the Appearance of a Report](#). But such changes do not affect the stored output in cache. Only changes to the report properties themselves would affect cache - property changes would force a new, full run of the report.

### 1.5.2.3 Changing the Appearance of a Report

You can edit the way your report output appears. The options that are available are based on the type of chart that you are viewing and your assigned rights.

**Note:** Changes in chart view are temporary and do not alter the original view of this chart. You can export changes, but to make permanent changes, you must have rights to edit chart properties.

**Note:** Most of the features discussed in this section are not supported on the Apple iPad, since Adobe Flash is not available on the iPad. The alternative chart output on the iPad may differ from Flash output in background color, bar colors, pie colors, column colors, area colors, line colors and y-axis value.

#### To change the appearance of a Report



1. View the chart that you want to change.
2. In the chart bar, make the appropriate changes. See [the Chart View field descriptions table](#) for specific chart bar option information.

The changes are immediately reflected in the chart view.

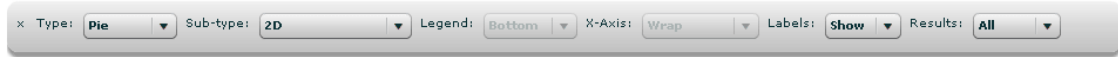


Chart Bar

The table below contains the available options for the different types of reports.

Chart View Field Descriptions

Field	Options
<b>Type</b>	<p>The type of chart in which you want to view the report data.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• <b>Bar</b></li> <li>• <b>Column</b></li> <li>• <b>Pie</b></li> <li>• <b>Line</b></li> <li>• <b>Area</b></li> </ul> <p>For specific information about chart types, see <a href="#">Types of Report Formats</a>.</p>
<b>Sub-type</b>	<p>The way the information appears graphically. The available options are based on the chart you select in the <b>Type</b> list as follows:</p> <ul style="list-style-type: none"> <li>• <b>Bar and Column charts</b>—Stacked, Clustered, Stacked 3D, or Clustered 3D</li> <li>• <b>Pie chart</b>—2D or 3D</li> <li>• <b>Area chart</b>—Stacked or Not Stacked</li> <li>• <b>Line chart</b>—With Marker or Without Marker</li> </ul>
<b>Legend</b>	<p>Location of the chart legend.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• <b>Bottom</b></li> <li>• <b>Right</b></li> <li>• <b>None</b></li> </ul>

<b>X-Axis</b>	<p>Location and orientation of the X-Axis label. Applies to Bar, Column, Line, and Area charts only.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• <b>None</b>—No axis labels are displayed.</li> <li>• <b>Wrap</b>—Displays text horizontally. This is the default.</li> <li>• <b>Rotate</b>—Displays text vertically.</li> <li>• <b>Slant</b>—Displays text at a 45-degree angle.</li> <li>• <b>Stagger</b>—Displays text in an alternating pattern; one above, one below.</li> </ul> <p><b>Wrap</b> and <b>Stagger</b> are not supported on the Apple iPad, since Adobe Flash is not available on the iPad.</p>
<b>Labels</b>	<p>Display or hide the values on the chart.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• <b>Show</b></li> <li>• <b>Hide</b></li> </ul>
<b>Results</b>	<p>How many results to show on the chart.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• <b>All</b>—Show all available results.</li> <li>• <b>Top</b>—Sort the results in descending order (highest to lowest) based on the first data series, then display the first X results based on the number selected in the spinner box.</li> <li>• <b>Bottom</b>—Sort the results in ascending order (lowest to highest) based on the first data series, then display the first X results based on the number selected in the spinner box.</li> </ul>

#### 1.5.2.3.1 Types of Report Formats

You can view report data as a graphic representation, such as a pie chart, or a tabular format which lets you view information as a spreadsheet. You can switch between these views at any time.

When using graphical charts, it is recommended that you match the type of data to the appropriate type of chart. For example, if you want to show trends over a period of time, it is best to use a line chart instead of a pie chart because a pie chart is designed to display information for a specific moment in time.

### Graphic Charts

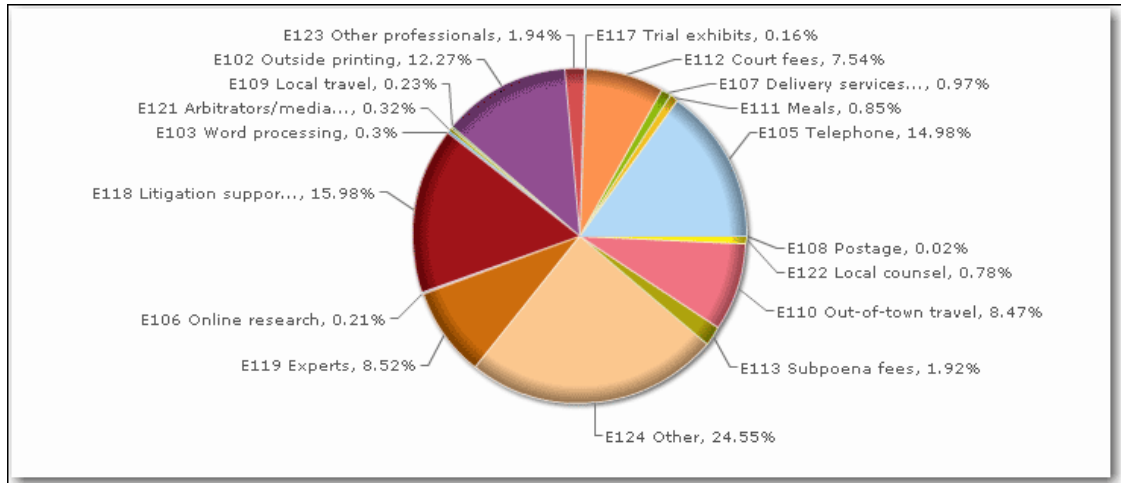
There are several types of default graphic reports as follows:

- Column

- Bar
- Pie
- Line
- Area

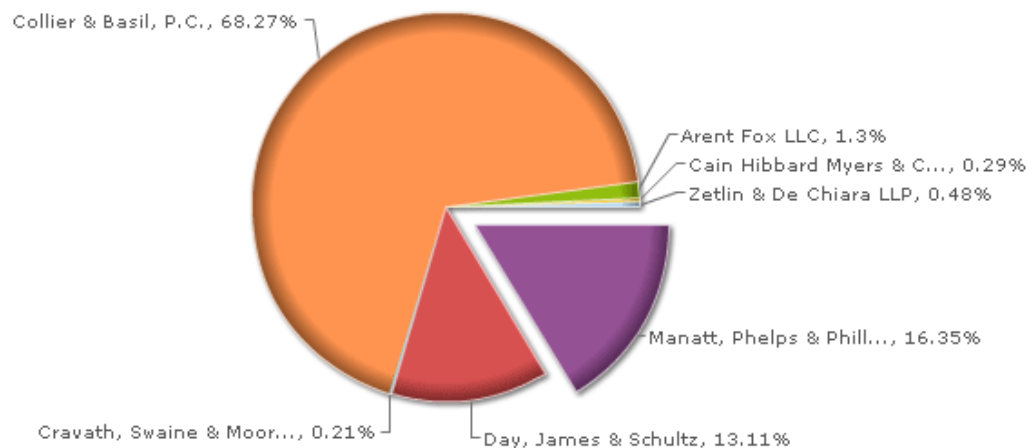
Once report data is open as a graphic chart, with the appropriate rights, you can make changes to the way the chart looks and export the graphic to a .pdf file. See [Changing the Appearance of a Report](#) for more information.

- **Pie Chart**—Displays information in a circular format with specific data represented as slices of the overall pie. Best used to display information for a specific point in time.



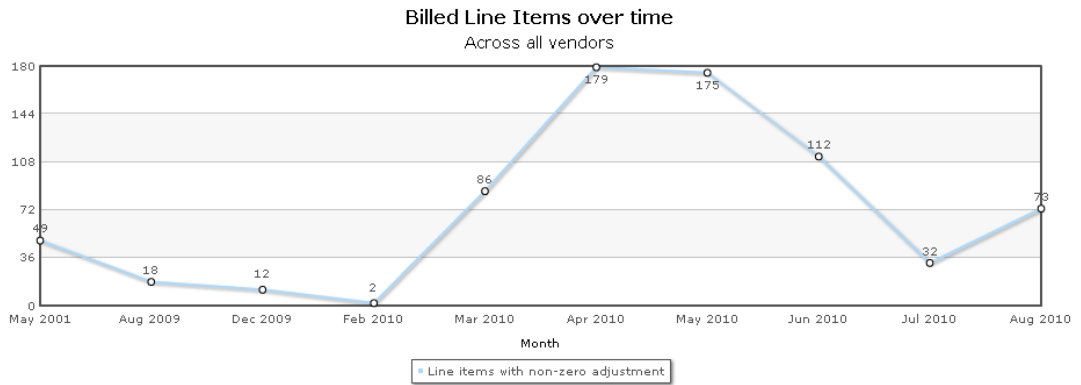
**Pie Chart**

*Tip: When working with pie charts, single click on a slice of the pie to separate it from the rest of the pie. Single-click again to move it back to the pie.*



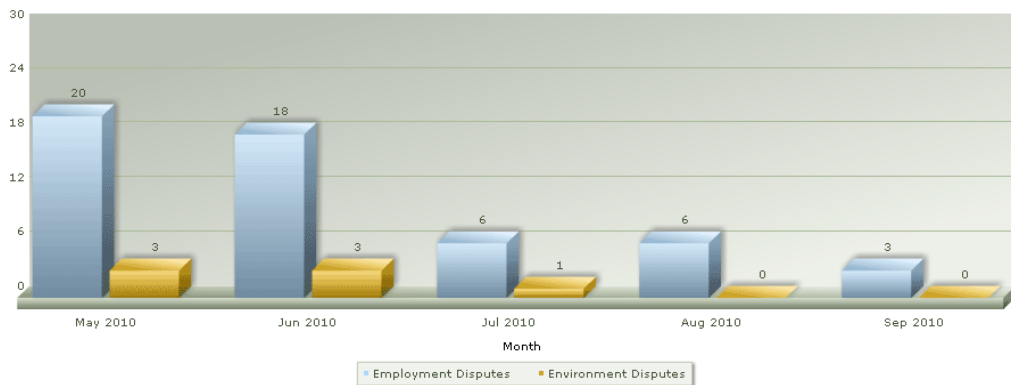
**Separating a Slice of Pie**

- **Line Chart**—Display information as points connected by a line. Best used to show trends in data over a period of time.



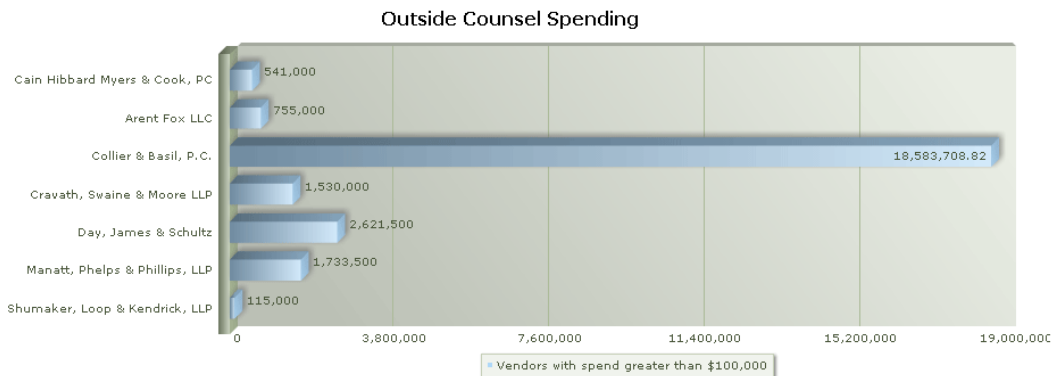
Line Chart

- **Bar chart**—Displays information in horizontal bars. Best used to display a small amount of information over time and under different conditions.



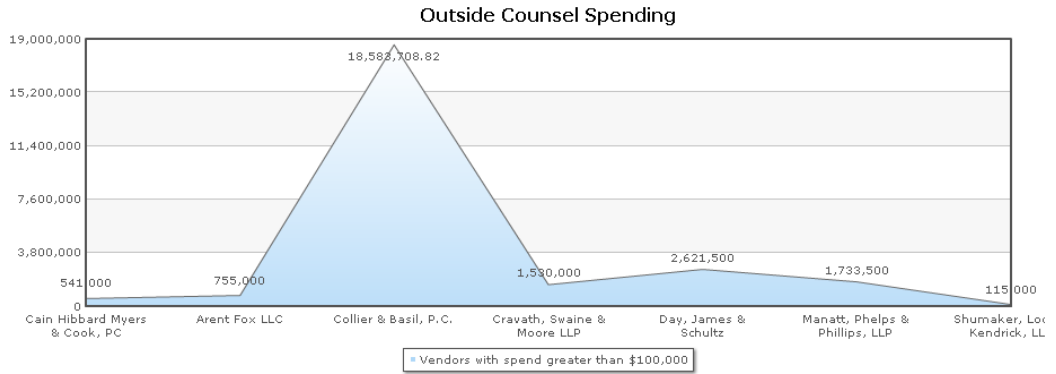
Bar Chart

- **Column chart**—Displays information in vertical bars. Best used to display a small amount of information over time and under different conditions.



3D Column Chart

- **Area chart**—Displays information in a filled area display. Best used for cumulative totals over time.



Area Chart

### Tabular Chart

Tabular chart data is displayed in a spreadsheet format. You can sort the information in ascending or descending order by clicking the column header. (Data which spans columns does not have a column header, thus cannot be used for sorting.) You can export tabular data to Excel or to a .pdf format.

#### To access a Tabular chart

If a chart is defined as a **Tabular** only, it will open in **Tabular View** by default.

Object Link	Main Assignee	Default Category
ADA (Americans with Disabilities Acts) (1 records)		
LIT-000276-Ferguson v. Javadzadeh	Neisen, Daniel	ADA (Americans with Disabilities Acts)
Age (9 records)		
DISP-000039-Acme Corporation vs Keldan Corp	Neisen, Daniel	Age
DISP-000115-Henderson v. Spectra	Larson, Brooke	Age
DISP-000149-Aluweb v. Keldan Corporation	Bultman, TJ	Age
DISP-000163-Descendoar v. Keldan	Zimmerman, John	Age
LIT-000212-New Version - Acme Corporation vs Keldan Corp	Neisen, Daniel	Age
LIT-000286-JM Family vs Acme Corp	Haake, Patrick	Age
LIT-000294-Novartis v. Smith	Lewis, Kyle	Age
LIT-000329-Descendoar v. Keldan v5	Zimmerman, John	Age
LIT-000385-Coleman, Jack v. Keldan (TX State Court)	Ballard, Franklin	Age
Air Quality (1 records)		
LIT-000278-Davis vs NBC Universal	Neisen, Daniel	Air Quality
Bankruptcy (3 records)		
DISP-000157-Griswold v. Spectra	Larson, Brooke	Bankruptcy
LIT-000348-New Matter with Cravath	Blackstone, Rick	Bankruptcy
LIT-000349-New Matter with Cravath phase 2	Blackstone, Rick	Bankruptcy

Tabular View

If you are in chart view, click the **View Results** link to present the chart in a tabular spreadsheet format. This link is in the upper right corner of the screen.

### 1.5.2.4 Exporting Reports

You can export a graphic chart to a .pdf format, and a tabular chart to both Excel and a .pdf format.

**Note:** To export to the .pdf format, you need at least Acrobat Reader version 9.4.0.

#### To export Reports

Run the report that you want to export. See [Running Reports](#) for more information.

- **For a Graphic chart**—Click **Export Chart**. The chart is saved and opened as a .pdf.
- **For a Tabular chart**—Click the **Export Results** drop-down list, and then select **Export as Excel** or **Export as PDF**. The data is saved or opened based on the action you select.

## 1.5.3 Building Reports

To build a report, you name the report, specify its properties, then save your work to one of the report folders. Once a report has been built, it is managed and executed as described in [Viewing Reports](#) and [Running Reports](#).

To edit an existing report, click the **Edit** button while viewing the report specification.

To build a new report, click the **New** button while viewing a listing of existing report specifications. The new report, when completed, will be saved by default in the folder which you had been viewing.

Whether building a new report or editing an existing report, you must edit the report's properties. There are five kinds of property pages. The General Properties, Data Series, and Chart pages have some properties that are mandatory. The report specification cannot be saved until at least those properties have values. Refer to the following topics:

- [General Properties](#)
- [Data Series](#)
- [Chart](#)
- [Security](#)
- [Scheduler](#)

### 1.5.3.1 General Properties

The **General** page allows you to view or edit general properties of a report specification. It appears by default when a new report specification is started, or when an existing report specification is clicked. You can also return to this page by clicking the **General** tab in the left pane of the report specification page.

The table below contains the general properties of a report specification. Field descriptions are listed in display order.

**General Properties Field Descriptions**

Field	Description
-------	-------------

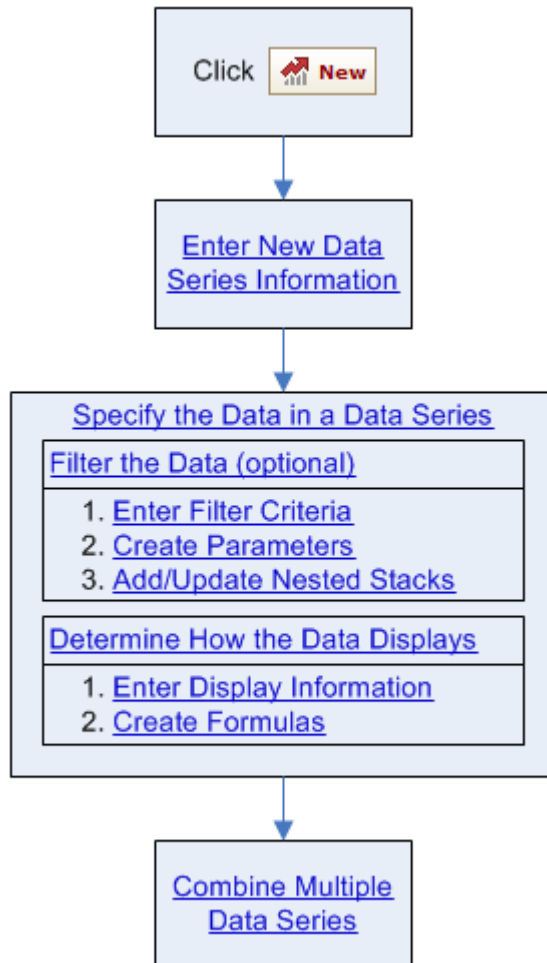
(* = required)	
<b>*Report Name</b>	Your name for this report specification. Each specification in a report folder must have a unique name within that folder. Maximum length: 250 characters.
<b>Unique Key</b>	<p>This text string is used for localizing reports - translating the report prompts to other languages. If your organization does not do localization, you should leave this field empty. If you plan to do localization, enter a string that is unique (a string that is not found in the Unique Key of any other reports.) Use only characters A-Z, a-z, and 0-9. Maximum length is 50 characters.</p> <p>This field is only editable if it is empty. Once you have supplied a text value to this field, it cannot be changed.</p> <p>For more information about localization, see <a href="#">Localizing Reports</a>.</p>
<b>Folder Path</b>	Describes where the report is found in the hierarchy of report folders. This is a read-only link. Clicking the link takes you out of the report edit page to the list page of that folder.
<b>Description</b>	Text describing the purpose of this report. Optional. Maximum length: 4000 characters.

When creating a new report, after you save your **General** values, the **Data** tab in the left pane is enabled. Click **Data** to begin creating one or more data series for the report.

### 1.5.3.2 Data Series

A data series is a group of data based on filtered criteria and fields you want to appear in the results. A report contains one or more data series. When you have more than one data series, you can combine them if necessary.

The following flowchart illustrates how to create a data series. Click areas of this flowchart for more information. **Note:** *The 'New' Icon has been redesigned for 5.x+ users. This does not affect functionality.*



The **Data** page of the report displays a row for each existing data series, plus a block with parameter information. A parameter is a value for the report that you enter into a prompt when the report runs. When you create a new report, this page is disabled until you specify the properties under the **General** tab and save. When the report specification is in **Edit** mode, the **Data** page allows you to edit and create data series.

You can delete an existing data series or parameter by selecting the check-box next to the row and clicking **Remove**.

Each row of a data series includes the following fields:

#### Data Series List Field Descriptions

Field	Description
<b>Order</b>	A number that controls the sequence of appearance for each data series that appears in the report output. If multiple data series have the same value, they appear in alphabetic sequence of <b>Name</b> .



<b>Name</b>	The name of the data series in the report output. Click the name to specify data for the data series, as described in <a href="#">Specifying Data for a Data Series</a> . This field is not editable.
<b>Unique Key</b>	<p>A text string used to localize reports and translate the report prompts and labels. If your organization does not localize, leave this field empty. If you plan to localize, enter a string that is not used as the <b>Unique Key</b> of any other data series in the report. Use only characters A-Z, a-z, and 0-9. The maximum length is 50 characters.</p> <p>This field is only editable if it is empty. Once you enter a text value for this field, you cannot change it.</p> <p>For more information about localization, see <a href="#">Localizing Reports</a>.</p>
<b>Record Type</b>	The types of records where fields in this data series originate. This field is not editable.
<b>Create a Copy</b>	Creates a new data series with the same information. After you create a copy, click the text next to the <b>Name</b> field in the <b>Properties</b> section to update it with a better description of the data series.

#### Parameter Field Descriptions

Field	Description
<b>Order</b>	A number that controls the sequence that the report prompts parameters at runtime. If multiple parameters have the same value, they appear in alphabetic sequence of <b>Name</b> .
<b>Name</b>	The name of the parameter. This name is not visible in the report output at runtime, solution developers may use it to embed the report in a custom page or portal pane. The name also appears in i18n keys for localizing reports. This field is not editable.
<b>Field Type</b>	The datatype of the parameter.
<b>Operator</b>	The numeric operator that tests the report data against the parameter value. This field is not editable.
<b>Default Value</b>	A value that the end user sees at runtime when prompted to enter parameter values.

**Note:** You cannot create new parameters using the **Parameters** block, only while defining filter criteria, as described in [Filtering the Data](#).

## 1.5.3.2.1 Entering New Data Series Information

After you click **New**, the **New Data Series** popup appears so that you can enter general information about a data series.

**To enter new data series information**

1. From the **New Data Series** popup, select a **Record Type**. The record type determines which types of fields the data series uses to create the report.
2. You cannot change the record type after you click **Ok**.
3. Enter a descriptive **Name** for the data series.
4. If you localize TeamConnect, enter a **Unique Key**.
5. (Optional) Enter a number in the **Order** field. If multiple data series have the same value in **Order**, they appear in alphabetic sequence of **Name** in the report output.
6. Click **Ok** to continue.

After you click **Ok**, you can edit the **Name** and **Order** of the data series.

**To edit the properties of a data series**

1. From an open data series, click **Edit Properties**.



Clicking **Edit Properties** to Edit Data Series Properties

2. Under **Properties** in the left pane, click the text next to **Name** and **Order** to edit the current text.

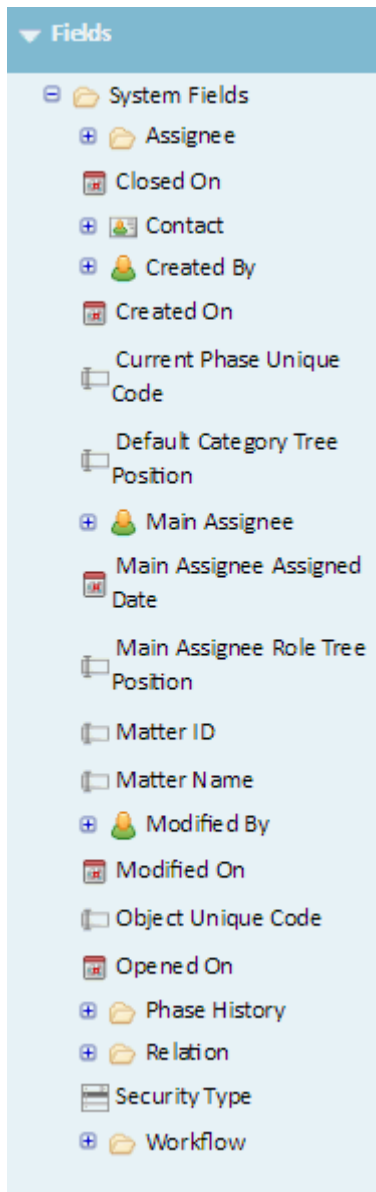


Editing the Properties of a Data Series

**Note:** You can also edit **Unique Key**, in addition to **Name** and **Order**, on the **Data** page of the report. See [Data Series List Field Descriptions](#) for more information.

### 1.5.3.2.2 Specifying Data for a Data Series

You specify data for each data series on the **Filter Criteria** and **Results Display** tabs. For both tabs, the Fields list in the left pane lists all fields for the chosen record type.



**Fields List**

Under the **Filter Criteria** tab, the **Fields** list includes all the fields that you can use to filter the data for a report so that only data that meets the filter criteria appears.

Under the **Results Display** tab, the **Fields** list includes fields that appear in the results.

## Selecting Fields

Drag a field from the **Fields** list to the work area under the tabs. Make sure a dotted outline appears for the field to stay.

Delete a field by clicking the  icon to the right of its name.

## Types of Field Folders

The fields are organized hierarchically within sections.

- **System Fields**

Standard fields that come with TeamConnect before any customizing is done.

- **Custom Fields**


Fields the solution developers in your organization add as they customize the TeamConnect design. In some cases, such as Legal Matter Management, custom fields are part of a pre-packaged design that is added to the standard TeamConnect design. Because custom fields are always associated with categories, this section is organized by category. Custom fields from the root category appear first. Expanding a category folder reveals the custom fields that belong to that category.

- **Related Fields**



Fields that pertain to the record type of the data series but that are not in the actual record type itself. For example, fields from Involved records appear here, as do fields from any child objects or embedded objects that relate to the data series record type. A folder represents each related record type. Expanding that folder reveals the fields that belong to the related record type.

## Field Specifics


In each folder, you might have the following types of fields:

- **Folders**—A  icon next to a field indicates additional levels of hierarchy. Click this icon to show additional fields at the next deeper level. For example, fields representing users, such as **Created by** and **Modified by** fields, include that icon. Click this icon to reveal the Contact fields associated with the user.

**Caution:** *Using fields from the deepest levels (more than three levels) may cause extensive database access that makes the report run slowly. Use good judgment in such cases, and talk to a solution developer if you have concerns that multi-level fields are causing slow report times.*

- **Parent Project**—If the record type for this data series is a child or an embedded object, you see a  icon named **Parent Project**. Expand the  icon to see the fields that pertain to the parent record.

For certain record types, the field has a different name from **Parent Project**. For example, if the record type is **Line Item** or **Non US Tax**, the parent folder is named **Line Item Invoice** or **Invoice** (respectively).

- **Object Link** (Results Display)—Using this field places a highlighted link with the name of the record in the results display at runtime. When clicked, the link opens the view page for the record. An Object Link is available in the field list of every record type.
- **Formula** (Results Display)—A number field you can edit. When you drag a  **Formula** field into the work area of the **Results Display**, you can [create a formula](#).

## 1.5.3.2.2.1 Filtering the Data

You filter the data by adding filter criteria so that only the data that meets the criteria appears in a report. To specify how data filters, use the **Filter Criteria** tab.

The visual layout of the **Filter Criteria** tab resembles a logic diagram, and you may find it helpful to have a logic diagram for your filters sketched out before you begin using this tab.

This tab enables the entry of a large, complex hierarchy of criteria, with mixed **AND** and **OR** specifications. You may also enter as few as one criterion if your filter needs are simple, or no filter criteria at all.


### To enter filter criteria for a field

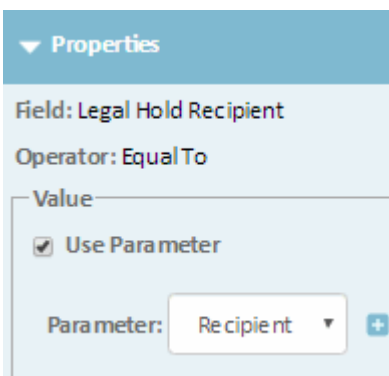
1. Drag a field from the Fields list to the dotted outline that appears.



Dragging a Field to the Dotted Outline

The newly dropped field appears with a default operator that you use to specify the criteria. For example, a date field appears with the operator **In the Last**.

2. Click the  icon to the right of the field to edit the **Properties** of the field.



Properties for a Field

3. Click the text next to **Operator** to select from a list of operators.
4. If the type of operator selected requires a value, specify a value for the operator in the **Value** section. You can enter one of the following types of values:
  - A literal value—Depending on the field, a list of values or Click to edit text appears. Use the list or text field to enter a value for the operator.

For example, for a date field with the **In the Last** operator, you could type 10, which would mean that you are filtering on dates that occurred within the most recent 10 days.

For a date field with the **Between** operator, if you want a report of all invoices dated after 1/1/2011, enter 1/1/2011 for the first value (the start date), and leave the second value (the end date) empty.

- A parameter—Select the **Use Parameter** check-box to indicate that the report should ask the user for a value at runtime. After checking this box, you may select an existing parameter from the adjacent drop-down list. To create a parameter, see the next section.

The screenshot shows a 'Properties' panel for a report field. The field is 'Legal Hold Recipient', the operator is 'Equal To', and the value is 'Recipient'. The 'Use Parameter' checkbox is checked, and there is a plus icon next to the parameter name 'Recipient'.

The Report Requires a Value Before Running When You Use a Parameter

### To create a parameter

1. After you have placed a check-mark in the **Use Parameter** check-box, click the **+** icon.
2. Enter a **Name** for the parameter. The name is an internal attribute that solution developers use to embed the report in a custom page or portal pane. Use a name that describes the purpose of the parameter. Each parameter name within a report must be unique.
3. Enter a **Label** for the parameter. The label is text that prompts the end user when they run the report.
4. (Optional) Enter a **Default Value** that the user sees when prompted to enter a parameter value.

**Note:** After you have applied your edits and closed this page, all parameters appear in the **Data** tab. At that point you can edit their **Order** property, which determines the order in which parameters appear to the end user at runtime. If you don't change the **Order** property of parameters, they appear to the end user at runtime in alphabetic sequence of the parameter labels.

The screenshot shows a 'Create Parameter' dialog box. The dialog has three input fields: 'Name' with the value 'ContactType', 'Label' with the value 'Type of Contact', and 'Default Value' with the value 'COMPANY'. There are 'Save' and 'Cancel' buttons at the bottom. The background shows a search filter with the criterion 'Legal Hold Recipient Equal To (Recipient)' and an 'Add Nested' button.

Using the Parameter Check-box and Plus Icon with a Criterion's Value

### To create a parameter for Document record types

If you are reporting on fields related to a Document record, your report might include data for more than one version of the document. To prevent multiple versions from appearing in a report, choose the following values for each field:

- **Field—Document Type**
- **Operator—Is**
- **Value—Document**

### Nested Stacks

The **Filter Criteria** tab includes one or more stacks, which are visual representations of a set of criteria. A nested stack uses the same rules and properties as the previous stack. Each stack includes at least two filter criteria or a field criterion and a nested stack, one level lower in the hierarchy. The following image includes an example of both.

### To change a relationship for a stack

The default relationship between filter criteria in a stack is **AND**. To change the relationship to **OR**, click the word **AND** for **OR** to appear. Click the word **OR** to change the relationship to **AND**.




Clicking **AND**  
Changes It to  
**OR**


Adding more criteria to an **AND** stack usually reduces the number of records in the report. Adding more criteria to an **OR** stack usually increases the number of records in the report.

**Note:** A very complex hierarchy of filter criteria, particularly nested OR criteria, may create complex database queries that run relatively slowly. If you suspect that your filter criteria may cause slow performance, ask a solution developer for advice about the best way to structure the criteria.

### To create a nested stack

Click  **Add Nested** in the current stack. A new stack appears, nested one level deeper than the current stack.

### To rearrange the hierarchy of stacks

You can move a filter criterion from one stack to another, and move stacks to different parts of the hierarchy. All stacks except the top-level stack display the  icon. Click the icon to drag and drop an entire stack elsewhere in the hierarchy. If the stack has any children, those also move.

### To delete a nested stack

Each stack except the top level contains the  icon. Click the icon to receive a confirmation prompt regarding the deletion of the stack. Click **OK** to delete the stack and any children of the stack in the hierarchy.

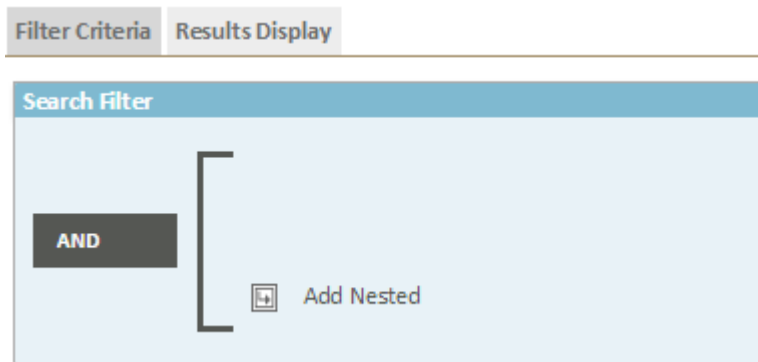
#### 1.5.3.2.2.2 Determining the Data that Displays

On the **Results Display** tab, you choose and arrange the data fields and formulas that appear in the report output.


**Note:** In the **Chart** tab, you will have the opportunity to display group statistics (record count, numeric field totals and averages, etc.). Fields used for grouping must also be present in the **Results Display**.

### To enter results display information for a field

1. Drag a field from the **Fields** list to the dotted outline that appears.



Dragging a Field to the Dotted Outline

2. Click the  icon to the right of the field to change the **Properties** of the field.
3. (Optional) Click the **Click to edit** text next to the **Label** field to determine the heading for the field in the report's output. If you leave the **Label** property empty, the default label associated with the field appears. All field labels within a data series must be unique.
4. (Optional) Click the **Click to edit** text next to the **Unique Key** field to enter a string used to localize column headings and translate the label to other languages. If you are using the default label, leave this property empty. If you override the default label by entering a value into **Label**, you should enter a **Unique Key** so that your overridden label can localize. Enter a string that is not used as the **Unique Key** of any other result fields in this data series. Use only characters A-Z, a-z, and 0-9. Maximum length is 50 characters.



This property is only editable if it is empty. Once you enter a text value for this field, you cannot change it.

For more information about localization, see [Localizing Reports](#).

- (Optional) Change the **Format** of numeric and date fields. The **Format** of most other fields is not editable.

Numeric fields include the following format options under **Properties**:


- **Decimal Points**—Select the number of digits that you would like to appear to the right of the decimal point.
  - **Total**—Select how the summary information for this field calculates the numbers in the column: **Sum**, **Average**, or **Count**.
- (Optional) Place a check-mark in the **Span Columns** check-box to make the field appear on a line of its own in the report. When this check-box is not checked, the field occupies a column in the report, with a heading at the top of the column. **Span Columns** is not available for numeric fields.


In a report, spanned fields appear below the non-spanned fields, and each of their values occupy a full row in the report, spanning all columns. If there are multiple spanned fields, the leftmost spanned field in the **Results Display** tab appears first, followed by the spanned field to its right, and so forth.

### To create formulas

- Drag the **Formula** field to the dotted outline that appears.
- Next to the **Formula** field, click the **Edit** icon. The **Edit Formula** pane appears below the **Results Display** work area with the **Properties** of the formula visible in the left pane.
- Drag at least two fields with a numeric data type to the dotted outlines in the **Edit Formula** pane.

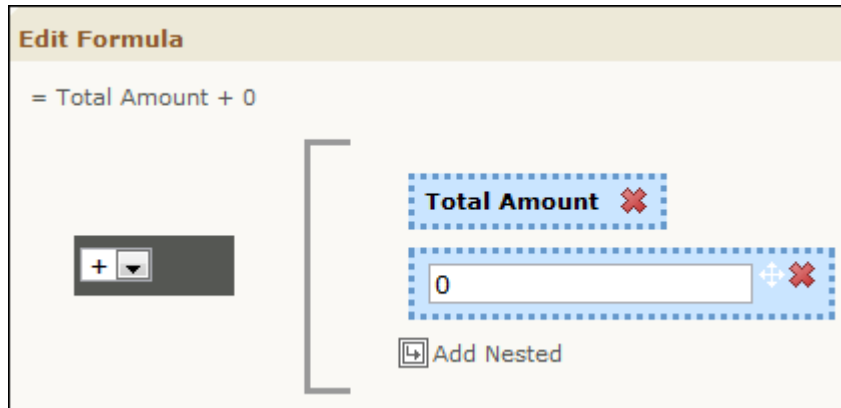
To enter a constant value for one of the fields, drag the **Constant Value** field. You may type up to 11 digits to the left of the decimal point and 6 digits to the right.

 **Formula**

 **Constant Value**

#### Constant Value Under the Formula Field

The formula expression displays as text just above the graphical layout of the operands and operators, and the text updates as you work in the graphical layout and properties pane.



Edit Formula Work Area

4. Choose an operator that indicates how the fields relate to each other. From the drop-down list, choose the plus, minus, multiply, or divide symbol as an operator.

That operator is applied to all the operands found at that level, and the left-to-right order of operations used in algebra is also used in formulas. For example, if a nested level has operands named Budget, Fee, and Expense, and the operator is a minus sign, the portion of the formula expression would be **(Budget - Fee) - Expense**.

The same principles of nesting that applied to Filter Criteria are used in the **Edit Formula** work area. Each level of nesting implies a set of parentheses in the final formula expression. See [Nested Stacks](#) for more information.

5. Under **Properties**, enter any additional information as you would for any other numeric field. See [To enter results display information for a field](#) for more information.

#### 1.5.3.2.3 Combining Multiple Data Series Using Cross-Object Reporting

After you create multiple data series for different objects, you can combine and display the data in a single report using cross-object reporting.

**Note:** Before you complete the following steps, save the report after you have created all data series.

#### To combine multiple data series

1. Check the **Combine Data Series** box on the **Data** page.
2. Click the **Results Display Configuration** button.  
The **Configuration** page appears.
3. Create join conditions. See [Join Conditions](#) for more information about the **Join Conditions** section.
4. Add results to display in the report. See [Results Display Fields](#) for more information about the **Results Display** section.

**Note:** If you later decide that you want multiple, individual data series, unchecking the box **Combine Data Series** returns your report to individual series.


#### 1.5.3.2.3.1 Join Conditions

After you click the **Results Display Configuration** button, you can specify the **Join Conditions**. Each **Join Conditions** section displays as an area into which you drag fields that connect two data series.

The number of join conditions is equal to the number of individual data series you have defined, minus one. For example, a report with four individual data series requires three join conditions. If a data series is not represented in a join condition, its data does not appear in the report.

Consider the following when you drag fields to the **Join Conditions**:

- The fields from each data series should be compatible, meaning that both data types match.
  - **Matching Project fields**—A general Project (relating to "All Projects") can match a specific Project (for one record type) or another general Project. However, two specific Project fields must share the same record type, or they do not match.
  - **Matching Object Link fields**—Object Link fields are always specific to one record type, so they can match a general Project or a specific Object Link of the same record type. Two specific Object Link fields of different record types do not match.
- Each join requires you to drag a field from one data series, and then a compatible field from another data series, into the join condition. Fields must be one of the following datatypes:
  - An Account
  - A User
  - A Contact
  - A Project (Custom Object)
  - An Object Link

Click  in the **Join Conditions** area to set the type of join from the **Properties**. The type of join can be an **Inner Join** or a **Left Outer Join**. These join types are equivalent to the types of joins found in your database server.

#### 1.5.3.2.3.2 Results Display Fields

Drag output fields from the individual data series in the left pane into the Results Display area. Edit the physical order, formatting, and other properties of those fields in the same way you edit an individual data series. See [Specifying Data for a Data Series](#) for more information.

You can drag fields from the **Join Conditions** into the **Results Display** area, and (if they are an eligible datatype) from the **Results Display** into the **Join Conditions**. The **Results Display** does not allow the same field, from the same data series, to be dragged in twice.

If you hover the mouse over a field in the **Join Conditions** area or the **Results Display** area, popup text shows the name of the field and the data series and path that contains the field.

#### 1.5.3.2.3.3 Sample for Combining Data Series

Combine data series when you want to see data from more than one record type in a single report.

A common use case for combining data series is a report that includes information about all the accounts associated with a project. An example of this report looks like the following figure.

#### To create the sample report

1. Create a report.
  - a. From the **Reports** tab, click **New**.
  - b. Enter a **Report Name**.
  - c. Save the report.
2. Click **Data** in the left pane.
3. Create a data series for a project.
  - a. From the **Data Series** page, click **New**.
  - b. From the **New Data Series** window, select a project type for **Record Type**.
  - c. Enter a **Name** for the data series.
  - d. Click **Ok**.
  - e. Select the **Results Display** tab.
  - f. From the **Systems Fields** folder, drag the following to the **Results Display** section:
    - **Object Link**
    - **Matter ID**
  - g. Click **Apply & Close** to save.
4. Create another data series for accounts.
  - a. From the **Data Series** page, click **New**.
  - b. From the **New Data Series** window, select **Account** for **Record Type**.
  - c. Enter a **Name** for the data series.
  - d. Click **Ok**.
  - e. Select the **Results Display** tab.

- f. From the **Systems Fields** folder, drag the following to the **Results Display** section:
  - **Project**
  - **Name**
  - **Total Used**
- a. Click **Apply & Close** to save.
5. Save the report.

When you create or update a data series, you must save the report before you use it to create join conditions.
6. Place a check-mark in the **Combine Data Series** check-box on the **Data** page.
7. Click the **Results Display Configuration** button.
8. Drag the following **Fields** to the **Join Conditions** section:
  - **Object Link** from the project data series folder
  - **Project** from the account data series folder
9. Drag all **Fields** in both folders to the **Results Display** section.
10. Click **Apply & Close** to save.
11. Click **Chart** in the left pane.
12. From the **Group By** field, select **Matter ID**.
13. Click **Save & View** from the save drop-down options.
14. Click **Run Report** to see the combined data series in a report, as shown in [the sample report](#).

#### 1.5.3.2.3.4 Best Practices for Combining Data Series

Follow these suggestions to improve the coordination between individual data series and combined data series:

- If you have done edits on individual data series, save your work before you begin work on a combined data series. If you leave work unsaved for individual data series, you will notice the following effects when working on combined data series:
  - Newly added, unsaved result fields in existing individual data series are not visible in the left pane.
  - Newly added, unsaved individual data series are not visible in the left pane, so they cannot appear in the combined data series report.
- Before you remove a result field from an individual data series, make sure that it is not being used in the combined data series, either as a result field or a join condition field. Otherwise, your attempt to remove the result field from the individual data series will cause an error message.

### 1.5.3.3 Chart

Use the **Chart** page to specify the visual characteristics of the report.

When using graphical charts, it is recommended that you match the type of data to the appropriate type of chart. For example, if you want to show trends over a period of time, it is best to use a line chart instead of a pie chart because a pie chart is designed to display information for a specific moment in time only.



The **Chart** page presents a Chart Data block for each data series in the report.

### Using the Chart Page

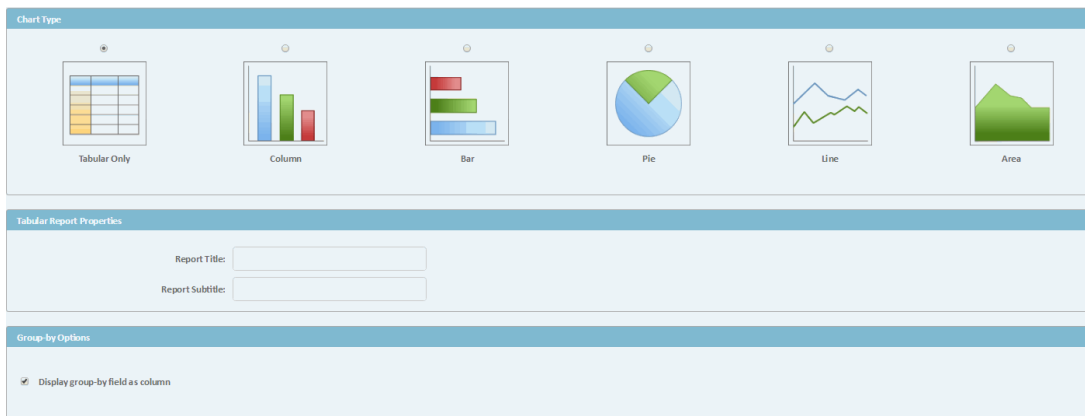
Some chart types aren't effective with certain kinds of data series. Be sure to choose an appropriate type. See [Types of Report Formats](#) for more information.

**Note:** If there are a very large number of records that fit the filter criteria, the chart will only show the number of search results that your TeamConnect administrator specified in the search results limit setting. For example, if the administrator specified a limit of 3000 search results, but there are 4400 records that fit your filter criteria, then the chart will be based on only 3000 records.

### To create a tabular report

1. In the **Chart Type** section, select **Tabular Only**.
2. Organize the table by groups of data. Use the **Order Group By** field to select one or more data fields. The options in the **Order Group By** drop-down lists are the non-numeric options in the **Results Display** section of each data series.
  - a. Under **Order Group By**, select the first field from the drop-down list that you want to group data by in the table.
  - b. Click the Add  icon one or more times to group by another data field within the previous field. You can group by up to five data fields.
  - c. Click the Remove  icon to remove an extra data field.

If you are combining multiple data series in a report, you can only group by one data field.



The screenshot shows the 'Chart Type' selection interface. The 'Chart Type' section has six radio buttons: 'Tabular Only' (selected), 'Column', 'Bar', 'Pie', 'Line', and 'Area'. Below this is the 'Tabular Report Properties' section with two text input fields: 'Report Title' and 'Report Subtitle'. The 'Group-by Options' section has a checked checkbox for 'Display group-by field as column'.

Organizing a Tabular Report by Groups of Data with Results of these Selections

3. If you select an option that is Date-related, such as **Opened On**, the **Summarize Date By** field appears so that you can specify how to group by the date.
4. If you select a category or a lookup table, the **Summarize Item by** field appears so that you can specify a category or lookup table option.
5. To hide detailed rows in the table, remove the check-mark from the **Show Details** check-box.  
  
By default, the **Show Details** check-box causes all the fields in the data series to appear in a row after each group of data. If you remove the check-mark, the row of details does not display.
6. Repeat steps 2 through 5 as necessary for each additional data series.
7. Enter a **Report Title** and **Report Subtitle** in the **Tabular Report Properties** section.
8. To remove columns for the data fields you are grouping by from the table, remove the check-mark from the **Display group-by field as column** check-box.  
  
Because each group has a separate summary row at the top of the group, you may not want to display that same value as a column in the report.
9. [Select a save option.](#)

#### To create a column, bar, pie, line, or area report

1. In the **Chart Type** section, select **Column, Bar, Pie, Line, or Area**.
2. In the Group By drop-down list, choose the field you want to appear as the axis or category on the chart.  
  
All chart types, except pie, contain properties for the x-axis. If a chart has multiple data series that use a date/time field for **Group By**, choose a date/time field for **Group By** to ensure a consistent look for the x-axis. The date/time field does not have to be the same field in each data series. Examples of date/time fields are **Opened On, Modified On, Closed On, and Inactive Date**.  
  
If the **Group By** field is not a date/time field, the same field type should be used in every data series.
3. If you select an option that is Date-related, such as **Opened On**, the **Summarize Date By** field appears so that you can specify how to group by the date.
4. If you select a category or a lookup table for the Group By field, the **Summarize Item by** field appears so that you can group by a category or lookup table option.
5. In the **Chart Value** drop-down list, choose a field for the value axis. The options in this list are the numeric options on the **Data** page, **Results** tab and are used for the value axis. In addition to numeric fields, you may use **Record Count** to specify a simple count of the number of detail records that make up the **Group By**.
6. Repeat steps 2 through 5 as necessary for each additional data series.
7. Use the **Chart Properties** section to complete the following as necessary for each data series:

If you have chosen a **Group By** field, the **Drill-down** field appears. This field is a drop-down list of names of other reports. You can select one of them as a "sub-report". The user who is running the current report can click on an element in the current graphical output (a bar, a pie slice, etc.), and this click action triggers execution of the sub-report that you have chosen. This sub-report uses the clicked-upon Group By as a parameter to limit the sub-report output to just that one category.

- A sub-report only appears in the **Drill-down** list if that sub-report has at least one parameter that is the same datatype as the **Group By** field.

When your current report is grouping records by date, and it presents data by exact date, then the sub-report must have at least one parameter of the Date datatype. If your current report presents data by month, quarter, or year, then the sub-report must have at least two Date parameters to be eligible to be shown in your choice of reports for drill-down.

- Once you have selected a sub-report in the **Drill-down** list, the parameters of that sub-report are displayed and you must specify settings for each parameter. If there are multiple parameters, you may choose between settings **Use group-by value** (when available) or **Use default value**. At least one of the parameters must be set to **Use group-by value**.
8. (optional) In the **Chart Properties** section, add or edit the appropriate options. The options that you see in this section depend on the type of chart that you select. See [Chart Page, Chart Properties Section](#) for field descriptions.
  9. In the **Data Ranking** section, complete the following options if desired. Ranking is based on the first **Chart Value** selected in the **Chart Data** section.
    - In the **Show Results** list, select one of the following:
      - **All**: Displays all results in no particular order. This is the default.
      - **Top**: Display results in ascending order.
      - **Bottom**: Display results in descending order.
    - If you selected **Top** or **Bottom**, in the **Maximum Number of Results** box, enter a number to define the number of results to display and in what order, such as Top 5 or Bottom 50. You can enter a number greater than 0, and less than 100.
  10. [Select a save option.](#)

The tables below contain the default fields for the Reports Chart page, Chart Properties section. The fields that you see may vary based on your system settings, assigned rights, and the type of chart that you are using.

#### Chart Page, Chart Properties Section

Field	Description
(* = required)	



<b>Chart Style</b>	<p>The way that you want the selected chart to appear. The available options are based on the type of chart you selected in the <b>Chart Type</b> section.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• Bar and Column charts—Stacked, Clustered, 3D Stacked, or 3D Clustered</li> <li>• Pie chart—2D or 3D</li> <li>• Area chart—Stacked or Not Stacked</li> </ul> <p>This option is not available for Line charts.</p>
<b>Chart Title</b>	<p>The name of the chart.</p>
<b>Chart Subtitle</b>	<p>An additional descriptive name that appears under the <b>Chart Title</b>.</p>
<b>Chart Category Label</b>	<p>The label name that you want to appear for the selected category.</p>
<b>Chart Value Label</b>	<p>The label name that you want to appear for the selected chart value.</p>
<b>Legend Position</b>	<p>Whether or not you want this chart to have a legend, and the placement of the legend.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• None</li> <li>• Bottom</li> <li>• Right</li> </ul>
<b>Show Labels</b>	<p>The way that you want the chart labels to appear:</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• Uncheck the <b>Chart Value</b> box if you do not want the chart values (or currency symbol if applicable) to appear. This option is checked by default.</li> </ul> <p>The following options only appear if you are creating a pie chart:</p> <ul style="list-style-type: none"> <li>• Uncheck the <b>Chart Category</b> box if you do not want labels to appear on your chart.</li> <li>• Check the <b>Chart Percentage</b> box if you also want to display a percentage for each slice of the pie chart.</li> </ul>

<b>X-Axis Label Orientation</b>	<p>The orientation of the X-Axis labels for Bar, Column, Line, and Area charts.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• <b>None</b>—No axis labels are displayed.</li> <li>• <b>Wrap</b>—Displays text horizontally. This is the default.</li> <li>• <b>Vertical</b>—Displays text vertically.</li> <li>• <b>Slant</b>—Displays text at a 45-degree angle.</li> <li>• <b>Stagger</b>—Displays text in an alternating pattern; one above, one below.</li> </ul> <p><b>Wrap</b> and <b>Stagger</b> are not supported on the Apple iPad, since Adobe Flash is not available on the iPad.</p>
<b>Minimum Chart Value</b>	<p>The lowest number that you want to appear as a value in your chart. You can enter negative numbers in this field.</p>

#### Chart Page, Group by Options Section

Field	Description
<b>Display group-by field as column</b>	<p>In a tabular report that is grouped, each group will have a separate summary row at the top of the group. This summary row will display the group-by field value. For this reason, you may find it redundant to display that same value as a column in the report. To display the value in a column, check this box (this is the default.) To suppress that column in the report, uncheck this box.</p>

The examples below show a Chart page and the resulting chart based on the selected options:

**Chart Data for Actuals**

Chart Category:

Chart Value:

Total:

**Chart Data for Allocated**

Chart Category:

Chart Value:

Total:

**Chart Properties**

Chart Style:

Chart Title:

Chart Subtitle:

Chart Category Label:

Chart Value Label:

Legend Position:

Show Labels:  Chart Value

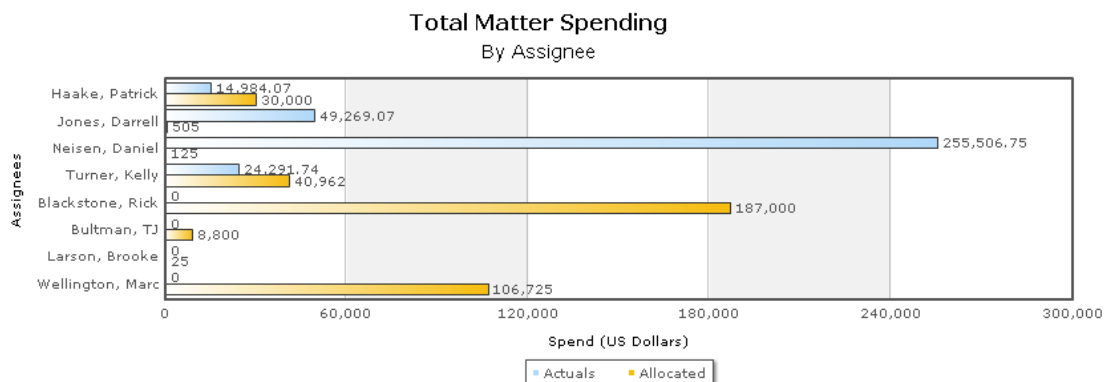
X-Axis Label Orientation:

Minimum Chart Value:

**Data Ranking**

Show Results:

Sample Chart Page



Sample Resulting Chart

### 1.5.3.4 Scheduler

Any existing report can be set up to run on an automatic schedule, with its output sent to a list of email recipients. Multiple schedules can be set up against the same report, using different recurrence date/time intervals, different parameters, and different email recipients.

**Note:** *To edit the Schedule tab in a report design, you need to have the right of report scheduling assigned to you by your TeamConnect administrator.*

**Important:** *When a report is run on demand, its output is localized to match the locale of the user who requested the run. However, when a report is scheduled, the report output will always be produced in the system default locale.*

#### 1.5.3.4.1 Edit Schedule Information

Each schedule requires information about recurrence, email recipients, and parameters. This information is entered in a similar way for both new and existing schedules. You can gain access to the Schedule tab in these ways:

- Edit a report. Click the Schedule tab.
- From the page that lists reports, click the properties (i) icon for a report. Click the Schedule tab.

To create a new schedule, click the New button.

To edit an existing schedule, click the name of the schedule. On the page that appears, click the Edit button.

#### To enter basic information about a schedule

1. Enter a value for the name of this schedule. Each schedule within a report must have a unique name.
2. Choose how frequently the report run recurs. Depending upon the frequency you choose, you must supply additional values (e.g. hours and minutes, day of the month, etc.)
3. Specify the start date and time at which this schedule shall become active.

This is not necessarily the date and time that the report will first run. For example, you might create a schedule that runs on the first day of each month, and specify a start date/time of the 20th of the current month. The report will not run on the 20th; instead, it will run on the first day of the next month. Exact information about the next run time is found in column "Next Run Time" when viewing a schedule (see [Viewing a Schedule](#).)

4. Choose when this schedule shall end (never, after a specified number of recurrences, or on a specified date.)

#### To enter information about email recipients

1. Choose an address type (To, CC, or BCC).
2. Select whether the recipient is an individual user, or a user group.

3. Choose the individual or group from the drop-down list.
4. Enter a subject for the email (defaults to the name of the report).
5. Enter a description, which will become the body of the email. HTML tags are permitted in the description.
6. Select whether the recipient should receive an Excel attachment or a .pdf attachment.
7. Select whether tabular data should be included with the chart output. For reports whose chart type is "Tabular", this option is pre-selected and not editable.
8. If there are more recipients to add, click the "plus" icon at the right edge of the recipient line to create a new recipient line, and follow the steps above.
9. You may remove existing recipients by clicking the "minus" icon at the right edge of the recipient line.

Each schedule requires at least one recipient.

**Caution:** *When a scheduled report runs, it uses the rights of the person who created the schedule to determine what information is included in the report. The report output is emailed to recipients who may or may not have rights to that information in TeamConnect. It is beyond the scope of the TeamConnect application to enforce security in the email recipient list. Use prudence when deciding which recipients get which reports.*

#### To enter information about parameters

1. Report parameters are displayed, one per line, in this block.
2. If no parameters are visible, that means that the report has no parameters defined for it, and you have no need to edit information in this block.
3. For each parameter that is visible, choose a value that should be passed to the report when it runs. For a new schedule, the values shown for parameters are the default values that were specified when the report was designed. You may enter different values. You may also clear a value for a parameter, leaving its value empty. The report will ignore an empty parameter (will not use that parameter as a criterion) when it runs.

##### 1.5.3.4.2 Viewing a Schedule

Edit a report, then choose the **Schedule** tab when in edit mode. Existing schedules are displayed, one per line. Click on the name of the schedule you wish to view.

##### 1.5.3.4.3 Activating or Deactivating a Schedule

Click on the **Activate** or **Deactivate** button at the top of the schedule. When a schedule is deactivated, all information about the schedule is retained, but the report does not run when a recurrence date/time is encountered. When a deactivated report is activated, no action occurs immediately, but the next time a recurrence date/time is encountered, the report runs as scheduled.

#### 1.5.3.4.4 Running a Scheduled Report Now

Click on the **Run Now** button at the top of the schedule. The report runs immediately, using the information specified in the schedule. This immediate run does not affect the schedule of recurring runs.

**Note:** *Run Now is not available if the schedule is currently deactivated.*

#### 1.5.3.4.5 Deleting a Schedule

Click on the **Delete** button at the top of the schedule.

#### 1.5.3.4.6 Changing an Existing Schedule's Run Time

After a schedule has been created and saved, it calculates its next run time, which is the value displayed in column **Next Run Time** when you are viewing the Schedule tab of a report. Each time a scheduled report runs, its next run time is recalculated.

If you later edit the schedule and change its frequency and/or its start time, the schedule must recalculate its next run time. It does so based on these rules:

If the schedule's frequency is changed to another value, the schedule refers to its Start date and time, and calculates the next run time which matches the new frequency, and that Start date and time. If that calculated next run time is in the past, the run time is advanced in increments until it is equal to or newer than the present date/time.

If the schedule's frequency is Hours and Minutes, and only the values of the Hours field and Minutes field are changed, the existing value of next run time will remain unchanged. However, when the report runs at that next run time, it will recalculate its run time, and during that recalculation the new values of Hours and Minutes will be used.

### 1.5.3.5 Security

This page permits you to explicitly declare which users and groups are allowed, or denied, access to the current report.

By default, access to a specific report is granted to any user who has View rights to the record type upon which a data series is based. You do not need to make changes on the Security page unless you wish to override that default behavior.

This page is used in exactly the same way that Security pages are used for other TeamConnect records. For details on how to use Security pages, see [Working with Record Security](#).

**Note:** *If a report has multiple data series which use different record types, a user must have rights to all of those record types. Otherwise the report is not visible to that user.*

## 1.5.4 Localizing Reports

When an end user's locale differs from the system default locale, report information is sometimes shown to the end user in their own locale, and sometimes in the system default locale. In general,

pages that relate to listing or executing reports will use the end user's locale, and pages that relate to the properties of a report design will use the system default locale.

"Static" text in report pages will always appear in the end user's locale. Static text includes things like the tab titles, block titles, labels of pushbuttons, labels of data fields, etc. But the data for an individual report is localized according to the rules below.

End user locale will be used for these pages:

- Report list
- Report execution
- Previously executed reports in Recent Items list

System default locale will be used for these pages:

- Report properties (whether editing or viewing)
- Report folder properties
- Previously edited report designs in Recent Items list

In order to localize report properties such as data series names, column headers, etc., there must exist an i18n key for each property. All "out-of-the-box" properties in TeamConnect already have such keys assigned to them. But since you can design brand-new reports that didn't exist in the out-of-the-box design, you need a way to assign such keys to the new report's properties.

This need is addressed by the **Unique Key** fields that are found in the various report design pages. If you do not plan to localize a report, you may leave these fields empty. But if you need to localize the report, each of these fields in the report must have a value. Each Unique Key field accepts characters A-Z, a-z, and/or 0-9. Spaces are not permitted. The values entered in these fields are later used to construct i18n keys that become associated with the report.

The Unique Key properties in the report design pages influence the creation of i18n keys as shown in the following example.

Tab	Description
<b>General</b>	The value entered in <b>Unique Key</b> becomes part of the i18n key. If you enter "TaskDueDate", the resulting i18n key becomes "report.TaskDueDate.name". The <b>Unique Key</b> value cannot be one that has been used by any other report.
<b>Data</b>	<p>If you enter a data series' <b>Unique Key</b> value of "Task1", the resulting i18n key becomes "report.TaskDueDate.dataSeries.Task1.name". Note the use of the report name's key as part of the data series' key. The <b>Unique Key</b> value cannot be one that has been used by any other data series within this report.</p> <p>Each field in the result display can have its own Unique Key. If a field in the result display has a field name of "Description" and a Unique Key value of "Descr", the resulting i18n key becomes "report.TaskDueDate.dataSeries.Task1.resultDisplay.Descr.label". The <b>Unique Key</b> value cannot be one that has been used by any other result field within this data series. Note that only result display fields that have an entry for their Unique Key</p>

property will not be exported for localization. Fields with that property empty will use the localized value of the default field label at runtime.

Parameters are automatically exported for localization, and you do not have the opportunity to manually enter a **Unique Key** for a parameter. Instead the i18n key is constructed automatically. For a parameter of name "start" the i18n key would be "report.TaskDueDate.parameter.start.label".

## 1.6 Appointments Calendar and Tasks

The appointment calendar is a daily planner that you can use to view, add, and edit all of your appointments and tasks.

Appointments may include internal meetings, meetings out of the office, or personal meetings, such as doctor visits or vacation days.

Tasks are various internal assignments that TeamConnect users have to complete, usually when working on projects, such as preparing reports, following up on cases, or reviewing documentation. Tasks may also be billable time items that need to be tracked by their respective rates on an hourly basis.

### 1.6.1 Appointments Calendar

Appointments are scheduled events, engagements, or meetings, usually with an assigned location and a specified time. Appointments are made only for TeamConnect users and can be easily associated with specific projects if necessary. Whenever you create an appointment, it is automatically displayed on the calendar screens of the users who are selected as attendees.

There are several different calendar views that you can use to manage your appointments, such as a 30-day calendar view, or a daily calendar view.

#### Appointment-specific Pages

Each appointment has the following pages on which you can view and edit information:

- **Attendees**—Invite or edit the individuals or groups of individuals that you invite to an appointment.
- **Resources**—Select the additional items that you need for the appointment, such as a conference room or a projector.

Each appointments record also has several other pages on which you can view, enter, and edit additional information that applies to an appointment, such as security rights, or the category for a selected appointment.

The following table shows how you can view, create, and edit appointments.

...Appointments	How...
Viewing	<ul style="list-style-type: none"> <li>• <a href="#">Viewing the Appointments Calendar</a></li> </ul>



	<ul style="list-style-type: none"><li>• <a href="#">Opening an Appointment</a></li><li>• <a href="#">Appointments Created Using Outlook</a></li></ul>
<b>Creating</b>	<ul style="list-style-type: none"><li>• <a href="#">Creating or Editing Appointments</a></li><li>• <a href="#">Using the Appointments Scheduler</a></li></ul>
<b>Updating</b>	<ul style="list-style-type: none"><li>• <a href="#">Adding and Removing Appointment Categories</a></li><li>• <a href="#">Adding Appointment Attendees</a></li><li>• <a href="#">Adding Appointment Resources</a></li><li>• <a href="#">Viewing or Creating Appointment History</a></li></ul>

### 1.6.1.1 Viewing the Appointments Calendar

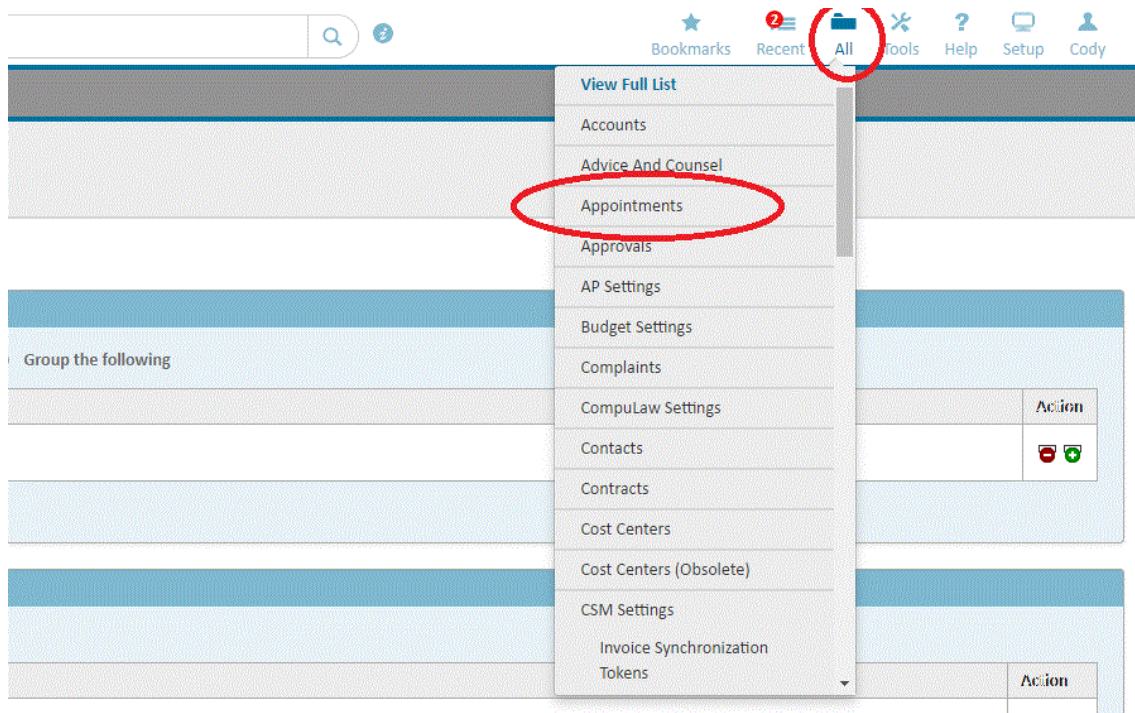
Use the appointments calendar to view the appointments that you scheduled, the appointments to which you are an attendee, and any other appointments that you have the rights to view.

You can also view appointment records in a collection or search for records. See [Viewing Collections of Records](#) and [Searching for Specific Records](#).

You can display appointments in several calendar views, such as the day or work week view. In any view, you can create a new appointment or click on an existing appointment to view or edit its details.

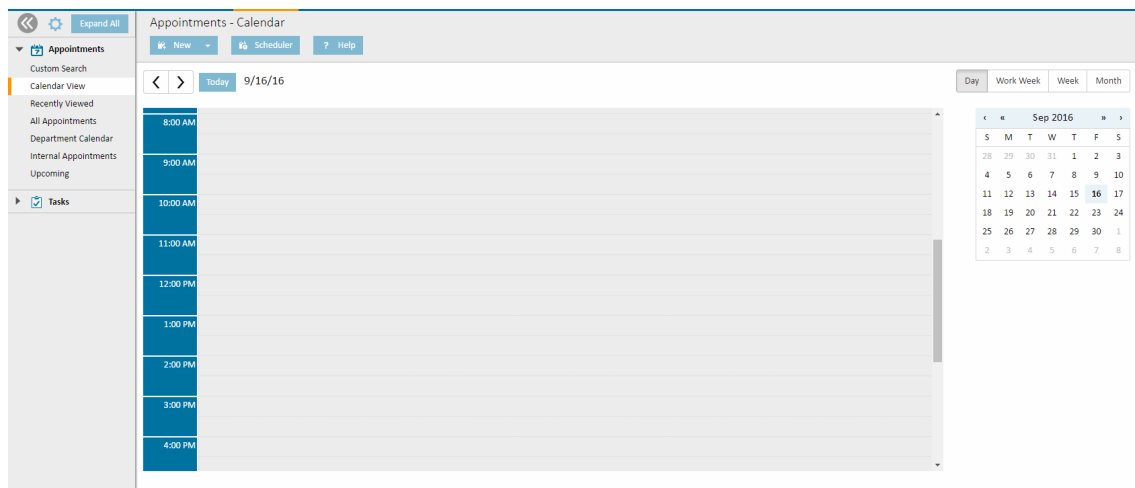
#### To view the appointments calendar

To view the appointments calendar, click the **All** then **Appointments** links located on the home screen.



Access the Appointments Calendar

If this is the first time you opened the calendar, it opens in the default **Day** view. Once you select your preferred view, it becomes the default view.



Calendar Day View

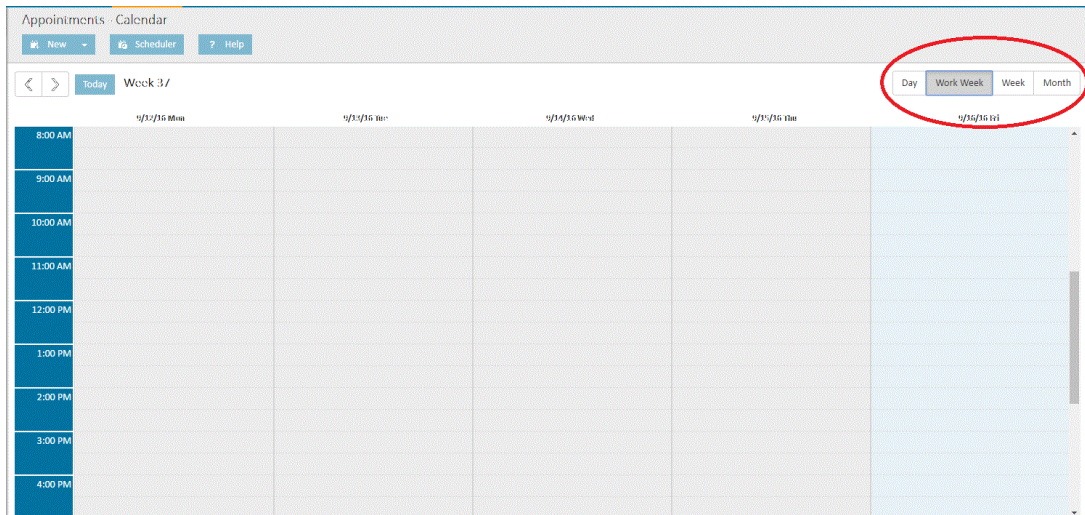
## Changing the Calendar View

You can change the way that you view the calendar and move backwards or forwards to view different days, weeks, or months.

**Note:** If you are currently in **List View**, click **Switch to Calendar View** to access the additional views.

### To change the calendar view

- Click **Work Week** to view your calendar and appointments in a Monday through Friday format, excluding weekends.

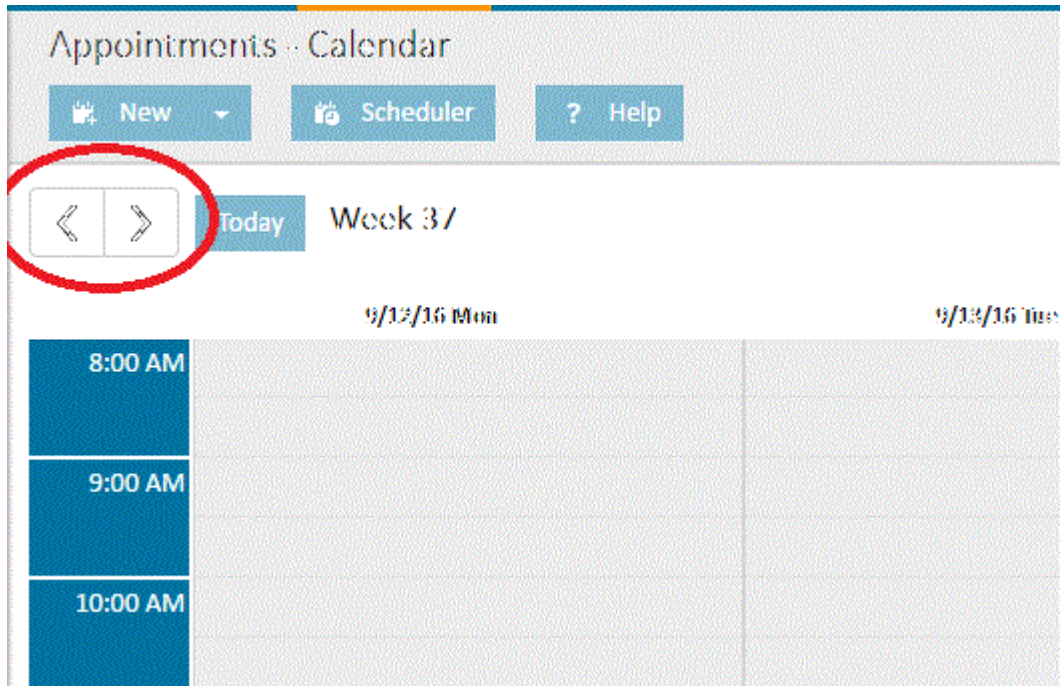


Calendar Work Week View

- Click **Week** to view your calendar and appointments in a 7-day format.
- Click **Month** to view your calendar and appointments in a format that allows you to view all of the days in a month.
- Click **Switch to List View** in any view to return to the default list view.

### Calendar Navigation

For information about the areas in the list view, see [User Interface](#).



Navigation Buttons in the Calendar View

#### 1.6.1.1.1 Opening an Appointment

You can open appointments that you created, all public appointments, and appointments that you have the rights to view.

##### To open an appointment

1. Click the **All** folder.
2. Click the **Appointments** link in the dropdown.
3. If necessary, switch to your preferred calendar view.
4. Locate the appointment that you want to open, and then click its link.

The appointment opens to its **General** page.

Once you open an appointment, you can edit or delete it, add or remove attendees, and make changes to the additional appointment pages. See [Appointments Calendar](#) for more information.

#### 1.6.1.1.2 Appointments Created Using Outlook

If you open an appointment whose **Attendees** page is read-only, this appointment was created using Outlook. See [Adding Appointment Attendees](#) for more information.



### 1.6.1.2 Creating or Editing Appointments

Creating an appointment may include scheduling internal and external meetings, project meetings, or personal appointments, such as a doctor's visit.

When you create an appointment, you set the date and time, and then, if necessary, add resources, such as a meeting room or a projector. Creating an appointment also blocks the time on your calendar so that others can see when you are not available.

You can also use the appointments pages to edit appointments, add categories to an appointment, or attach documents that you want users to review before a meeting.

The following topics apply to creating or editing appointments:

- [Creating or Editing an Appointment Manually](#): If you use this option, you must wait for individual attendees to accept or decline the meeting invitation. You can also use this option to block time for personal appointments, such as a doctor's visit or your vacation so that users can easily see when you are not available.
- [Using the Appointments Scheduler](#): Use the scheduler to check the calendar of each attendee to find an appointment date and time that fits all of the attendees schedules.

You can also schedule appointments that relate to a project. See [Adding Appointments to Projects](#).

### Using a Wizard

Your system administrator may provide one or more wizards to help you create new appointments using a guided, step-by-step process. A wizard may be one of the available options or the only option. If you click **New** and a wizard opens, complete the required fields on each page, and then click **Next** to progress to the end of the wizard. When you click **Finish** at the end of the wizard, the record is available in the list view.

**Note:** *Your company may only allow you to create an account using a wizard, so the option to create an account manually may not be available.*

#### 1.6.1.2.1 Creating or Editing an Appointment Manually

Use the **Appointments Calendar General** page to create or edit an appointment and add attendees and resources.

#### To create or edit an appointment manually

1. Click the **All** button.
2. Click the **Appointments** link under the **All** button.
3. [Switch to your preferred calendar view](#), if necessary.
4. Select one of the following options:
  - To create a new appointment, click **New**.
  - To edit an existing appointment, click the appropriate appointment in your preferred calendar view.

The appointment **General** page opens.

5. On the **General** page, enter or edit the general appointment information. See [the Appointments General page fields table](#) for more information.

To enter date and time information for this appointment, use the **Start Time** and **End Time** fields.

6. If necessary, complete the following pages:
  - Click the **Attendees** link to select the individual users, address books, or user groups that you want to attend the appointment. See [Adding Appointment Attendees](#) for more information.
  - Click the **Resources** link to select a location or reserve equipment for the appointment. See [Adding Appointment Resources](#) for more information.
7. If necessary, use the additional appointment pages to enter or edit supplementary information, such as categories or historical information. See [Appointments Calendar](#) for more information.
8. [Select a save option.](#)

An email notification is automatically sent to the attendees if their TeamConnect Preferences are set to receive email notifications. See [Setting Up Email Notifications](#) for more information.

The table below contains the default fields for the Appointments **General** page. The fields that you see may vary based on your system settings and assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

**Appointments General Page Field Descriptions**

Field (* = required)	Description
<b>All Day Event</b>	Indicates whether or not this appointment lasts a full workday.  If you choose to create an All Day Event, you cannot specify a start or an end time. (If necessary, you can specify such information in the <b>Notes</b> field.)
<b>Location</b>	The location where the appointment is to take place, such as conference room.  This field supports up to 250 characters.
<b>Notes</b>	Additional comments that apply to this appointment.
<b>Project</b>	The project that you want to associate with the appointment, if any.  Click the search icon to locate the appropriate project.
<b>Start Time</b>	The time frame of the entire event.

<b>End Time</b>	<ul style="list-style-type: none"> <li>• Click the calendar icons next to the date to open a calendar to select start and end dates.</li> <li>• Click the time drop-down list to select the beginning and ending time for the appointment.</li> </ul> <p>The system reads the current time from the application server. It automatically rounds time to the nearest hour. For example, at 9:02 a.m., it displays 10:00 a.m. as the start time for the event. By default, all appointments are set to last one hour.</p> <p>If you go back and change these times, and attendees or resources have been scheduled for only part of the appointment, you may need to adjust their attendance times.</p> <p>This field supports the following formats:</p> <ul style="list-style-type: none"> <li>• MM/DD/YYYY</li> <li>• MM/DD/YY</li> <li>• HH:MM</li> <li>• A.M. or P.M.</li> </ul>
<b>*Subject</b>	<p>The name or reason for the appointment.</p> <p>This field supports up to 250 characters.</p>

#### 1.6.1.2.2 Using the Appointments Scheduler

The Scheduler is a wizard that lets you search for an available date and time on anyone's calendar, check the availability of the necessary resources, create the appointment, and then add the appointment to the attendees' calendars.

#### To use the appointments scheduler

1. On your home page, click the **All** button. Click **Appointments** from the resulting dropdown.
2. You can create a new appointment from the main calendar view by clicking the **Scheduler button**.

The first page of the scheduler wizard opens.

3. On the first screen complete the fields in the [Appointment Information Field Descriptions table](#).

First Page of Scheduler

4. Click **Next**

The **Attendees and Resources** page opens.

Scheduler Wizard Page Two

5. In the **Attendees** field, select from the list of attendees that automatically appears on your page. The selected user or contact appears in the attendees list. To invite additional users, click **Add New Item** and continue adding users until your list is complete.
6. In the **Resource** drop-down list, select the resources that you may need, such as a conference room or a projector. To select additional resources, click **Add Another Item**.
7. Click **Next**.

Based on the criteria that you entered, the system searches the calendar of each attendee to find a date and time when everyone is available to meet, and then returns one or more results.



	From	To
<input type="radio"/>	9/20/16 9:00 AM	9/20/16 10:00 AM
<input type="radio"/>	9/20/16 9:30 AM	9/20/16 10:30 AM
<input type="radio"/>	9/20/16 10:00 AM	9/20/16 11:00 AM
<input type="radio"/>	9/20/16 10:30 AM	9/20/16 11:30 AM
<input type="radio"/>	9/20/16 11:00 AM	9/20/16 12:00 PM

Scheduler Wizard Page Three

8. On the **Select Available Time Slot** page, select the option that you prefer.

If you want to open the appointment to its **General** page after the appointment is saved, select the **Open the scheduled appointment** check-box.

9. Click **Finish**.

An email notification is automatically sent to the attendees if their preferences are set to receive email notification. See [Setting Up Email Notifications](#) for more information.

In response, attendees can use the appointment calendar to accept or decline the invitation. If an attendee accepts this invitation, the item is automatically added to their calendar.

#### Appointment Information Field Descriptions

Field	Description
<b>Subject</b>	Enter the name of the appointment or meeting.
<b>Duration</b>	Enter how long the appointment or meeting will last.
<b>Days(s)</b>	Click the boxes for the days that you want to check on everyone's calendar.
<b>Week(s)</b>	Enter the span of weeks that you want to check on everyone's calendar.
<b>Time</b>	Enter the time span that you want to check on everyone's calendar.

#### 1.6.1.3 Adding and Removing Appointment Categories

Use the Appointment **Categories** page to add a category to an appointment. Appointment categories such as Conference, Deposition, or Trial help attendees understand the nature of the appointment they are attending and allow them to search for appointments by category.

Categories may have one or more sub-categories and are listed in a hierarchical view with the main category at the top and all of its sub-categories listed below it.

Based on your access rights, you can view, add, change, or delete categories.

See [Setting Category Details](#) for more information.

#### 1.6.1.4 Adding Appointment Attendees

Use the Appointment **Attendees** page to invite the appropriate people or groups of people to an appointment. With the appropriate rights, you can add the following types of attendees to appointments:

- Individual contacts who are also TeamConnect users
- Address Books that contain the people that you want to attend the appointment

For address books, only members who are also TeamConnect users are added as appointment attendees. Members who are not TeamConnect users are not added or notified.

- User Groups that contain the people that you want to attend the appointment

The user who creates the appointment is automatically added as an attendee.

##### To add appointment attendees

1. Click the **All** button, and then click the **Appointments** drop-down link located on the left side of the screen.
2. Click the **All Appointments** link in the left pane.
3. Locate the appropriate appointment and open it.
4. In the left pane, click the **Attendees** link.

The **Attendees** page opens.

You are automatically added as the appointment creator. If you are scheduling this appointment for others and you do not want to attend, delete yourself from the attendee list.

5. Click **Edit**.
6. Click **Add New Item**.
7. In the **User** drop-down list, select the type of attendee that you want to add. Options are **Address Book**, **Group**, and **User**.
8. In the **Attendee** drop-down list, select the appropriate attendee. Your choices are based on the option you selected in the **Type** drop-down list.
9. Repeat steps 4 and 5 until all of the appropriate users are added.

***Tip:** By default, the option in the **Attendance** drop-down list is set to **Will Attend** for all attendees. You do not have to change it.*

10. [Select a save option.](#)

The appointment automatically appears on the calendars of all attendees.

### To remove appointment attendees

To delete attendees, select the check-box for the attendees that you want to remove, and then click **Remove**.

#### 1.6.1.5 Adding Appointment Resources

Use the Appointment **Resources** page to add a location for the appointment, or equipment that you may need, such as a projector.

### To add appointments resources

1. Click the **All** button.
2. Click the **Appointments** link under the tab bar.
3. Locate the appropriate appointment and open it.
4. In the left pane, click the **Resources** link.
5. Click **Edit**.
6. In the **Resource** drop-down list, select the appropriate item or location.
7. If you want to add additional items, click **Add New Item**, and then select the additional items or locations.
8. [Select a save option.](#)

### To delete resources

To delete resources, select the check-box for the resource that you want to remove, and then click **Remove**.

#### 1.6.1.6 Viewing or Creating Appointment History

Use the Appointment **History** page to view and write historical details for an appointment, such as the points discussed in a meeting, or the details of a previous meeting.

See [Working with Record History](#) for more information.

## 1.6.2 Tasks

Tasks in TeamConnect are internal assignments that TeamConnect users have to complete, usually when working on projects, such as preparing reports, following up on cases, or reviewing documentation. A group of tasks is a list of items, often in order of importance.

Tasks can be one of the following:

- **Non-billable**—Tasks that a user must complete that have no associated money costs, such as preparing a report.
- **Billable**—Tasks that are tracked by their rates on an hourly basis. All billable tasks can be posted against certain accounts according to the posting criteria that is set for each individual account. See [Posting Tasks](#) for more information.

Each task record lets you enter all the information related to a given task on the same record page.

## Types of Tasks

There are two types of tasks:

- **Project-specific**—Tasks that must be performed to move a project forward to completion.
- **Global**—Not associated with any project, such as preparing payroll reports.

## Task-specific Pages

Each task record has the following task-specific pages:

- **General**—Create and edit tasks information, including the subject, due date, and task assignee.
- **Transactions**—Displays a history of the task's posting and voiding statuses and allows you to access the related account record.

Each task record also has additional pages on which you can view and edit specific information, such as a **Documents** page where you can store documents that are related to a task. See [Viewing Documents](#) for more information.

The following table shows how you can view, create, and edit tasks.

...Tasks	How...
Viewing	<ul style="list-style-type: none"> <li>• <a href="#">Viewing Tasks</a></li> <li>• <a href="#">Viewing Task Transactions</a></li> </ul>
Creating	<ul style="list-style-type: none"> <li>• <a href="#">Creating or Editing Tasks</a></li> <li>• <a href="#">Creating a Task Manually</a></li> <li>• <a href="#">Creating Multiple Tasks</a></li> </ul>
Updating	<ul style="list-style-type: none"> <li>• <a href="#">Assigning or Reassigning Tasks to Users</a></li> <li>• <a href="#">Posting Tasks</a></li> <li>• <a href="#">Voiding Tasks</a></li> <li>• <a href="#">Viewing or Creating Task History</a></li> </ul>

	<ul style="list-style-type: none"><li>• <a href="#">Using the Task Entry Tool</a></li><li>• <a href="#">Editing Multiple Tasks</a></li><li>• <a href="#">Posting Multiple Tasks</a></li><li>• <a href="#">Deleting Multiple Tasks</a></li></ul>
--	---

### 1.6.2.1 Using the Task General Page

On the **General** page, you can perform the following actions:

- [Copying a Record](#)
- [Assigning or Reassigning Tasks to Users](#)
- [Posting Tasks](#)
- [Voiding Tasks](#)

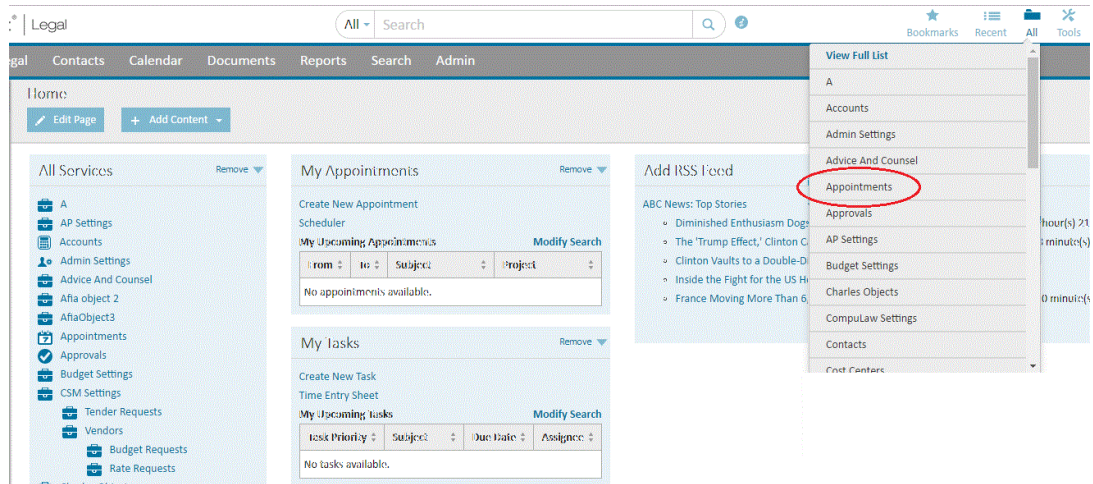
### 1.6.2.2 Viewing Tasks

You can view tasks in a list view.

You can also view tasks in a collection or search for records. See [Viewing Collections of Records](#) and [Searching for Specific Records](#).

#### To view a task

1. Click the **All > Appointments** tab.
2. Click the **Tasks** link under the tab bar. The **Tasks** list page opens.  
If necessary, click a task collection link in the left pane, such as **Recently Viewed** or **All Tasks**.
3. To open a task, click its link in the **Subject** column.



**Tasks List Page**

For information about the areas in the contact views, see [User Interface](#).

### 1.6.2.3 Creating or Editing Tasks

Use the **General** page to create and edit tasks.

A task record can serve as a simple reminder to work on an assignment, or it can be a billable time item, which you will eventually post against a related account.

When you create a task, you can track the time for a project or vendor, and then post the task.

The following topics apply to creating or editing tasks:

- [Creating a Task Manually](#)—Create a task starting from a blank Task record.
- [Copying a Record](#)—Use an existing Task record to create a new task when many of the settings are the same.

### Using a Task Wizard

Your system administrator may provide one or more wizards to help you create new tasks using a guided, step-by-step process. A wizard may be one of the available options or the only option.

If you click **New** and a wizard opens, complete the required fields on each page, and then click **Next** to progress to the end of the wizard. When you click **Finish** at the end of the wizard, the new record is saved.

**Note:** Your company may only allow you to create an account using a wizard, so the option to create an account manually may not be available.

### Other Areas Where You Can Create Tasks

There are several areas in TeamConnect where you can create tasks:

- The **Tasks List** page—Based on your rights, you can create or edit a task. See [Viewing Tasks](#) for more information.

- The **Tasks** page for a project--Open a project and use its **Tasks** page to create or edit a task. See [Viewing Tasks](#) for more information.
- The Task Entry Tool--Create and edit multiple tasks in one operation. See [Using the Task Entry Tool](#) for more information.

#### 1.6.2.3.1 Creating a Task Manually

Use the Tasks **General** page to create a task starting from a blank record.

#### To create or edit a task manually

1. Click the **All** button.
2. Click the **Tasks** link in the dropdown menu.  
The **Tasks** List page opens.
3. Select one of the following options:
  - To create a task, click **New**.
  - To edit a task, click the pencil icon to the left of a task name.

The task record opens to its **General** page.

#### Tasks General Page

4. On the **General** page, enter or edit the appropriate information in the available sections, including billing information if appropriate. See the [the Task General page fields table](#) for more information.

The following fields are required:

- Subject

- Assignee

Based on your rights, this page may also contain custom fields that appear based on the option that you select in the **Category** drop-down list. Complete these fields according on your company policies.

5. [Select a save option.](#)

The tables below contains the default fields for the Tasks General page. The fields that you see may vary based on your system settings and assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

#### Task Information

Field Name (* = Required)	Description
<b>Activity</b>	The appropriate activity type for the task.  <i>Note: Typically, this field is used as a subcategory for the fee.</i>
<b>*Assignee</b>	The user assigned to the task. If you are creating a task, this field displays your name. For an explanation of how to assign a task, see <a href="#">Assigning or Reassigning Tasks to Users</a> .
<b>Category</b>	A project category for the task, such as follow up, review file, or status report. If the current task has no project (global task), use the system category <b>Task</b> . Inactive categories, if any, are displayed but cannot be selected.  The category that you select may allow custom fields to appear on the <b>General</b> page. Complete these fields based on your company policies.  <i>Important: Each task may have only one category.</i>
<b>Completed Date</b>	The date when the task is completed. Select it by clicking the <b>Calendar</b> icon next to the field.  If you enter the current date as the <b>Completed Date</b> then save, the <b>Status</b> changes to <b>Completed</b> and the <b>Percentage Completed</b> to 100.
<b>Contact</b>	Search for and select the appropriate contact.  You may use this field, for example, to let the task track the time spent working on assignments by a specific contact.



	<p><b>Note:</b> This field serves as a vendor posting criterion, if you decide to post the task. For more information, see <a href="#">Posting Tasks</a>.</p>
<b>Due Date</b>	<p>The date when the task is due. Select it by clicking the <b>Calendar</b> icon next to the field.</p> <p><b>Note:</b> Certain tasks may have no specific due date. These tasks are known as floating tasks.</p>
<b>Estimated Hours</b>	<p>The approximate number of hours needed to complete the task.</p>
<b>Hours</b>	<p>The number of hours required to complete the task, such as number of hours or number of items. This applies to billable hours only.</p>
<b>Percentage Completed</b>	<p>The approximate percentage of the task already completed.</p> <p>When you enter 100 then save, the <b>Completed Date</b> field automatically displays the current date and the <b>Status</b> field changes to <b>Completed</b>.</p>
<b>Priority</b>	<p>One of the following priority levels for the task: <b>Highest, High, Normal, Low, Lowest</b>. The default is <b>Normal</b>.</p>
<b>Project or Project name</b>	<p>Displays one of the following types of information:</p> <ul style="list-style-type: none"> <li>If you are working within a project, this field is read-only and displays the name of that project, for example, <b>Claim</b> or <b>Program</b>.</li> <li>If you access the current task using <b>Calendar &gt; Tasks</b>, an optional field <b>Project</b> allows you to choose a project to associate with the task.</li> </ul> <p>Select a project for the task if there is no current project.</p>
<b>Start Date</b>	<p>The date when the assignee is supposed to start working on the task. Select it by clicking the <b>Calendar</b> icon next to the field.</p>
<b>Status</b>	<p>One of the following descriptions to indicate the status of the task: <b>Started, Not Started, Completed</b>.</p> <p><b>Note:</b> When you choose <b>Completed</b> then save, the <b>Percentage Completed</b> field automatically changes to 100, and the <b>Completed Date</b> displays the current date.</p>
<b>*Subject</b>	<p>A brief description of the task.</p>

#### Billing Information

Field Name	Description
<b>Hours</b>	The amount of time in hours billed to the task.
<b>Posting Status</b>	<p>Displays task's posting status, which may be:</p> <ul style="list-style-type: none"> <li>• <b>Not Posted</b>—The task has not been posted, or it has been voided.</li> <li>• <b>Posted</b>—The task is posted to an account.</li> <li>• <b>Failed</b>—An unsuccessful attempt was made to post the task.</li> <li>• <b>Posted For Approval</b>—The task has been submitted for approval to be posted.</li> </ul> <p>The text in this field is read-only.</p>
<b>Rate</b>	<p>The amount charged per task unit.</p> <p><i>Note: If the task has a contact, and that contact has a specified rate entered, TeamConnect automatically displays that rate here. For more information on contacts and rates, see <a href="#">Setting Contact Rates</a>.</i></p>
<b>Total Amount</b>	<p>The total cost of the task after you save, calculated by the system using the rate and units shown on this page.</p> <p>The text in this field is read-only.</p>

**Access Information**

Field Name	Description
<b>Created By</b>	The name of the user who created the task.
<b>Created On</b>	The date the task was added to TeamConnect.
<b>Modified By</b>	The name of the user who last modified the task.
<b>Modified On</b>	The date when the task was last modified.

The **Notes** section allows you to enter comments about the current task. For information about the **Assignment History** section, see [Assigning or Reassigning Tasks to Users](#).

### 1.6.2.4 Assigning or Reassigning Tasks to Users


Use the **General** page to assign an individual user to a task. Each task must have an assignee who is responsible for completing the task. If you create a task, you are the assignee by default, but you may reassign the task to a different user if necessary.

#### To assign or reassign a task

1. Click the **Calendar** tab, and then click the **Tasks** link under the tab bar.

The **Tasks List** page opens.

2. Select one of the following options:

- To assign a task, click **New**.
- To reassign a task, click the edit icon .

The **General** page of the task opens allowing you to enter information. For details, see the [Task General Page Field Descriptions](#).

3. Use the drop-down list in the **Assignee** field to choose the user you want to assign to this task.
4. [Select a save option](#).

**Note:** If the appropriate preferences are set, TeamConnect automatically sends an e-mail notification to a new assignee.

The following table explains the fields in this section of the **General** page. The text in this section is read-only.

#### Assignee History

Field or control	Description
<b>Assignee</b>	Displays a list of names of the task's assignees in the order in which they were added, with the first added at the top of the list.  <b>Note:</b> You cannot click on the <b>Assignees</b> heading to sort assignees alphabetically.
<b>Assigned On Date</b>	Displays the date when an assignee was assigned to the task.

#### Points to Remember

The following list summarizes key information about task assignments:

- By default, the name of a user creating a task displays in the **Assignee** field.

- You may only assign users to a task.
- A task can only have one active assignee at a time, the current assignee.
- The **Assignee History** section of the **General** page is read-only and displays all previous assignments.
- Make reassignments as follows:
  - *Individually*, by choosing a new user name in the **Assignee** field.
  - *In batch*, using the **Reassignment** tool accessible from the **Tools** drop-down list in the **All** tab.

#### 1.6.2.5 Posting Tasks

Use the toolbar and the **General** page to post tasks against accounts. When a task posts successfully, TeamConnect debits its expense total from any linked account's current balance.

Before posting a task, you must make sure the following conditions are met:

- The task must be billable.
- The task's status is **Completed**. For more information, see [Task General Page Field Descriptions](#).
- One or more accounts are linked to the task by way of the task's category.
- Any linked account is active and allows posting.
- Any linked account can accept posting by this task, for example:
  - There must be adequate account funds.
  - The task's completion date must be within any linked account's time period.
  - You must have rights to any linked account.

You may also create and save a billable task, and then post it later. TeamConnect stores the record until you are ready to post or delete it. You can also use the General page to edit a posted task.

#### To post a task

1. Open the **General** page of the task.

If the task's posting status is **Not Posted**, **Failed**, or **Posted For Approval**, the **Post** button appears in the toolbar.

2. Click **Post**.

The total for the task is debited from any linked account. After posting a task, you may view the corresponding account transaction entries. To do so, go to the **Transactions** pages of the task and any linked account. For more information, see [Viewing Task Transactions](#).

**Note:** *After posting, the task's system fields become read-only. Depending on system settings and your rights, one or more custom fields may be editable. If the task is voided, you*

*may edit the task's system fields, if necessary. For more information, see your administrator.*

### To post multiple tasks

TeamConnect allows you to post many tasks in one operation. See [Using the Task Entry Tool](#) for more information.

### To edit a posted task

1. Void the task using its **General** page. For details, see [Voiding Tasks](#).
2. Edit the task, as necessary. For details, see [Creating or Editing Tasks](#).
3. Post the task again, if necessary.

#### 1.6.2.6 Voiding Tasks

Open the **General** page of a task to void it.

Voiding a task lets you cancel a posting for any reason, such as if you received a refund and need to back out the original posting, or found a mistake in the dollar amounts. Voiding the original account credits the account by the expensed amount.

#### To void a task

1. Open the **General** page of the task that you want to void.  
If the task's posting status is **Posted**, the **Void** button appears in the toolbar.
2. Click **Void**.

The total for the task is credited to any linked account.

After voiding the task, you may view the corresponding account transaction entries on the **Transactions** page for this task. See [Viewing Task Transactions](#) for more information.

If necessary, you may [edit the task's fields](#), and then post the task again.

#### 1.6.2.7 Viewing Task Transactions

Use the **Transactions** page to view a history of posting transactions for a given task. Posted tasks appear in a convenient list that includes information, for example, the date the task was posted. For another example, if you need information on a task and want to know who posted it, you may use this list to view that user's name.

Transactions you may or may not view depend on the access rights to the related task and any of its linked accounts.

### To view task transactions

1. Display the **General** page of the task.
2. Click **Transactions** in the left pane.  
A list page opens showing the task's previous transactions.
3. If necessary, click an account name link to view information on any linked account.

**Note:** The **Transactions** page is read-only. You may only edit tasks using the appropriate **General** page.

#### 1.6.2.8 Viewing or Creating Task History

Use the Task **History** page to create a history for a task. A history is one or more text notes about changes to the task. You may also use this page to view a list of history records for that task or to do an advanced search on a task's history entries.

You always have the rights to a task history you created. However, you may regulate rights that other users have to this history.

For more information, see [Working with Record History](#).

#### 1.6.2.9 Using the Task Entry Tool

You may perform the following operations using the Task Entry Tool:

- [Creating Multiple Tasks](#)
- [Editing Multiple Tasks](#)
- [Posting Multiple Tasks](#)
- [Deleting Multiple Tasks](#)

##### 1.6.2.9.1 Creating Multiple Tasks

The Task Entry Tool provides a convenient way to create multiple tasks quickly in one TeamConnect page. The **Task Entry Tool** can be accessed through the **Tools** button.

**Caution:** When using the **Task Entry Tool** page, if you navigate away from the page, you lose all work done since your last **Add** or **Save** action. If you go back to the **Task Entry Tool** page after navigating away, you must create a new set of tasks and a new work session to continue working in the page. For example, if you want to edit a copy of a task you created in this page, the best practice is to finish all the work you want to do during your current page session before you navigate away to edit the copy.

When you finish a session and save your work, you may then edit the task copies you made during the session. You may access tasks previously created in this page using the **Task** list page or **General** page.

### To create multiple tasks

1. On the toolbar of the Task list or on the General page, click **Task Entry Tool**.  
The **Task Entry Tool** page opens.
2. For each task, enter information in the **Task** pane. See [the Task Entry Tool field table](#) for field descriptions.
3. Click **Clear** to clear all field entries, if necessary.
4. Click **Add** to create and save each task.  
The task appears in the List pane below.  
For more information, see [the Task General page field table](#).
5. Click the **Copy** button to create and save additional copies of the current task.  
Each copied task appears in the list pane.

The table below contains the default fields for the Task Entry Tool. The fields that you see may vary based on your system settings and assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

#### Task Entry Tool Field Definitions

Field Name (* = Required)	Description
<b>Activity</b>	The activity type for the task, if applicable.  <i>Note: Typically, this field is used as a subcategory for the fee, if applicable.</i>
<b>*Billed By</b>	The name of the user (same as <b>Assignee</b> ) assigned to the task. If the task is billable, this is the contact billing the task.
<b>Category</b>	A project category for the task, if applicable, such as follow up, review file, or status report. If the current task has no project (global task), use the system category <b>Task</b> .  <i>Important: If assigned, each task may have only one category.</i>
<b>Completed Date</b>	The date when the task is completed. Select it by clicking the <b>Calendar</b> icon next to the field.
<b>Contact</b>	Search for and select the appropriate contact.

	<p>You may use this field, for example, to let the task track the time spent working on assignments by a specific contact.</p> <p><b>Note:</b> <i>This field serves as a vendor posting criterion, if you decide to post the task, and it is billable. For more information, see <a href="#">Posting Tasks</a>.</i></p>
<p><b>Project or Project name</b></p>	<p>Displays one of the following types of information:</p> <ul style="list-style-type: none"> <li>• If you are working within a project, this field is read-only and displays the name of that project, for example, <b>Claim</b> or <b>Program</b>.</li> <li>• If you access the current task using <b>Calendar &gt; Tasks</b>, an optional field <b>Project</b> allows you to choose a project to associate with the task.</li> </ul> <p>If you want, select a project for the task if there is no current project.</p>
<p><b>Rate</b></p>	<p>The amount charged per task unit, only for a billable task.</p> <p><b>Note:</b> <i>If the task has a billable contact or <b>Billed By</b> assignee, and that person has a specified rate entered, TeamConnect automatically displays that rate here. For more information on contacts and rates, see <a href="#">Setting Contact Rates</a>.</i></p>
<p><b>*Subject</b></p>	<p>A brief description of the task used as a task name.</p>
<p><b>Units</b></p>	<p>The number of units (for example, hours or items) required to complete the task. Only applies to billable tasks.</p>


#### 1.6.2.9.2 Editing Multiple Tasks

The Task Entry Tool provides a convenient way to edit multiple tasks quickly in one TeamConnect page.

#### To edit multiple tasks

1. Click **Task Entry Tool** in the toolbar of the **Task** List or **General** page.

The **Task Entry Tool** page opens.


2. Click  on the right, next to the appropriate task in the List pane.

The field entries for the task appear in the **Task** pane.

3. Edit and enter information in the **Task** pane according to descriptions listed under [Task Entry Tool Field Definitions](#).



When you are editing a task, the **Add** and **Clear** buttons change to **Ok** and **Cancel**.

4. Click **Cancel** to discard all your edits, if necessary.
5. Click **Ok** [to save your edits in the task.](#)
6. Click  to create and save additional copies of the current task, if necessary.

Each copied task appears in the List pane.

#### 1.6.2.9.3 Posting Multiple Tasks

The Task Entry Tool provides a convenient way to post multiple tasks quickly in one TeamConnect page.

##### To post tasks

1. Click **Task Entry Tool** in the toolbar of the **Task** List or **General** page.  
The **Task Entry Tool** page opens.
2. Select the check-boxes in the List pane on the left, next to the tasks you want to post, or to post all tasks, click the check-box at the top of the list.
3. Click **Post**.

Any corresponding account transactions display on the **Transactions** pages of the tasks and any associated account.

**Note:** *You may not edit or delete a posted task. If you want to void a task, you must view the task using the **General** page then void the task from there.*

#### 1.6.2.9.4 Deleting Multiple Tasks

The Task Entry Tool provides a convenient way to delete multiple tasks quickly in one TeamConnect page.

##### To delete tasks

1. Click **Task Entry Tool** in the toolbar of the **Task** List or **General** page.  
The **Task Entry Tool** page opens.
2. Select the check-boxes in the List pane on the left, next to the tasks you want to delete, or to delete all tasks, click the check-box at the top of the list.
3. Click **Remove**.

## 1.7 Finance

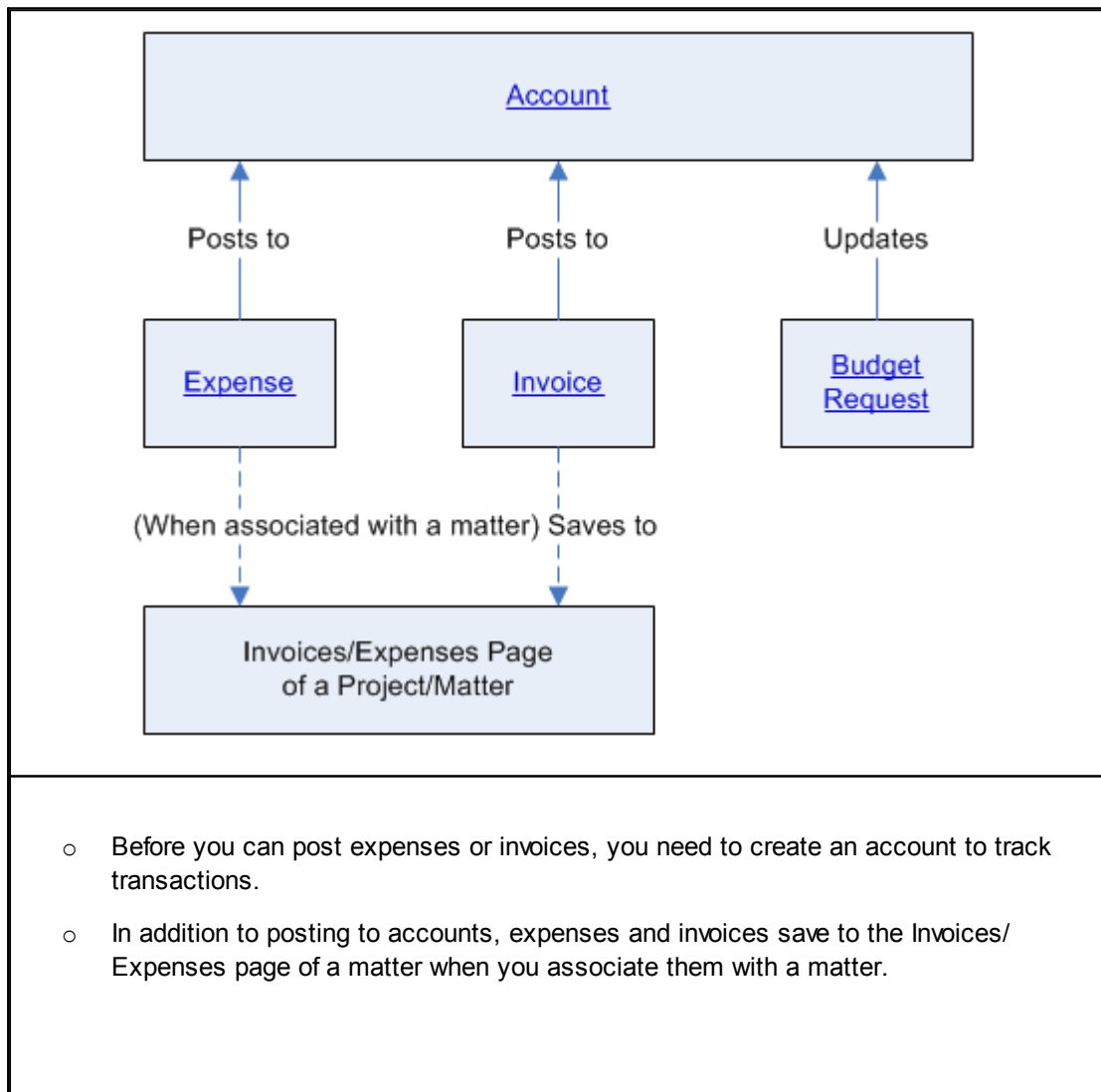
TeamConnect can track various financial components of a project, such as attorney time and vendor bills (Invoices) and internal business costs (Expenses). You must set the appropriate budgets and reserves (Accounts) to use invoices and expenses.

Accounts play a central role in the TeamConnect finance area. Each account can track transactions according to the desired posting criteria, which can be a combination of projects, vendors, involved contacts, and types of transactions.

If you have e-billing through Collaborati Spend Management (CSM), you can also use budget requests. Through Collaborati, vendors approve budgets for a matter.

**Note:** The finance area in TeamConnect can track financial transactions electronically but is not tied to real monetary accounts.

The following diagram illustrates how all the financial parts of TeamConnect work together. Click areas of this flowchart for more information.



- If you use CSM, you can submit a budget request for each combination of a vendor and a matter. After you submit a budget request and the vendor approves it, the accounts listed on the budget update the Total Allocated amount for the account.

**Note:** The pages where financial information save depend on your system configuration.

## Tasks

Tasks are a type of transaction that can also post to designated accounts. For more information, see [Tasks](#).

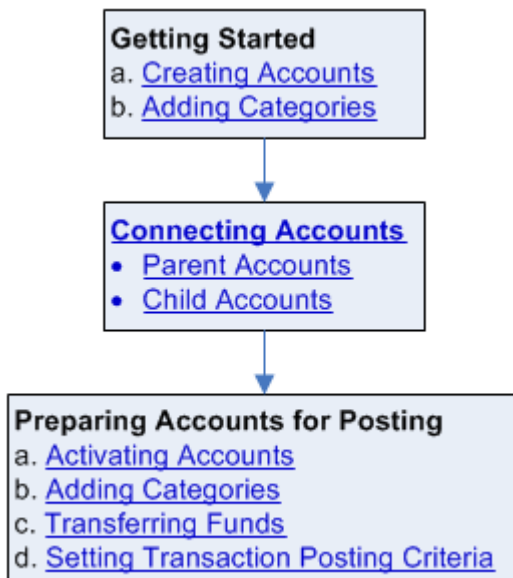
### 1.7.1 Accounts

*Accounts* in TeamConnect let you electronically track posted transactions, such as expenses, tasks, and invoices. By default, TeamConnect accounts are not tied to actual monetary accounts or billing systems and are provided for record-keeping purposes. There are two types of accounts:

- **Budget**—An amount of money set aside by the organization or its individual departments to do business for a specific period of time, such as the fiscal year.
- **Reserve**—Money allocated from the budget for the settlement of matters, such as paying outside counsel fees and other costs.

All transactions (tasks, expenses, invoices) can be posted to their designated accounts. Each account can be set to track transactions according to the desired posting criteria, such as a combination of projects, vendors, involved contacts, or types of transactions. If an invoice posts to an account, the **Transactions** page of both the account and invoice displays the monetary amounts that are debited or credited to the account as a result.

The following flowchart illustrates how you can use accounts. Click areas of this flowchart for more information.



## Account Record Pages

There are four account-specific pages:

- The **General** page lets you enter information about the account. In addition, you can deposit funds, withdraw funds, or transfer funds.
- The **Transactions** page displays a history of the account's posting and voiding status.
- The **Posting Criteria** page lets you set criteria for transactions that can post to the account, such as expenses, tasks, invoice expenses, or invoice tasks.
- The **Child Accounts** page displays an account's related child accounts.

### 1.7.1.1 Viewing Accounts

You can view accounts by browsing Account collections or using the search feature to locate an account.

See [Viewing Collections of Records](#) and [Searching for Specific Records](#).

#### To view accounts

1. Click the **All** button.
2. Click the **Accounts** link in the left tab bar.

The recently viewed accounts are listed in the right pane. To open an account record, click an account link in the **Name** column.

For information about the areas in the Accounts List page, see [User Interface](#).

### 1.7.1.2 Creating or Editing an Account

You can create an account manually using a blank account record or a wizard that guides you through the process step-by-step. You can also create an account record by using an existing account record as a template.

There are several locations where you can add new Account records:

- From the **Accounts** list by clicking the **New** button.
- From an existing account's **Child Accounts** page.

After creating an account you must activate the account. Afterward, you can deposit funds, transfer funds, and set transaction posting criteria.

#### Using a Wizard

Your system administrator may provide one or more wizards to help you create new accounts. A wizard may be one of the available options or the only option.

If a wizard is your only option, when you click **New** the wizard opens. Complete the required fields on each page, and then click **Next** to progress to the end of the wizard. When you click **Finish** at

the end of the wizard, the record is available in the appropriate list view and collection. See [Viewing Accounts](#) for more information.


**Note:** *If a wizard is your only option, and you want to use an existing account record to create a new account record, a wizard will not open. A blank account record opens and you can complete the fields manually. See [Copying a Record](#) and [Creating or Editing an Account Manually](#) for more information.*

#### 1.7.1.2.1 Creating or Editing an Account Manually

Use the account **General** page to begin manually creating an account record. Once you save the record, you can edit it at any time.

**Note:** *Your company may allow you to create an account using a wizard only, so the option to create an account manually may not be available. See [Using a Wizard](#) for more information.*

#### To create or edit an account manually

1. Click the **Finance** tab.
2. Click the **Accounts** link on the left tab bar.
3. Select one of the following options:
  - To create an account, click New.
  - To edit an account, click the Edit Record icon  to the left of an account.
4. From the **General** page, the following fields in the **Account Information** section are required:
  - **Name—Enter** a name for the account.
  - **Account Period Begins**—Click the **Calendar** icon to select the start date.
  - **Account Period Ends**—Click the **Calendar** icon to select the end date.

Transactions can post to the account only between the **Account Period** start and end dates.
  - **Allocation Limit**—Enter the maximum amount of money that can be allocated (deposited or transferred) to the account. If you do not want an account to have an allocation limit, set the Allocation Limit to zero (\$0.00).
5. (optional) To enter information in fields that have been created for your company:
  - Click the **Categories** link in the left pane.

The **Categories** page opens. Select the category to add (enable).
  - Click the **General** link in the left pane.

A section displays below **Notes**. Enter the appropriate information in the corresponding fields.

6. For additional field descriptions, see [Account General Page Field Descriptions](#).
7. For information about adding categories and details, see [Adding or Removing Categories from an Account](#).
8. For information about setting posting criteria, see [Setting Transaction Posting Criteria](#).
9. For information about adding child accounts, see [Viewing or Creating Child Accounts](#).
10. [Select a save option](#).

After saving an account, click **Activate** to add funds to the account.

## Account General Page Field Descriptions

Some field descriptions that follow may not display based on system settings and your assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

### Account Information Section Field Descriptions

Field (*required)	Description
<b>*Account Period Begins</b>	The start of the date range during which transactions can be posted against the account.  <i>Note: Transactions can only post to the account between the <b>Account Period Begins</b> and <b>Account Period Ends</b> date range.</i>
<b>*Account Period Ends</b>	The end of the date range during which transactions can be posted against the account.  <i>Note: Transactions can only post to the account between the <b>Account Period Begins</b> and <b>Account Period Ends</b> date range.</i>
<b>Account Type</b>	Can be one of the following: <ul style="list-style-type: none"> <li>• <b>Budget</b>—An amount of money set aside by the organization or its individual departments to do business for a specific period of time, such as the fiscal year.</li> <li>• <b>Reserve</b>—Money allocated from the budget for the settlement of matters, such as paying the outside counsel fees and other costs.</li> </ul>
<b>Active</b>	Can be one of the following:

	<ul style="list-style-type: none"> <li>• <b>Yes</b>—Indicates that the account has been activated. An account must be activated before you can add funds to, subtract funds from, or transfer funds between this account and another account.</li> <li>• <b>No</b>—Indicates that the account has not been activated or has been inactivated.</li> </ul>
<b>*Allocation Limit</b>	The maximum amount of money that can be allocated (deposited or transferred) to the account. The amount sets a cap for the <b>Total Allocated</b> . If you do not want an account to have a limit, set the <b>Allocation Limit</b> to zero (\$0.00).
<b>Allow Posting</b>	If checked, expenses, tasks, or invoices can post against the account record according to the criteria specified on the <b>Posting Criteria</b> tab. See <a href="#">Setting Transaction Posting Criteria</a> .
<b>*Name</b>	The account name.
<b>Notes</b>	Comments for the account.
<b>Overdraft Type</b>	<p>One of the following:</p> <ul style="list-style-type: none"> <li>• <b>Do not allow negative account</b>—No transactions will be posted if the account balance will be negative.  If you select this option, make sure to deposit enough funds to the account to post transactions. Keep in mind, you can only deposit funds up to your Allocation Limit.</li> <li>• <b>Allow negative account</b>—Transactions can still be posted even if the account balance is or will be in the negative.</li> <li>• <b>Automatic overdraft protection</b>—Available only when you have a parent account.  The necessary funds will be automatically transferred from the parent account to the child account so that transactions continue to be posted.</li> </ul> <p><b>Note:</b> The action selected in the <b>Overdraft Type</b> takes effect based on the amount in the <b>Balance</b> field.</p>
<b>Parent Account</b>	<p>The parent account.</p> <p><b>Note:</b> Required when the Automatic overdraft protection option is selected for the Overdraft Type. See also <a href="#">Parent and Child Accounts</a>.</p>

<b>Total Accruals</b>	The sum of all of the transaction amounts created when you post an Accrual invoice.
-----------------------	---

#### Account Summary Section Field Descriptions

Field	Description
<b>Add</b>	Click to deposit funds to the account. See <a href="#">Account Add Funds Screen Field Descriptions</a> for field descriptions on the resulting screen.
<b>Balance</b>	The difference between the <b>Total Allocated</b> amount and the <b>Total Used</b> amount:  <b>Balance = Total Allocated - Total Used</b>
<b>Subtract</b>	Click to withdraw funds from the account. See <a href="#">Account Subtract Funds Screen Field Descriptions</a> for field descriptions on the resulting screen.
<b>Total Allocated</b>	The sum of all deposited, transferred funds, and automatic overdrafts from the parent account. The respective activity details are displayed on the <b>Transactions</b> tab.  <i><b>Important:</b> The <b>Total Allocated</b> cannot exceed the <b>Allocation Limit</b>.</i>
<b>Total Used</b>	The total amount of transactions posted, withdrawn, or transferred from the account record.
<b>Transfer From</b>	Click to transfer funds from another account to the current account.  <i><b>Note:</b> You can transfer between parent and child accounts only.</i>
<b>Transfer To</b>	Click to transfer funds from the current account to another account.  <i><b>Note:</b> You can transfer between parent and child accounts only.</i>

#### Account Add Funds Screen Field Descriptions

Field (* = required)	Description
<b>*Amount</b>	The numeric <b>Amount</b> to deposit.



<b>Balance</b>	The difference between the <b>Total Allocated</b> amount and the <b>Total Used</b> amount:  <b>Balance = Total Allocated - Total Used</b>
<b>Description</b>	Notes about the transaction.

**Account Subtract Funds Screen Field Descriptions**

<b>Field</b> (* = required)	<b>Description</b>
<b>*Amount (-)</b>	The numeric <b>Amount</b> to withdraw.
<b>Balance</b>	The difference between the <b>Total Allocated</b> amount and the <b>Total Used</b> amount:  <b>Balance = Total Allocated - Total Used</b>
<b>Description</b>	Notes about the transaction.

**Account Notes Section Field Description**

<b>Field</b>	<b>Description</b>
<b>Notes</b>	Comments for the account.  <i>Note: If you enter more than 250 characters without any spaces, you will need to scroll horizontally on the record to view all sections.</i>

**Account Access Information Section Field Descriptions**

<b>Field</b>	<b>Description</b>
<b>Created By</b>	The name of the user who created the account.
<b>Created On</b>	The date that the account was added to TeamConnect.
<b>Modified By</b>	The name of the user who last modified the account.
<b>Modified On</b>	The date on which the account was modified.

## 1.7.1.2.2 Creating a Shadow Invoice Account

A shadow invoice is an invoice that tracks all fees but does not reflect what you actually pay. In order to post shadow invoices to an account, another account must already exist with the same category and posting criteria as the shadow invoice. Most of the time, this account already exists and you do not need to create one.

To find out if a shadow invoice account already exists for this fiscal period, run a custom search from the **Accounts** page. Under the **Filter Criteria**, select **Account Category** as the **Field**, **Is** as the **Operator**, and **Shadow Invoice** as the **Value**. Include **Account Period Begins** and **Account Period Ends** as the **Results Display** fields to determine the fiscal periods of any existing records.

If there is no shadow invoice account for the appropriate fiscal period, you can create the account manually using the **Accounts** pages.

**To create a shadow invoice account manually**

1. From the **Finance** tab, click **Accounts**.
2. Click the **New** button to create an account record.
3. From the **General** page, enter or select the following values:
  - **Name**: Shadow Invoice
  - **Account Type**: Budget
  - **Account Period Begins/Ends**: [your fiscal period start date] and [your fiscal period end date]
  - **Overdraft Type**: Allow negative account
  - **Allow Posting**: checkbox selected
4. From the **Posting Criteria** page, scroll down to the **Invoice** field and select **Shadow Invoice** as the primary category.
5. Save the account.
6. From the open account click the **Activate** button at the top of the page.
7. From the **Posting Criteria** page, click **Edit**, scroll down to the **Invoices** field, and enter or select the following values:
  - **Post Invoice Tasks to this Account**: checkbox selected
  - **Post Invoice Tasks of Type**: (ANY)
  - **Invoice Task Percent**: 100%
  - **Post Invoice Expenses to this Account**: checkbox selected
  - **Post Invoice Expenses of Type**: (ANY)
  - **Invoice Expense Percent**: 100%
8. Save the account.

**Note:** If you change your preferences from budgeting by fiscal period to budgeting without a fiscal period, a shadow invoice might post to two accounts rather than one. In this case, manually deactivate the original shadow invoice account (the one that budgets by fiscal period).

The screenshot shows the 'Account - Kristen's Account' settings page. The left sidebar contains a navigation menu with options like General, Transactions, Posting Criteria, Categories, Documents, Security, Workflow, Child Accounts, History, Invoices, Expenses, Budget Requests, Time Entry Settings, Cost Centers, and Accounts. The main content area is titled 'Invoice' and contains the following settings:

- Post Invoice Type:  Standard,  Credit Note,  Accrual, OR  Shadow Invoice
- Post Invoice Tasks to this Account
  - Prevent Posting Child Invoice Task Categories
  - Post Invoice Task of Type: (Any)
  - Invoice Task Percent: 100 %
- Post Invoice Expenses to this Account
  - Prevent Posting Child Invoice Expense Categories
  - Post Invoice Expense of Category: (Any)
  - Invoice Expense Percent: 100 %

**Creating A Shadow Invoice**

### 1.7.1.3 Adding or Removing Categories from an Account

Categories organize similar types of accounts or act like filters that display or hide custom fields related to an account.

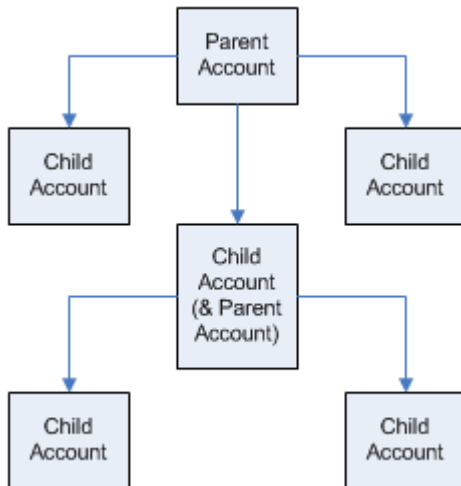
Depending on the type of account you are creating or editing, you might add a category that results in the display of related custom fields. Afterward, you can enter information in those custom fields.

After you add a category on an account's **Categories** page, you can click the **General** link in the left pane and find a **Details** section for that category's custom fields. For example, if you added a category called **Department**, then the **General** page would be updated to include a **Department Details** section with related custom fields.

Similarly, if you remove a category from an account, the related custom field(s) will no longer display. For more information, see [Setting Category Details](#).

#### 1.7.1.4 Parent and Child Accounts

Accounts can be independent or related through a parent-child relationship. Each parent account can have several child accounts, but each child account can have only one parent account. A child account can also be a parent for other accounts. The following flowchart illustrates the parent-child relationship.



Parent accounts are often set to protect their child accounts when there are insufficient funds in the child accounts to cover transactions. You can set up child accounts with **Automatic overdraft protection**, enabling automatic withdrawal of funds from its parent account to avoid a negative balance.

Take into consideration the following when working with parent and child accounts:

- You cannot deposit funds directly to child accounts.
- You can transfer funds from a parent account to a child account, or transfer funds from a child account to a parent account.
- If you deactivate a parent account, make sure to change the **Overdraft Type** of the child accounts' **General** page.
- The parent account must be able to support its child (children), by either having enough funds (**Total Allocated**), high or unlimited **Allocation Limit**, or it must be set up to allow for a negative balance (**Overdraft Type = Allow negative account**).
- From a parent account, you can only create new accounts from the Child Accounts page. To add child accounts that are already existing, open the child account and select the desired parent account.

##### 1.7.1.4.1 Assigning Parent Accounts

On the **General** page, you can assign a parent account to an existing account. For more information, see [Parent and Child Accounts](#).

#### To assign a parent account to an existing account

1. Search for an account that will be the child account and open it.
2. On the **General** page, **Parent Account** field, click the **Search Accounts** icon. Search and select a parent account.
3. [Select a save option.](#)

The screenshot shows the 'Account Information' form for 'Account - Kristen's Account'. The form is displayed in a light blue theme. The left sidebar contains a navigation menu with options: General, Transactions, Posting Criteria, Categories, Documents, Security, Workflow, Child Accounts, History, Invoices, Expenses, Budget Requests, Time Entry Settings, and Cost Centers. The main form area has a title bar with 'Account Information' and a toolbar with buttons for Save, Cancel & View, Create a Copy, Activate, Share, and Print. The form fields are as follows:

- \*Name: Kristen's Account
- Active: No
- Parent Account: [Search icon]
- Account Type: Budget
- Overdraft Type: Do not allow negative Account
- \*Account Period Begins: 10/6/2016
- \*Account Period Ends: 10/31/2016
- \*Allocation Limit: 1,000.00
- Allow Posting

#### Assigning Parent Accounts

#### 1.7.1.4.2 Viewing or Creating Child Accounts

On an account's **Child Accounts** page, you can view its existing child accounts or create a new account and set it as a child account of the original account. For more information, see [Parent and Child Accounts](#).

#### To view or create a child account

1. Search for an account that will be the parent account and open it.
2. Click **Child Accounts** in the left pane.
3. Select one of the following options:
  - To view existing child accounts either browse the table or click an account name link to open the child account record.
  - To create a child account, click **New Child Account**.

A new **Account** record is displayed with the original account's name displayed in the **Parent Account** field.

4. Enter account information as described in [Creating or Editing an Account](#).
5. [Select a save option.](#)

The newly created account is displayed as a child account on the **Child Accounts** screen of the originally selected account. The original account is set as the parent account for the newly created child account.

#### 1.7.1.5 Activating or Deactivating Accounts

Use the **General** page to activate an existing account. If you are creating an account, you need to save and then activate the account before any transactions can be posted to the account. You must also activate the account before depositing funds or transferring funds between this account and another account. You can also deactivate an account, preventing future transactions from posting to the account.

##### To activate an account

1. Search for an account and open it.
2. From the **General** page, click **Activate**.

##### To deactivate an account

1. Search for an account and open it.
2. From the **General** page, click **Deactivate**.

#### Activation for Budget Requests

If you are using the Budget Request feature in Collaborati Spend Management, you must pay special attention to the sequence of activating accounts. The specific event that generates an automatic budget request is the activation of child accounts within a matter. For this reason, you must insure that all other conditions for budget requests have been met before you activate a child account.

The full set of conditions required for automatic generation of budget requests is described in the *CSM Administration Guide*.

#### 1.7.1.6 Adding and Subtracting Funds

The ability to add and subtract funds from an account depends on:

- Whether the account is independent or has a parent-child relationship--You can add funds to a parent account but not to a child account. You can transfer funds from the parent account to its child accounts. You can also transfer funds between child accounts that share the same parent account.
- The account's **Overdraft Type** setting on the **General** page:
  - If **Do not allow negative account** is selected--Prevents subtracting an amount greater than the current balance.
  - If **Allow negative account** is selected--No restriction on subtracting from the account.

- If **Automatic overdraft protection** is selected--Allows subtracting an amount covered by the parent account's balance.


#### To add or subtract funds to an account


1. Search for an account and open it.
2. Verify that the account has been activated. If an **Activate** button appears, click **Activate**. If a **Deactivate** button appears, the account is already active.
3. Select one of the following options:
  - **Add funds to the account:**
    - i. In the **Account Summary** section, click the **Add** link.
    - ii. On the **Add Funds** screen, enter the numeric **Amount** to amount.
    - iii. Click **Add**.
  - Subtract funds from the account:
    - i. In the **Account Summary** section, click the **Subtract** link.
    - ii. On the **Subtract Funds** screen, enter the numeric **Amount** to subtract. For example, to subtract \$2,500, enter 2500.
    - iii. Click **Subtract**.

#### 1.7.1.7 Transferring Funds Between Accounts

You can only transfer funds between accounts that have parent-child relationships. For example, you can transfer funds from a parent account to its child accounts. You can also transfer funds from a child account to its parent account.

#### To transfer funds between accounts

1. Search for an account and open it.
2. Click the **Account Summary** page.
3. Complete the following steps, depending on whether you are transferring funds to or from accounts:
  - **To transfer funds from the current account to another account**
    - a. Click **Transfer To**.
    - b. On the **Transfer Funds** page, click the Search Accounts icon  next to the **To** field to search and select an account.
  - **To transfer funds from another account to the current account**
    - a. Click **Transfer From**.

- b. On the **Transfer Funds** page, click the Search Accounts icon  next to the **From** field to search for, and then select an account.
4. Enter the numeric value to transfer in the **Amount** field.
  5. Click **Transfer**.

## Account Transfer Funds Screens Field Descriptions

Some field descriptions that follow may not display based on system settings and your assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

### Account - Transfer Funds To Screen Field Descriptions

Field (*required)	Description
<b>*Amount</b>	The numeric value to transfer.
<b>Description</b>	Notes about the transaction.
<b>From</b>	The current account name.
<b>*To</b>	The account to which to transfer funds. Click the Find icon to search and select an account to transfer funds to.

### Account - Transfer Funds From Screen Field Descriptions

Field (*required)	Description
<b>*Amount</b>	The numeric value to transfer.
<b>Description</b>	Notes about the transaction.
<b>*From</b>	The account to transfer funds from.
<b>To</b>	The current account name.



### 1.7.1.8 Setting Transaction Posting Criteria

Use the **Posting Criteria** page to define conditions under which transactions (expenses, tasks, invoice line items) can be posted to the account. Before a transaction is posted to an account, it is compared to the Posting Criteria. You must enable posting tasks, expenses, or invoices to an account on the **Posting Criteria** page before you can post a transaction.

The following procedures are described below:

You can perform one or more of the following procedures per account.

- [To set transaction posting criteria for tasks](#)
- [To set transaction posting criteria for expenses](#)
- [To set transaction posting criteria for invoice line items](#)
- [To post matter-level taxes to accounts](#)

The following screenshot shows general groupings of transaction posting criteria for posting tasks or expenses.

The screenshot displays the 'Account Posting Criteria Page Excerpt' with four main sections: Project, Task, Expense, and Involved. Each section contains specific configuration options for posting transactions to an account.

- Project:** Includes a dropdown for 'Posting Project to this Account' (set to '(Any)'), a search field for 'This Project', and a dropdown for 'Post Projects of Type' (set to 'A') with a '(Select)' button.
- Task:** Includes checkboxes for 'Post Tasks to this Account' and 'Prevent Posting Child Task Categories'. It also has a dropdown for 'Post Tasks of Type' (set to '(Any)') and a 'Task Percent' field (set to '100 %').
- Expense:** Includes checkboxes for 'Post Expenses to this Account' and 'Prevent Posting Child Expense Categories'. It has a dropdown for 'Post Expenses of Category' (set to '(Any)') and an 'Expense Percent' field (set to '100 %').
- Involved:** Includes a dropdown for 'Post Account to this Involved' (set to '(Any)') and a search field for 'This Involved'.

Account Posting Criteria Page Excerpt (Task and Expense Posting Criteria)

The following screenshot shows general groupings of transaction posting criteria for posting invoice line items (expense or task type). Although the Projects section is not shown below, you can also add additional posting criteria for an invoice line item's associated project.

The screenshot displays the 'Account Posting Criteria Page Excerpt (Invoice Line Item Posting Criteria)'. It is divided into two main sections: 'Vendor' and 'Invoice'.

**Vendor Section:**

- 'Post Account to this Vendor:' with a dropdown menu set to '(Any)'.
- 'This Vendor:' with an empty text input field and a search icon.

**Invoice Section:**

Select the Invoice Types that can be posted to this account.

**Post Invoice Type:**

- Standard
- Credit Note
- Accrual

OR

- Shadow Invoice

**Task and Expense Criteria:**

- Post Invoice Tasks to this Account
- Prevent Posting Child Invoice Task Categories
- Post Invoice Expenses to this Account
- Prevent Posting Child Invoice Expense Categories

**Task Posting Criteria:**

- 'Post Invoice Task of Type:' with a dropdown menu set to '(Any)'.
- 'Invoice Task Percent:' with a text input '100' and a '%' symbol.

**Expense Posting Criteria:**

- 'Post Invoice Expense of Category:' with a dropdown menu set to '(Any)'.
- 'Invoice Expense Percent:' with a text input '100' and a '%' symbol.

Account Posting Criteria Page Excerpt (Invoice Line Item Posting Criteria)

## Tips

- If you allow tasks and/or expenses to post to an account, you can also specify additional posting criteria by the associated project and/or task contact or expense contact.
- If you allow invoice tasks and/or invoice expenses to post to an account, you also can specify additional posting criteria by the associated project and by vendor contact. For invoice line items, you can also restrict posting by line item task type (category), line item expense type (category), and by timekeeper.
- If you allow posting by more than one transaction type (for example, if you select **Post Tasks to this Account** and also select **Post Expenses to this Account**), then any posting criteria you specify under the **Project** section or **This Involved** section will apply to all allowed transactions. For example, if you specify posting criteria to transactions associated with a specific project, then only tasks associated with that project or expenses associated with the same project will be able to post.

### To set transaction posting criteria for tasks

1. Search for an account and open it.
2. Click **Edit** on the toolbar or click the **Edit** link in the section that you want to update.
3. Click **Posting Criteria** in the left pane.
4. To allow Tasks to post to this account, in the **Task** section, select the **Post Tasks to this Account** check-box.  

This allows all tasks to post to the current account.
5. Specify how tasks can post to the account in one or more of the following ways:
  - By a task category—In the **Task** section, select a category from the **Post Tasks of Type** drop-down list.
  - By a percent amount of the task—Type a numeric percent value in the **Task Percent** field. For example, if you enter 10 and the task total is \$70, then 10% of \$70 (or \$7) will post to the account.
  - By a task contact—In the **Involved** section, for the **Post Account to this Involved** drop-down list, select **One**. Click the Search Contacts icon next to the **This Involved** field to search and select a contact.
  - By an associated project—In the **Project** section, for the **Post Project to this Account** drop-down list, select **One**. Click the Search Projects icon next to the **This Project** field to search and select a project.
  - By an associated project category—In the **Project** section, for the **Post Project to this Account** drop-down list, select **By Type**. In the **Post Projects of Type** field, select the custom record type (for example, Project). Also select the project category to restrict transactions to.
6. [Select a save option.](#)

Any posting criteria selected under the Project or Involved sections would also apply to posting expenses or invoice line items.

#### To set transaction posting criteria for expenses

1. Search for an account and open it.
2. Click **Edit** on the toolbar or click the **Edit** link in the section that you want to update.
3. Click **Posting Criteria** in the left pane.
4. To allow expenses to post to this account, in the **Expense** section, select the **Post Expenses to this Account** check-box.  

This allows all expenses to post to the current account.
5. Specify how expenses can post to the account in one or more of the following ways:
  - By an expense category—Select a category in the **Post Expenses of Category** drop-down list.

- By a percent amount of an expense—Type a numeric percent value in the **Expense Percent** field. For example, if you enter 10 and the expense total is \$70, then 10% of \$70 (or \$7) will post to the account.
  - By an expense contact—In the **Involved** section, for the **Post Account to this Involved** drop-down list, select **One**. Click the Search Contacts icon next to the **This Involved** field to search and select a contact.
  - By an associated project—In the **Project** section, for the **Post Project to this Account** drop-down list, select **One**. Click the Search Contacts icon next to the **This Project** field to search and select a project.
  - By an associated project category—In the **Project** section, for the **Post Project to this Account** drop-down list, select **Type**. In the **Post Projects of Type** field, select the custom record type (for example, Project). Also select the project category to restrict transactions to.
6. [Select a save option.](#)
- Any posting criteria selected under the Project or Involved sections would also apply to posting tasks or invoice line items.

#### To set transaction posting criteria for invoice line items

1. Search for an account and open it.
2. Click **Edit** on the toolbar or click the **Edit** link in the section that you want to update.
3. Click **Posting Criteria** in the left pane.
4. To specify the types of invoices that can post to the account, in the **Invoice** section, select one or more checkboxes in the **Post Invoice Type** fields. The options that you select are the only type(s) of invoices that will post to this account. Options are **Standard**, **Accrual**, **Shadow**, or **Credit Note**. See [Account Invoice Section Field Descriptions](#) for more information.

**Important:** *If you do not select an option in the **Post Invoice Type** area, nothing will post to this account, regardless of the other options that you select on this page.*

There are special considerations for posting Shadow Invoices:

- An account that allows Shadow Invoices to post cannot have a parent account, or child account, that does not allow Shadow Invoices to post. In other words, the treatment of Shadow Invoices must be consistent throughout a parent-child hierarchy.
  - An account that has not had any invoices posted to it is eligible to have Shadow Invoices posted to it. However, once an invoice that is not a Shadow Invoice has been posted to that account, Shadow Invoices can no longer be posted to it. Conversely, if a Shadow Invoice is the first invoice posted to an account, thereafter no other invoice type will be allowed to post to that account.
5. To allow invoice task/fee line items to post to this account, in the **Invoice** section, select the **Post Invoice Tasks to this Account** check-box.

This allows all invoice task/fee line items to post to the current account.

6. Specify how invoice task/fee line items can post to the account in one or more of the following ways:
  - By a task category—Select a category in the **Post Invoice Task of Type** drop-down list.
  - By a percent amount of an invoice task line item—Type a numeric percent value in the **Invoice Task Percent** field. For example, if you enter 10 and the task total is \$70, then 10% of \$70 (\$7) will post to the account.
  - By an associated project—In the **Project** section, for the **Posting Project to this Account** drop-down list, select **One**. Click the Search Projects icon next to the **This Project** field to search and select a project.
  - By an invoice vendor—In the **Vendor** section, for the **Post Account to this Vendor** drop-down list, select **One**. Click the Search Contacts icon next to the **This Vendor** field to search and select a contact.
  - By an invoice line item timekeeper—In the **Involved** section, for the **Post Account to this Involved** drop-down list, select **One**. Click the Search Contacts icon next to the **This Involved** field to search and select a contact.
7. To allow Invoice expense line items to post to this account, in the Invoice section, select the **Post Invoice Expenses to this Account** check-box.

This allows all expense line items to post to the current account.

8. Specify how invoice expense line items can post to the account in one or more of the following ways:
  - By an expense category—Select a category in the **Post Invoice Expense of Category** drop-down list.
  - By a percent amount of an invoice expense line item—Type a numeric percent value in the **Invoice Expense Percent** field. For example, if you enter 10 and the expense total is \$70, then 10% of \$70 (or \$7) will post to the account.
  - By an associated project—In the **Project** section, for the **Posting Project to this Account** drop-down list, select **One**. Click the Search Projects icon next to the **This Project** field to search and select a project.
  - By an invoice vendor—In the **Vendor** section, for the **Post Account to this Vendor** drop-down list, select **One**. Click the Search Contacts icon next to the **This Vendor** field to search and select a contact.
  - By an invoice line item timekeeper—In the **Involved** section, **Post Account to this Involved** drop-down list, select **One**. Click the Search Contacts icon next to the **This Involved** field to search and select a contact.
9. [Select a save option.](#)

Any posting criteria selected under the Project or Involved sections would also apply to posting tasks or expenses.

### To set transaction posting criteria for non-US tax accounts

1. Search for an account and open it.
2. Click **Edit** from the toolbar or click the **Edit** link for the section to update.
3. Click **Posting Criteria** from the left pane.
4. Click the **Post Non-US Tax to this Account** to allow non-US Tax categories to post to the account.
5. In the **Post Non-US Tax of Type** field, select the non-US Tax categories that are allowed to post to this account. Once you select a category, the category and all of its child categories are allowed for this account.
6. In the **Non-US Tax Percent** field, enter the percentage of the tax amount that will post to the account.
7. [Select a save option.](#)

### To post matter-level taxes to accounts

*Note: If an invoice uploaded from Collaborati has matter-level taxes, you must use these steps to post the invoice to an account.*

1. Search for an account and open it.
2. Click **Edit** on the toolbar.
3. Click **Posting Criteria** on the left pane.
4. Depending on whether you need to post fee or expense taxes, place a checkmark in the **Post total matter fee tax to this account** or the **Post total matter expense tax to this account** check-boxes.
5. Select a save option.

### To allow accounts to be associated with budget requests

1. Search for an account and open it.
2. Click **Edit** on the toolbar.
3. Click **Posting Criteria** on the left pane.
4. From **Posting Project to this Account**, select **One**.
5. From **This Project**, select the matter or project you want to associate with this account.
6. From **This Vendor**, select the vendor you want to associate with this account. The vendor you select must be an involved party of the project you select in the previous step.

When you create the budget request, you must specify the same project and vendor for the budget request that you specified for the account. See [Creating a Budget Request](#) for more information.

7. Specify whether you want to post invoice tasks or expenses to this account by placing a check-mark in one or both of the following check-boxes:
  - **Post Invoice Tasks to this Account**
  - **Post Invoice Expenses to this Account**
8. Save the account.

**Note:** The account must be current for you to associate a budget request with the account. You must also activate the account after you create the budget request. See [Activating or Deactivating Accounts](#) for more information.

## Account Posting Criteria Page Field Descriptions

Some field descriptions that follow may or may not display based on system settings and your assigned rights. The following tables provide field descriptions for the Posting Criteria page. The field descriptions are in the order that they appear on the screen. The field descriptions for the Projects, Involved, and Vendor sections are at the bottom of the topic because they are used as secondary filters to the posting criteria for expenses, tasks, and invoice line items (expense and fee).

### Account Task Section Field Descriptions

Field	Description
<b>Post Tasks to this Account</b>	If this check-box is selected, tasks will automatically post against the account.  <i>Note: If the check-box is cleared, disregard the other two fields in this section.</i>
<b>Post Tasks of Type</b>	The task category used as criteria for posting a task to this account. You can select a specific category or select <b>Any</b> .  Once you select a category, the category and all of its child categories are allowed for this account.
<b>Task Percent</b>	The percent amount of a task that can post to this account.

### Account Expense Section Field Descriptions

Field	Description
<b>Post Expenses to this Account</b>	If this check-box is selected, expenses will automatically post against the account.

	<b>Note:</b> If the check-box is cleared, disregard the other fields in this section.
<b>Post Expenses of Category</b>	The expense category used as criteria for posting an expense to this account.
<b>Expense Percent</b>	The percent amount of an expense that can post to this account.

#### Account Project Section Field Descriptions

Field	Description
<b>Post Project to this Account</b>	<p>Can be one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Any</b>—(default) Indicates that there is no restriction for transactions posting to the account based on the associated project. If you select this option, you can leave the other two fields in this section blank.</li> <li>• <b>One</b>—Allows you to specify a single project whose associated transactions can be posted against the account. If you select this option, specify This Project field value below.</li> <li>• <b>By Type</b>—Allows you to specify the project type (category) whose associated transactions can be posted against the account. If you select this option, specify the <b>Post Projects of Type</b> field value below.</li> </ul>
<b>This Project</b>	<p><b>Important:</b> Required if the <b>Post Project to this Account</b> drop-down lists's option, <b>One</b>, is selected.</p> <p>The project used as criteria for posting a transaction to this account.</p> <p>Only transactions associated with the project can post against the account. Also the project's Accounts page displays a link to the account.</p>
<b>Post Projects of Type</b>	<p><b>Important:</b> Required if the <b>Post Project to this Account</b> drop-down lists's option, <b>By Type</b>, is selected.</p> <p>The project category used as criteria for posting a transaction to this account.</p> <p>Only transactions associated with projects that have this category type enabled can post against the account.</p> <p>You can select a specific category or select <b>Any</b>.</p>



Once you select a category, the category and all of its child categories are allowed for this project.

#### Account Involved Section Field Descriptions

Field	Description
<b>Post Account to</b>	<p>Can be one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Any</b>—(default) Indicates that there is no restriction for tasks or expenses posting to the account based on the associated <b>Contact</b>. If you select this option, you can leave the other field in this section blank.</li> <li>• <b>One</b>—Allows you to specify the contact associated with tasks or expenses that can post against the account. If you select this option, specify the <b>This Involved</b> field value below.</li> </ul>
<b>This Involved</b>	<p><i><b>Important:</b> Required if the <b>Post Account to</b> drop-down lists's option, <b>One</b>, is selected.</i></p> <p>Find and select the desired contact.</p> <p>Afterward, only tasks or expenses where this contact is specified in the <b>General</b> page, <b>Contact</b> field, can post against the account.</p>

#### Account Vendor Section Field Descriptions

Field	Description
<b>Post Account to</b>	<p>Can be one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Any</b>—(default) Indicates that there is no restriction for invoice tasks or invoice expenses posting to the account based on the associated Vendor contact. If you select this option, you can leave the other field in this section blank.</li> <li>• <b>One</b>—Allows you to specify the Vendor contact associated with invoice tasks or invoice expenses that can post against the account. If you select this option, specify the <b>This Vendor</b> field value below.</li> </ul>
<b>This Vendor</b>	<p><i><b>Important:</b> Required if the <b>Post Account to</b> drop-down list's option, <b>One</b>, is selected.</i></p> <p>Find and select the desired vendor contact.</p> <p>Afterward, only invoice tasks or invoice expenses where this vendor is specified in the <b>General</b> page, <b>Vendor</b> field, can post against the account.</p>

## Account Invoice Section Field Descriptions

Field	Description
<b>Post Invoice Type</b>	<p>The type of invoices that can post to this account.</p> <p>In addition to the typical <b>Standard</b> invoice, additional options are as follows:</p> <ul style="list-style-type: none"> <li>• <b>Credit Note</b>—A legal instrument that corrects a Non-US Tax invoice.</li> <li>• <b>Accrual</b>—An estimate of an upcoming bill.</li> <li>• <b>Shadow Invoice</b>—A detailed invoice of work performed that is not intended to be actually paid, because an alternate fee arrangement exists between client and vendor. The invoice is posted to allow the client to do financial analysis.</li> </ul> <p>If you do not select an option, no invoices will post to this account.</p>
<b>Post Invoice Tasks to this Account</b>	<p>If this check-box is selected, invoice task line items will automatically post against the account.</p> <p><i>Note: If the check-box is cleared, disregard the following two fields in this section.</i></p>
<b>Post Invoice Tasks of Type</b>	<p>The invoice line item task category used as criteria for posting an expense to this account.</p> <p>You can select a specific category or select <b>Any</b>.</p> <p>Once you select a category, the category and all of its child categories are allowed for this invoice.</p>
<b>Invoice Task Percent</b>	<p>The percent amount of an invoice task line item that can post to this account.</p>
<b>Post Invoice Expenses to this Account</b>	<p>If this check-box is selected, invoice expense line items will automatically post against the account.</p> <p><i>Note: If the check-box is cleared, disregard the following two fields in this section.</i></p>
<b>Post Invoice Expenses of Type</b>	<p>The invoice line item expense category used as criteria for posting an expense to this account.</p>
<b>Invoice Expenses</b>	<p>The percent amount of an invoice task line item that can post to this account.</p>

Percent	
---------	--

## Account Taxes Section Field Descriptions

Field	Description
<b>Post Non-US Tax to this Account</b>	Check this check-box to automatically post non-US tax amounts against this account.  <i>Note: If the check-box has no check mark, disregard the <b>Post Non-US Tax of Type</b> and <b>Non-US Tax Percent</b> fields.</i>
<b>Post Non-US Tax of Type</b>	Select a type of non-US tax amounts to this account. <b>Any</b> indicates no restriction on tax types.
<b>Non-US Tax Percent</b>	Enter the percent of the non-US tax amount that can post to this account.
<b>Post total matter fee tax to this account</b>	Check this check-box to automatically post matter-level fee taxes against this account. The tax amounts come from invoices uploaded from Collaborati.
<b>Post total matter expense tax to this account</b>	Check this check-box to automatically post matter-level expense taxes against this account. The tax amounts come from invoices uploaded from Collaborati.
<b>Line Item Category Required for Detail Level Non-US Tax to Post</b>	Select the desired radio button to refine the types of Non-US taxes that can be posted to the account. For more detail, see the section below on Refining Non-US Tax Posting. <i>Only available in TeamConnect 4.0.9+</i>

 **Refining Non-US Tax Posting**

Beginning in TeamConnect 4.0.9, users are able to refine account posting criteria to ensure Detail-Level Non-US taxes are posting optimally and as intended. The radio buttons in the image below demonstrate the options available for ensuring what type of Non-US taxes are applied to invoices on this account.



Category Detail Option	Result
All Categories	The Non-US tax chosen to be applied will be applied to all categories on the invoice.
<b>Expense Only</b>	The Non-US tax on the invoice will be applied only to the invoice expense category.
<b>Task Only</b>	The Non-US tax on the invoice will be applied only to the invoice fee category.
<b>Either Expense or Task</b>	The Non-US tax on the invoice will be applied to the expense or fee (task) set for the invoice.

**Note:** Bold items in the table above will be most useful to users with Financial Management. The All Categories option is best suited for users without Financial Management that utilize Detail-Level Non-US taxes on line items.

#### 1.7.1.9 Viewing Transactions History for an Account

Use the **Transactions** page to view all financial transactions for an account.

Transactions that are internal to the account include fund additions, fund subtractions, and transfer of funds.

Transactions that are triggered outside of the account can include posting and voiding expenses, posting and voiding tasks, and posting and voiding invoices. These externally triggered transactions are dependent on an account's **Posting Criteria** page settings. For more information, see [Setting Transaction Posting Criteria](#).

### Modifying Account Transactions

You cannot delete or modify account transactions from the **Transactions** page. If you need to change invoice, task, and expense transactions, you have to void each transaction, edit the respective record, and post the record again.

To change fund additions, subtractions, and transfers you must post reverse transactions. For example, if you added \$150,000 to an account and then decide that only \$75,000 should have been added, you can make the correction by subtracting \$75,000 from the account.

#### To view transactions history for an account

1. Search for an account and open it.
2. Click **Transactions** from the left pane. For more information about fields, see the following table.

#### Account Transactions Page Field Descriptions

Field	Description
<b>Search Transactions</b>	Click this link to perform a search using multiple transaction field values as search criteria.
<b>Date</b>	Displays the date the transaction occurred.
<b>Name</b>	Displays the type and number of the record that was posted or voided. If applicable, the invoice type is in parenthesis next to the invoice name.
<b>Vendor</b>	Click the link to view the associated vendor's contact record.
<b>Used For</b>	Click the link to view the record that was posted or voided.
<b>Debit</b>	Displays the monetary amount that was debited from (posted against) the account.
<b>Credit</b>	Displays the monetary amount that was credited to (or voided from) the account.
<b>Transactions per page</b>	Determines the number of transactions that appear on each page. Click the drop-down arrow to change the number.  <b>Note:</b> To change the default number of records that display per list page across all record types, see <a href="#">Defining Search Result Settings</a> .

#### 1.7.1.10 Viewing or Creating Account History

From the **History** page, you can add notes about changes made to an account.

For more information, see [Working with Record History](#).

## 1.7.2 Expenses

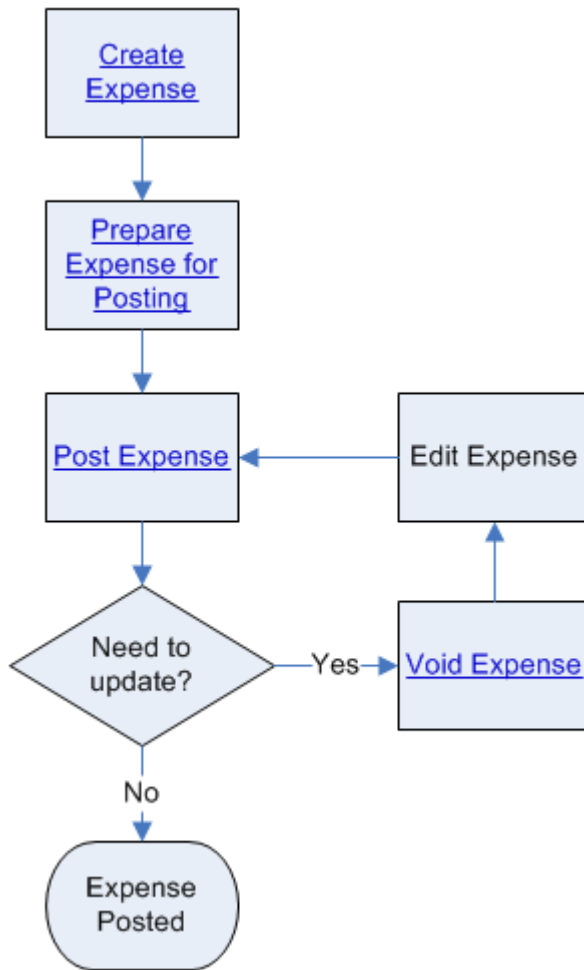
Expenses in TeamConnect are the internal costs of doing business at your organization. Typically, expenses include the cost of the merchandise bought and the cost of the services received, for example, office supplies, transportation, phone calls, and meals.

Two types of expenses exist:

- *Project-specific*—Associated with a particular project.
- *Global*—Not associated with projects.

All expenses can post against certain accounts according to the posting criteria you set for each individual account. See [Posting an Expense](#).

The following flowchart illustrates how you can use expenses. Click areas of this flowchart for more information.



## Expense Entry Tool

You can also add multiple expenses at the same time. The Expense Entry Tool allows you to enter and post multiple expenses into the system, all from one screen. TeamConnect creates the corresponding individual records automatically for each expense. See [Working with the Expense Entry Tool](#).

## Expense Record Pages

Two expense-specific pages exist:

- The **General** page lets you enter information about the expense, such as a Description, Date, Contact who is responsible for an expensed item, Unit Price, and Quantity.
- The **Transactions** page displays a history of the expense's posting and voiding statuses.

### 1.7.2.1 Viewing Expenses

You can view expenses by browsing expense collections. Alternatively, you can search for an expense, for example, from the **Custom Search** page.

See [Viewing Collections of Records](#) and [Searching for Specific Records](#) for more information.

#### To view expenses

1. Click the **All** button.
2. Click the **Expenses** link under the tab bar.

The recently viewed accounts are listed in the right pane. To open an account record, click an account link in the **Name** column.

A summary for each expense in a collection displays in a list pane to the right.

3. To open an expense, in the right pane, click an expense name link.

For more information about the areas on the **Expenses List** page, see [User Interface](#).

### 1.7.2.2 Creating or Editing an Expense

You can create or edit an expense manually using a blank expense record, or a wizard that guides you through the process step-by-step. You can also create an expense record by using an existing expense record as a template. See [Copying a Record](#) for more information.

When you purchase merchandise or services necessary for conducting your business, you enter them as expenses into TeamConnect. Once you enter expenses, you can perform the following tasks:

- Post expenses against the respective accounts.
- Track expenses by certain projects and by specific vendors.

There can only be one expense item for each expense record. To make multiple expense entries, see [Working with the Expense Entry Tool](#).

You can create expenses in the following locations:

- On the **Expenses List** page.
- On a project's **Expenses** page.
- Using the Expense Entry Tool on the **All** page.

#### Using a Wizard

Your system administrator may provide one or more wizards to help you create new expenses using a guided, step-by-step process. A wizard may be one of the available options or the only option.

If a wizard is your only option, when you click **New** the wizard opens. Complete the required fields on each page, and then click **Next** to progress to the end of the wizard. When you click **Finish** at the end of the wizard, the record is available in the appropriate list view and collection. See [Viewing Expenses](#) or [Viewing Collections of Records](#) for more information.


**Note:** If a wizard is your only option, and you want to use an existing expense record to create a new account record, a wizard will not open. A blank expense record opens and you can complete the fields manually. See [Copying a Record](#) for more information.

#### 1.7.2.2.1 Creating or Editing an Expense Manually

Use the expense **General** page to begin manually creating an expense record. Once you save the record, you can edit it at any time.

**Note:** Your company may allow you to create an expense using a wizard only, so the option to create an account manually may not be available. See [Using a Wizard](#) for more information.

#### To create or edit an expense manually

1. Click the **Finance** tab.
2. Click the **Expenses** link under the tab bar.  
The **Expenses List** page opens.
3. Select one of the following options:
  - To create an expense, click **New**.
  - To edit an expense, click the **Edit Record** icon  to the left of an expense.
4. From the **General** page, the following fields from the **Account Information** section are required:
  - **Description**
  - **Expense Date**
  - **Expensed By**
5. (optional) To enter information in fields that have been created especially for your company:
  - From the **Category** drop-down list, select an option. Inactive categories, if any, are displayed but cannot be selected.
  - A **Details** section will display below **Notes**. Enter the appropriate information in the corresponding fields.
6. (optional) To enter billing information:
  - Enter a numeric value in the **Unit Price** field.
  - Enter a numeric value in the **Quantity** field.
7. For additional field descriptions, see [Expense Information Section Field Descriptions table](#).
8. [Select a save option](#).



## Expense General Page Field Descriptions

Some field descriptions that follow may not display based on system settings and your assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

### Expense Information Section Field Descriptions

Field (* = required)	Description
<b>Category</b>	The expense type (category), such as Transportation, Meals, or Calls. An expense record can only be associated with one category. When you select a <b>Category</b> , additional fields that are related to this expense category may appear in a Details section. Some of these new fields may be required and must be completed before you can save the record.  Categories are usually pre-defined by your system administrator.
<b>Contact</b>	The contact who is charging for the expense item.  Typically, this contact is a vendor. This field value may be validated against the associated account's Vendor posting criterion. See <a href="#">Setting Transaction Posting Criteria</a> .
<b>*Description</b>	The expense description.
<b>*Expensed by</b>	The user who is responsible for the expenditure. The default value is the creator's user name.  For an existing expense, displays a link to the user's contact record.
<b>*Expense Date</b>	The <b>Calendar</b> icon allows selection of a date to associate with the expense, for example, when the expense was incurred or the date the record was made.  The default date is the current date.
<b>Project</b>	The project associated with the account. The Find icon allows searching and selecting a project. This field value may be validated against the associated account's Project posting criteria. See <a href="#">Setting Transaction Posting Criteria</a> .  <i>Note: A link to the project record displays.</i>

### Expense Billing Information Section Field Descriptions

Field	Description
-------	-------------

<b>Posting Status</b>	The posting status for an expense can be one of the following: <ul style="list-style-type: none"> <li>• <b>Not Posted</b>—The expense has not been posted or it has been voided.</li> <li>• <b>Posted</b>—The expense posted to an account.</li> <li>• <b>Failed</b>—An unsuccessful attempt was made to post the expense.</li> <li>• <b>Posted For Approval</b>—The expense has been submitted for approval to be posted.</li> </ul>
<b>Quantity</b>	The total quantity of expense items.
<b>Total</b>	This amount is automatically calculated by TeamConnect. <b>Total = Quantity x Unit Price.</b>
<b>Unit Price</b>	The price per unit (item) for the expense.

#### Expense Notes Section Field Description

Field	Description
<b>Notes</b>	Comments for the expense.  <i>Note: If you enter more than 250 characters without any spaces, you will need to scroll horizontally on the record to view all sections.</i>

#### Expense Access Information Section Field Descriptions

Field	Description
<b>Created By</b>	The name of the user who created the expense.
<b>Created On</b>	The date that the expense was added to TeamConnect.
<b>Modified By</b>	The name of the user who last modified the expense.
<b>Modified On</b>	The date on which the expense was modified.

### 1.7.2.3 Preparing an Expense for Posting

Before posting an expense, you must prepare the expense and the account associated with it. The following table describes requirements for posting.

Requirement	Description
-------------	-------------

The expense must be associated with an account.

From **Accounts**, search for all accounts associated with the project listed on the expense. This project can be a dispute or other type of matter. If you want to post an expense to an account, it must be associated with a project.

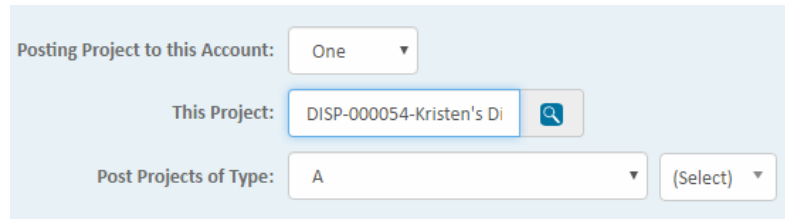
For example, enter DISP-000054 for the name value in the search. Select the account you want to post to, in this case, one with "Expenses" in the **Name**. For example, "Allocated - DISP-00098 - OC Expenses - 2013."

Edit	Name	Type
	Kristen's Account	Account
	Child Account 1	Account

**Selecting Accounts for Posting Expenses**

If no accounts exist for the project, create an account and specify the following fields on the **Posting Criteria** page.

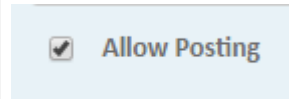
- Posting Project to this Account: Select **One**
- This Project: Enter the name of the project



**Posting Projects to an Account**

The account must allow posting.

From the **General** page of the account, place a checkmark in the **Allow Posting** check-box if one does not exist.

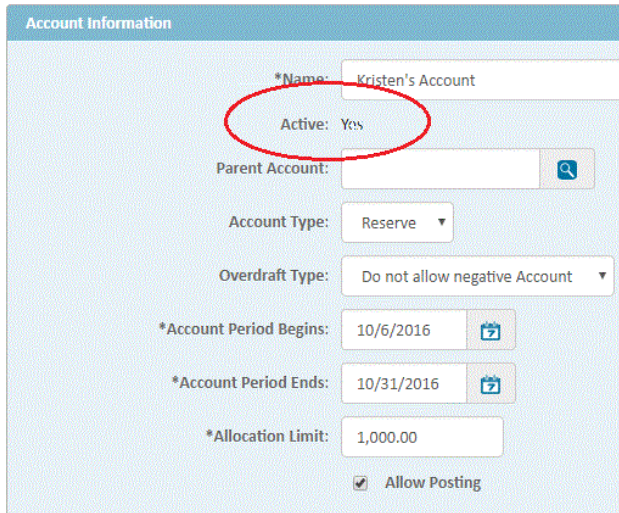
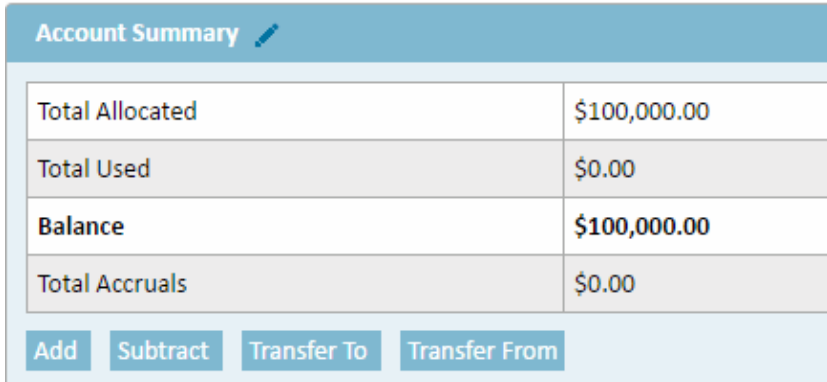
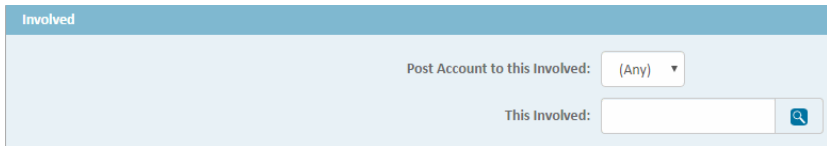


**Allowing Posting to an Account**

The account must allow expenses to be posted.

1. From the **Posting Criteria** page of the account, place a checkmark in the **Post Expenses to this Account** check-box if one does not exist.
2. From the **Post Expenses of Category** drop-down, update the type of expenses you want to be able to post.
3. From the **Expense Percent** field, enter a number for the percentage of the expense you want to post. Keep at 100% if you are only posting to one account. Update the percentage if you want to split expense amounts between multiple accounts.

	<div data-bbox="587 197 1420 443"> <p><b>Expense</b></p> <p><input checked="" type="checkbox"/> Post Expenses to this Account</p> <p><input type="checkbox"/> Prevent Posting Child Expense Categories</p> <p>Post Expenses of Category: (Any) ▾</p> <p>Expense Percent: 100 %</p> </div> <p style="text-align: center;"><b>Allowing Expenses to Post an Account</b></p>
<p>The date of the expense must be within the account period.</p>	<p>The <b>Expense Date</b> on the expense should be within the account period, which can be found on the <b>General</b> page of the associated account</p> <div data-bbox="587 577 1420 1243"> <p><b>Account Information</b></p> <p>*Name: Kristen's Account</p> <p>Active: Yes</p> <p>Parent Account: <input type="text"/> 🔍</p> <p>Account Type: Reserve ▾</p> <p>Overdraft Type: Do not allow negative Account ▾</p> <p>*Account Period Begins: 10/6/2016 📅</p> <p>*Account Period Ends: 10/31/2016 📅</p> <p>*Allocation Limit: 1,000.00</p> <p><input checked="" type="checkbox"/> Allow Posting</p> </div> <p style="text-align: center;"><b>Account Period on the General Page</b></p>
<p>The account must be active.</p>	<p>The status of the <b>Active</b> field on the <b>General</b> page of the account is <b>Yes</b>. See <a href="#">Activating or Deactivating Accounts</a> for more information.</p>

	 <p style="text-align: center;"><b>Account Period on the General Page</b></p>
<p>The account (or the parent's account) must have a balance of the same amount as the expense or more.</p>	<p>From the <b>Account Summary</b> page of the account, check whether the <b>Balance</b> has enough funds to cover the expense. If the account has no balance, check the <b>Account Summary</b> page of the parent account or the parent's parent account. You can continue checking parent accounts until you find a balance. If the balance does not have enough funds, add money to one of the accounts.</p>  <p style="text-align: center;"><b>Balance on the Account Summary Page</b></p>
<p>The expense must include the same Involved contact that is on the account.</p>	<p>From the <b>Posting Criteria</b> page, if the associated account specifies an <b>Involved</b> contact in the <b>This Involved</b> field, the <b>Contact</b> field of the expense record must include the same contact.</p> <p><b>Note:</b> <i>The account does not need to specify an Involved contact, even if the expense includes one.</i></p>  <p style="text-align: center;"><b>Specifying an Involved Contact for Posting</b></p>

#### 1.7.2.4 Posting an Expense

When an expense posts successfully, TeamConnect debits the budget of an account associated with the expense.

**Note:** *If TeamConnect cannot locate the appropriate budget, you may need to associate the expense with an account's budget. See [Preparing an Expense for Posting](#).*

##### To post an expense

1. Search for an expense and open it.
2. Click **Post**.

The account transaction displays on the **Transactions** page of the expense.

**Note:** *After posting, the expense's system fields are not editable. Depending on system settings and your rights, the expense's custom fields may be editable. Click **Void** to void the transaction and make the system fields editable again.*

#### 1.7.2.5 Voiding Expenses

You can void an expense to clear a post transaction from the related account. The posted expense total will be credited to the related account.

##### To void expenses

1. Search for an expense and open it.

Depending on the expense status, a **Post** or **Void** button displays on the toolbar.

2. Click **Void**.

Afterward, the corresponding account transactions will be removed from the **Transactions** page of the expense and associated account.

3. Once an expense is voided, the expense can be reposted. After the expense is voided, the expense fields can be edited. Afterward, the expense can be posted again.

#### 1.7.2.6 Viewing Expense Transactions

Use the **Transactions** page to view a history of the expense posting to an account.

The **Transactions** page is read-only. To make changes to a posted expense, you must void the expense, modify it, and post it again. For more information, see [Posting an Expense](#), [Creating or Editing an Expense](#), or [Voiding Expenses](#).

##### To view expense transactions

1. Search for an expense and open it.

2. Click **Transactions** in the left pane.
3. Click the account name link to view the account details, if necessary.

### 1.7.2.7 Viewing or Creating Expense History

Use an expense's **History** page to create a history (containing notes about changes to the expense), view a list of history records for that expense, or perform an advanced search for a history.

For more information, see [Working with Record History](#).

### 1.7.2.8 Working with the Expense Entry Tool

The Expense Entry Tool allows you to enter and post multiple expenses into the system from one screen.

Refer to the following image of the Expense Entry Tool. Click each square for more information about how you use the buttons.

Expense Entry Tool

? Help

\*Expensed By: Sackey, Kristen

\*Description:

\*Expense Date: 10/7/2016

\*Category: Expense

Contact:

Project:

Unit Price: \$ 0.00

Quantity: 0.00

Total: \$0.00

Line	Expensed By	Description	Expense Date	Category	Contact	Project	Unit Price	Quantity	Total	Posting Status	Action
No records available											

Using the Expense Tool

#### 1.7.2.8.1 Entering Expenses

The Expense Entry Tool provides a convenient way to create multiple expenses from one TeamConnect page. The **Expense Entry Tool** page has the standard functionality of a screen. See [Working with the Expense Entry Tool](#) for more information.

#### To enter expenses

1. Open the **Expense Entry Tool** in one of the following locations:
  - Click the **Tools** icon in the top right corner of the screen.

Click **Expense Entry Tool**.

The Expense Entry Tool opens.

2. For each expense, enter the appropriate information. See [Expense Entry Tool Field Definitions](#) for more information.
3. Click **Add**.
4. (optional) To post expenses, select the check-box to the left of the expense(s), and then click **post**.

## Expense Entry Tool Field Descriptions

Some field descriptions that follow may not display based on system settings and your assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

### Expense Entry Tool Fields

Field (* = required)	Description
<b>*Category</b>	The expense type (category), such as transportation, meals, or calls.
<b>Contact</b>	<p>The contact who is charging for the expense item, for example, a vendor.</p> <p>The Find icon allows searching and selecting the contact who is charging you for the expense item.</p> <p>This value may be validated against an associated account's Contact posting criterion.</p>
<b>*Description</b>	<p>The expense description.</p> <p><i>Tip: To facilitate the searching process for the record, try to make the description unique and descriptive.</i></p>
<b>Expense List</b>	<p>The list of entered expenses.</p> <p>Check-boxes on the left of expense items allow selecting expenses for editing, deleting or posting.</p>
<b>*Expensed By</b>	<p>The user who is responsible for the expenditure. By default, it displays the creator's name.</p> <p><i>Note: Depending on how your version of TeamConnect is configured, this field may be a drop-down list or a Search Module. For more information, see <a href="#">Using Search Modules</a>.</i></p>




<b>*Expense Date</b>	<p>The date when the expense was incurred or the date the record was created.</p> <p>The <b>Calendar</b> icon allows selecting the desired date for the expense.</p> <p>When creating an expense, the default value is the current date (when creating an expense).</p>
<b>Post</b>	<p>This button posts the selected expenses to associated accounts. See <a href="#">Posting an Expense</a> for details.</p> <p><b>Note:</b> To void a posted expense, you must open the expense record and void on the <b>General</b> page. See <a href="#">Voiding Expenses</a>.</p> <p><b>Tip:</b> Click the corresponding expense link in the displayed list to open its <b>General</b> page.</p>
<b>Posting Status</b>	<p>The posting status for an expense can be one of the following, depending on whether you clicked the <b>Post</b> button or not:</p> <ul style="list-style-type: none"> <li>• <b>Not Posted</b></li> <li>• <b>Posted</b></li> </ul>
<b>Project</b>	<p>The project associated with the expense.</p> <p>The Find icon allows searching and selecting the appropriate project.</p> <p>This value may be validated against an associated account's Project posting criterion.</p>
<b>Quantity</b>	<p>The total quantity of expense items.</p>
<b>Total</b>	<p>This amount is automatically calculated by Team Connect:</p> <p><b>Total = Unit Price x Quantity.</b></p>
<b>Unit Price</b>	<p>The price per unit (item) for expense item.</p>

#### 1.7.2.8.2 Copying Expenses

The Expense Entry Tool provides a convenient way to create additional expenses from one TeamConnect page. By copying an existing expense, you can edit necessary fields and then add the expense as a new record.


#### To copy expenses

1. Open the **Expense Entry Tool** in one of the following locations:
2. On an existing expense, click the Copy icon .
3. Edit the resulting fields as necessary.
4. Click **Add**.

#### 1.7.2.8.3 Editing Expenses

The Expense Entry Tool provides a convenient way to edit multiple expenses from one TeamConnect page.

##### To edit expenses

1. Open the **Expense Entry Tool**:
2. On the table of existing expenses, under the **Action** column, click the Edit icon  next to the expense to edit.
3. Make changes to fields described in [Expense Entry Tool Field Definitions](#).
4. Click **OK** for each updated expense.

#### 1.7.2.8.4 Posting Expenses

The Expense Entry Tool provides a convenient way to post multiple expenses from one TeamConnect page.

##### To post expenses

1. Open the **Expense Entry Tool**:
2. To post all expenses, select the top check-box, or select the check-boxes for expenses to post.
3. Click **Post**.
4. Afterward, the corresponding account transactions will display on the **Transactions** page of the expense and associated account.

#### 1.7.2.8.5 Deleting Expenses

The Expense Entry Tool provides a convenient way to delete multiple expenses from one TeamConnect page.

##### To delete expenses

1. Open the **Expense Entry Tool**:

2. To delete expenses you have created with this tool, select the top check-box or select individual check-boxes for expenses to delete.
3. Click **Remove**.

### 1.7.2.9 Updating Expense Unit Prices

Use the **Expense Unit Prices** tool to update one or more prices that are linked to active expense codes.

**Note:** Typically, expense codes are set up in advance and you cannot delete or change them. You cannot use this tool to add new expense codes - see your system administrator if you want to add or change codes.

You can update expense unit prices in the following ways:

- Upload a .csv or Excel file—Create a data file that contains the new prices, and then upload the file to TeamConnect.
- Enter new prices manually—Type the modified rates on the Expense Unit Prices page.

#### To access the Expense Unit Prices tool

1. Click the **Tools** button in the top right corner of the page.
2. Click **Expense Unit Prices** from the drop-down menu.

The **Expense Unit Prices** page opens.

Expense Unit Prices

[Update](#) [? Help](#)

**Upload Expense Unit Prices**

To update Expense Unit Prices, upload a spreadsheet or enter them in the table below.

Upload Expense Unit Prices: (.csv or Excel files) [Download Sample File](#)

No file chosen

**Expense Unit Prices**

Expense Unit Prices 1 - 27 of 27

Code	Description	Unit Price
OCEX	Expense:Outside Counsel Expenses	0.00
E100	Expense:Outside Counsel Expenses:E100 Expenses	0.00
E101	Expense:Outside Counsel Expenses:E100 Expenses:E101 Copying	0.00
E102	Expense:Outside Counsel Expenses:E100 Expenses:E102 Outside printing	0.00
E103	Expense:Outside Counsel Expenses:E100 Expenses:E103 Word processing	0.00
E104	Expense:Outside Counsel Expenses:E100 Expenses:E104 Facsimile	0.00
E105	Expense:Outside Counsel Expenses:E100 Expenses:E105 Telephone	0.00
E106	Expense:Outside Counsel Expenses:E100 Expenses:E106 Online research	0.00
E107	Expense:Outside Counsel Expenses:E100 Expenses:E107 Delivery services/messengers	0.00
E108	Expense:Outside Counsel Expenses:E100 Expenses:E108 Postage	0.00

#### Expense Unit Prices Page

## Creating Expense Unit Price Data Files

Before you can upload prices into TeamConnect, you must first create the file that you want to upload. You can upload an Excel spreadsheet or a .csv file.

### To create an expense unit price data file

Create an Excel spreadsheet or .csv file based on the following guidelines:

- The first row must contain labels only.
- Additional rows must contain expense unit price data only

**Note:** TeamConnect provides a sample Excel file. To access this file, click **Download Sample File** on the Expense Unit Prices page. The sample file opens in Excel.

### Sample Excel File

A completed expense unit price Excel file may appear as follows:

	A	B
1	CODE	UNIT PRICE
2	E101	0.1
3	E102	0.2
4	E103	0.3
5	E104	0.4
6		

### Sample .csv File

A completed expense unit price .csv file may appear as follows:

```
CODE,UNIT PRICE
E101,2.1
E102,4.2
E103,2.3
E104,3.4
```

## Uploading Expense Unit Price Data

Once you create the data files, you can upload them to TeamConnect.

### To upload expense unit price data

1. With the **Expense Unit Prices** page open, click **Browse** and locate the file that you want to upload.
2. Click **Upload**.

The **Unit Price** column is updated.

**Note:** You can only update prices for codes that currently exist. If you try to upload prices for codes that do not exist, an error message appears and none of the unit prices are updated.

## Updating Expense Unit Prices Manually

You can manually update expense unit prices on the Expense Unit Prices page.

### To update expense unit prices manually

1. In the **Unit Price** column, modify the appropriate prices.
2. Click **Update** to save the changes.

## 1.7.3 Invoices

Invoices in TeamConnect represent bills sent to your organization by outside counsel and other vendors who provide work on your projects and other services. Invoices can be posted against accounts and projects, according to the posting criteria you set for each individual account.

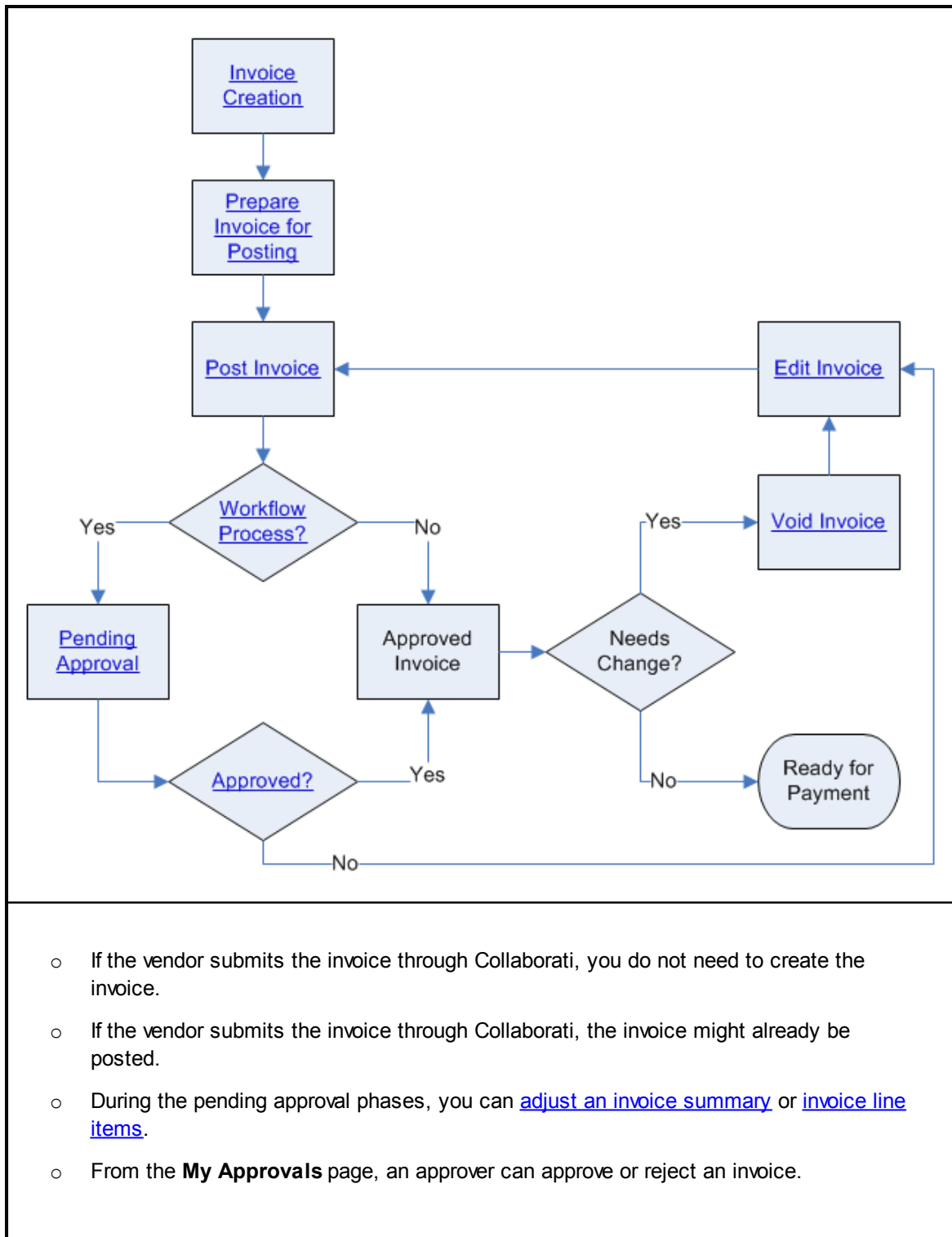
**Important:** Certain invoice fields and functionality depend on a combination of system settings and your user rights. As a result, your invoices may look different from the descriptions in this section and some features described may not be available. Contact your TeamConnect administrator if you have questions.

Two ways for invoices to appear in TeamConnect exist:

- Manually created directly in TeamConnect
- Uploaded from Collaborati, if you subscribed to Collaborati Spend Management service

Invoices are transferred from another system as per custom integration your organization may have in place (e.g. with a 3rd party e-billing system)

During the invoice life cycle, multiple users inside and outside your organization can perform an action for the same invoice. The following flowchart illustrates this life cycle. **Click areas of the flowchart for more information.**



## Invoice Record Pages

There are four invoice-specific pages:

- The **General** page lets you enter the following information about the invoice:
  - **Header Information**—Specifies general information about the invoice, such as a **Description, Date, Vendor, Billing Start Date, Billing End Date**.
  - **Tax Information**—Specifies tax types, levels, and rates.
  - **Invoice Summary**—Displays totals for invoice expense line items, fee line items, and the invoice total (all line items).
  - **Line Items**—Individual expenses or fees that are being billed on the invoice.
- The **Summary Breakdowns** page lets you view a summary of line items grouped together by Timekeeper or Category, and organized by matter. If the invoice is associated with more than one matter, an **All Matters** section also appears on this page, showing all of the associated matters grouped by Timekeeper or Category.
- The **Transactions** page displays a history of the invoice's posts and voids.
- The **Versions** page is read-only, and displays links to prior versions of this invoice, if more than one version of an invoice exists. This page only applies to previously rejected invoices that were resubmitted.

**Header Information**

<p><b>*Invoice Number:</b> <input type="text" value="inv0626_01"/></p> <p><b>*Vendor:</b> <input type="text" value="The Bakery"/> </p> <p><b>*Invoice Date:</b> <input type="text" value="6/26/2008"/> </p> <p>Received Date: <input type="text"/></p> <p>Project: <input type="text"/></p> <p>Posting Status: Not Posted</p>	<p>Billing Start Date: <input type="text"/> </p> <p>Billing End Date: <input type="text"/> </p> <p>Submitted Total (USD): <input type="text" value="\$0.00"/></p> <p>Submitted Currency: <input type="text" value="United States, Dollars"/> </p> <p>Tax Rate: <input type="text" value="0.00"/> %</p> <p><b>Line Item Warnings:</b> None</p>
---	---

**Invoice Summary**

Type	Rate x Unit	Discount	Adjustment	Tax	Amount
Fees	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Expenses	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
<b>Invoice Total (USD)</b>	<b>\$0.00</b>	<b>\$0.00</b>	<b>\$0.00</b>	<b>\$0.00</b>	<b>\$0.00</b>

**Description**

Invoice - General Page (Header Information, Invoice Summary, Description Sections)

**Comments to Requestor**

**Line Items**

Date:

Item Type:

Category:

Activity:

Project:

Timekeeper:

Description:

	Rate	Unit	Discount	Amount
Original	<input type="text" value="0.00"/>	<input type="text" value="0.00"/>	<input type="text" value="0.00"/>	<input type="text" value="0.00"/>
Current	<input type="text" value="0.00"/>	<input type="text" value="0.00"/>	<input type="text" value="0.00"/>	<input type="text" value="0.00"/>
<b>Adjustment Total(USD):</b>				<b>\$0.00</b>

Reason for Adjustment:

Comments to Requestor:

In House Comments:

Line Item 1 - 1 of 1

<input type="checkbox"/>	Item	Date	Item Type	Category	Timekeeper	Original Rate	Original Unit	Original Discount	Original Amount	Current Rate	Current Unit	Current Discount	Current Amount	Action
<input type="checkbox"/>	1	6/30/08	Expense	Expense		\$30.00	3.00	\$0.00	\$90.00	\$30.00	3.00	\$20.00	\$70.00	

Project: [car\\_collision0613\\_01](#) [More Details](#)

Activity:

Description:

Line Items per page

**Invoice - General Page (Comments to Requestor, Line Items Sections) Continued**

### 1.7.3.1 Viewing Invoices

You can view invoices by browsing invoice collections. Alternatively, you can search for an invoice. See [Searching for Specific Records](#) for more information.

When viewing an invoice, the **Line Items** section displays fields that are different from the **Line Items** section in edit-mode. The fields from the **Invoice Information** and **Invoice Summary** sections are the same as described in the [Creating an Invoice Manually](#), with the exception of the following fields that may appear under certain circumstances in the **Header Information** section:

- **Appeal Invoice Number**—Displays the Invoice Number of the Appeal Invoice associated with the current invoice. Clicking the **Appeal Invoice Number** opens the Appeal Invoice.  
  
This field only appears if an Appeal Invoice based on the current invoice has been created and synced to TeamConnect.
- **Original Invoice Printable View**—Displays the Invoice Number of the original invoice associated with the current Appeal Invoice. This field only appears if the current invoice is of type Appeal. It displays the Invoice Number of the invoice the Appeal Invoice is based on.

Clicking the **Original Invoice Printable View** link launches the printable view pop-up for the original invoice. The **Appealed Line Items** check-box appears directly below the **Line Items** check-box of the printable view pop-up. When selected, the printable view will only return line items from the original invoice that have been appealed in the new Appeal Invoice.

Select the desired **Sections** check-boxes, then click **Ok** to open the original invoice's printable view in a new browser window.



**Note:** For sections with a search results display, the default setting is **Display all result pages**.

There may be several types of invoices, for example Non-US Tax Type.

### To view an invoice

1. Click the **Finance** tab.
2. Click the **Invoices** link under the tab bar.

The recently viewed invoices are listed in the right pane.

For more information, see [the Invoice, General Page, Line Items Section fields table](#).

A summary for each invoice in a collection displays in a list pane to the right.

Invoices of type Appeal have a flag that identifies the invoice as an Appeal Invoice. This is intended to allow workflows to be configured differently for Appeal Invoices as opposed to other types of invoices.

3. To open an invoice, in the right pane, click an invoice name link.

For information about the **Invoice Header** section, see [Invoice Header Information Section table](#).

When viewing an invoice, the **Line Items** section displays fields that are different from the invoice when creating or editing an invoice, or adjusting the line item. For more information, see [the Invoice, General Page, Line Items Section fields table](#).

For more information about the areas on the Invoices List page, see [User Interface](#).

#### Invoice, General Page, Line Items Section Field Descriptions (read-only)

Field	Description
<b>Adjust</b>	(Batch adjustment or In-Line & Bulk adjustment only) Click to adjust one or more invoice line item.
<b>Show Filter/Hide Filter</b>	Click the link or green arrow to toggle between displaying/hiding the invoice line items filter conditions.
<b>View</b>	From the drop-down list, select one of the following: <ul style="list-style-type: none"> <li>• <b>All Line Items</b> (default)</li> <li>• <b>Line Item Warnings</b></li> </ul>
The following fields apply per invoice line item row	
<b>Action</b>	A <b>Copy</b> icon allows copying the line item. An <b>Edit</b> icon allows editing the line item.

<b>Additional information</b>	If available, the following information displays: <ul style="list-style-type: none"> <li>• Project</li> <li>• Activity</li> <li>• Description</li> </ul>
<b>Category</b>	Displays the category associated with the Line Item type (Task/Fee Category or Expense Category).
<b>Date</b>	Displays the date on which the services were provided or expenses were incurred.
<b>Item</b>	Click a link to view the individual line item record. Message icons indicate associated warnings, errors, or information messages. See the <b>Messages</b> field description below.
<b>Item Type</b>	Displays either <b>Fee</b> or <b>Expense</b> for the line item type.
<b>Original Amount</b>	This amount is automatically calculated when you add the line item. <b>Amount = (Rate x Units) - Discount</b> . Displays the current total for a Line Item.
<b>Original Discount</b>	Displays the discount amount. The discount will display as a flat amount.
<b>Original Rate</b>	The hourly rate for a billed task or per unit cost for an expense.
<b>Original Unit</b>	Displays the number of hours for fee items or units for expense items.
<b>Tax Type</b>	Displays the tax code(s), if any, that were selected for the line item. This field only appears if the invoice has a Non-US tax type.
<b>Timekeeper</b>	The Timekeeper or contact who performed the services.

#### 1.7.3.1.1 Searching for Invoice Line Items

You can filter line items that display for an invoice. For example, you can display line items associated with a specific timekeeper or task code.

This is especially useful when reviewing large electronic invoices that have hundreds of line items. Two search views for line items are available by default.

For invoices with a large number of line items, you can select search criteria from a list unless the number of items in the list exceeds the maximum number of items allowed as specified in the Admin Settings. In this case, a search box is provided that lets you search across all values in the database.

### To search for invoice line items

1. Search for an invoice and open it.




The **General** page opens.

2. In the **Line Items** section, select a search view from the **View** drop-down list:

- **All Line Items**—Displays all Line Items for the invoice (default selection).
- **Line Item Warnings**—Displays only Line Items that have warnings (produced by invoice validation rules, as specified by your TeamConnect administrator).

The list might include additional search views.

3. To apply additional search criteria, click the **Show Filter** link.

- Enter additional Line Items search criteria from the displayed fields. For example, if the search view is **All Line Items**, you can enter the following:
  - To display only Fee Line Items, in the **Item Type** row, leave the first drop-down list option as **Equal To**. In the second drop-down list, select **Fee**.
  - To display only Expense Line Items, in the **Item Type** row, leave the first drop-down list option as **Equal To**. In the second drop-down list, select **Expense**.
  - To display only Fee line items with a specific task code (category), in the **Task Code** row, leave the first drop-down list option as **Equal To**. In the second drop-down list, select a category.
  - To display only Fee line items with a specific activity code (category), in the **Activity Item** row, leave the first drop-down list option as **Equal To**. In the second drop-down list, select a category.
  - To display only Line Items for a specific project, in the **Project Name** row, leave the first drop-down list option as **Equal To**. In the second drop-down list, select the project record type (for example, Dispute). In the third field, search for the project record name.
  - To display only Line Items with specific Warning message content, in the **Warning Description** field, type the word or text to search.
- If the Search icon is next to the search box, you must enter an existing value.
- You can enter multiple values for fields with the Add  icon.
  - After you enter a value, click the Add  icon to enter another value.
  - To remove a value from the search, click the Remove  icon next to the value. To remove the first value you entered, select the text with the mouse and delete it.

4. Click **Search**.

## 1.7.3.1.2 Appeals Invoices

Enabling the Appeals Invoices functionality allows vendors recourse when a submitted invoice is not entirely approved as submitted in CSM. This setting can be enabled on the **General** tab of the **CSM Settings** page.

### ▣ Appeals Invoices Setup and Details

For the appeals to work as designed, TeamConnect users at a given stop must be able to edit invoices before posting.

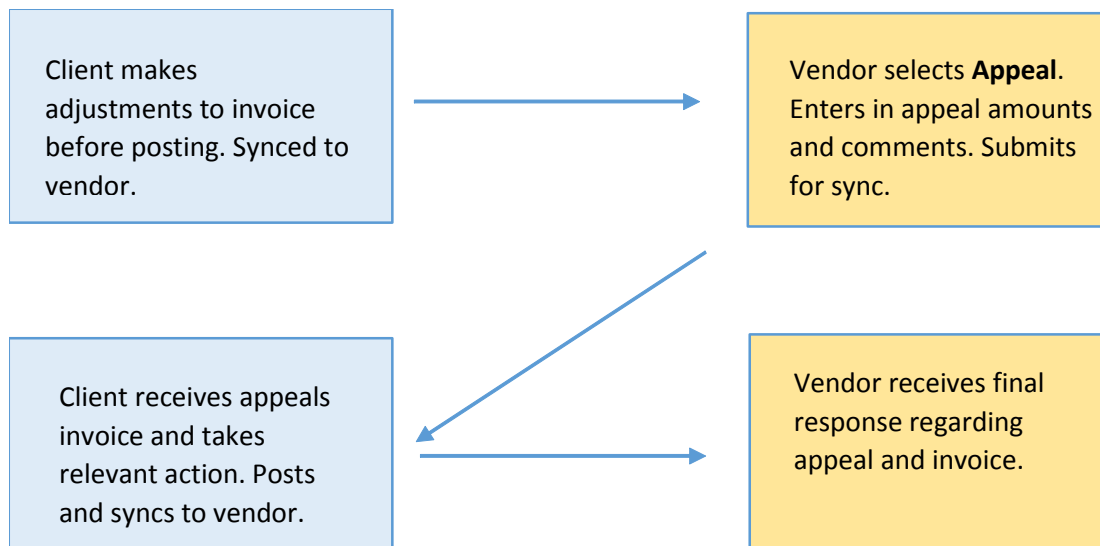
Vendors are only able to submit one appeal per invoice. The appeal will only include line items where the original amount was not approved. See the table below for examples.

Original Invoice Amount	Amount Approved	Amount Visible on Appeals Invoice
\$200.00	\$150.00	\$50.00
\$100.00	\$80.00	\$20.00
\$50.00	\$50.00	N/A, will not show

Vendors are also able to provide a comment for each line item adjustment detailing the reason behind the appeal.

### ▣ Appeals Invoice Workflow

The flowchart below demonstrates an appeal invoice workflow.



### Client-side Process

1. Enable Appeals Invoices on the **General** tab of the **CSM Settings** page.
2. Upon receipt of an invoice from Collaborati, make adjustments to line items *before posting*, if desired.
3. Submit and sync.
4. If vendor appeals, an appeals invoice will generate. This invoice will have the same number as the original invoice along with "Appeal" added to the end of it.



5. After viewing the adjustments and comments made by the vendor, make the desired edits, save, and submit the appeals invoice. Sync with Collaborati. **Note:** *Only the items that were not fully approved on the initial invoice will be included. For an example, see the table in [Setup and Details](#).*

After this submission, the vendor will not be able to send another appeal invoice for that same original invoice.



### Configuration Tips

The following best practices will assist in setting up appeals invoices:

- Configure object views to place appealed line items block directly under the line items block for easy comparison.
- Enable the approval rules to allow users to edit items before posting.

However, for optimized configuration, contact Mitratech Services at [professionalservices@mitratech.com](mailto:professionalservices@mitratech.com).

### 1.7.3.2 Differences Among Invoices

Fields that display on invoice pages may vary based on the following scenarios:

#### Line Item Adjustment Types

Depending on TeamConnect system settings, when you adjust invoice line items, either the [Detail Adjustment format](#), the [Batch Adjustment format](#), the [In-Line & Bulk Adjustment format](#), or the [Bulk Adjustment format](#) are available. This system setting affects invoice fields that display on the **General** page, **Line Items** section when adjusting line items.

The ways to tell which line item adjustment setting applies to your system are:

- On an existing invoice's **General** page, if an **Adjust** link displays near the **Line Items** section bar, then the setting is set for Batch Adjustment or In-Line & Bulk Adjustment.

Line Items [Edit](#) [Adjust](#)

View:  [Show Search Options](#)

Line Items 1 - 2 of 2

Item	Type	Date	TK	Task/Exp	Act	Hrs/Units	Rate	Disc	Adj	Amount
1	Expense	06/07/2008		Photocopy		3.00	30.00	0.00	0.00	90.00
	<b>Project:</b> <b>Activity:</b> <b>Description:</b>									<a href="#">More Details</a>
2	Fee	06/09/2008		Research	Default	5.00	25.00	0.00	0.00	125.00
	<b>Project:</b> <b>Activity:</b> Default <b>Description:</b>									<a href="#">More Details</a>

Line Items per page

#### Invoice General Page

- On an existing invoice's **General** page, if the line item numeric values (**Rate**, **Unit**, **Amount**) display as links, then the setting is set for Detail Adjustment.

Line Items [Edit](#)

View:  [Show Search Options](#)

Line Items 1 - 2 of 2

Item	Type	Date	TK	Task/Exp	Act	Hrs/Units	Rate	Disc	Adj	Amount
<a href="#">1</a>	Expense	06/07/2008		Photocopy		<a href="#">3.00</a>	<a href="#">30.00</a>	<a href="#">0.00</a>	<a href="#">0.00</a>	<a href="#">90.00</a>
	<b>Project:</b> <b>Activity:</b> <b>Description:</b>									<a href="#">More Details</a>
<a href="#">2</a>	Fee	06/09/2008		Research	Default	<a href="#">5.00</a>	<a href="#">25.00</a>	<a href="#">0.00</a>	<a href="#">0.00</a>	<a href="#">125.00</a>
	<b>Project:</b> <b>Activity:</b> Default <b>Description:</b>									<a href="#">More Details</a>

Line Items per page

Invoice General Page, Line Items Section with Numeric Value Links for Detail Adjustment

## Invoice Duplicates

Based on system settings TeamConnect can check for duplicate invoices in the system to prevent creation of duplicates. For more information, see [Invoice Duplicates](#).

## Invoice Versions

If a previously entered invoice was rejected in TeamConnect, when the invoice is re-entered, it is not considered a duplicate. Resubmitting an invoice creates a new version of the original invoice. For more information, see [Invoice Versions](#) and [Viewing Previous Invoice Versions](#).

## Invoices Submitted in Different Currencies

If an invoice is submitted in a currency that is different from your default system currency, you can view that invoice's values in either your system currency or the submitted currency. For more information, see [Viewing Invoices Submitted in Different Currencies](#).

## Invoice Workflow

The invoice fields that you see and the functionality that is available to you depend on the following:

- Whether Workflow approval is required. See either [Invoice Procedures \(without Workflow\)](#) or [Invoice Procedures \(with Workflow\)](#) for the list of tasks you can perform depending on whether workflow is required.
- Whether [the invoice sender uses a different currency](#) from the adjustor.
- Whether a solution developer created invoice or line item customizations. Contact the TeamConnect administrator for more information.

#### 1.7.3.2.1 Invoice Duplicates

Depending on system settings, TeamConnect may prevent you from creating duplicate invoices. A duplicate invoice is an invoice that has identical values for the following fields as an existing invoice:

- Invoice number
- Vendor ID
- Invoice Date that falls within 12 months of the existing invoice
- Invoice Type

If a new invoice is detected to be a duplicate, the system will display an error message. Contact the TeamConnect administrator for questions about saving duplicate invoices.

#### 1.7.3.2.2 Invoice Versions

An invoice can have multiple versions stored in TeamConnect. When an invoice submitted for posting has been rejected by an approver in an invoice workflow, and the rejected invoice is later modified and submitted again for approval, the invoice will be saved in TeamConnect as a newer version. Different invoice versions will be differentiated with a version number.

### Points to Remember

- You will be able to view the previously rejected invoice and its workflow history while reviewing the re-submitted invoice.
- On an invoice search view results list, only the latest version of an invoice with a rejection history displays. To view previous versions of the invoice, you must open the latest version and then select the **Versions** page. Click the link of the invoice version to view it.
- On an invoice's **Versions** page, the originally rejected invoice displays with a version number 1.
- Prior versions of invoices are read-only and cannot be adjusted or updated. Only the current invoice version can be adjusted, as long as the invoice is in the appropriate state and you have appropriate adjustment rights. See [Adjusting an Invoice Summary](#) or [Adjusting Invoice Line Items](#) for more information on adjusting your invoice.
- On an invoice's **Versions** page (**Invoice Version** field), the version number displays in parentheses in the **Invoice No.** field. The **Versions** page is read-only and displays links to previously rejected invoice(s), or related invoice versions.
- If the invoice has been rejected and resubmitted multiple times, you will see all previously rejected invoice versions under the **Invoice Version** column.

#### 1.7.3.2.2.1 Viewing Previous Invoice Versions

On the **Versions** tab (**Invoice Version** field) and on a previously rejected invoice's **General** tab, the version number displays in parentheses in the **Invoice No.** field. The **Versions** tab is read-only and displays links to previously rejected invoice(s), or related invoice versions.

On this page, you can click the **Invoice Version** link to view the versioned invoice.

If the invoice is rejected and resubmitted multiple times, you will see all previously rejected invoice versions in the **Invoice Version** column.



### To view previous invoice versions (workflow)

1. Search for an invoice and open it.
2. Click the **Versions** link in the left pane.
3. To view a previous version of an invoice, click its link in the **Invoice Version** column.

#### 1.7.3.2.3 Viewing Invoices Submitted in Different Currencies

If an invoice has been submitted in a currency other than your system currency, by default, the invoice's values will display in the submitted currency. You can change the invoice to display in the system TeamConnect currency. The rate value & the conversion date will be additionally displayed under the Exchange Rate field.

### Tips and Warnings

- If the invoice display has been updated to show the system TeamConnect currency, when an approver adjusts the invoice, values will display and be calculated against the originally submitted currency values.
- If the invoice has not posted and you change the view to a different currency, the current exchange rate will apply, as specified in your multi-currency table. Contact your Administrator for details.
- If the invoice has posted and you change the view to a different currency, the conversion rate will be based on the exchange rate at the time of posting.

### To view an invoice submitted in a different currency

1. Search for an invoice and open it.
2. Click **Edit**.
3. From the **Submitted Currency** drop-down list, select your TeamConnect system currency.
4. Afterward the invoice values are converted from the submitted currency to your system currency.

**Caution:** *If you adjust the invoice Summary or Line Items, adjustment values display and are calculated in the submitted currency values.*

### Example

For example, the invoice was submitted in Pounds (Great Britain). The approver changes the invoice's currency view to US Dollars (USD). All invoice currency values will display in the equivalent USD values.

If the approver adjusts the invoice, the adjustment screen will display values in the originally submitted currency (GBP). Adjustments will be applied in GBP as well.

After the approver makes adjustments and clicks OK, the Invoice General Tab view will revert to displayed values in USD (as selected in the View in drop-down list).

#### 1.7.3.2.4 Invoice Procedures (w without Workflow )

The tasks below are related to working with invoices when no workflow process is required.

- [Searching for Specific Records](#)
- [Viewing Invoices](#)
- [Creating an Invoice Manually](#)
- [Editing Invoices](#)
- [Adjusting Invoice Line Items](#)
- [Posting an Invoice](#)
- [Voiding an Invoice](#)
- [Deleting Records](#)

When there is no invoice workflow process, the general life cycle of an invoice is like:

1. A new invoice has the status, **Not Posted**.
2. After the invoice information is entered, including line items and related project/account information, you can post the invoice. Afterward, if all line items post successfully, the invoice status displays **Posted**. If all line items do not post successfully, the invoice status displays **Failed**.
3. After an invoice has posted, you can click **Void** to void the invoice. Afterward, the invoice status displays **Not Posted** again.

#### 1.7.3.2.5 Invoice Procedures (w ith Workflow )

When an approval workflow is in place for invoices, there are typically at least 4 roles for invoice users (creator, adjustor/approver, reviewer, process manager). The tasks below are related to working with invoices when a workflow process is required.

##### **Invoice Creator**

- [Searching for Specific Records](#)
- [Viewing Invoices](#)
- [Creating an Invoice Manually](#)
- [Editing Invoices](#)
- [Posting an Invoice](#)

##### **Adjustor and/or Approver**

- [Searching for Specific Records](#)

- [Viewing Invoices](#)
- [Adjusting an Invoice Summary](#)
- [Adjusting Invoice Line Items](#)
- [Approving Invoices](#)
- [Rejecting Invoices](#)
- [Voiding an Invoice](#)

## Reviewer

- [Viewing Invoices](#)
- [Reviewing an Invoice](#)

## Process Manager

- [Managing Workflow Processes](#)

When there is an invoice workflow process (for example, approval is required for posting an invoice), the general life cycle of an invoice is like:

1. A new invoice has the status, **Not Posted**.
2. After the invoice creator has entered invoice information, including line items and related project/account information, the creator can click Post. Afterward, the invoice status displays **Pending Approval**.  
  
While an invoice is pending approval, an adjustor can adjust the invoice summary or the invoice line items.
3. Invoice approvers can do the following:
  - Approve the invoice to post. Afterward, if all line items post successfully, the invoice status displays **Posted**. If all line items do not post successfully, the invoice status displays **Failed**.
  - Reject the invoice. Afterward, the invoice status displays **Rejected**. Then the invoice creator can edit the invoice and submit it to post again.
  - Send the invoice to another user for review. The invoice status continues to display **Pending Approval**.
4. After an invoice has posted, you can click **Void** to void the invoice. Afterward, the invoice status displays **Not Posted** again.

### 1.7.3.3 Creating an Invoice Manually

Use the invoice **General** page to start manually creating an invoice record.

When creating invoices, you can add general information as well as add invoice line items. You can also adjust line items (for scenarios when you are entering a paper invoice with line item adjustments into TeamConnect).

You can also create new invoice records by copying similar existing records. See [Copying a Record](#) for more information.

Your system administrator may provide one or more wizards to help you create new invoices using a guided, step-by-step process. A wizard may be one of the available options or the only option.

### To create an invoice using a wizard

**Note:** *If a wizard is your only option, and you want to use an existing invoice record to create a new invoice record, a wizard will not open. A blank invoice record opens and you can complete the fields manually. See [Editing Invoices](#) for more information.*

1. Click the **Finance** tab.
2. Click the **Invoices** link under the tab bar.
3. Click the **New**, and then select **New Invoice Wizard** from the drop-down list.

The **New Invoice** wizard opens to the **General** page.

4. Complete all of the required fields and any other fields for which you have information. See [Invoice General Page Field Descriptions](#) for more information.
5. Click **Next** to continue to the next page, or click the appropriate link in the left pane.

If a **Finish** button appears on any page, click it if you want to save the matter with only the information that is currently entered. This is useful if you do not have complete matter details or if you only need to generate a matter ID. Once the matter is saved, you can add additional information whenever you receive it.

If you need to make changes on a previous page of the wizard, click the **Back** button or the appropriate page in the left pane; if you want to cancel the wizard, click **Cancel**.

The **Line Items** page opens.

6. For each line item, complete all required fields and any other fields for which you have information. See [Invoice General Page Line Items Section Field Descriptions](#) for more information.
7. To add more line items, click **Add**, and then repeat the procedure above.
8. Click **Next**.

The invoice saves and its **General** page displays.

### To create an invoice manually

1. Click the **Finance** tab.
2. Click the **Invoices** link under the tab bar.
3. Click **New**.
4. On the **General** page, the following fields from **Invoice Header** section are required:
  - **Invoice Number**

- **Vendor**—Click the Search Contacts icon to select a vendor contact.
  - **Invoice Date**—Click the Calendar icon next to this field to select the date when the invoice was issued.
5. For additional information about **Invoice Header** section fields, see [Invoice General Page Field Descriptions](#).
  6. The **Invoice Summary** section is read-only during invoice creation. After adding invoice line items, the **Invoice Summary** section automatically updates with total summaries for line items.
  7. Complete the **Description** and **Comments to Requestor** fields if necessary.
  8. To enter invoice line items, see [Adding Invoice Line Items](#).
  9. [Select a save option](#).

## Invoice General Page Field Descriptions

Some field descriptions that follow may or may not display based on system settings and your assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

### Invoice Header Information Section

Field (* = required)	Description
<b>Billing End Date</b>	Click the <b>Calendar</b> icon next to the field to select the end date for the invoice period.
<b>Billing Start Date</b>	Click the <b>Calendar</b> icon next to the field to select the start date for the invoice period.
<b>Display Currency</b>	The type of currency in which you want to use to view this invoice.  This field only appears if the <b>Submitted Currency</b> is different from the default system currency.
<b>Exchange Rate</b>	The exchange rate that applies to this invoice.  This field only appears if the <b>Submitted Currency</b> is different from the <b>Display Currency</b> .
<b>*Invoice Date</b>	Displays the invoice date, or if in edit mode, click the <b>Calendar</b> icon next to this field to select the date when the invoice was issued.
<b>*Invoice Number</b>	Displays the invoice number, or if in edit mode, allows you to enter the invoice number.

<b>Invoice Type</b>	<p>The type of invoice being created or edited.</p> <p>In addition to the typical <b>Standard</b> Invoice, additional options are as follows:</p> <ul style="list-style-type: none"> <li>• <b>Credit Note</b>—A legal instrument that corrects a Non-US Tax invoice.</li> <li>• <b>Accrual</b>—An estimate of an upcoming bill.</li> <li>• <b>Shadow Invoice</b>—A detailed invoice of work performed that is not intended to be actually paid, because an alternate fee arrangement exists between client and vendor. The invoice is posted to allow the client to do financial analysis. For example, if vendors receive a fixed-fee or retainer fee as payment, clients use shadow invoices to keep a record of their work and perform financial analysis. See <a href="#">Creating a Shadow Invoice Account</a> for more information.</li> </ul> <p><i>Note: When editing an invoice, this field only appears for Accrual or Credit Note invoices.</i></p>
<b>Line Item Warnings</b>	<p>Displays the number of line items with warnings based on your validation rules. If there are no rules activated or there are no warnings, no values display.</p>
<b>Posting Status</b>	<p>The posting status always displays in read-only mode.</p> <p>The posting status for an invoice can be one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Not Posted</b>—The invoice has not been posted or it has been voided. Displays as the default status for new invoices.</li> <li>• <b>Rejected</b>—The invoice has been rejected during workflow approval.</li> <li>• <b>Posted</b>—The invoice is posted.</li> <li>• <b>Failed</b>—The invoice could not post because of an issue.</li> </ul> <p>For example, the related account did not exist or the invoice total exceeded the allocated budget. For more details, see <a href="#">Posting an Invoice</a> and <a href="#">Voiding an Invoice</a>.</p>
<b>Project</b>	<p>Displays one of the following:</p> <ul style="list-style-type: none"> <li>• If this invoice is allocated to only one project, displays a link to the project record in the upper-right corner.</li> <li>• If this invoice is allocated towards multiple projects, displays Multiple Projects in upper-right corner (non-clickable). In this scenario, projects are displayed in the first column of the line items.</li> </ul>

<b>Received Date</b>	Displays the date the invoice was received, or if in edit mode, allows you to click the <b>Calendar</b> icon next to this field to select the date when the invoice was received.
<b>Reference Number</b>	The reference number for an invoice that is also a Credit Note. Many invoices may use the same reference number.  This field only appears if the invoice type is Credit Notes.
<b>Submitted Currency</b>	This field is available when the invoice was submitted in a currency other than the default system currency. Select a currency to apply to the invoice values.  <i>Note: The invoice can be adjusted only in the submitted currency.</i>
<b>Submitted Total</b>	(optional) Enter the total amount of the invoice. If this value does not match the actual line items total, you may not be able to save the invoice.  <i>Note: For CSM invoices, this field is automatically populated by the invoice total.</i>
<b>*Vendor</b>	Displays the vendor name, or if in edit mode, allows you to find and select the contact record of the vendor who sent the invoice.

**Tax Information Section**

Field		Description
<b>Tax Type</b>		Specify the type of taxes the invoice contains with one of the following options: <ul style="list-style-type: none"> <li>• <b>US</b>—Indicates the invoice does not contain foreign taxes.</li> <li>• <b>Non-US</b>—Indicates the invoice contains foreign taxes or any additional type of tax.</li> </ul> Save the invoice after you change the <b>Tax Type</b> .
<b>For a US tax type</b>	<b>Tax Rate</b>	Enter a numeric value for the percentage to be applied to line items. For example, enter 8.25 for an 8.25% tax.
<b>For a Non-US tax type</b>	<b>Tax Level</b>	Specify how to apply non-US taxes to the invoice with one of the following options: <ul style="list-style-type: none"> <li>• <b>Summary Level</b>—Allows you to enter taxes for the invoice as a whole. To add summary level taxes, save</li> </ul>

		<p>the invoice with at least one line item associated with a project.</p> <ul style="list-style-type: none"> <li>• <b>Line Item Level</b>—Allows you to enter taxes for individual line items.</li> </ul> <p>Save the invoice after you change the <b>Tax Level</b>.</p> <p><b>Note:</b> <i>You only have the <b>Tax Level</b> option if line item tax details are enabled for your system. To enable line item tax details for Collaborati invoices, contact Mitrtech Support. To enable line item tax details for manual invoices, contact your system administrator.</i></p>
	<b>Non-US Tax Code</b>	Select a tax code. If you want to select a tax code that is not available, ask your system administrator to add it to the system.
	<b>Project (at the Summary Level)</b>	Select a project. Only projects associated with line items in the invoice appear in this field.
	<b>Amount (at the Summary Level)</b>	Enter the tax amount.
	<b>Tax Rate (at the Line Item Level)</b>	Enter a numeric value for the percentage to be applied to line items. You can enter more than one <b>Tax Rate</b> , which you can apply to different line items.
	<b>Action</b>	Click <input type="button" value="+"/> to add another tax code to the table, and click <input type="button" value="-"/> to remove a tax code from the invoice.

#### Invoice Summary Section

Field	Description
<b>Type</b>	The type of invoice. Options are <b>Fees</b> or <b>Expenses</b> .
<b>Rate x Units</b>	The total cost of the rate multiplied by the number of units.
<b>Discount</b>	The discount if applicable.
<b>Adjustment</b>	The adjustment based on the discount.



<b>Tax</b>	The tax on the amount if applicable.
<b>Amount</b>	The total based on the discount, adjustment, and tax. Click on this link to make an adjustment.

#### 1.7.3.3.1 Timekeeper Rates

When adding a fee line item to an invoice, the name in the timekeeper field corresponds to a contact record. If a rate for the contact is specified under **Invoice Tasks Rate** on the contact's **Rates** page, this rate will automatically populate the invoice line item's **Original Rate** field. For more information, see [Setting Contact Rates](#).

The following conditions apply:

1. If the Contact record > **Rates** Page > **Default Rates** section is specified and the **Invoice Task Rates** section is blank, then the following applies to invoice fee line item rates:
  - In an invoice Line Items section, if the following are true:
    - Item Type** is **Fee**.
    - Timekeeper** is the contact with set rate.
  - Then the **Original Rate** field should be automatically populated with the **Default Rate Value** (from Contact record > **Rates** page).
2. If the Contact record > **Rates** Page > **Invoice Task Rates** section specifies a rate, then the following applies to invoice fee line item rates:
  - In an invoice **Line Items** section, if the following are true:
    - Item Type** is **Fee**.
    - Line Item **Category** matches the **Task** value (from Contact record > **Rates** page > **Invoice Task Rates** row).
    - Timekeeper** is the contact with set Invoice Task Rate.
    - Service date** falls within the **From-To** date range (from Contact > **Rate** page > **Invoice Task Rates** row).
  - Then the **Original Rate** field should be automatically populated with the Invoice Task Rate (from the contact's **Rate** page).

If the **Default Rate Value** and an **Invoice Task Rate** are both specified, the **Invoice Task Rate** will override the **Default Rate Value**.

#### 1.7.3.3.2 Adding Invoice Line Items

You can add invoice line items on an invoice's **General** page in the **Line Items** section. Line items can be added when creating invoices or editing invoices that have not been submitted for posting. After the invoice is saved, each invoice line item appears as a row in the Line Items section table. In addition, each invoice line item is stored as a separate record.

### When creating fee line items

- If the timekeeper's contact record has rates set for specific task categories on the **Rates** page, the appropriate rate is automatically displayed in the **Rate** field when you select a category from the **Category** drop-down list. Inactive categories, if any, are displayed but cannot be selected.
- If the timekeeper has no rate assigned for the task category of the line item, or the rate period does not fall within the service date, the default rate value of the contact is displayed instead.
- You can overwrite the automatically displayed rate if necessary.
- The currency of the timekeeper is disregarded. Only the rate amount from the timekeeper's contact record is displayed automatically, even if the contact's currency is not the same as the invoice.

### When creating expense line items

- Depending on system settings, the rate for expense line items may include 2-5 digits after the decimal points. This is useful for rates that use fractions of a cent, such as photocopying at \$0.055 per page.
- If you enter values in excess of the specified number of decimal places for expense line items, the value will be rounded. For example, if the system specifies 2 digits after the decimal place and you enter 0.0345, the value will be rounded to 0.03. If you enter 0.0379, it would be rounded to 0.04.
- If you enter values with less than the system setting for the number of decimal places, the value will be padded with zeros. For example, if the system specifies 5 digits after the decimal place and you enter a value such as 0.03, the value will be padded to 0.03000.

For information about adding line items by copying existing line items, see [Copying an Invoice Line Item](#).

**Note:** You can enter negative values in **Line Item** section fields.

### To add invoice line items


1. If it is not already open, open the invoice to which you want to add a new line item or, on the Invoice List page, click the Edit Record icon.
2. On the invoice's **General** page in the **Line Items** section, complete the appropriate fields. See [Invoice General Page Line Items Section Field Descriptions](#) for more information.
3. Click **Add** to add the line item to the invoice.
4. Repeat steps 2 and 3 to add additional line items.
5. [Select a save option](#).

### Invoice General Page Line Items Section Field Descriptions

Some field descriptions that follow may not display based on system settings and your assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

#### Invoice General Page Line Items Fields

Label	Description
<b>Activity</b>	(optional) If you selected <b>Fee</b> from the <b>Item Type</b> drop-down list, select the appropriate activity category from this drop-down list.
<b>Adjusted Amount</b>	This field is automatically calculated after a Line Item is adjusted. <b>Amount = (Rate) x (Hrs/Units) - Discount</b>
<b>Adjusted Discount</b>	This field automatically displays the Original <b>Discount</b> value. You can change the value to adjust the Line Item. For more information, see <a href="#">Adjusting Line Items with Discounts</a> .
<b>Adjusted Rate</b>	This field automatically displays the Original <b>Rate</b> value. You can change the value to adjust the Line Item rate.
<b>Adjusted Unit</b>	This field automatically displays the Original <b>Unit</b> value. You can change the value to adjust the Line Item hours/units.
<b>Adjustment Total (USD)</b>	(Batch Adjustment only) Displays an automatically calculated value. <b>Total Adjustment = Adjusted Amount - Original Amount</b>
<b>Amount</b>	Displays an automatically calculated value after you add a line item. <b>Amount = (Rate x Unit) - Discount</b>
<b>Category</b>	<ul style="list-style-type: none"> <li>If you selected <b>Expense</b> from the <b>Item Type</b> drop-down list, then select a corresponding Expense category from this drop-down list.</li> <li>If you selected <b>Fee</b> from the <b>Item Type</b> drop-down list, then select a corresponding Task category from this drop-down list. (Inactive categories, if any, are displayed but cannot be selected.)</li> </ul>
<b>Comments to Vendor</b>	Enter adjustment notes. You can enter up to 2000 characters, including spaces.  <b>Note:</b> For CSM invoices, this information will be sent to the vendor.

<b>Date</b>	<p>Click the Calendar icon or enter the date for the expense or services provided.</p> <p>This field may be pre-populated based on one of the following criteria:</p> <ul style="list-style-type: none"> <li>• If you are creating a new line item and other line items exist on this invoice, the <b>Date</b> field displays the date of the most recently created line item, even if it is blank.</li> <li>• If the invoice doesn't have existing line items, and the invoice has a <b>Billing Date</b>, the <b>Date</b> field is populated with the invoice's <b>Billing Start Date</b>.</li> <li>• If neither of the above criteria applies, the <b>Date</b> field displays the invoice date.</li> </ul> <p>You can delete and change a pre-populated date.</p>
<b>Description</b>	Additional information.
<b>Discount</b>	Enter a discount to be applied to the original price or rate for the line item. The discount will display as a flat amount.
<b>In House Comments</b>	Notes for internal staff and TeamConnect users. You can enter up to 2000 characters, including spaces.
<b>Item Type</b>	Select either <b>Fee</b> or <b>Expense</b> to describe the item type.
<b>Non-US Tax Code</b>	<p>Select a tax code from the drop-down list. The tax rate associated with the code applies to the line item. You specify tax rates in the <b>Header Information</b> of the invoice.</p> <p>Click the  button to include additional tax codes. You cannot include the same tax code twice.</p> <p><b>Note:</b> <i>You can only enter taxes for each line item if line item tax details are enabled for your system. To enable line item tax details for Collaborati invoices, contact Mitratach Support. To enable line item tax details for manual invoices, contact your system administrator.</i></p>
<b>Project</b>	The project associated with this line item.
<b>Rate</b>	<p>Enter the original rate for services rendered or unit price for an expense.</p> <p>Also see <a href="#">Timekeeper Rates</a>, <a href="#">When creating fee line items</a>, and <a href="#">When creating expense line items</a> for exceptions.</p>

<b>Reason for Adjustment</b>	If you adjusted the original Rate, Unit, or Amount, select a reason from the drop-down list.
<b>Taxable Item</b>	Select this check-box to indicate whether the tax rate information in the <b>Tax Information</b> section should apply to the line item.  <i>Note: This check-box only appears if the invoice has a US tax type.</i>
<b>Timekeeper</b>	Find and select the contact who performed the service for a Fee line item.  If this contact already has a rate defined, the screen refreshes and updates rate field with timekeeper's established rate.
<b>Unit</b>	Enter the number of hours of a service or the number of units for an expense.

#### 1.7.3.3.3 Copying an Invoice Line Item

Creating a new line using an existing line item as a template is recommended when the new line item is very similar to an existing one, and you will not have to change many of the values that are automatically populated.

#### To copy an invoice line item

1. Search for an invoice and open it.  
The **General** page opens.
2. In the **Line Items** section, **Action** column, click the Copy icon for the line item to copy.
3. Edit the line item as required.
4. Click **Add**.  
The line item is added to the **Line Items** table.
5. [Select a Save option](#).

#### 1.7.3.4 Editing Invoices

Invoice fields that you can edit vary. Contact the TeamConnect Administrator for more information.

#### Tips and Warnings

- **If the invoice status is Not Posted**—Based on your invoice update rights, all Invoice **Header Information** and Line Items section fields are editable and the **Post** button is available.

- **If an invoice has posted**—You can only edit custom fields if your system allows edits to custom fields after posting. Contact your system administrator to enable this setting. The invoice's system fields are not editable.
- **If the invoice is in a workflow process** (submitted for posting and pending approval)—Based on your invoice update rights, all Invoice **Header Information** section fields are editable, the **Invoice Summary** section **Amount** values display as adjustable links, and **Line Items** section fields are not editable.
- **If you upgraded from TeamConnect 2.3 or earlier**—For pre-existing invoices that did not previously require vendor contact information, the invoice vendor field will be populated with the "System user" contact. This is so that invoices will not be voided due to lack of required information.
- **For invoices that are received through Collaborati Spend Management (CSM)**—Only custom fields are editable in the **Invoice Header** section and **Line Items** section. You can also edit the **Comments to Requestor** field.
- **If a Line Item has been adjusted**—You can edit fields except the **Original Rate**, **Original Quantity**, **Original Discount**, and **Original Amount** (total). For invoices that were generated in e-billing tools, system fields are not editable.

#### To edit an invoice in TeamConnect

1. Search for an invoice and open it.
2. Click **Edit** on the toolbar or click the **Edit** link in the section that you want to update.
3. On the **General** page, edit fields in the **Header Information** section. For more information, see [Invoice Header Information Section](#).

***Note:** The **Invoice Summary** section is read-only during invoice editing. After editing invoice line items, the Invoice Summary section automatically updates with total summaries for line items.*

4. [Select a save option.](#)

To edit invoice line items, see [Editing Invoice Line Items](#).

To add invoice line items, see [Adding Invoice Line Items](#).

To delete invoice line items, see [Deleting Invoice Line Items](#).

#### 1.7.3.4.1 Changing the Displayed Currency on an Invoice

If an invoice is submitted in a currency other than your system currency, by default, the invoice's values display in the submitted currency. Matter Management lets you temporarily change the invoice to display all invoice amounts in a different currency.

For example, for an invoice submitted in Pounds (Great Britain), an approver changes the invoice's currency view to US Dollars (USD). As a result, all invoice currency values are displayed in the equivalent USD values. If the approver adjusts the invoice, the adjustment screen displays values in the originally submitted currency (GBP). All Adjustments are applied in GBP as well.

## Notes

- If the invoice display is updated to show a currency other than the original currency, when an approver adjusts the invoice, values are displayed and calculated against the originally submitted currency values.
- If the invoice has not posted and you change the view to a different currency, the current exchange rate applies, as specified in your multi-currency table. Contact your Administrator for details.
- If the invoice has posted and you change the view to a different currency, the conversion rate is based on the exchange rate at the time of posting.

### To change the displayed currency on an invoice

1. Search for an invoice and open it.
2. Click **Edit**.

In the **Submitted Currency** drop-down list, select the appropriate type of currency.

The invoice values are displayed in the selected currency.

**Caution:** *If you adjust the invoice Summary or Line Items, adjustment values display and are calculated in the submitted currency values.*

#### 1.7.3.4.2 Editing Invoice Line Items

This procedure describes editing invoice line items from an invoice record. While editing an invoice, you can update existing line items.

### To edit an invoice line item

1. Do one of the following:
  - Search for an invoice and open it.
  - Click the **Invoice** tab. On the **Invoice List** page, click the Edit Record icon for an invoice.
2. On the **General** page, **Line Items** section, **Action** column, click the Pencil icon for a line item.

**Line Items**

Date: 6/9/2008

Item Type: Fee

Category: Research

Activity: Default

Project:

Timekeeper:

Description:

	Rate	Unit	Discount	Amounts
Original	25.00	5.00	0.00	125.00
Current	25.00	5.00	0.00	125.00
<b>Adjustment Total(USD):</b>				<b>\$0.00</b>

Reason for Adjustment: (Select)

Comments to Vendor:

In House Comments:

Line Items 1 - 2 of 2

<input type="checkbox"/>	Item	Date	Item Type	Category	Timekeeper	Original Rate	Original Unit	Original Discount	Original Amount	Current Rate	Current Unit	Current Discount	Current Amount	Action
<input type="checkbox"/>	1	6/8/08	Expense	Photocopy		\$30.00	3.00	\$0.00	\$90.00	\$30.00	3.00	\$0.00	\$90.00	
	<b>Project:</b> <b>Activity:</b> <b>Description:</b>													
<input type="checkbox"/>	2	6/9/08	Fee	Research		\$25.00	5.00	\$0.00	\$125.00	\$25.00	5.00	\$0.00	\$125.00	
	<b>Project:</b> <b>Activity:</b> Default <b>Description:</b>													

Line Items per page 10

#### Invoice General Page, Line Items Section--Editing Line Item #2

- Afterward, you can edit that line item's fields under the **Line Items** section bar. See [Invoice General Page Line Items Section Field Descriptions](#) for more information.
- Click **OK**.
- [Select a Save option.](#)

#### 1.7.3.4.3 Deleting Invoice Line Items

When editing invoices, you may want to delete line items from the invoice.








#### To delete invoice line items

- Search for an invoice and open it.  
The **General** page opens.
- From the **Line Items** section, select the check-box of the line item to delete.
- Click **Remove**.
- [Select a Save option.](#)



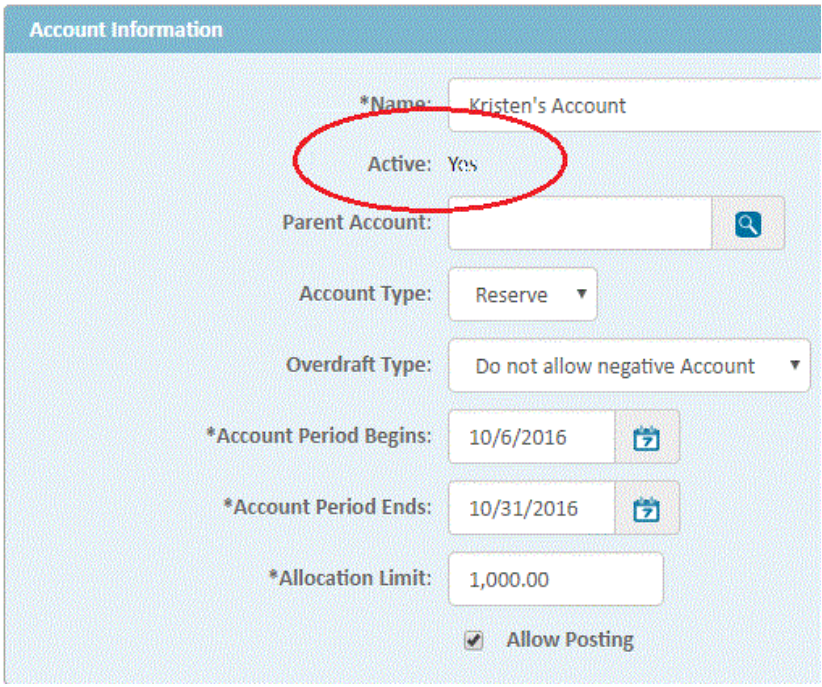
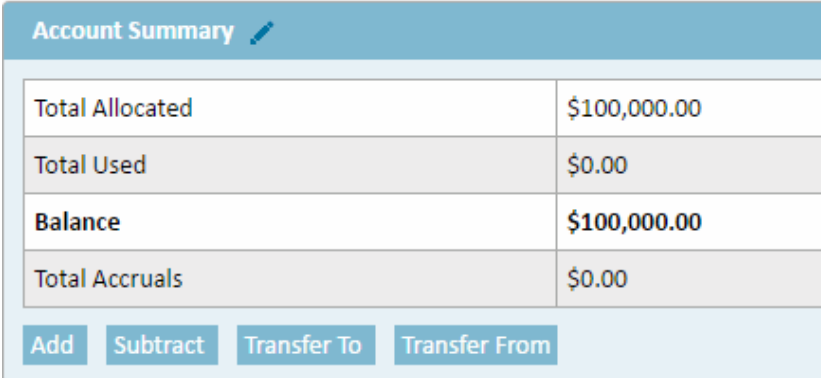
1.7.3.5 Preparing an Invoice for Posting

Before you post an invoice, you must prepare the invoice and its associated account. The following table describes requirements for posting.

Requirement	Description									
<p>The invoice must be associated with an account.</p>	<p>From <b>Accounts</b>, search for all accounts associated with the project listed on the invoice. This project can be a dispute or other type of matter. If you want to post an invoice to an account, it must be associated with a project.</p> <p>For example, enter DISP-000098 for the name value in the search. Select the account to which you want to post. For example, "Allocated - DISP-00098 - OC Expenses - 2013."</p> <table border="1" data-bbox="596 678 1422 797"> <thead> <tr> <th>Edit</th> <th>Name</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td></td> <td>Kristen's Account</td> <td>Account</td> </tr> <tr> <td></td> <td>Child Account 1</td> <td>Account</td> </tr> </tbody> </table> <p style="text-align: center;"><b>Selecting Accounts for Posting Invoices</b></p> <p>If no accounts exist for the project, create an account and specify the following fields on the <b>Posting Criteria</b> page.</p> <ul style="list-style-type: none"> <li>• <b>Posting Project to this Account:</b> Select <b>One</b></li> <li>• <b>This Project:</b> Enter the name of the project</li> </ul> <div data-bbox="644 1032 1422 1249" style="border: 1px solid #ccc; padding: 10px; background-color: #f0f8ff;"> <p>Posting Project to this Account: <input type="text" value="One"/></p> <p>This Project: <input type="text" value="DISP-000054-Kristen's Di"/> </p> <p>Post Projects of Type: <input type="text" value="A"/> (Select)</p> </div> <p style="text-align: center;"><b>Posting Projects to an Account</b></p>	Edit	Name	Type		Kristen's Account	Account		Child Account 1	Account
Edit	Name	Type								
	Kristen's Account	Account								
	Child Account 1	Account								
<p>The account must allow posting.</p>	<p>From the <b>General</b> page of the account, place a checkmark in the <b>Allow Posting</b> check-box if one does not exist.</p> <div data-bbox="596 1406 884 1503" style="border: 1px solid #ccc; padding: 5px; background-color: #f0f8ff;"> <p><input checked="" type="checkbox"/> <b>Allow Posting</b></p> </div> <p style="text-align: center;"><b>Allowing Posting to an Account</b></p>									
<p>The account must allow invoices to be posted.</p>	<ol style="list-style-type: none"> <li>1. From the <b>Posting Criteria</b> page of the account, depending on the type of invoices you want to post, place checkmarks in the <b>Post Invoice Type</b> check-boxes if necessary.</li> <li>2. Depending on the type of line items you want to post, place checkmarks in the <b>Post Invoice Tasks</b> to this Account or <b>Post</b></li> </ol>									

	<p><b>Invoice Expenses</b> to this Account check-boxes if necessary. You can also select both check-boxes.</p> <ol style="list-style-type: none"> <li>From the <b>Post Invoice Task of Type</b> or <b>Post Invoice Expense of Category</b> drop-down, update the type of tasks or expenses you want to be able to post.</li> <li>From the <b>Invoice Task Percent</b> or <b>Invoice Expense Percent</b> field, enter a number for the percentage of the task or expense you want to post. Keep at 100% if you are only posting to one account. Update the percentage if you want to split invoice amounts between multiple accounts.</li> </ol> <div data-bbox="598 582 1420 1384" style="border: 1px solid black; padding: 5px;"> <p><b>Invoice</b></p> <p>Select the Invoice Types that can be posted to this account.</p> <p>Post Invoice Type: <input checked="" type="checkbox"/> Standard  <input type="checkbox"/> Credit Note  <input type="checkbox"/> Accrual</p> <p>OR</p> <p><input type="checkbox"/> Shadow Invoice</p> <p><input type="checkbox"/> Post Invoice Tasks to this Account</p> <p>Post Invoice Task of Type: (Any)</p> <p>Invoice Task Percent: 100 %</p> <p><input checked="" type="checkbox"/> Post Invoice Expenses to this Account</p> <p>Post Invoice Expense of Category: (Any)</p> <p>Invoice Expense Percent: 100 %</p> </div> <p style="text-align: center;"><b>Allowing Invoices to Post to an Account</b></p>
<p>The account must allow non-US taxes to be posted if the invoice has non-US taxes.</p>	<ol style="list-style-type: none"> <li>From the <b>Posting Criteria</b> page of the account, if the invoice has a non-US <b>Tax Type</b>, place a checkmark in the <b>Post Non-US Tax to this Account</b> check-box if one is not present.</li> <li>From the <b>Post Non-US Tax of Type</b> drop-down, update the type of tax you want to be able to post.</li> <li>From the <b>Non-US Tax Percent</b> field, enter a number for the percentage of the tax you want to post. Keep at 100% if you are only posting to one account. Update the percentage if you want to split tax amounts between multiple accounts.</li> </ol>

	<p><b>Taxes</b></p> <p><input type="checkbox"/> Post Non-US Tax to this Account</p> <p>Post Non-US Tax of Type: (Any) ▾</p> <p>Non-US Tax Percent: 100 %</p> <p><b>Allowing Invoices with a Non-US Tax Type to Post to an Account</b></p>
<p>The date of the invoice must be within the account period.</p>	<p>The <b>Invoice Date</b> on the invoice should be within the account period, which can be found on the <b>General</b> page of the associated account</p> <p><b>Account Information</b></p> <p>*Name: Kristen's Account</p> <p>Active: Yes</p> <p>Parent Account: <input type="text"/> 🔍</p> <p>Account Type: Reserve ▾</p> <p>Overdraft Type: Do not allow negative Account ▾</p> <p>*Account Period Begins: 10/6/2016 📅</p> <p>*Account Period Ends: 10/31/2016 📅</p> <p>*Allocation Limit: 1,000.00</p> <p><input checked="" type="checkbox"/> Allow Posting</p> <p><b>Account Period on the General Page</b></p>
<p>The account must be active.</p>	<p>The status of the <b>Active</b> field on the <b>General</b> page of the account is <b>Yes</b>. See <a href="#">Activating or Deactivating Accounts</a> for more information.</p>

	 <p style="text-align: center;"><b>Account Period on the General Page</b></p>
<p>The account (or the parent's account) must have a balance of the same amount as the invoice or more.</p>	<p>From the <b>Account Summary</b> page of the account, check whether the <b>Balance</b> has enough funds to cover the invoice. If the account has no balance, check the <b>Account Summary</b> page of the parent account or the parent's parent account. You can continue checking parent accounts until you find a balance. If the balance does not have enough funds, add money to one of the accounts.</p>  <p style="text-align: center;"><b>Balance on the Account Summary Page</b></p>
<p>The invoice must include the same Vendor contact that is on the account.</p>	<p>From the <b>Posting Criteria</b> page, if the associated account specifies a <b>Vendor</b> contact in the <b>This Vendor</b> field, the <b>Vendor</b> field of the invoice record must include the same vendor.</p>

	<p><b>Note:</b> The account does not need to specify a vendor, even if the invoice includes one.</p> <div data-bbox="595 327 1390 528" style="border: 1px solid black; padding: 5px;"> <p><b>Vendor</b></p> <p>Post Account to this Vendor: (Any) ▾</p> <p>This Vendor: <input type="text"/> 🔍</p> </div> <p style="text-align: center;"><b>Specifying a Vendor for Posting</b></p>
<p>The invoice line items must include the same Involved contact that is on the account.</p>	<p>From the <b>Posting Criteria</b> page, if the associated account specifies an <b>Involved</b> contact in the <b>This Involved</b> field, the <b>Timekeeper</b> field of each line item in the invoice must include the same contact.</p> <p><b>Note:</b> The account does not need to specify an Involved contact, even if the invoice line items include one</p> <div data-bbox="595 837 1422 972" style="border: 1px solid black; padding: 5px;"> <p><b>Involved</b></p> <p>Post Account to this Involved: (Any) ▾</p> <p>This Involved: <input type="text"/> 🔍</p> </div> <p style="text-align: center;"><b>Specifying an Involved Contact for Posting</b></p>

### 1.7.3.6 Posting an Invoice

When an invoice posts successfully, TeamConnect debits the budget of accounts associated with the projects associated with line items of an invoice.

**Note:** If TeamConnect cannot locate the appropriate budget, invoice posting fails because you may need to associate the line item's of the invoice with the budgets of accounts. See [Preparing an Invoice for Posting](#). You can repost the invoice after preparing the invoice for posting correctly.

When posting an invoice unsuccessfully, you might receive the following errors:

- After a user clicks the **Post** button on an invoice, if any line item could not post, the entire invoice will not be posted, and the status of the invoice is changed to Failed.
- If an electronically submitted invoice is approved for Posting, but fails for reasons such as insufficient funds, the **Post** button will show in the invoice and the invoice can be re-posted manually once all errors are resolved.

#### To post an invoice

1. Search for an invoice and open it.
2. Click **Post**.

**Note:** After an invoice posts, you can edit custom fields but you cannot edit system fields.

The following table explains what can happen after you click the **Post** button.

Workflow process?	All line items able to be posted?	Invoice currency the default currency ?	Description
Yes	Yes	Yes	<ul style="list-style-type: none"> <li>An <b>Adjust</b> button becomes available and the <b>Posting Status</b> in the <b>Header Information</b> section on the <b>General</b> page is marked as <b>Not Posted to Pending Approval</b>.</li> </ul> <p><i>Note: The invoice must be approved by the final approver before an invoice can be posted.</i></p>
No	Yes	Yes	<ul style="list-style-type: none"> <li>The <b>Posting Status</b> in the <b>Header Information</b> section on the <b>General</b> page is marked as <b>Posted</b>.</li> <li>The <b>Transactions</b> page of the invoice and the associated account records updates with the corresponding entries.</li> <li>The <b>Void</b> button becomes available.</li> </ul>
No	Yes	No	<ul style="list-style-type: none"> <li>The amount on the invoice converts to the default currency and displays on the <b>Transactions</b> page of the invoice and the account records.</li> <li>The <b>Exchange Rate</b> displays in the <b>Header Information</b> section of the <b>General</b> page.</li> </ul>
Yes	Yes	No	<ul style="list-style-type: none"> <li>The exchange rate at the time that the invoice is finally approved is the rate that converts the invoice amounts to the default rate—not the exchange rate when you first attempt to post it.</li> </ul>
No	No	Yes	<ul style="list-style-type: none"> <li>The entire invoice does not post and an error message informs you which line items have issues.</li> <li>The <b>Posting Status</b> in the <b>Header Information</b> section on the <b>General</b> page is marked as <b>Failed</b> and the <b>Post</b> button continues to display.</li> </ul>

			<p><b>Note:</b> If you are unable to post an invoice, review <a href="#">Preparing an Invoice for Posting</a> and attempt to re-post the invoice after making necessary changes.</p>
--	--	--	--

### 1.7.3.7 Adjusting an Invoice Summary

For invoices that are pending workflow approval, you can adjust the invoice summary values, such as total fees, total expenses, or the total invoice amount.

**Note:** You cannot make negative adjustments to the Invoice Summary that result in a final invoice Total (Fees Amount total or Expenses Amount total) that is greater than or equal to the original total Amount. However, you can make negative adjustments to individual Line Items.

You can adjust the invoice summary values by:

- Adjusting the total Fees. For more information, see [Adjusting by Total Fees](#). See a screenshot example below.
- Reducing the total Expenses. For more information, see [Adjusting by Total Expenses](#).
- Reducing the total invoice amount. For more information, see [Adjusting by Total Amount](#).

**Header Adjustment**

**Adjust Fee | [Adjustment History](#)**

**Adjustment**

Adjust Fee: New Amount ▼ \$ 0.00

<b>Invoice Fee</b>		
Original (USD)		\$ 80.00
Adjusted (USD)		\$ 80.00
<b>Total Adjustment (USD)</b>		<b>\$ 0.00</b>

**Comments to Requestor**

Reason for Adjustment: (Select) ▼

**Internal Comments**

Save
Cancel

Invoice Summary Adjustment - Adjust Fee Screen

Afterward, the adjustment amount is applied proportionally to each line item according to the adjustment type. For example, if you adjust the total Fees, the adjustment amount is applied proportionally to line items of type, Fee.

### To adjust an invoice summary by total Fees

1. Search for an invoice and open it.

The following steps apply to the **Invoice Summary** section.

2. To adjust invoice Fees

- a. In the Invoice Summary section, click the Fees Amount numeric value.

- b. On the Adjust Fee screen, do one of the following:

- To adjust the **Fees** total to a different amount, from the **Adjust Fee** drop-down list, select **New Amount**. Enter the amount in the field to the right.

If you click on the screen, the amount you entered should display in the **Adjusted** row.

- To reduce the **Fees**, from the **Adjust Fee** drop-down list, select **Reduce by Amount**. Enter the amount in the field to the right.

If you click on the screen, the amount you entered should display in the **Total Adjustment** row.

- To change the **Fees** total amount by a percentage value, from the **Adjust Fee** drop-down list, select **Reduce by Percentage**. Enter the amount in the field to the right.

If you click on the screen, the **Adjusted** row should display the difference between the Original fee total and the Original fee total \* Percentage you entered. For example, if the Original fee total is 100 and the percentage is 10%, the **Adjusted** row displays 90. Also for the example, the **Total Adjustment** would display the Original fee total \* Percentage you entered ( $100 * 0.10$ ) or 10.



**Header Adjustment**

**Adjust Fee | [Adjustment History](#)**

**Adjustment**

Adjust Fee:  \$

**Invoice Fee**

Original (USD)	\$ 80.00
Adjusted (USD)	\$ 80.00
<b>Total Adjustment (USD)</b>	<b>\$ 0.00</b>

**Comments to Requestor**

Reason for Adjustment:

**Internal Comments**

**Adjusting by Total Fees**

- c. The Total Adjustment row should display the difference between the Original and Adjusted fee amounts (Original fee amount - Adjusted fee amount).
  - d. Select a Reason for Adjustment from the drop-down list.
  - e. Enter Comments to Vendor (Requestor).
  - f. Enter Internal Comments.
3. Click **Save**.

**To adjust an invoice summary by total Expenses**

1. Search for an invoice and open it.  
The following steps apply to the **Invoice Summary** section.
2. To adjust invoice Expenses
  - a. From the **Invoice Summary** section, click the Expenses Amount numeric value.
    - To adjust the **Expenses** total to a different amount, from the **Adjust Expense** drop-down list, select **New Amount**. Enter the amount in the field to the right.  
If you click on the screen, the amount you entered should display in the **Adjusted** row.
    - To reduce the **Expenses**, from the **Adjust Expense** drop-down list, select **Reduce by Amount**. Enter the amount in the field to the right.

If you click on the screen, the amount you entered should display in the **Total Adjustment** row.

- To change the **Expenses** total amount by a percentage value, from the **Adjust Expense** drop-down list, select **Reduce by Percentage**. Enter the amount in the field to the right.

If you click on the screen, the **Adjusted** row should display the difference between the Original expense total and the Original expense total \* Percentage you entered. For example, if the Original expense total is 100 and the percentage is 10%, the Adjusted row would display 90. Also for the example, the **Total Adjustment** would display the Original expense total \* Percentage you entered (100\*0.10) or 10.

**Header Adjustment**

**Adjust Expense | Adjustment History**

**Adjustment**

Adjust Expense:  \$

**Invoice Expense**

Original (USD)	\$ 120.00
Adjusted (USD)	\$ 120.00
<b>Total Adjustment (USD)</b>	<b>\$ 0.00</b>

**Comments to Requestor**

Reason for Adjustment:

**Internal Comments**

#### Adjusting by Total Expenses

- The **Total Adjustment** row should display the difference between the Original and Adjusted expense amounts (Original expense amount - Adjusted expense amount).
  - Select a **Reason for Adjustment** from the drop-down list.
  - Enter **Comments to Vendor** (Requestor).
  - Enter **Internal Comments**.
- Click **Save**.

#### To adjust an invoice summary by total amount

- Search for an invoice and open it.

The following steps apply to the **Invoice Summary** section.

2. To adjust the invoice total Amount:
  - a. From the **Invoice Summary** section, click the **Invoice Total Amount** numeric value.
  - b. From the Adjust Total screen, do one of the following:
    - To adjust the **Invoice Total** to a different amount, from the **Adjust Total** drop-down list, select **New Amount**. Enter the amount in the field to the right.  
If you click on the screen, the amount you entered should display in the **Adjusted** row.
    - To reduce the **Invoice Total**, from the **Adjust Total** drop-down list, select **Reduce by Amount**. Enter the amount in the field to the right.  
If you click on the screen, the amount you entered should display in the **Total Adjustment** row.
    - To change the **Invoice Total** amount by a percentage value, from the **Adjust Total** drop-down list, select **Reduce by Percentage**. Enter the amount in the field to the right.  
If you click on the screen, the **Adjusted** row should display the difference between the Original Invoice total and the Original Invoice total \* Percentage you entered. For example, if the Original Invoice total is 100 and the percentage is 10%, the Adjusted row would display 90. Also for the example, the **Total Adjustment** would display the Original Invoice total \* Percentage you entered (100\*0.10) or 10.
  - c. The **Total Adjustment** row should display the difference between the Original and Adjusted Invoice Total amounts (Original Invoice Total amount - Adjusted Invoice Total amount).
  - d. Select a **Reason for Adjustment** from the drop-down list.
  - e. Enter **Comments to Vendor** (Requestor).
  - f. Enter **Internal Comments**.
3. Click **Save**.

The following field descriptions apply to the all of the Invoice Summary adjustment types (Adjust Fees, Adjust Expenses, and Adjust Total screens).

#### Invoice Summary Adjustment, Adjust Fee Screen

Label	Description
<b>Adjust Fee</b>	Drop-down list options include: <ul style="list-style-type: none"> <li>• <b>New Amount</b>—If selected, the numeric amount you enter in the field to the right will become the Adjusted (total).</li> <li>• <b>Reduce by Amount</b>—If selected, the numeric amount you enter in the field to the right will become the <b>Total Adjustment</b> amount.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Reduce by Percentage</b>—If selected, the numeric amount you enter in the field to the right will be used to calculate the <b>Total Adjustment</b> amount. The Total Adjustment amount will be (the Original Invoice Fee total * Percentage). For example, if you enter 10 in the numeric field, the <b>Total Adjustment</b> amount will be (the Original Invoice Fee total * 10%).</li> </ul>
<b>Adjustment History</b>	Links to a screen that displays a Invoice Summary adjustment history.
<b>Comments to Requestor</b>	<ul style="list-style-type: none"> <li>• <b>Reason for Adjustment</b>—Drop-down list displays the option, Other, by default. Additional options may be available depending on your system customizations. Contact your TeamConnect administrator for more information.</li> <li>• A text box is available to enter notes to the requestor. You can enter up to 2000 characters, including spaces.</li> </ul>
<b>Invoice Fee</b>	<p>The following rows display:</p> <ul style="list-style-type: none"> <li>• <b>Original</b>—The Invoice Summary section, Fees (total) Amount.</li> <li>• <b>Adjusted</b>—The amount that the Fees (total) Amount will be adjusted to.</li> <li>• <b>Total Adjustment</b>—The difference between the Original and Adjusted amounts (described above).</li> </ul>

#### Invoice Summary Adjustment, Adjust Expense Screen

Label	Description
<b>Adjust Expense</b>	<p>Drop-down list options include:</p> <ul style="list-style-type: none"> <li>• <b>New Amount</b>—If selected, the numeric amount you enter in the field to the right will become the Adjusted (total).</li> <li>• <b>Reduce by Amount</b>—If selected, the numeric amount you enter in the field to the right will become the <b>Total Adjustment</b> amount.</li> <li>• <b>Reduce by Percentage</b>—If selected, the numeric amount you enter in the field to the right will be used to calculate the <b>Total Adjustment</b> amount. The Total Adjustment amount will be (the Original Invoice Expense total * Percentage). For example, if you enter 10 in the numeric field, the <b>Total Adjustment</b> amount will be (the Original Invoice Expense total * 10%).</li> </ul>
<b>Adjustment History</b>	Links to a screen that displays a Invoice Summary adjustment history.
<b>Comments to Requestor</b>	<ul style="list-style-type: none"> <li>• <b>Reason for Adjustment</b>—Drop-down list displays the option, Other, by default. Additional options may be available depending on your system</li> </ul>

	<p>customizations. Contact your TeamConnect administrator for more information.</p> <ul style="list-style-type: none"> <li>• A text box is available to enter notes to the requestor. You can enter up to 2000 characters, including spaces.</li> </ul>
<b>Invoice Expense</b>	<p>The following rows display:</p> <ul style="list-style-type: none"> <li>• <b>Original</b>—The Invoice Summary section, Expense (total) Amount.</li> <li>• <b>Adjusted</b>—The amount that the Expense (total) Amount will be adjusted to.</li> <li>• <b>Total Adjustment</b>—The difference between the Original and Adjusted amounts (described above).</li> </ul>

#### Invoice Summary Adjustment, Adjust Invoice Total Screen

Label	Description
<b>Adjustment History</b>	Links to a screen that displays a Invoice Summary adjustment history.
<b>Adjust Total</b>	<p>Drop-down list options include:</p> <ul style="list-style-type: none"> <li>• <b>New Amount</b>—If selected, the numeric amount you enter in the field to the right will become the Adjusted (total).</li> <li>• <b>Reduce by Amount</b>—If selected, the numeric amount you enter in the field to the right will become the <b>Total Adjustment</b> amount.</li> <li>• <b>Reduce by Percentage</b>—If selected, the numeric amount you enter in the field to the right will be used to calculate the <b>Total Adjustment</b> amount. The Total Adjustment amount will be (the Original Invoice Total * Percentage). For example, if you enter 10 in the numeric field, the <b>Total Adjustment</b> amount will be (the Original Invoice Total * 10%).</li> </ul>
<b>Comments to Requestor</b>	<ul style="list-style-type: none"> <li>• <b>Reason for Adjustment</b>—Drop-down list displays the option, Other, by default. Additional options may be available depending on your system customizations. Contact your TeamConnect administrator for more information.</li> <li>• A text box is available to enter notes to the requestor. You can enter up to 2000 characters, including spaces.</li> </ul>
<b>Internal Comments</b>	Notes that display internally but do not display to the requestor (for example, vendor). You can enter up to 2000 characters, including spaces.
<b>Invoice Total</b>	<p>The following rows display:</p> <ul style="list-style-type: none"> <li>• <b>Original</b>—The Invoice Summary section, Invoice Total Amount.</li> </ul>

- **Adjusted**—The amount that the Invoice Total Amount will be adjusted to.
- **Total Adjustment**—The difference between the Original and Adjusted amounts (described above).

### 1.7.3.8 Adjusting Invoice Line Items

Three procedures for adjusting invoice line items in TeamConnect are available but only one is available for use at a time. The active line item adjustment setting is defined in the TeamConnect Setup screen. For more information on changing the available line item adjustment setting, see [Setting up Invoice Line Items Adjustment Behavior](#).

From an invoice, **Line Items** section bar, if an **Adjust** link displays, then either [the Batch Adjustment setting](#) or [the In-Line & Bulk Adjustment setting](#) is active. Otherwise, [the Detail Adjustment setting](#) is active.

**Note:** A line item adjustment is considered an edit to the invoice. The invoice will be locked against other edits until the adjustment is completed.

**Note:** The procedures and fields described in this section may not display based on your rights. If you have questions, contact your TeamConnect administrator.

### Adjusting Line Items with Discounts

Beginning in TeamConnect 5.0, users are able to [revert line item adjustments](#) and [reject line items](#) altogether. These processes operate much like how applying discounts functioned in 4.x versions. It is important to note that there is no functional difference between reverting and line item and applying a correlating discount to negate the adjustment. The reversion serves to simplify this process. For more information, see Reverting Line Items on this page.

The discount always displays as currency value. However, based on system settings, it may be interpreted as either a flat discount, or as a value calculated from a percentage discount.

**Important:** During line item adjustments where the original discount was entered as a percentage, the system will automatically adjust the discount amount if the original rate or units are adjusted.

When adjusting the Invoice Summary or Line Items, the following messages could display at the bottom of the adjustment screen:

#### Invoice Discount Information Message

Information Message	Description
Original discount was applied as percentage. Discount value is recalculated accordingly if	This means that in the system settings, Discounts are set to be interpreted as percentage values. The Original Discount value will be used to calculate the discount percentage, and then the same

Rate or Hours/Units are adjusted.	percentage would be applied to the final discount after your adjustments.
Adjustment to amount will be reflected in the discount.	This means that any adjustment to the line item total will display the difference between the new total and the original total as a change in the discount.

**Note:** When adjusting Line Items in batch adjustment mode, the messages above display if there is at least one Line Item with a non-zero Original Discount and the Line Item Adjusted Amount has not been manually adjusted.

### To adjust invoice line items (Detail Adjustment)

1. Search for an invoice and open it.
2. To adjust a line item **Rate**:
  - a. From the Line Items section, under the Rate column, click the link for the value to adjust.
  - b. From the Adjust Rate screen:
    - To adjust the Rate to a different amount, from the **Adjust Rate** drop-down list, select **New Amount**. Enter the amount in the field to the right.  
If you click on the screen, the amount you entered should display in the **Adjusted Rate** table cell.
    - To reduce the Rate, from the **Adjust Rate** drop-down list, select **Reduce by Amount**. Enter the amount in the field to the right.  
If you click on the screen, the following calculation applies. **Adjusted Rate = Original Rate - Reduce by Amount** value.
    - To change the Rate by a percentage value, from the **Adjust Rate** drop-down list, select **Reduce by Percentage**. Enter the amount in the field to the right.  
If you click on the screen, the following calculation applies. **Adjusted Rate = Original Rate - (Original Rate\*Reduce by Percentage value/100)**.
    - Select a **Reason for Adjustment** from the drop-down list.
    - Enter adjustment notes to the vendor in the **Comments to Vendor** field.
    - Enter **Internal Comments** for your company.
3. To adjust a line item **Unit**:
  - a. From the **Line Items** section, under the **Units** column, click the link for the value to adjust.
  - b. From the **Adjust Units** screen:
    - To adjust the Units to a different amount, from the **Adjust Units** drop-down list, select **New Amount**. Enter the amount in the field to the right.

If you click on the screen, the amount you entered should display in the **Adjusted Unit** table cell.

- To reduce the Hours or Units, from the **Adjust Hrs/Units** drop-down list, select **Reduce by Amount**. Enter the amount in the field to the right.

If you click on the screen, the following calculation applies. **Adjusted Unit = Original Unit - Reduce by Amount** value.

- To change the Hours or Units by a percentage value, from the **Adjust Hrs/Units** drop-down list, select **Reduce by Percentage**. Enter the amount in the field to the right.

If you click on the screen, the following calculation applies. **Adjusted Unit = Original Unit - (Original Unit\*Reduce by Percentage value/100)**.

- Select a **Reason for Adjustment** from the drop-down list.
- Enter adjustment notes to the vendor in the **Comments to Vendor** field.
- Enter **Internal Comments** for your company.

4. To adjust a line item total **Amount**:

- a. From the **Line Items** section, under the **Amount** column, click the link for the value to adjust.

- b. From the **Adjust Amount** screen:

- To adjust the total Amount to a different value, from the **Adjust Amount** drop-down list, select **New Amount**. Enter the amount in the field to the right.

If you click on the screen, the amount you entered should display in the **Adjusted Amount** table cell.

- To reduce the total Amount, from the **Adjust Amount** drop-down list, select **Reduce by Amount**. Enter the amount in the field to the right.

If you click on the screen, the following calculation applies. **Adjusted Amount = Original Amount - Reduce by Amount** value.

- To change the total Amount by a percentage value, from the **Adjust Amount** drop-down list, select **Reduce by Percentage**. Enter the amount in the field to the right.

If you click on the screen, the following calculation applies. **Adjusted Amount = Original Amount - (Original Amount\*Reduce by Percentage value/100)**.

- Select a **Reason for Adjustment** from the drop-down list.
- Enter adjustment notes to the vendor in the **Comments to Vendor** field.
- Enter **Internal Comments** for your company.

5. Click **Save**. The list of invoice line items appear. The line item that you were working on is visible and has a focus frame around it.

The adjustment will then be displayed on the line item along with the name of the user who made the change and the time of the change. Manual adjustments (as described above) will have a



different icon next to the change than automated adjustments. In the case of multiple adjustments, each will be shown in the order of the adjustment.

Item	Date	Type	Category	TK	Rate	Units	Tax Type	Disc	Adj	Amt
1	6/21/16	Expense	E101 Copying		₩1.00	1.00		₩0.00	₩-9.00	₩10.00
Project: DISP-000000-Charles Dispute 1 Adjustment Type: Manual Adjustment: 6/30/16 - Rate adjusted to 2.00 KRW - Khan, Afia Adjustment: 7/6/16 - Quantity adjusted to 5.00 hours - Passaman, Clayton										

**Manual Line Item Adjustment**

Adjustments that have been completed via automated rule will have the icon shown below as well as details on the change.

Item	Date	Type	Category	TK	Rate	Units	Disc	Adj	Amt	
1	9/30/16	Fee	Fee		\$500.00	1.00	\$0.00	\$0.00	\$500.00	
Project: DISP-000000-Charles Dispute 1										
2	9/30/16	Expense	E100 Expenses	purple associates, Other Timekeeper	\$45.00	2.00	\$0.00	\$90.00	\$0.00	
Project: DISP-000000-Charles Dispute 1 Description: pink Adjustment Type: Auto Adjustment: 9/30/16 - Discount adjusted to 0.00 USD - system, system										

**Automatically Rejected Line Item**

**To adjust invoice line items (Batch Adjustment)**

This procedure allows you to perform adjustments on a list of line items in a batch, using different types of adjustments. If instead you wish to apply a single adjustment type to multiple line items, you can [use Bulk Adjustment](#).

1. Search for an invoice and open it.
2. Use Line Item Search by clicking **Show Filter** to locate and display the particular items that need adjustment.
3. From the **Line Items** section bar, click the **Adjust** link. The adjustment window is displayed.
4. If the invoice contains a large number of line items, there will be multiple pages, each containing several line items. You can control the number of line items per page by changing the value in the **Line Items per page** drop-down list. For each line item to adjust:
  - o Edit the **Current Rate**
  - o Edit the **Current Unit**
  - o Edit the **Current Discount**
  - o Edit the **Current Amount**
  - o Select an **Adjustment Reason** from the drop-down list.

- Type adjustment explanations to the vendor in the **Comments to Vendor** field.
- Type **Internal Comments** for your company but not for the Vendor to view.

5. Click **Save**.

The adjustment will then be displayed on the line item along with the name of the user who made the change and the time of the change. Manual adjustments (as described above) will have a different icon next to the change than automated adjustments. In the case of multiple adjustments, each will be shown in the order of the adjustment.

Item	Date	Type	Category	TK	Rate	Units	Tax Type	Disc	Adj	Amt
1	6/21/16	Expense	E101 Copying		₩1.00	1.00		₩0.00	₩-9.00	₩10.00
Project: DISP-000000-Charles Dispute 1 Adjustment Type: Manual Adjustment: 6/30/16 - Rate adjusted to 2.00 KRW - Khan, Afia Adjustment: 7/6/16 - Quantity adjusted to 5.00 hours - Passaman, Clayton										

#### Manual Line Item Adjustment

Adjustments that have been completed via automated rule will have the icon shown below as well as details on the change.

Item	Date	Type	Category	TK	Rate	Units	Disc	Adj	Amt	
1	9/30/16	Fee	Fee		\$500.00	1.00	\$0.00	\$0.00	\$500.00	
Project: DISP-000000-Charles Dispute 1										
2	9/30/16	Expense	E100 Expenses	purple associates, Other Timekeeper	\$45.00	2.00	\$0.00	\$90.00	\$0.00	
Project: DISP-000000-Charles Dispute 1 Description: pink Adjustment Type: Auto Adjustment: 9/30/16 - Discount adjusted to 0.00 USD - system, system										

#### Automatically Rejected Line Item

### To adjust invoice line items (In-Line & Bulk Adjustment)

This procedure allows you to perform adjustments on a list of line items in a batch, using different types of adjustments. If instead you wish to apply a single adjustment type to multiple line items, you can [use Bulk Adjustment](#).

1. Search for an invoice and open it.
2. Use Line Item Search by clicking **Show Filter** to locate and display the particular items that need adjustment.
3. From the **Line Items** section bar, click the **Adjust** link. The adjustment window is displayed.

Line Items											
Expand All		Collapse All									
Project: DISP-000000-Charles Dispute 1											
Item	Date	Type	Category	Timekeeper	Rate	Units	Discount	Adjustment	Total	Comments	
1	10/10/2016	EXPENSE	E101 Copying	Holden, Cody	Current	\$ 10.00	10.00	\$ 0.00	-	\$ 100.00	
					Original	\$10.00	10.00	\$0.00	-	\$100.00	
2	10/10/2016	FEE	Bankruptcy Code set	Timekeeper, Michael	Current	\$ 10.00	10.00	\$ 0.00	-	\$ 100.00	
					Original	\$10.00	10.00	\$0.00	-	\$100.00	
Potential duplicate expense on :current invoice (line item 1)											
3	10/10/2016	EXPENSE	E121 Arbitrators/mediators	Pate, Atish	Current	\$ 10.00	80.00	\$ 0.00	-	\$ 800.00	
					Original	\$10.00	80.00	\$0.00	-	\$800.00	
Potential duplicate expense on :current invoice (line item 1,2)											
Potential duplicate expense on :current invoice (line item 1)											
Bulk Adjustment		All line items shown.									

4. A list of the line items displays in an embedded box. This is so the column headers of the line items are visible even as you scroll down to view the items on the bottom of the list. For each line item to adjust:

- Edit the **Current Rate**
- Edit the **Current Units**
- Edit the **Current Discount**
- Edit the **Current Total**
- Click the **Comments** drop-down menu to open a pop-up in which you can add the following items:
  - **Reason for Adjustment** from the drop-down list.
  - Adjustment explanations to the vendor in the **Comments to Vendor** field.
  - **In House Comments** visible to your company but not to the Vendor.

5. Click **Save**.

The adjustment will then be displayed on the line item along with the name of the user who made the change and the time of the change. Manual adjustments (as described above) will have a different icon next to the change than automated adjustments. In the case of multiple adjustments, each will be shown in the order of the adjustment.

Item	Date	Type	Category	TK	Rate	Units	Tax Type	Disc	Adj	Amt	
1	6/21/16	Expense	E101 Copying		₩1.00	1.00		₩0.00	₩-9.00	₩10.00	
	Project: DISP-000000-Charles Dispute 1										
	Adjustment Type: Manual Adjustment: 6/30/16 - Rate adjusted to 2.00 KRW - Khan, Afa Adjustment: 7/6/16 - Quantity adjusted to 5.00 hours - Passaman, Clayton										

**Manual Line Item Adjustment**

Adjustments that have been completed via automated rule will have the icon shown below as well as details on the change.

Item	Date	Type	Category	TK	Rate	Units	Disc	Adj	Amt
1	9/30/16	Fee	Fee		\$500.00	1.00	\$0.00	\$0.00	\$500.00
	Project: DISP-000000-Charles Dispute 1								
2	9/30/16	Expense	E100 Expenses	purple associates, Other Timekeeper	\$45.00	2.00	\$0.00	\$90.00	\$0.00
	Project: DISP-000000-Charles Dispute 1 Description: pink  Adjustment Type:  Auto Adjustment: 9/30/16 - Discount adjusted to 0.00 USD - system, system								

#### Automatically Rejected Line Item

### To adjust invoice line items (Bulk Adjustment)

1. Search for an invoice and open it.
2. Use Line Item Search to locate and display the particular items that need adjustment.
3. From the **Line Items** section bar, click the **Adjust** link. This link appears if the invoice line item adjustment setting is set to either Batch Adjustment or In-Line & Bulk Adjustment. See [Setting up Invoice Line Items Adjustment Behavior](#) for more information.
4. Click the check-box for all of the line items that will be affected by the bulk adjustment.
5. Click the **Bulk Adjustment** button. The Bulk Adjustment window is displayed.

**Bulk Adjustment**

3 Line items to be adjusted

**Adjustment**

Adjust: Rate

Adjust Type: New Amount

Adjust Amount:

Reason for Adjustment: (Select)

Comments to Requester:

\* In House Comments:

Maintain Selection After Adjustment:

Ok Cancel

**The Bulk Adjustment Window for Line Items**

- Complete the appropriate fields. The fields are a subset of those used in Batch Adjustment or In-Line & Bulk Adjustment, and vary depending on which setting is active. The fields behave in the same way as when used in their respective in-line adjustment setting. See [the Batch Adjustment and In-Line & Bulk Adjustment Line Items fields table](#).
- Click **OK**.  
The adjustments are applied to the line items that you checked.
- Select a save option to save the changes.
- The adjustment will then be displayed on the line item along with the name of the user who made the change and the time of the change. Manual adjustments (as described above) will have a different icon next to the change than automated adjustments. In the case of multiple adjustments, each will be shown in the order of the adjustment.

Item	Date	Type	Category	TK	Rate	Units	Tax Type	Disc	Adj	Amt
1	6/21/16	Expense	E101 Copying		₩1.00	1.00		₩0.00	₩-9.00	₩10.00
Project: DISP-000000-Charles Dispute 1 Adjustment Type: Manual Adjustment: 6/30/16 - Rate adjusted to 2.00 KRW - Khan, Afa Adjustment: 7/6/16 - Quantity adjusted to 5.00 hours - Passaman, Clayton										

**Manual Line Item Adjustment**

Adjustments that have been completed via automated rule will have the icon shown below as well as details on the change.

Item	Date	Type	Category	TK	Rate	Units	Disc	Adj	Amt	
1	9/30/16	Fee	Fee		\$500.00	1.00	\$0.00	\$0.00	\$500.00	
Project: DISP-000000-Charles Dispute 1										
2	9/30/16	Expense	E100 Expenses	purple associates, Other Timekeeper	\$45.00	2.00	\$0.00	\$90.00	\$0.00	
Project: DISP-000000-Charles Dispute 1 Description: pink Adjustment Type: Auto Adjustment: 9/30/16 - Discount adjusted to 0.00 USD - system, system										

**Automatically Rejected Line Item**

**To revert invoice line item adjustments**

Invoice line item adjustments can be reverted by approvers. Click the **Revert Selected** button after checking the desired adjustment checkbox to revert the adjustment change. This function operates much like a more efficient method of applying and adjusting discounts on a line item.

**Note:** Reverting an adjustment will also revert all adjustments made after that selected point. For example, if there are three adjustments (respectively dated 10/1/16, 10/3/16, and 10/5/16) and the 10/3/15 adjustment is reverted, the 10/5/16 adjustment will automatically be reverted as well. The 10/1/16 adjustment will remain unchanged.

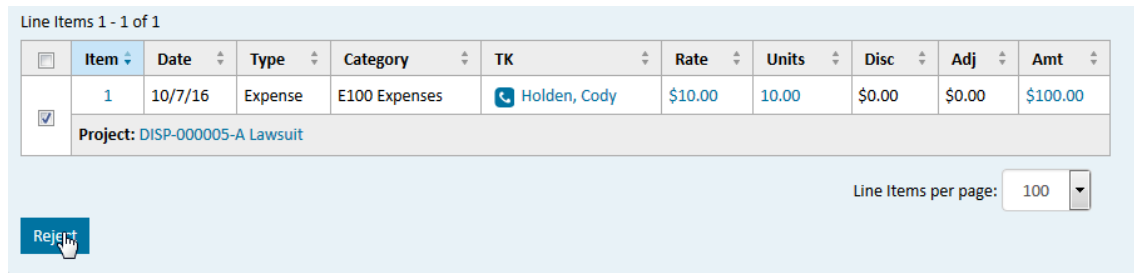
Item	Date	Type	Category	TK	Rate	Units	Disc	Adj	Amt	
1	10/7/16	Expense	E100 Expenses	Holden, Cody	\$10.00	10.00	\$0.00	\$100.00	\$0.00	
Project: DISP-000005-A Lawsuit Adjustment Type: Manual <span style="color: red;">✘ Rejected</span> <input checked="" type="checkbox"/> <b>Revert Selected</b> <input checked="" type="checkbox"/> <b>Adjustment:</b> 10/7/16 - Discount adjusted to 100.00 USD - Holden, Cody Reason for Adjustment: Blended Hourly Rate In House Comments: Bad Idea										

Line Items per page: 100

**Reverting a Line Item Adjustment**

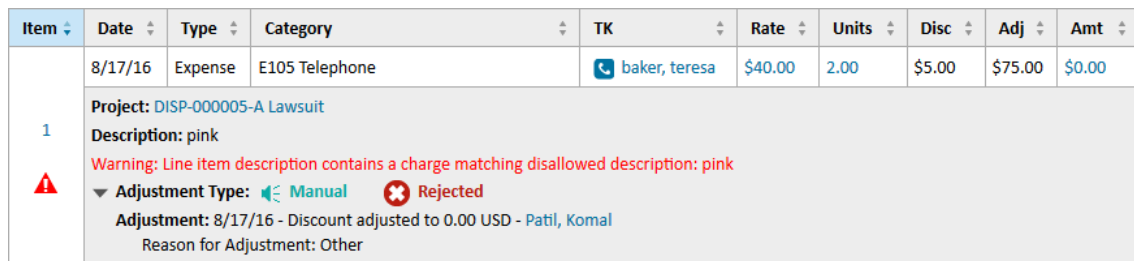
**To reject invoice line items**

Invoice line items can be rejected in total by approvers. After an invoice moves into workflow, simply highlight the line item and click the **Reject** button.



Manually Rejecting A Line Item

The line item will be marked as "Rejected." Line items can also be rejected automatically as per designated rules.



Automatically Rejected Line Item

### (Detail Adjustment) Line Items Field Descriptions

The following field descriptions apply to screens that display when adjusting Line Items section with the Detail Adjustment setting (Adjust Rate, Adjust Unit, Adjust Amount screens). Some field descriptions that follow may not display based on system settings and your assigned rights.

Field descriptions are listed in alphabetical order and may display in a different order in the application.

#### (Detail Adjustment) Invoice Line Item, Adjust Rate Screen

Label	Description
<b>Adjust Rate</b>	Drop-down list options include: <ul style="list-style-type: none"> <li><b>New Amount</b>—If selected, the numeric amount you enter in the field to the right will become the Adjusted Rate.</li> <li><b>Reduce by Amount</b>—If selected, the numeric amount you enter in the field to the right will be applied in the following calculation: <b>Adjusted Rate = Original Rate - Reduce by Amount</b> value.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Reduce by Percentage</b>—If selected, the numeric amount you enter in the field to the right will be applied in the following calculation: <b>Adjusted Rate = Original Rate - (Original Rate*Reduce by Percentage value/100)</b>. For example, if you enter 10 in the numeric field and the <b>Original Rate</b> is 50, the <b>Adjusted Rate</b> = 50 - (50*0.10) = 45</li> </ul>
<b>Adjustment History</b>	Links to a screen that displays a Invoice Line Item adjustment history.
<b>Comments to Requestor</b>	<ul style="list-style-type: none"> <li>• <b>Reason for Adjustment</b>—Drop-down list displays the option, Other, by default. Additional options may be available depending on your system customizations. Contact your TeamConnect administrator for more information.</li> <li>• A text box is available to enter notes to the requestor. You can enter up to 2000 characters, including spaces.</li> </ul>
<b>Internal Comments</b>	Notes that should only be read by other approvers and TeamConnect users. You can enter up to 2000 characters, including spaces.
<b>Line Item table</b>	<p>The following display:</p> <ul style="list-style-type: none"> <li>• <b>Original row</b>—Original values entered for the line item, including Original Rate, Original Unit, Original Discount. Original Amounts = Original Rate * Original Unit - Original Discount</li> <li>• <b>Adjusted row</b>—Reflects adjustments to the rate. Original Rate, Original Unit, Original Discount. Adjusted Amounts = Adjusted Rate * Adjusted Unit - Adjusted Discount</li> <li>• <b>Total Adjustment</b>—The difference between the Original and Adjusted amounts (described above).</li> </ul>

**(Detail Adjustment) Invoice Line Item, Adjust Unit Screen**

Label	Description
<b>Adjust Unit</b>	<p>Drop-down list options include:</p> <ul style="list-style-type: none"> <li>• <b>New Amount</b>—If selected, the numeric amount you enter in the field to the right will become the Adjusted Unit.</li> <li>• <b>Reduce by Amount</b>—If selected, the numeric amount you enter in the field to the right will be applied in the following calculation: <b>Adjusted Unit = Original Unit - Reduce by Amount</b> value.</li> <li>• <b>Reduce by Percentage</b>—If selected, the numeric amount you enter in the field to the right will be applied in the following calculation: <b>Adjusted Unit = Original Unit - (Original Unit*Reduce by Percentage value/100)</b>. For example, if you enter 10 in the numeric field and the <b>Original Unit</b> is 50, the <b>Adjusted Unit</b> = 50 - (50*0.10) = 45</li> </ul>



<b>Adjustment History</b>	Links to a screen that displays a Invoice Line Item adjustment history.
<b>Comments to Requestor</b>	<ul style="list-style-type: none"> <li>• <b>Reason for Adjustment</b>—Drop-down list displays the option, Other, by default. Additional options may be available depending on your system customizations. Contact your TeamConnect administrator for more information.</li> <li>• A text box is available to enter notes to the requestor. You can enter up to 2000 characters, including spaces.</li> </ul>
<b>Internal Comments</b>	Notes that should only be read by other approvers and TeamConnect users. You can enter up to 2000 characters, including spaces.
<b>Line Item table</b>	<p>The following display:</p> <ul style="list-style-type: none"> <li>• <b>Original row</b>—Original values entered for the line item, including Original Rate, Original Unit, Original Discount. Original Amounts = Original Rate * Original Unit - Original Discount</li> <li>• <b>Adjusted row</b>—Reflects adjustments to the rate. Original Rate, Original Unit, Original Discount. Adjusted Amounts = Adjusted Rate * Adjusted Unit - Adjusted Discount</li> <li>• <b>Total Adjustment</b>—The difference between the Original and Adjusted amounts (described above).</li> </ul>

**(Detail Adjustment) Invoice Line Item, Adjust Amount Screen**

<b>Label</b>	<b>Description</b>
<b>Adjust Amount</b>	<p>Drop-down list options include:</p> <ul style="list-style-type: none"> <li>• <b>New Amount</b>—If selected, the numeric amount you enter in the field to the right will become the Adjusted Amount.</li> <li>• <b>Reduce by Amount</b>—If selected, the numeric amount you enter in the field to the right will be applied in the following calculation: <b>Adjusted Amount = Original Amount - Reduce by Amount</b> value.</li> <li>• <b>Reduce by Percentage</b>—If selected, the numeric amount you enter in the field to the right will be applied in the following calculation: <b>Adjusted Amount = Original Amount - (Original Amount*Reduce by Percentage value/100)</b>. For example, if you enter 10 in the numeric field and the <b>Original Amount</b> is 50, the <b>Adjusted Amount = 50 - (50*0.10) = 45</b></li> </ul>
<b>Adjustment History</b>	Links to a screen that displays a Invoice Line Item adjustment history.

<b>Comments to Requestor</b>	<ul style="list-style-type: none"> <li>• <b>Reason for Adjustment</b>—Drop-down list displays the option, Other, by default. Additional options may be available depending on your system customizations. Contact your TeamConnect administrator for more information.</li> <li>• A text box is available to enter notes to the requestor. You can enter up to 2000 characters, including spaces.</li> </ul>
<b>Internal Comments</b>	Notes that should only be read by other approvers and TeamConnect users. You can enter up to 2000 characters, including spaces.
<b>Line Item table</b>	<p>The following display:</p> <ul style="list-style-type: none"> <li>• <b>Original row</b>—Original values entered for the line item, including Original Rate, Original Unit, Original Discount. Original Amounts = Original Rate + Original Unit + Original Discount</li> <li>• <b>Adjusted row</b>—Reflects adjustments to the rate. Original Rate, Original Unit, Original Discount. Adjusted Amounts = Adjusted Rate + Adjusted Unit + Adjusted Discount</li> <li>• <b>Total Adjustment</b>—The difference between the Original and Adjusted amounts (described above).</li> </ul>

### (Batch Adjustment and In-Line & Bulk Adjustment) Line Items Section Field Descriptions

The following field descriptions apply to the screen for adjusting the Line Items section with the Batch Adjustment and In-Line & Bulk Adjustment setting. Some field descriptions that follow may not display based on system settings and your assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

#### Line Item Screen

Label	Description
<b>Activity</b>	For Fee line items, displays an activity category.
<b>Adjustment Total</b>	Displays any adjustments made to the line item.
<b>Category</b>	Displays the category associated with the Line Item type (Task/Fee Category or Expense Category).
<b>Comments to Vendor</b>	Enter adjustment notes.

	<b>Note:</b> For Collaborati Spend Management invoices (sent from Collaborati), this information will be sent to the vendor. You can enter up to 2000 characters, including spaces.
<b>Current Amount</b>	This amount is automatically calculated when you add the line item: <b>Amount = (Rate) x (Hrs/Units) - Discount</b> .  Displays the total for a Line Item.
<b>Current Discount</b>	Displays the discount amount. The discount will display as a flat amount.
<b>Current Rate</b>	Displays the hourly rate for a billed task or per unit cost for an expense.
<b>Current Unit</b>	Displays the number of hours for fee items or units for expense items.
<b>Date</b>	Displays the date on which the services were provided or expenses were incurred.
<b>Description</b>	Displays the line item description.
<b>In-house Comments</b>	Notes that should only be read by other approvers and TeamConnect users. You can enter up to 2000 characters, including spaces.
<b>Item Type</b>	Displays either Fee or Expense for the line item type.
<b>Original Amount</b>	Displays an automatically calculated value after you add a line item. <b>Amount = (Rate x Unit) - Discount</b>
<b>Original Discount</b>	Displays the discount to be applied to the original price or rate for the line item. The discount will display as a flat amount.
<b>Original Rate</b>	Displays the original rate for services rendered or unit price for an expense.
<b>Original Unit</b>	Displays the number of hours of a service or the number of units for an expense.
<b>Reason for Adjustment</b>	Select an option from the drop-down list.
<b>Taxable Item</b>	Displays Yes or No.

**Timekeeper**

Displays the Timekeeper or contact who performed the services.

**1.7.3.9 Line Item General**

The following links provide detail on adding, customizing, and configuring invoice line items:

[Invoice Overview](#)

[Adjusting Invoice Line Items](#)

[Adding Invoice Line Items](#)

[Setting up General Invoice Line Items Behavior](#)

[Rejecting Line Items](#)

[Reverting Line Item Adjustments](#)

**1.7.3.10 Approving Invoices**

This topic applies if an invoice is in a workflow process only. You can approve invoices in a batch (from an Approvals list).

Home Finance Contacts Calendar Documents Admin All Services

My Approvals

Requests to Approve

Requests to Review

Requests to Approve

1 - 2 of 2

<input type="checkbox"/>	Record	Request	Type	Requester	Due Date	Last Approver	Action
<input type="checkbox"/>	invoice0609_02	Invoice for 155.00 USD	Invoice	Walker, Carol	6/16/08	---	Reject
<input type="checkbox"/>	invoice0609_03	Invoice for 630.00 USD	Invoice	Holt, Nan	6/16/08	---	Reject

Approve Send For Review

My Approvals Area, Requests to Approve Page

**To approve invoices in a batch**

- To access the My Approvals page, do one of the following:
  - Click one of the hyperlinks in your My Approvals portal pane on your home page.
  - Select **Approvals** from the **All** tab.
 

The **My Approvals** screen opens with the **Requests to Approve** tab showing.
- Select the check-box(es) next to the record(s) you wish to approve.
- Click **Approve**. The Invoice status changes from **Not Posted** to **Posted**, or the invoice will go to the next approver, depending on the workflow settings.

The approval history record is recorded on the **Workflow** page of the invoice. All other requests for review or other workflow related events are recorded on the **Workflow** page as well.

### To approve an individual invoice

1. Search for an invoice and open it.
2. From the top of the page, click **Approve**.
3. From the **Confirm Approval** screen, you can enter **Internal Comments**.
4. Click **Confirm Approval**.

#### 1.7.3.10.1 Rejecting Invoices

This topic applies if an invoice is in a workflow process only. You can reject invoices from an Approvals summary list or from an invoice record.

The screenshot shows the 'My Approvals' page with a navigation bar at the top containing 'Home', 'Finance', 'Contacts', 'Calendar', 'Documents', 'Admin', and 'All Services'. Below the navigation bar, there is a 'My Approvals' section with a sidebar on the left containing 'Requests to Approve' and 'Requests to Review'. The main content area is titled 'Requests to Approve' and shows a table with 2 records. Below the table are buttons for 'Approve' and 'Send For Review'.

<input type="checkbox"/>	Record	Request	Type	Requester	Due Date	Last Approver	Action
<input type="checkbox"/>	<a href="#">invoice0609_02</a>	Invoice for 155.00 USD	Invoice	<a href="#">Walker, Carol</a>	6/16/08	---	<a href="#">Reject</a>
<input type="checkbox"/>	<a href="#">invoice0609_03</a>	Invoice for 630.00 USD	Invoice	<a href="#">Holt, Nan</a>	6/16/08	---	<a href="#">Reject</a>

My Approvals Area, Requests to Approve Page

### To reject an invoice (from Approvals)

1. To access the **My Approvals** page, do one of the following:
  - Click one of the hyperlinks in your **My Approvals** portal pane on your home page.
  - Select **Approvals** from the All tab.

The **My Approvals** screen opens with the **Requests to Approve** tab showing.

2. Select the check-box next to the record you wish to reject.
3. Enter a reason for the rejection in the **Internal Comment** field.
4. Click **Reject**. A confirmation pop-up opens.
5. Select a **Rejection Reason**. The list of reasons is determined by your TeamConnect administrator.

6. Enter comments in the **Comment to Requestor** field.

If you select a **Rejection Reason** option other than **Default**, on the invoice's **General** tab, the **Comment to Requestor** box will contain both the selected **Rejection Reason** option and the **Comment to Requestor** from the **My Approvals - Reject Invoice** screen.

For example, if the "Reason for Reject" is "Invoice contains multiple violations" and "Comments to Requestor" is "Too many billing guidelines violation, including line items 2-10. Please resubmit per discussion", then the "Comments to Requestor" of the Invoice will show the following:

*Invoice contains multiple violations.*

*Too many billing guidelines violation, including line items 2-10. Please resubmit per discussion.*

7. Click **Confirm Rejection**. The invoice status changes from Not Posted to Rejected.

The approval history record is recorded on the **Workflow** tab. All other requests for review or other workflow related events are recorded here as well.

#### To reject an invoice (from invoice record)

1. Search for an invoice and open it.
2. From the top of the page, click **Reject**.
3. From the **Confirm Rejection** screen, you can:
  - Select the **Rejection Reason** from the drop-down list.
  - Enter comments in the **Comment to Requestor** field.
4. Click **Confirm Rejection**.

If you select a **Rejection Reason** option other than **Default**, the resulting invoice, **General** page, the **Comment to Requestor** field will contain the selected **Rejection Reason** option and the **Comment to Requestor** from the **My Approvals - Reject Invoice** screen.

For example, if the "Reason for Reject" is "Invoice contains multiple violations" and "Comments to Requestor" is "Too many billing guidelines violation, including line items 2-10. Please resubmit per discussion", then the "Comments to Requestor" of the Invoice will show:

*Invoice contains multiple violations.*

*Too many billing guidelines violation, including line items 2-10. Please resubmit per discussion.*

5. Click **Confirm Rejection**. The invoice status changes from **Not Posted** to **Rejected**.

#### 1.7.3.10.2 Sending an Invoice for Review

Depending on system configuration and your user rights, you may have the ability to send approval requests for review to other users.

When you send an invoice for review, the time allotted for you to approve or reject the request is not paused; it continues to run while the invoice is in review. As the approver, you have the option to approve or reject the invoice before it is reviewed. The review simply allows someone who may be more familiar with the project to evaluate and make comments about the invoice.

Invoice Review Requests can be forwarded in batch using the Approvals summary list, or individually using an invoice record.

### To send a batch of invoices for review

1. To access the **My Approvals** page, do one of the following:
  - Click one of the hyperlinks in your **My Approvals** portal pane on your home page.
  - Select **Approvals** from the **All** tab.

The **Requests to Approve** tab of the **My Approvals** screen is displayed.

2. Select the check-box next to the record you want to send for review.
3. Click **Send for Review**.

(Optional) Enter a comment in the **Internal Comment** box.

A pop-up prompts you to specify the user who should review the invoice.

4. Make your selection and click Yes.

The **My Approvals** screen will update and state that you forwarded the invoice to another user for review.

The reviewer will see the request for review on his or her home page under **My Approvals, Requests to Review** tab. The reviewer does not need to post a review in order for you to approve or reject the invoice.

The reviewer's comments will be stored in the Approval History for the invoice, and in the **Workflow** tab.

5. From the **My Approvals** screen, click the invoice record. Under the **Approval History of Active Requests** on the **Workflow** page, you will see the reviewers comments.
6. Make your approval or rejection selection, or send to another user for review.

**Note:** You can send the same invoice for review to multiple users.

### To send an invoice for review (from an invoice)

1. Search for an invoice and open it.
2. From the top of the page, click **Send for Review**.
3. From the **Send for Review** screen, you can:
  - Select the **Recipient Type** (User or Group).
  - Select the **Recipient** from the drop-down list.

- Enter an **Internal Comment**.
4. Click **Send**.

#### 1.7.3.10.3 Reviewing an Invoice

Depending on your role in the organization, and how approval processes are defined, you may receive requests to review invoices that are pending approval.

##### To review an invoice

1. To access the **My Approvals** page, do one of the following:
  - Click one of the hyperlinks in your **My Approvals** portal pane on your home page.
  - Select **Approvals** from the **All** tab.
  - The **My Approvals** screen opens with the **Requests to Approve** tab displayed.
2. Click the **Requests to Review** link from the left pane.
3. Click **Submit Review** on the invoice record row you wish to review.

Enter notes in the **Comment to Requester** field. These comments will be stored in the Approval History for the invoice and on the **Workflow** page.

4. Click **Submit**. The comments will be submitted to the approver.

Alternatively, on the **My Approvals** screen, select the check-box for the record you reviewed, enter any comments in the **Comment** field, and click **submit review**.

**Note:** *The approver has the option to make a decision about the invoice status before the request expires. If the approver makes a decision before you have submitted comments, your request for review will automatically be removed from your pending requests.*

#### 1.7.3.11 Voiding an Invoice

After an invoice has been posted, you can void it.

##### To void an invoice transaction

1. Search for an invoice and open it.
2. Click **Void**.

##### What Happens After You Click Void (no workflow)

The following changes occur when you void an invoice.

- The **Post** button becomes available. The invoice can be posted again if necessary.



- The **Posting Status** changes to **Not Posted** in the **Header Information** section on the **General** page.
- The transaction listed on the **Transactions** page, indicating the posting of the invoice, is removed and replaced with a **Voided** transaction.
- The change in balance is reflected in the **Balance** field in the **Account Summary** section of the **General** page in the corresponding account(s).

### What Happens After You Click Void (with workflow)

- The **Posting Status** changes to **Pending Approval** in the **Header Information** section on the **General** tab.

***Important:** If there is an approval process in place for voiding invoices, NONE of the following items occur until the invoice void is approved by the final individual who must approve it.*

- The **Post** button becomes available. The invoice can be posted again if necessary.
- The **Posting Status** changes to **Not Posted** in the **Header Information** section on the **General** page.
- The transaction listed on the **Transactions** page, indicating the posting of the invoice, is removed and replaced with a **Voided** transaction.
- The change in balance is reflected in the **Balance** field in the **Account Summary** section of the **General** page in the corresponding account(s).

### Voiding Invoices in Other Currencies

To void an invoice that is in a different currency from the default currency, follow the same procedure, and the resulting changes described in the [What Happens After You Click Void \(no workflow\)](#) section still apply.

In addition, the following points are important to remember:

- The amount that was posted is the amount that is voided, regardless of the current exchange rate.
- When you void and then re-post an invoice that is in a different currency than the default currency, the second posting uses the most current exchange rate, which may have changed since the first time the invoice was posted. Therefore, the posted amount may be different.

#### 1.7.3.12 Using the Invoice Summary Breakdowns Page

Use the **Summary Breakdowns** page to view a summary of line items for the selected invoice. You can view summary information in the following ways:

- **By Timekeeper**—View a summary of invoice line items that are in the Fee category and are assigned to a specific timekeeper.
- **By Fee/Expense Category**—View a summary breakdown of invoices grouped by fee or expense category.

You can also use this page to perform adjustments. See [Adjusting an Invoice Summary](#) for more information.

### To view breakdowns by timekeeper

1. Search for an invoice and open it.
2. In the left pane, click **Summary Breakdowns**.
3. At the top of the page, click **By Timekeeper**. The timekeeper summary breakdown information is displayed as follows:
  - All Matters—Lists all timekeepers who are assigned to any matter associated with this invoice, along with their related totals. This section only appears if this is a multi-matter invoice.
  - Individual Matters—Lists each timekeeper under the matter to which they are assigned, along with their related totals.

See [Summary Breakdowns Field Descriptions](#) for specific information about the information in each column.

Timekeeper	Avg Net Rate	Net Units	Total Adj	Net Amt	% of Total Fees
<b>All Matters</b>					
Apple, George	\$75.00	4.00	\$0.00	\$300.00	46.88%
Lexington, Dan	\$60.00	6.00	\$20.00	\$340.00	53.13%
<b>000454-Gravin, Alexander v ABC Inc</b>					
Apple, George	\$75.00	2.00	\$0.00	\$150.00	23.44%
Lexington, Dan	\$60.00	3.00	\$0.00	\$180.00	28.13%
<b>000455-Thompson, Tim v ABC Inc</b>					
Apple, George	\$75.00	2.00	\$0.00	\$150.00	23.44%
Lexington, Dan	\$60.00	3.00	\$20.00	\$160.00	25.00%

Summary Breakdowns Page - By Timekeeper

### To view invoice summary breakdowns by fee or expense category

1. Search for an invoice and open it.
2. In the left pane, click **Summary Breakdowns**.
3. At the top of the page, click **By Fee/Expense Category**. The Fee/Expense summary breakdown information is displayed as follows:
  - All Matters—Lists all fees and/or expense line items for any matter associated with this invoice, along with their related totals. This section only appears if this is a multi-matter invoice.
  - Individual Matters—Lists each fee and/or expense category under its associated matter in a hierarchical representation that includes its associated line items and totals.

See [Summary Breakdowns Field Descriptions](#) for specific information about the information in each column.

Invoice - New Back to Invoices

[Edit](#) [Approve](#) [Send for Review](#) [Reject](#) [Post](#) [Void](#) [Create a Copy](#) [Delete](#) [Share](#) [Printable View](#) [Help](#)

[General](#) | **Summary Breakdowns** | [Transactions](#) | [Documents](#) | [Security](#) | [Workflow](#) | [History](#)

Summary Breakdowns [View Adjustment History](#)

By Timekeeper | By Fee/Expense Category

Fee/Expense Category	Avg Net Rate	Net Units	Total Adj	Net Amt	% of Total
<b>All Matters</b>					
Fee	\$0.00	0.00	\$0.00	\$0.00	0.00%
Outside Counsel Fees	\$0.00	0.00	\$0.00	\$0.00	0.00%
Bankruptcy Code set	\$0.00	0.00	\$0.00	\$0.00	0.00%
B100 Administration	\$0.00	0.00	\$0.00	\$0.00	0.00%
B110 Case Administration	\$0.00	0.00	\$0.00	\$0.00	0.00%
B170 Fee/Employment Objections	\$0.00	0.00	\$0.00	\$0.00	0.00%
Litigation Code set	\$0.00	0.00	\$0.00	\$0.00	0.00%
L100 Case Assessment, Dev and Adm	\$0.00	0.00	\$0.00	\$0.00	0.00%
Expense	\$0.00	0.00	\$0.00	\$0.00	0.00%
Outside Counsel Expenses	\$0.00	0.00	\$0.00	\$0.00	0.00%
E100 Expenses	\$0.00	0.00	\$0.00	\$0.00	0.00%
E101 Copying	\$0.00	0.00	\$0.00	\$0.00	0.00%
E111 Meals	\$0.00	0.00	\$0.00	\$0.00	0.00%
<b>000454-Gravin, Alexander v ABC Inc</b>					
Fee	\$0.00	0.00	\$0.00	\$0.00	0.00%
Outside Counsel Fees	\$0.00	0.00	\$0.00	\$0.00	0.00%
Bankruptcy Code set	\$0.00	0.00	\$0.00	\$0.00	0.00%
B100 Administration	\$0.00	0.00	\$0.00	\$0.00	0.00%
B110 Case Administration	\$0.00	0.00	\$0.00	\$0.00	0.00%
B170 Fee/Employment Objections	\$0.00	0.00	\$0.00	\$0.00	0.00%
Expense	\$0.00	0.00	\$0.00	\$0.00	0.00%

Summary Breakdowns Page - By Fee/Expense Category

### Summary Breakdowns Field Descriptions

The following field descriptions apply to the **Summary Breakdowns** screen for both the **By Timekeeper** and **By Fee/Expense Category** options. Some field descriptions that follow may not display based on system settings and your assigned rights. Field descriptions are listed in alphabetical order and may display in a different order in the application.

#### Summary Breakdowns Fields

Label	Description
<b>% of Total Fees</b>	<ul style="list-style-type: none"> <li><b>By Timekeeper</b> only: The percent ratio of the Net Amount of this row to the Total Net Fee amount for this invoice. (% = Net Amount:Total Net Fee)</li> </ul>
<b>% of Total</b>	<ul style="list-style-type: none"> <li><b>By Fee/Expense Category</b> only: Percent ratio of the Net Amount of this row to the Total Net Amount for this invoice.</li> </ul>
<b>Avg Net Rate</b>	<ul style="list-style-type: none"> <li><b>By Timekeeper</b>: The average of the Original Rate of the grouped line items for this timekeeper.</li> <li><b>By Fee/Expense Category</b>: The average of the Original Rate of the grouped line items for this category. For an Expense category, appropriate decimal places are displayed.</li> </ul>
<b>Net Amt</b>	<ul style="list-style-type: none"> <li><b>By Timekeeper</b>: The sum of the total amounts of the grouped line items for this timekeeper. The tax amount is excluded.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>By Fee/Expense Category:</b> The sum of the total amounts of the grouped line items for this category. The tax amount is excluded.</li> </ul>
<b>Net Discount</b>	<ul style="list-style-type: none"> <li>• <b>By Timekeeper:</b> The total of all of the Original Discounts of the grouped line items for this timekeeper.</li> <li>• <b>By Fee/Expense Category:</b> Total of the Original Discounts of the grouped line items for this category.</li> </ul>
<b>Net Units</b>	<ul style="list-style-type: none"> <li>• <b>By Timekeeper:</b> The total of the Original Units of the grouped line items for this timekeeper.</li> <li>• <b>By Fee/Expense Category:</b> The total of the Original Units of the grouped line items for this category.</li> </ul>
<b>Total Adj</b>	<ul style="list-style-type: none"> <li>• <b>By Timekeeper:</b> The sum of the adjustments of the grouped line items for this timekeeper.</li> <li>• <b>By Fee/Expense Category:</b> The sum of the adjustments of the grouped line items for this category.</li> </ul>

#### 1.7.3.12.1 Adjusting Net Amounts on the Summary Breakdowns Page

Use the **Summary Breakdowns** page to adjust the net amount for specific timekeepers or categories. Once you make an adjustment, it is distributed across the affected line items.

For example, on the **By Timekeeper** page, on Invoice A, the total amount of the fee line items for Timekeeper Dave is \$1,000.00. If you adjust the amount to \$800.00, and then save this information, the adjusted amount of \$200.00 is distributed across the fee line items that apply to timekeeper Dave based on the ratio of each line item amount over the total fee amount prior to the adjustment.

If there are expense line items that also apply to Timekeeper Dave, they are not affected, because the adjustment is made on Fee line items only.

#### To adjust net amounts

1. Search for an invoice and open it.
2. In the left pane, click **Summary Breakdowns**.
3. At the top of the page, click **By Fee/Expense Category** or **By Timekeeper**.
4. In the **Net Amount** column, click the amount that you want to adjust.

An Adjustment window opens.

5. Enter the appropriate adjustments and information, and then click **Save**. See [Adjusting an Invoice Summary](#) for specific information.

The adjustment is distributed across affected line items.

### 1.7.3.13 Viewing and Creating History for an Invoice

Use an invoice's **History** page to create a history (containing notes about changes to the invoice), view a list of history records for that invoice, or perform an advanced search for a history.

For more information, see [Working with Record History](#).

#### 1.7.3.13.1 Viewing Invoice Summary Adjustment History

You can view a history of Invoice Summary adjustments. For each adjustment, the following information displays:

- The adjustor's name
- Descriptions of the adjustments made
- Adjustment date

#### To view an invoice summary adjustment history

1. Search for an invoice and open it.
2. From the **Invoice Summary** section, under the **Amount** column, click a numeric value link.
3. From the resulting adjustment screen, click **Adjustment History**.
4. The resulting **Invoice Adjustment History** screen displays the adjustor name, adjustment type, adjustment amount, and date of adjustment.
5. Click the **Cancel** button to return to the invoice **General** page.

#### 1.7.3.13.2 Viewing Line Item Adjustment History

Within each invoice, you will find a history of any invoice adjustments and line item adjustments. For each adjustment, the following information displays:

- Adjustment dates
- The user who made the adjustment
- Descriptions of the numeric adjustments made
- In-house comments
- Adjustment reasons
- Comments to vendor

#### To view an invoice line item's adjustment history

1. Search for an invoice and open it.
2. From the **Line Items** section, under the **Item** column, click an item number link.
3. From the line item record left pane, click the **Adjustment History** link.

4. A history of that line item's adjustments displays.

#### 1.7.3.13.3 Viewing Invoice Transactions History

Posted invoice line item amounts are recorded on the corresponding **Transactions** page of the invoice. Transactions are displayed in the default currency, regardless of the currency of the invoice.

You cannot delete or modify invoice transactions on the **Transactions** page. If changes are necessary, you must void the invoice, modify the invoice line items, and post the invoice again.

For more information on invoice transactions, see [Posting an Invoice](#), [Voiding an Invoice](#), and [Setting Transaction Posting Criteria](#).

#### To view invoice transactions

1. Search for an invoice and open it.
2. In the left pane, click **Transactions**.
3. View the transaction records.

#### 1.7.3.13.4 Viewing Invoice Workflow

You can view information about an invoice's posting status from both the General page and Workflow page.

- From an invoice's **General** page, the workflow status of an invoice displays at the top.
- From an invoice's **Workflow** page, you can view information about an invoice's active and past approval processes.
  - From the **Active Requests** section, you can find the approver(s) for an action that is pending approval.
  - From the **Completed Requests** section, you can review the history of approvals, rejections, and canceled approval processes for the invoice.

The table below describes the possible invoice statuses and related adjustment options. Manual Invoice refers to an invoice created in TeamConnect. CSM Invoice refers to an invoice that was transferred from Collaborati to TeamConnect.

#### Invoice status messages and descriptions

Invoice Status	Description	Can an Invoice be Adjusted in this State?
<b>Not Posted</b>	The invoice is not posted.	<ul style="list-style-type: none"> <li>• Manual Invoices—Yes</li> <li>• CSM Invoices—No</li> </ul>

		<b>Note:</b> CSM Invoices' custom fields can be adjusted in this state
<b>Posted</b>	The invoice has been posted.	<ul style="list-style-type: none"> <li>• Manual Invoices—No</li> <li>• CSM Invoices—No</li> </ul> <b>Note:</b> All invoices' custom fields can be adjusted in this state
<b>Pending Approval</b>	The invoice has been submitted for posting and is awaiting appropriate approvals before it can be posted.	<ul style="list-style-type: none"> <li>• Manual Invoices—Yes</li> <li>• CSM Invoices—Yes</li> </ul>
<b>Rejected</b>	The invoice has been rejected in the approval process.	<ul style="list-style-type: none"> <li>• Manual Invoices—Yes</li> <li>• CSM Invoices—No</li> </ul> <b>Note:</b> CSM Invoices' custom fields can be adjusted in this state
<b>Failed</b>	The invoice failed while attempting to post.	<ul style="list-style-type: none"> <li>• Manual Invoices—Yes</li> <li>• CSM Invoices—No</li> </ul> <b>Note:</b> CSM Invoices' custom fields can be adjusted in this state

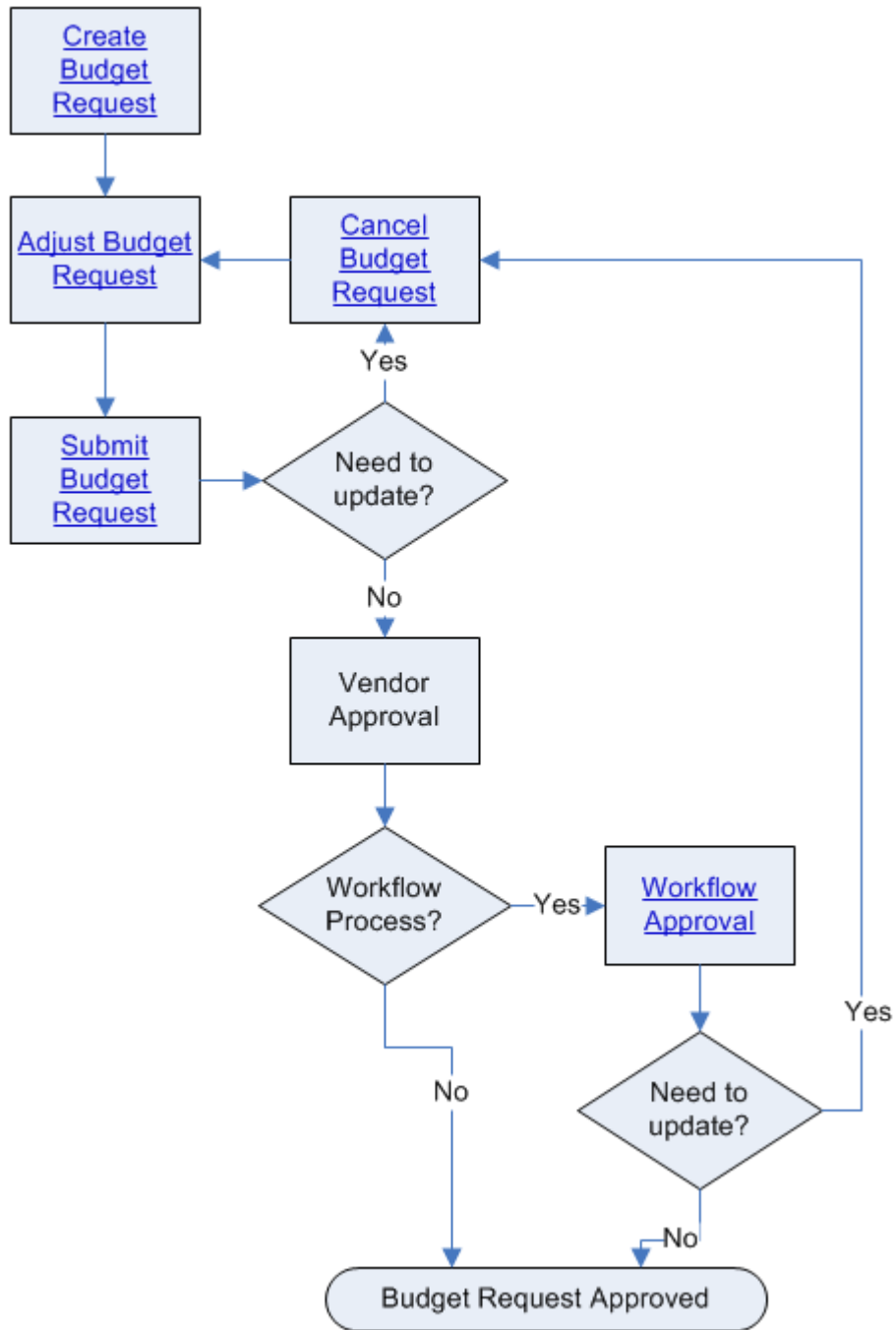
#### 1.7.4 Budget Requests

If you are using the e-billing features of Collaborati Spend Management (CSM), you can send budget requests to vendors for approval. Each combination of vendor and project (a Matter or Dispute) uses one budget request. The cost estimates for an approved budget post to the appropriate Account records for that project.

After a vendor receives a budget request and enters their cost estimates, they submit the request back to you, the client. When CSM receives the budget request, it can go through an approval workflow, much like invoice approvals do. Instructions for handling workflow are found in [Workflow](#).

Depending on your CSM configuration, TeamConnect can automatically generate and send budget requests to vendors when activating the accounts for a matter. See the CSM Administration Guide for this configuration process.

You create budget requests manually or you receive them from Collaborati. If you receive them from Collaborati, see [Budget Requests Created by Vendors](#) for more information. If you create budget requests manually, refer to the following flowchart. Click areas of this flowchart for more information.



You can access budget requests from several locations in TeamConnect:

- CSM Settings
- A portal pane in your home page
- A Matter or project record
- A Vendor record



- Pending approvals
- Budgets Requests link under Finance

#### 1.7.4.1 Creating a Budget Request

Budget requests are often created automatically, but you can create them manually if necessary.

See the *Collaborati Spend Management Configuration Guide* for information about the settings that affect automatic budget request creation.

##### To create a new budget request

1. Select one of the following options:
  - On the **Budget Requests** page, click the **New** button.
  - Click the **New Budget Request** link from the CSM Settings page or from a Vendor record page.
2. Complete the general fields. See [Budget Request Fields table](#) for field information.
3. Enter the amounts you want to allocate for the account in the **Starting Amount** column.
4. After you complete the general fields, click **Save**.

The budget names information in a new budget request does not become editable until the general information is saved.

Finally, the **Notes** field allows you to enter instructions or comments that will be visible to you and the vendor.

### Budget Request Field Descriptions

The tables below contains Budget Request field descriptions. The fields that you see may vary based on your system settings and assigned rights.

#### Budget Request Fields

Field	Description
<b>Name</b>	Assign a name to this request. It is helpful to have matter identification as part of the name, and to use a name that will be familiar to the vendor.
<b>Vendor</b>	Choose a vendor for the request. (If you clicked the New Budget Request from a Vendor record page, this field is pre-populated for you.) Auto-suggest and search features are available for this field.
<b>Matter</b>	Choose a matter for the request. (If you clicked the New Budget Request from a Matter record page, this field is pre-populated for you.) Auto-suggest and search features are available for this field.

<b>Request Date</b>	Type in a date value, or click the calendar icon to use the date picker.
<b>Due Date</b>	Type in a date value, or click the calendar icon to use the date picker.
<b>Status</b>	This field is not editable. The initial status for all new budget requests is <b>Not Submitted</b> .

#### 1.7.4.1.1 Submitting a New Budget Request

Budget requests are submitted by changing the phase of the request to **Submit**. The request will be sent to the vendor during the next synchronization performed by Collaborati Spend Management.

#### 1.7.4.1.2 Budget Requests Created by Vendors

Under some conditions, vendors using Collaborati are allowed to rebudget a matter-to create a new budget request for the matter. Although uncommon, it is possible that the vendor is doing a rebudget process at the same time that you are creating a budget request for the same matter. This conflict will be detected during synchronization, and handled as follows:

- If you have created a budget request, but have not submitted it, the vendor's budget request will take precedence over the one that you created. Your budget request will be changed to status "Cancelled".
- If you have created a budget request and submitted it, yours will take precedence over the one that the vendor just created. The vendor's budget request will be changed to status "Cancelled". Your budget request will appear in Collaborati.
- If you recently deleted one or more matter accounts, those accounts will be removed from any vendor-created budget request during synchronization. The budget request itself will remain open.

#### 1.7.4.2 Editing a Budget Request

A budget request can be edited when its status is **Not Submitted** or **Pending Approval**.

**Vendor** and **Matter** cannot be edited after the request is created.

**Name** and **Request Date** can only be edited if the phase is **Not Submitted**.

##### 1.7.4.2.1 Adjusting Budget Requests

The Budget Account section lets you make changes to budget requests while in the workflow process.

#### To adjust budget requests

1. Open the appropriate budget request.
2. In the left pane, click the **Budget** page.

3. The **Budget Accounts** section displays all budget accounts associated with the matter that appears in the **Matter** field.
4. Adjust budget amounts in the **Starting Amount** field.

You can only make changes to budget accounts if the budget request status is **Not Submitted**.

The **Budget** and **Period** columns are always visible, but the visibility of other columns in this section vary based on the status of the request, the stored value, and whether or not you are in edit mode. Additionally, the amounts in a field may shift to different field when you save the budget request.

#### Budget Accounts Section

Field	Description
<b>Budget</b>	The name of the budget account associated with the matter. This field is not editable.
<b>Period</b>	The period start and end date for this account. This field is not editable.
<b>Starting Amount</b>	The amount proposed by the client. In edit mode, you can revise the information in this field if the current status is <b>Not Submitted</b> . This field is always visible and is populated when a budget request is created automatically or manually.
<b>Vendor Proposed</b>	The amount proposed by the vendor in Collaborati. This field is not editable. This field is blank when a budget request is created automatically.
<b>Previously Adjusted</b>	The amount that may be adjusted by a user, only during workflow. An amount may appear in this field if the status is Pending Approval and at least one budget account has a value for that field. This field is not editable, and is blank when a budget request is created automatically.
<b>Accepted Amount</b>	The amount suggested by the client after the request is submitted. This amount overrides the current value. If a budget request is in Approved status, this column always displays a value. This value may be from the <b>Accepted Amount</b> , <b>Vendor Proposed</b> , or <b>Starting Amount</b> column, depending on which of these columns displays a value. This field is not editable, and is blank when a budget request is created automatically.

<b>New Accepted</b>	<p>The new amount that is suggested by the client. This amount is saved to the <b>Accepted Amount</b> field after the request is submitted.</p> <p>For a new budget request, this field is blank. If an amount is entered in this field, and you click <b>Save</b>, the following happens:</p> <ul style="list-style-type: none"><li>• If the amount in the <b>Accepted Amount</b> field was saved by an approver other than the approver who is currently logged in, this amount moves into the <b>Previous Adjusted Amount</b> field.</li><li>• The amount in the <b>New Accepted</b> field moves into the <b>Accepted Amount</b> field</li></ul> <p>In edit mode, you can revise the information in this field if the current status is <b>Pending Approval</b>.</p>
---------------------	---

### Submitted, Approved, or Rejected Status

After a budget request is submitted, the following applies:

- If the status is **Submitted**, only the **Starting Amount** column is visible.
- If the status is **Approved**, only the **Starting Amount**, **Vendor Proposed**, and **Accepted Amount** columns are visible and accounts are adjusted.
- If the status is **Rejected**, only the **Accepted Amount** column is visible.

Final values are passed back to the Collaborati vendor.

#### 1.7.4.3 Canceling a Budget Request

A budget request can have its phase changed to **Cancel** if its status is presently one of the following:

- **Submitted**
- **Vendor Review**
- **Pending Approval**
- **Rejected**
- **Error**

If a **Submitted** budget request is canceled, the status becomes **Not Submitted**.

In all other cases, the status becomes **Canceled**.

#### 1.7.4.4 Deleting a Budget Request

A budget request can be deleted if its status is presently one of the following:

- **Not Submitted**
- **Approved** [and Synchronized]

- **Canceled** [and Synchronized]

#### 1.7.4.5 Handling Budget Request Workflow

Budget requests that are returned from a vendor become pending approvals in workflow, in the same manner that invoices are subject to approval. For details about how to approve or reject a workflow item, see [Workflow](#).

Workflow information about a budget request is available from the **Workflow** link while viewing a specific request. The request also has a **Vendor Notes** field that may contain comments from the vendor.

When you approve a budget request, the **Current Budget** amounts in the request are posted to the related Account records. When you reject a budget request, its status becomes Rejected, and the rejection is sent to the vendor during the next Collaborati Spend Management synchronization.

#### 1.7.5 Updating Exchange Rates

The Exchange Rate tool lets you update exchange rates for the currencies that you commonly use. An exchange rate is the value of a foreign nation's currency in terms of the home nation's currency. In TeamConnect, it is the value of the foreign currency based on the selected default currency.

Once you open the Exchange Rate tool, use one of the following options to update exchange rates:

- **Upload a CSV or Excel file**—Create a data file that contains the exchange rates that you want to change, and then upload the file to TeamConnect.
- **Enter exchange rates manually**—Type the modified rates on the Exchange Rates page.

**Note:** Typically, currency codes are set up in advance and you cannot delete or change them. You cannot use this tool to add new currency codes. See your system administrator if you want to add or delete currency codes.

##### To access the Exchange Rates tool

1. Click the **All** tab.
2. Under **Tools**, click **Exchange Rates**.

The **Exchange Rates** page opens.

**Exchange Rates**

[Update](#) [Help](#)

**Default Currency:** USD United States, Dollars

To update Exchange Rates, upload a spreadsheet or enter values in table below. Exchange Rates are specified with respect to the Default Currency e.g. for Default Currency USD, if 1 EUR is equal to 1.49 USD then the Exchange Rate is 1.49.

Upload Exchange Rates:  [Browse...](#) (.csv or Excel files) [Download Sample File](#)

[Upload](#)

Exchange Rates 1 - 7 of 7

Name	Currency Code	Exchange Rate	Symbol	Last Modified
Belize, Dollars	BZD	<input type="text" value="1.51317"/>		3/16/2010 11:41 PM
Canada, Dollars	CAD	<input type="text" value="0.984155"/>		3/16/2010 11:41 PM
Dirhams	AED	<input type="text" value="1.37229"/>		3/16/2010 11:41 PM
Guilders	ANG	<input type="text" value="0.914561"/>		3/16/2010 11:41 PM
Jamaica, Dollars	JMD	<input type="text" value="0.33234"/>		3/15/2010 5:14 PM
Jordan, Dinars	JOD	<input type="text" value="0.23451"/>		3/15/2010 5:14 PM
United Kingdom, Pounds	GBP	<input type="text" value="1.2345"/>	£	3/15/2010 5:14 PM

Exchange Rates Page

## Creating/Uploading Exchange Rate Data Files

Before you can upload exchange rates into TeamConnect, you must first create the file that contains the data that you want to upload. You can upload an Excel spreadsheet or a CSV (.csv) file.

### To create an exchange rate data file

Create an Excel spreadsheet or CSV file based on the following guidelines:

- The first row must contain labels only.
- Additional rows must contain exchange rate data only.
- The first column must contain currency codes only.
- The second column must contain exchange rates only.
- The file extension must be .xls, or .csv.

**Note:** TeamConnect provides a sample Excel file. To access this file, click **Download Sample File**. The sample file opens in Excel and you can use it to enter the information that you want to change.

	A	B
1	CURRENCY CODE	EXCHANGE RATE
2	GBP	1.51317
3	CAD	0.984155
4	EUR	1.37229
5	AUD	0.914561
6		

Sample Excel File

```
CURRENCY CODE, EXCHANGE RATE
GBP, 1.51317
CAD, 0.984155
EUR, 1.37229
AUD, 0.914561
```

Sample CSV File

## Uploading Exchange Rate Data Files

Once you create the data file, you can upload it to TeamConnect.

### To upload exchange rate data

1. With the Exchange Rates page open, click **Browse** and locate the file that you want to upload.
2. Click **Upload**.

The **Exchange Rate** column is updated to display the modified rates.

**Note:** You can only update exchange rates for currency codes that currently exist. If you need to add additional currency codes, contact your system administrator.

## Updating Exchange Rates Manually

You can manually update exchange rates on the **Exchange Rate** page.

### To update exchange rates manually

1. In the **Exchange Rate** column, modify the appropriate exchange rates.
2. Click **Update** to save the changes.

## 1.8 Documents

A document is any type of item that is added to TeamConnect and is accessible on a **Documents** page, such as a Microsoft Word document, a Microsoft Excel spreadsheet, image files, text files, and email messages.

A document can also be a reference, such as a link to a Web page or a shortcut link that opens a document in a different location.

The following table shows how you can view, add, and update documents.

...Documents	How...
Viewing	<ul style="list-style-type: none"> <li>• <a href="#">Searching for Existing Documents</a></li> </ul>

	<ul style="list-style-type: none"> <li>• <a href="#">Documents and Global Search</a></li> <li>• <a href="#">Viewing Documents</a></li> <li>• <a href="#">Opening Documents</a></li> <li>• <a href="#">Sharing Documents using Email</a></li> <li>• <a href="#">Exporting a List of Documents in a Folder</a></li> <li>• <a href="#">Downloading Documents</a></li> <li>• <a href="#">Viewing Document Versions</a></li> <li>• <a href="#">Reverting to a Previous Version</a></li> <li>• <a href="#">Viewing or Creating Documents History</a></li> </ul>
Adding	<ul style="list-style-type: none"> <li>• <a href="#">Adding a Document to TeamConnect</a></li> <li>• <a href="#">Copying Documents</a></li> <li>• <a href="#">Creating a Document Shortcut</a></li> <li>• <a href="#">Creating a Link to a Web Page</a></li> <li>• <a href="#">Generating Documents Using TeamConnect Data</a></li> </ul>
Updating	<ul style="list-style-type: none"> <li>• <a href="#">Checking Out a Document</a></li> <li>• <a href="#">Checking In a Document</a></li> <li>• <a href="#">Canceling a Document Check Out</a></li> <li>• <a href="#">Viewing or Editing General Information of a Document</a></li> <li>• <a href="#">Adding and Removing Document Categories</a></li> </ul>
Organizing	<ul style="list-style-type: none"> <li>• <a href="#">Creating a Document Folder</a></li> <li>• <a href="#">Moving Documents</a></li> <li>• <a href="#">Deleting Documents and Document Folders</a></li> </ul>

## Document Property Pages

Each document has the following document-specific property pages on which you can view and edit information:














- **General**—View the name, type, and check-out information for a document.
- **Version**—View the various versions of a document. A new version of a document is created each time a document is checked out and then checked back in. You can also use this page to revert to an earlier version of a document if necessary.

There are also additional document property pages that you can use to enter supplementary information, such as security rights for a selected document.



## Document Task Buttons and Icons

You can perform the same documents-related tasks on both the global **Documents** page and the **Documents** page for a record. Sometimes the icons that you use to perform tasks are located in different areas, depending on the page.

Global Documents Page (Record Toolbar)	Documents Page for a Record (Task Toolbar)	Task
 Upload File		<b>Upload a file:</b> Adds a new file to TeamConnect. See <a href="#">Adding a Document to TeamConnect</a> for more information.
 New Folder		<b>Create a new folder:</b> Creates a folder to hold related documents. See <a href="#">Creating a Document Folder</a> for more information.
 New URL		<b>Create URL:</b> Creates a link to a Web page. See <a href="#">Creating a Link to a Web Page</a> for more information.
 More Actions ▾		<b>Move Document:</b> Moves a document from one location in TeamConnect to a different location in TeamConnect. See <a href="#">Moving Documents</a> for more information.
		<b>Copy Document:</b> Copies a document from one location in TeamConnect to a different location in TeamConnect. See <a href="#">Copying Documents</a> for more information.
		<b>Create Shortcut:</b> Creates a link from your <b>Documents</b> page to a document in a different location in TeamConnect. See <a href="#">Creating a Document Shortcut</a> for more information.
		<b>Generate Document:</b> Creates a form document that uses information currently in TeamConnect, such as the name and address of the recipient of the document. See <a href="#">Generating Documents Using TeamConnect Data</a> for more information
 Delete		<b>Delete:</b> Removes a selected document. This document cannot be recovered.

		See <a href="#">Deleting Documents and Document Folders</a> for more information.
		<b>Help:</b> Opens the help for the selected <b>Documents</b> page.
		<b>Find:</b> Searches for documents by name, specified search criteria, or text that may be in the document. See <a href="#">Searching for Existing Documents</a> .
		<b>View/edit document properties:</b> Opens the <b>General</b> , <b>Versions</b> , <b>Categories</b> , <b>Security</b> , and <b>History</b> property pages. Find this icon in the <b>Action</b> column.
		<b>Check out a document:</b> Downloads the document to your computer to edit it. Find this icon in the <b>Action</b> column. See <a href="#">Checking Out a Document</a> .
		<b>Check in a document:</b> Returns a document to TeamConnect when you are done editing it. Find this icon in the <b>Action</b> column when the related document is checked out. See <a href="#">Checking In a Document</a> .
		<b>Undo Check-out:</b> Returns a checked-out document to unchecked-out. Find this icon in the <b>Action</b> column when the related document is checked out. See <a href="#">Canceling a Document Check Out</a> .

### 1.8.1 Searching for Existing Documents

You can locate documents or folders by performing a search using all or part of a document or folder name, text contained in the document, or the name of the person who created the document.

You can search documents in the following ways:

- **Custom Search**—Search for a document using multiple field values as criteria. For example, you can search for documents by name or date on which the document was created.
- **Record Search**—Opens a records **Documents** page to search for specific documents.

Before you add a document to TeamConnect, it is recommended that you perform a search to ensure that the document is not already available.

#### To search for a document using Custom Search

1. Click the **Documents** tab.
2. Click the **Custom Search** link. This link is located in the body of the list view in the right corner.
3. Select your search criteria, and then click **Search**.

The documents that match your criteria appear in a list view.

#### To search for a document within a specific record

1. Open the **Documents** [page for a record](#).
2. Click the **Search Documents** icon.
3. If necessary, in the **View** drop-down list, select a view type.
4. Select an **Operator** and/or **Value**, and then click **Search**.

The documents that match your criteria appear in a list view.

**Note:** The criterion **Document Body** will search the content of the document if that document is in a file format that is recognized by TeamConnect.

#### Tip

If you do not see the document that you want, click **Search Again** and expand or narrow your search criteria. For example, use "Contains" as a search criteria instead of "Begins With", "Ends With", or "Equal To".

For more information about search fields and search criteria, see [Searching for Specific Records](#).

## 1.8.2 Documents and Global Search

### Documents and Global Search

Users with Global Search enabled are able to search on both indexed documents, their metadata, and the contents of their attachments. Documents present on the TeamConnect instance but not indexed will not deliver full-text search capabilities. **Note:** For more information on indexing, see *Global Search Index Tool Help*.

## Extensions

When using Global Search, supported attachments that have matching metadata keywords or matching content will be returned. The following attachment extensions are supported:

- HTML/XHTML
- XML
- Email files (.eml, .msg)
- Microsoft Office File Formats (.xls, .xlsx, .doc, .docx, .pptx, .one, .pub, .mdb)
- PDF
- RTF, TXT
- ODF
- EPUB
- JAR

## Documents and File Size

Admins are able to configure the maximum size for documents to be indexed as well as searched. If a document that exceeds the preset value (e.g., a 100MB document with a 50MB maximum size limit), full-text searching will not be available. However, the document keywords and metadata will still return results (e.g., document title, author, etc.).

If documents are indexed in a batch and only one exceeds the maximum value, all documents within the size parameters will be uploaded and the larger file will not. There will not be an error message on the screen; however, users with logging turned on will be able to see the failed document in their logs.

### 1.8.3 Viewing Documents

You can view documents on the global **Documents** page or on the **Documents** page for a specific record type. To open a document, click its link on any page.

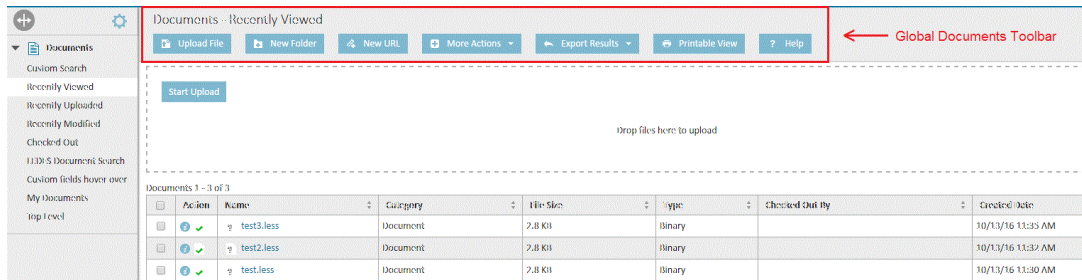
Additionally, if you receive an email message that contains a link to a document or document folder, clicking on the link immediately opens the document or document folder. This depends on whether or not you are logged into TeamConnect and have the appropriate rights.

#### Viewing the Global Documents Area

Based on your access rights, the global **Documents** area allows you to access all of the documents that you have the rights to view in TeamConnect. You can also use this page to subscribe to a collection of documents that you want to track, such as all documents that are currently checked out for editing, or all documents that were recently added to TeamConnect.

#### To view the global Documents area

1. Click the **Documents** tab.
2. Use the Global Documents Toolbar to perform most document-related tasks.



Global Documents Page

## Viewing the Documents Page for a Record

Several record types have a documents page where you can manage documents that relate to the specified type of record. You can add many types of records to a Documents page, such as court documents, letters, and presentations.

The following record types have a **Documents** page:

- Accounts
- Appointments
- Contacts
- Expenses
- Invoices
- Projects
- Tasks

Related record types **Involved** and **History** also have **Documents** pages, but those record types must be accessed through one of the record types listed above.

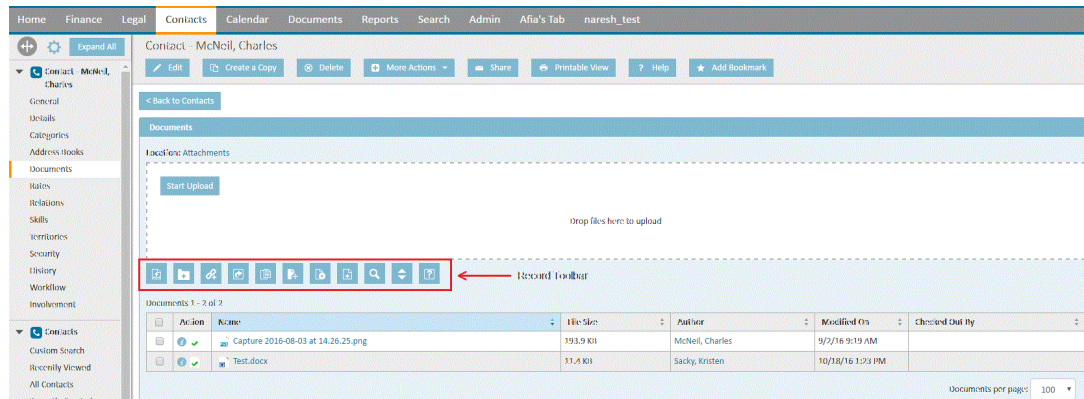
### To view the Documents page for a record

1. From your menu, select one of the following options to access the **Documents** page:
  - For **Accounts**—Click the **Finance** tab, and then click the **Accounts** link under the tab bar. In the **Name** column, click the appropriate account.
  - For **Appointments**—Click the **Calendar** tab, and then click the **Appointments** link under the tab bar. Switch to your preferred calendar view and then click on the appropriate appointment.
  - For **Contacts**—Click the **Contacts** tab, and then click a contact in the **Name** column.
  - For **Expenses**—Click the **Finance** tab, click the **Expenses** link under the tab bar, and then click a link in the **Description** column.

- For **Invoices**—Click the **Finance** tab, click the **Invoices** link under the tab bar, and then click a link in the **Invoice Number** column.
- For **Projects**—Click the **All** tab, and then click the project type that contains the appropriate project. When the project type window opens, click a project in the **Project** column.
- For **Tasks**—Click the **Calendar** tab, click the **Tasks** link under the tab bar, and then click a link in the **Subject** column.

The selected record opens to its **General** page.

2. In the left pane, click the **Documents** page link.



Documents Page for a Contact

### 1.8.3.1 Opening Documents

You can open any documents that you created, all public documents, documents to which you have the appropriate rights, and document folders. Once you open a document, you can save it or print it. Once you open a document folder, you can manage the documents to which you have the appropriate rights.

You can also open a document to edit it if you have the related software. See [Editing Documents](#) for more information.

**Note:** If the document does not open, you may not have its application installed on your computer. For example, if you want to open a .pdf file, you must have Adobe Reader installed on your computer.

#### To open a document

You can use the following options to open documents:

- On any **Documents** page for a record, or global **Documents** page, locate the document that you want to open and click its link in the **Name** column. The document opens in a separate window.

See [Searching for Existing Documents](#) for more information.

- If available, the **Open in Explorer** link opens the Explorer Web browser that links to the Attachment Root directory of the current record. You can then use the browser to manage your TeamConnect documents. See [Using Windows Explorer with Documents \(WebDAV\)](#) for more information.

### 1.8.3.2 Sharing Documents using Email

TeamConnect provides a **Share** button that lets you cut-and-paste a document URL into an email message. Once the email message is received, the recipient can click the URL to access the TeamConnect document or document folder.

#### To share documents in email

1. Select one of the following options:
  - **Documents** page for a record—Open a record, and then click the **Documents** link in the left pane.
  - Global **Documents** tab—Click the **Documents** tab. If necessary, in the left pane, click the link that contains the document that you want to share.
2. Click the **Properties** icon for the document that you want to share.

The **General** page opens for the selected document.
3. Click the **Share** button.

The **Share Record** window opens. This window contains the document URL of the document that you want to share.
4. In the **Share Record** window, select the entire URL.
5. Copy the link.

***Tip:** To copy, with the URL highlighted, right-click and select **Copy**.*

6. Paste the link into an email message.

#### To open a document URL received in an email message

If you are logged on to TeamConnect and have the appropriate rights, clicking a document URL in an email message opens the document in a browser window.

If the document URL links to a document folder or a document that contains other files (such as a .zip file), clicking the link opens the TeamConnect **Documents** page where the folder or file resides.

***Note:** If you are not logged-in to TeamConnect, the login page opens. Log in to access documents. Browser settings determine the way that documents are displayed. For*

*example, if the browser is set to open a new tab whenever a document is opened, clicking a document URL link always opens the document in a separate tab.*

## 1.8.4 Using Documents Collections

The global **Documents** area provides easy access to all of the documents in TeamConnect without having to search for them. Document collections let you view specific groups of documents, such as all of the documents that were recently added to TeamConnect or all of the documents that are currently checked out.

### To view and open Documents collections

1. Click the **Documents** tab. The list view opens and collections are listed in the left pane.
2. To open a collection and view its contents, click its link in the left pane.

By default, the following document collections are available in the left pane:

- **Recently Viewed**—Displays the last 50 documents that you opened and viewed. You cannot delete this collection or change its order. It is always the first collection listed in the left pane.
- **Recently Uploaded**—Displays all of the documents that were added to TeamConnect within the last 30 days.
- **Recently Modified**—Displays a list of documents that were edited within the last 30 days.
- **Checked Out**—Displays a list of documents that are currently checked out of TeamConnect.
- **My Documents**—Displays a list of documents that you added to TeamConnect, or document references that you created. Since you are the default author of these documents, there is no **Author** information displayed.
- **Top Level**—Displays the TeamConnect file structure:
  - **Attachments** navigates to the **Documents** page for a project, appointment, contact, history, task, account, expense, or invoice record.
  - **Users** navigates to the personal **Documents** folders for all TeamConnect users.
  - **System** navigates to the **Documents** page for system folders.

In addition to the default document collections, there may be other collections to which you can subscribe. When you subscribe to a collection, its link appears in the collection list and you have access to its contents. You can change the order in which your collections are listed and remove collections that you no longer need.

## 1.8.5 Managing Documents

A document in TeamConnect is any type of item that is added to TeamConnect and is accessible on a **Documents** page, such as a Microsoft Word document, a Microsoft Excel spreadsheet, image files, and text files.



A document is also a reference to a document, such as a link to a Web page, or a shortcut link that opens to a document in a different location.


You can perform the following tasks to manage documents in TeamConnect:

- [Adding a Document to TeamConnect](#)—Upload a document to a Documents page.
- [Editing Documents](#)—Edit a document by first checking it out, and then checking it back in when you are done editing it.
- [Creating a Document Folder](#)—Create a folder to hold related documents.
- [Exporting a List of Documents in a Folder](#)—Export a list of documents to an Excel spreadsheet or as a .pdf file.
- [Copying Documents](#)—Copy documents from one location in TeamConnect to a different location in TeamConnect.
- [Moving Documents](#)—Move documents from one location in TeamConnect to a different location in TeamConnect.
- [Creating a Document Shortcut](#)—Create a bookmark that allows you to access a document that is in a different location in TeamConnect.
- [Creating a Link to a Web Page](#)—Create a link that allows you to access a Web page.
- [Downloading Documents](#)—Copy a document from TeamConnect to your computer.
- [Generating Documents Using TeamConnect Data](#)—Use the Document Generator to create a form document, such as a standard status letter that you send to clients.
- [Using Windows Explorer with Documents \(WebDAV\)](#)—Use the Windows Explorer browser to view, copy, move, edit, and delete TeamConnect documents.

The tasks that you can perform depend on your assigned rights.

#### 1.8.5.1 Adding a Document to TeamConnect

To add a new document to a **Documents** page, you upload it from a local computer to TeamConnect.

This can be achieved via two methods, either the Upload File icon  or drag-and-drop file addition.

##### To add a document to TeamConnect via Drag-and-Drop

If adding multiple files, each file will be restricted by the maximum upload size set by your admin; the aggregate size of the documents can exceed the limit. Likewise, if a single upload exceeds the maximum, the files within the upload size limit will not be impeded.

1. Click the box for drag-and-drop functionality, or simply drag-and-drop to begin the upload.
2. After uploading the files to the open box area, click **Start Upload** to begin the process.

The image below demonstrates two files successfully added for upload and one which exceeds the administrator maximum file size.



A document in TeamConnect may be many different types, such as a word-processed file, a spreadsheet, image files, text files, or email messages. A document is also a reference to a document, such as a link to a Web page, or a shortcut that opens to a document in a different location.

After you add a document to TeamConnect, you are listed as its author. As the author, you can edit or delete your documents and assign security rights that deny or allow other users to access it.

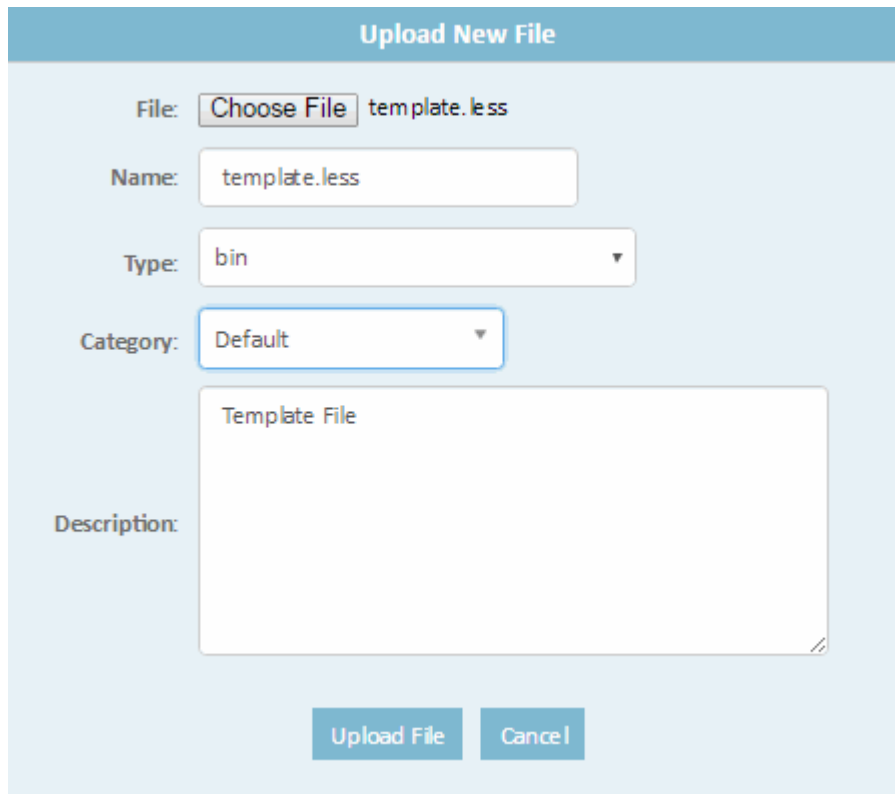
If you added the document using the **Documents** tab, all of the documents that you add are automatically listed in the **My Documents** collection, unless you navigate to a specific record. See [Using Documents Collections](#) for more information.

**Important:** Your system administrator can limit the size of documents that you can add to TeamConnect. An error message appears if you exceed the limitation. The size of the documents that you can add also depends on the amount of memory that is currently available on your company's server. You may have difficulty adding documents if the server memory is low. Contact your system administrator if you are having difficulty adding documents.

### To add a document to TeamConnect via File Upload

1. Open the **Documents** page where you want to add the new document. If you want to add a document to a folder, click the folder to open it.
2. Click the **Upload File** button or icon. Likewise, users can [drag-and-drop](#) files for upload. After adding the desired files, click **Start Upload** to begin the process. See [Sharing Documents using Email](#) for more information.

The **Upload New File** window opens.



The screenshot shows a dialog box titled "Upload New File". It contains the following fields and controls:

- File:** A button labeled "Choose File" followed by the text "template.less".
- Name:** A text input field containing "template.less".
- Type:** A dropdown menu with "bin" selected.
- Category:** A dropdown menu with "Default" selected.
- Description:** A large text area containing "Template File".
- Buttons:** "Upload File" and "Cancel" at the bottom.

Upload New File Window

3. Click **Browse** to locate the file that you want to add.
4. In the **Name** field, enter the name of the document as you want it to appear on the **Documents** page. The name defaults to the current file name if you do not change it.
5. In the **Type** drop-down list, click the type of file that you are adding. TeamConnect selects the file type if you do not select one. See the [Supported Document Types](#) for more information.  
  
If you do not see the file type that you want, speak to your system administrator to see if it can be added.
6. (optional) In the **Category** drop-down list, select the category that applies to this document.
7. (optional) In the **Description** box, enter a description of the file.
8. Click **Upload File**.
  - If you logged on using your personal login id and password, and you are on the **Documents** tab, the file is visible in the **My Documents** collection. See [Using Documents Collections](#) for more information.
  - If you added this document to a [Documents](#) page for a record, the document is visible on the selected page.
9. If necessary, complete the associated document property pages to add additional information, such as the category to which this document belongs or its access rights.

## Supported Document Types

The list below provides the default document types that you can add to TeamConnect. Based on your policies, your list may include additional document types. The document types are listed in alphabetical order.

Document Type	File Extension
Adobe Acrobat	pdf
Audio (Aiff)	aiff
Audio(wav)	wav
Binary	bin
Data Mapping File	dm
GIF	gif
HTM	htm
HTML	html
Image-jpg	jpg
Image-pcx	pcx
Image-tiff	tiff
JAR File	jar
JavaClass	class
JavaScriptCode	js
JSP	JSP
Microsoft Access	mdb
Microsoft Excel	xls, xlsx

Microsoft Power Point	ppt, pptx
Microsoft Word	doc, docx
Quicktime Video	mov
Rich Text Format	rtf
Standard Email	eml
Text	txt
Video	mpg
XML	xml
ZIP	zip

### 1.8.5.2 Editing Documents

To edit a document that is currently in TeamConnect, you must use the following process:

1. **Check out the document**—When a document is checked out, no other users can edit it. See [Checking Out a Document](#) for more information.

*Note: Super users are able to cancel document check out for other users.*

2. **Open the document and make the necessary changes**—You must have the associated application to open and edit a document.
3. **Check in the edited document**—Upload the edited version of the document to TeamConnect. See [Checking In a Document](#) for more information.

The edited document is now the latest version in TeamConnect. You can edit documents that you own or to which you have the appropriate rights.

If you check out a document and decide not to make any changes, you can cancel the check out. When you cancel a check out, the document remains as it was before you checked it out. See [Canceling a Document Check Out](#) for more information.

### Document Versions

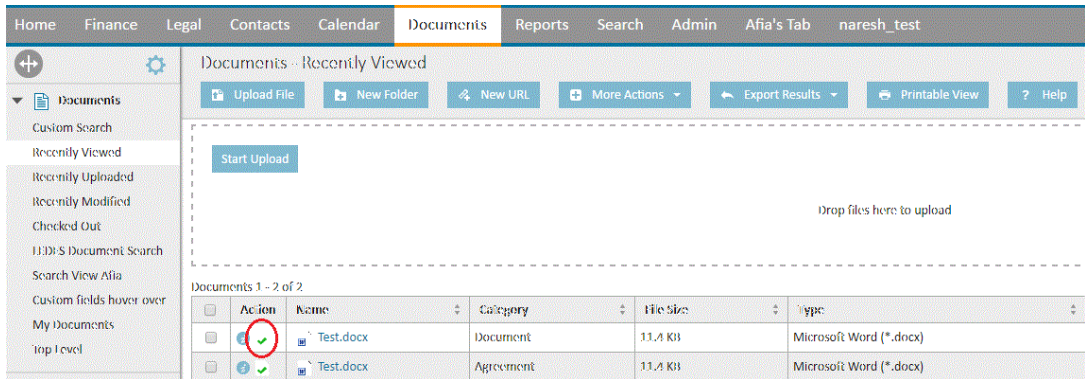
Each time you check out a document and then check it back in, a new version is created and added to the document **Version** page. See [Viewing Document Versions](#) for more information.

## 1.8.5.2.1 Checking Out a Document

To edit a document, you must first check it out of TeamConnect so that others cannot edit it at the same time.

### To check out a document

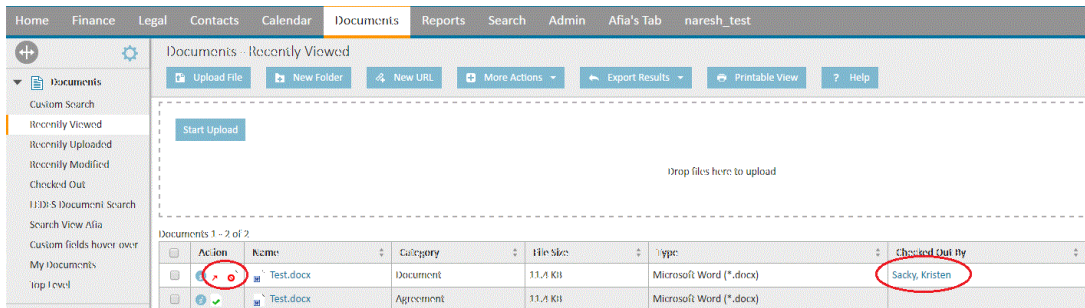
1. [Open](#) the **Documents** page that contains the document that you want to edit.
2. In the **Action** column, click the check out icon next to the document that you want to edit.



Documents Page - Check Out a Document

3. If a window appears with options to save or open the document, select the appropriate option. The document is handled as specified.

Once the document is checked out, in TeamConnect, your name is listed in the **Checked Out By** column on the selected **Documents** page, and only the check-in and undo icons are visible in the **Action** column.



Documents Page - File Checked Out

4. Make the necessary changes to the document.


When you are done making changes to the document, you can check it back in to TeamConnect. See [Checking In a Document](#) for more information.

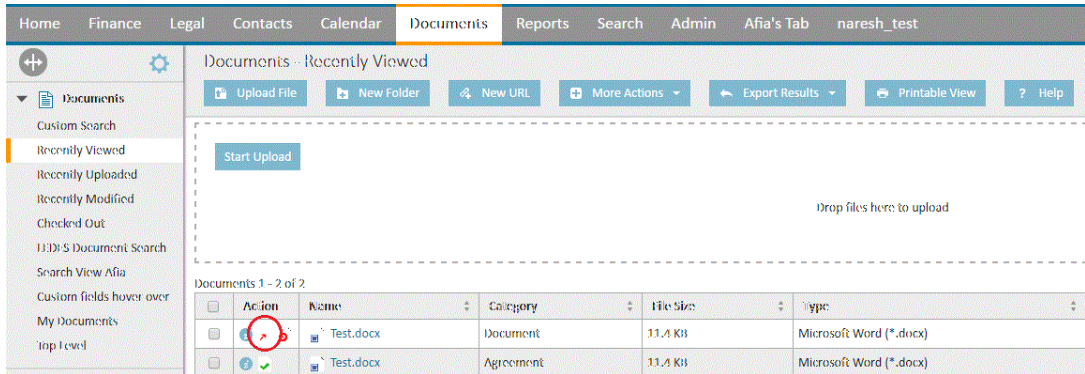
## 1.8.5.2.2 Checking In a Document

Once you are done editing a document, you can check it back in to TeamConnect.



### To check in a document after editing

1. In TeamConnect, [locate the document that you checked out](#).

2. In the **Action** column, click the check in icon  .



The screenshot shows the 'Documents' page in TeamConnect. The 'Documents' tab is selected, and the 'Recently Viewed' section is active. A table lists two documents:

Documents 1 - 2 of 2	Action	Name	Category	File Size	Type
<input type="checkbox"/>		Test.docx	Document	11.4 KB	Microsoft Word (*.docx)
<input type="checkbox"/>		Test.docx	Agreement	11.4 KB	Microsoft Word (*.docx)

**Documents Page - File Checked Out**

3. In the **Please Select a File** window, browse to the file that you want to check in, and then click **Check in**.



The edited document is saved to the original **Documents** page, your name is removed from the **Checked Out By** column, and the check out icon is visible again.

Additionally, this new version of this document is added to the **Versions** page. See [Viewing Document Versions](#) for more information.


**Caution:** If the document is not the same as the one that you checked out, the document that you check in overwrites the document that is currently in TeamConnect.

#### 1.8.5.2.3 Canceling a Document Check Out

If you decide not to make changes to a document that you checked out, you can cancel the check out.

##### To cancel a document checkout

1. In TeamConnect, [locate the document that you checked out](#).

2. In the **Action** column, click the undo icon  .

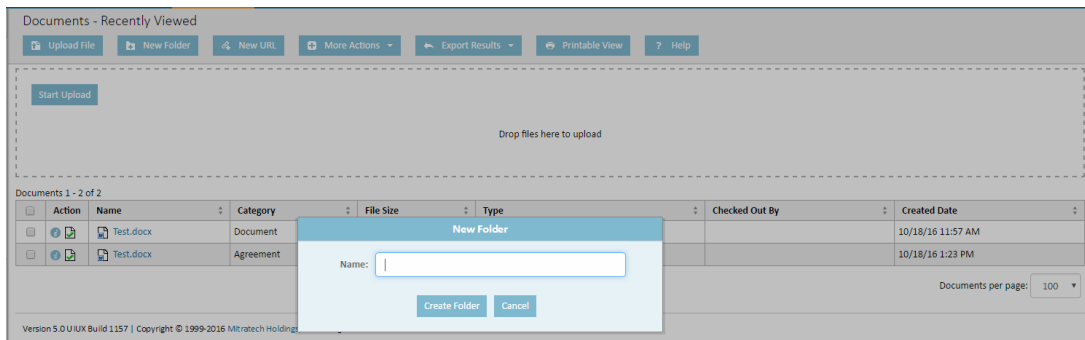
### 1.8.5.3 Creating a Document Folder

You can create a document folder to store related documents.

#### To create a document folder

1. Open the **Documents** page where you want to create the new folder.
2. Click the **New Folder** button or icon.

The **New Folder** window opens, and the **Documents** page is grayed-out in the background.



Documents New Folder Window

3. In the **New Folder** field, enter the name of the new folder, and then click **Create Folder**.

**Caution:** You cannot use special symbols or characters in folder names. You can only use alphanumeric letters, numbers, and the underscore character, for example *Status123\_today*. If you use special symbols or characters in folder names, they are replaced with hyphens (-). For example, if you enter *Status12\*\*@today*, the folder name will appear as *Status12--today*.

The new folder is created. To view the folder:

- If you created the new folder in the global **Documents** area, click the **My Documents** collection link in the left pane to view it.
- If you created the new folder on the **Documents** page for a record, your folder is visible in the selected location.

### 1.8.5.4 Exporting a List of Documents in a Folder

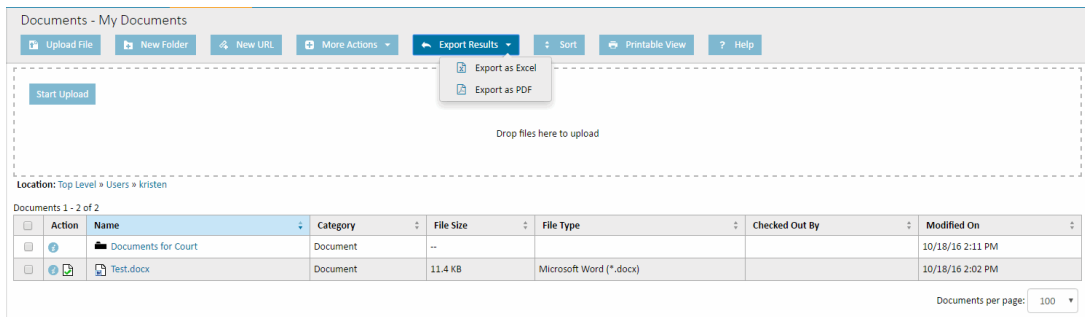
You can export a list of folders on a **Documents** page and/or a list of the contents in a document folder. You can export the list in an .xls format that you can open in Microsoft Excel, or a .pdf format that you can open in Adobe Acrobat Reader.

#### To export a document folder list

1. Open the Documents folder that contains the list of documents that you want to export. See [Viewing Documents](#) for more information.
2. Click **Export Results**, and then select one of the following options:



- Export as Excel
- Export as PDF



#### Export Results Drop-Down List

3. When the File Download window opens, select one of the following options:
  - **Save**—Saves the file to the selected location without opening it.
  - **Open**—Immediately opens the file in the related application. If you select this option, you must save this file if you want to access it later.

#### 1.8.5.5 Copying Documents

You can copy documents and document folders from one location in TeamConnect to a different location in TeamConnect. You can only copy documents that you own or to which you have the appropriate rights.

**Note:** To copy documents, you must have View and Create document rights.

#### What is Copied

When you successfully copy a document or folder, the following information is copied from the original document record:

- The assigned categories on the **Categories** page
- Everything on the **Details** page
- Everything on the **General** page
- The User Rights and Groups Rights on the **Security** page

#### What is Not Copied

The following items are not copied from the original document record:

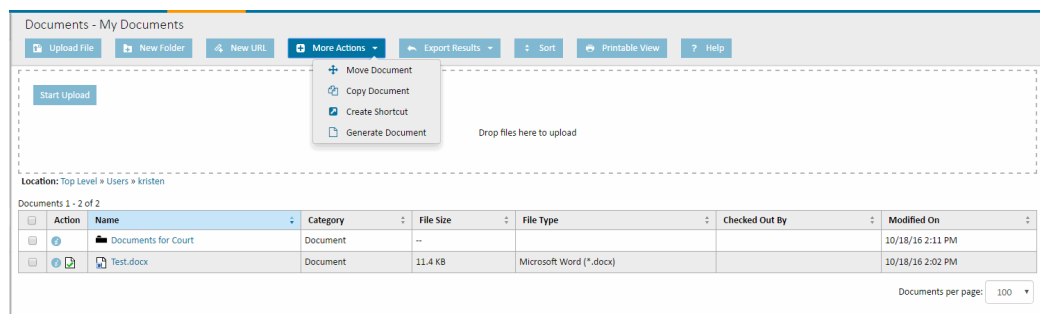
- Historical entries on the **History** page
- Versions that are listed on the **Version** page
- Approvals on the **Workflow** page

## Copying Do's and Don't's

- You can copy documents from a system **Documents** collection, such as **Recently Viewed**, **Recently Modified**, or **Checked Out**.
- You cannot copy documents into a system documents collection.
- You can copy documents into your personal **My Documents** folder or go to the **Top Level** folder > **Users** folder and locate the appropriate users **Documents** folder.
- You can copy documents between different record types. If you have the appropriate rights. For example, you can copy a document from a contact's **Documents** page to a project's **Documents** page.

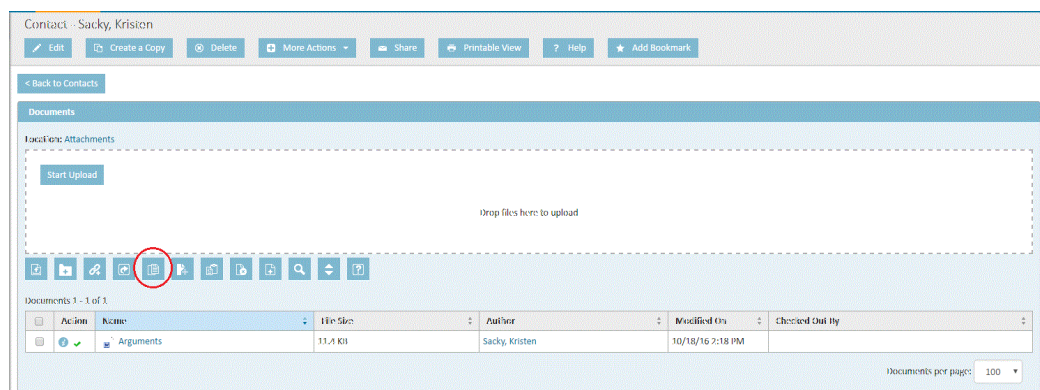
### To copy documents in TeamConnect

1. Open the **Documents** page that contains the document that you want to copy.
2. Select the check-box for the document that you want to copy. You can select more than one check-box if necessary, but only if you want to copy all of the selected documents to the same location.
3. Select one of the following options:
  - If you are on the global **Documents** page, click **More Actions**, and then select **Copy Document**.



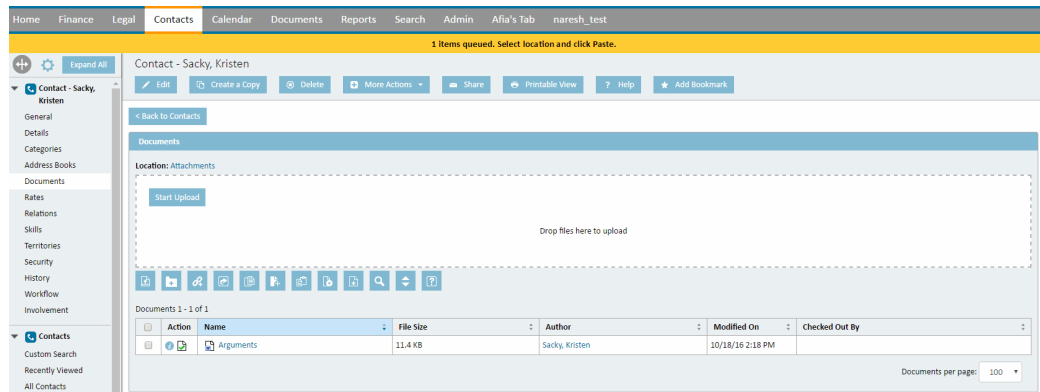
Global Documents Page - Copy Document

- If you are using the **Documents** page for a record, click the **Copy** icon.



Documents Page for a Record - Copy Icon

A confirmation message appears at the top of the screen confirming the number of items that you want to copy.

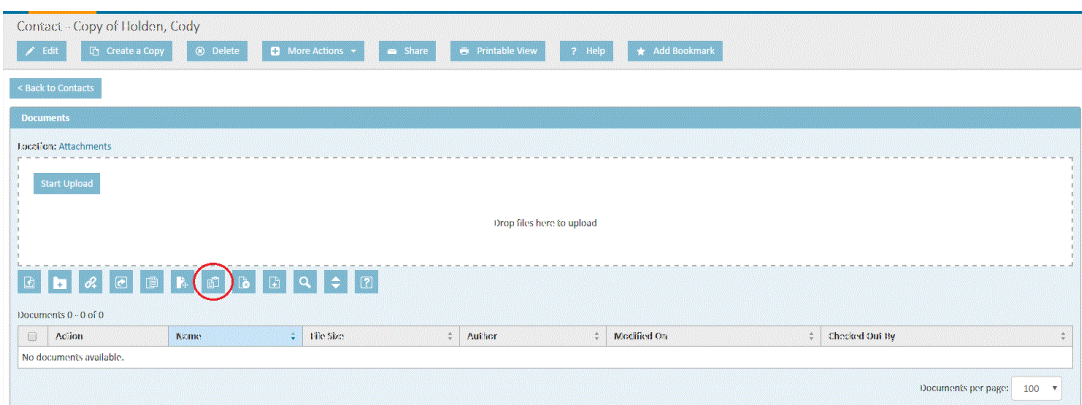


**Documents Page - Copy Confirmation**

4. Select one of the following options to select the location where you want to copy the document(s).
  - To copy the document to your own **Documents** page, click the **Documents** tab, and then, in the left pane, click **My Documents**.
  - To copy a document to another user, click the **Documents** tab, and then, in the left pane, click **Top Level**. On the Documents -Top Level page, in the **Name** column, click the **Users** folder and then click the name of the user to whom you want to copy the document.

*Tip: To alphabetize the list of users, on the Documents - Top Level page, click on the Name label.*

- To copy documents to a different record type, locate the appropriate **Documents** page. For example, if you want to copy a document from your personal Documents page to the Documents page for an Invoice, click the **Finance** tab, click the **Invoices** sub-tab, and then click the **Documents** link in the left pane.
5. Click **Paste**.



**Documents Tab - Paste**

A copy of the selected document is now available on the selected **Documents** page.

On the Global Documents page, the **Paste** button is located on the Global Toolbar; on a Documents page for a record, the Paste button is located on the Record Toolbar. See [Sharing Documents using Email](#) for more information.

### If a Document Copy Fails

If one or more of the documents fails to copy, you will see an error message listing the failed documents. A document copy may fail for several reasons, such as the document that you want to copy is checked out by another user.

If the document is not visible in the selected location, you may not have rights to view or copy documents in TeamConnect. Contact your system administrator.

#### 1.8.5.6 Moving Documents

You can move documents and folders from one location in TeamConnect to a different location in TeamConnect. Once you move a document, it is no longer available in the old location.

You can only move documents and folders that you created or to which you have the appropriate rights.

***Note:** To move documents, you must have View and Create document rights.*

### Moving Do's and Don't's

- You cannot move documents from a system documents collection, such as **Recently Viewed**, **Recently Modified**, or **Checked Out**.
- You cannot move documents into a system Documents collection.
- You can move documents into your own **My Documents** folder or go to the **Top Level** folder > **Users** folder and locate the appropriate users documents folder.
- You can move documents between different record types. For example, you can move a document from a contact's **Documents** page to a project's **Documents** page.

### To move documents in TeamConnect

1. [Open](#) the **Documents** page that contains the document that you want to move, and then select its corresponding check-box. You can select more than one check-box only if you intend to move all of the selected documents to the same location.
2. Select one of the following options:
  - On the global **Documents** page, click **More Actions**, and then select **Move Document**.
  - On the **Documents** page for a record, click the Move icon.

A confirmation message appears at the top of the screen confirming the number of items that you want to copy.

3. Select one of the following options to select the location where you want to move the document(s).
  - To move documents to your own Documents page, click the **Documents** tab, and then, in the left pane, click **My Documents**.
  - To move a document to another users, click the **Documents** tab, and then, in the left pane, click Top Level. On the **Documents -Top Level** page, in the **Name** column, click the **Users** folder and then click the name of the user to whom you want to move the document.

*Tip: To alphabetize the list of users, on the **Documents - Top Level** page, click on the **Name** label.*

- To move documents to a different record type, locate the appropriate **Documents** page. For example, if you want to move a document from your personal **Documents** page to the **Documents** page for an Invoice, click the **Finance** tab, click the **Invoices** sub-tab, and then click the **Documents** link in the left pane.
4. Click **Paste**.

The document is moved to the selected **Documents** page.

If one or more of the documents fails to move, you will see an error message. A document move may fail for several reasons, such as the document that you want to move is checked out by another user.

If the document is not visible in the selected location, you may not have rights to move documents in TeamConnect. Contact your system administrator.

#### 1.8.5.7 Creating a Document Shortcut

Create a shortcut if you want to provide a link that lets a user access a document that is on your **Documents** page.

#### Shortcut Do's and Don't's

- You cannot create shortcuts to a document in a system documents collection, such as **Recently Viewed**, **Recently Modified**, or **Checked Out**.
- You cannot create shortcuts in a system documents collection.
- You can create shortcuts in your own **My Documents** folder or go to the **Top Level** folder > **Users** folder and locate the appropriate users documents folder.
- You can create shortcuts between document types, For example, you can copy a document from a contact's **Documents** page to a project's **Documents** page.

***Important:** To create a shortcut, you must have View and Create document rights. In order for a user to access your document using a shortcut, you must allow them View access on the Documents **Security** page. See [Working with Record Security](#) for more information.*

**To create a document shortcut**

1. [Locate the document or folder](#) for which you want to create the shortcut link, and then select its check-box. You can select more than one check-box, but only if you intend to create all of the shortcuts in the same location.
2. Select one of the following options:
  - On the **Documents** tab, click **More Actions**, and then click **Create Shortcut**.
  - On the **Documents** page for a record, click the Shortcut icon.

A confirmation message appears at the top of the screen confirming the number of items.

3. Select one of the following options to select the location where you want to create the shortcut(s).
  - To add the shortcut to your own **Documents** page, click the **Documents** tab, and then, in the left pane, click **My Documents**.
  - To create a shortcut on another users page, click the **Documents** tab, and then, in the left pane, click **Top Level**. On the **Documents -Top Level** page, in the **Name** column, click the **Users** folder and then click the name of the appropriate user.

*Tip: To alphabetize the list of users, on the **Documents - Top Level** page, click on the **Name** label.*

- To create a document shortcut on a different record type, locate the appropriate **Documents** page. For example, if you want to create a document shortcut from your personal **Documents** page to the **Documents** page for an Invoice, click the **Finance** tab, click the **Invoices** sub-tab, and then click the **Documents** link in the left pane.
4. Click **Paste**.

The shortcut is added to the selected location.

If the shortcut is not visible in the selected location, you may not have rights to view documents in TeamConnect. Contact your system administrator.

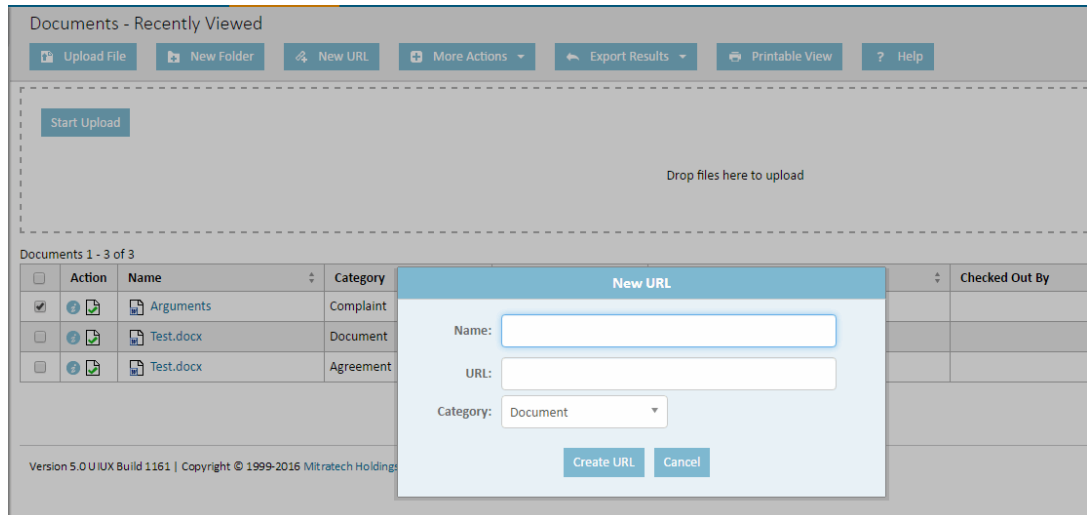
**1.8.5.8 Creating a Link to a Web Page**

You can create a document link that opens to a Web page.

**To create a link to a Web page**

1. Open the **Documents** page where you want to create a link to a Web page.
2. Click the **New URL button** or icon.

The **New URL** box opens and the **Documents** page is grayed-out in the background.



Documents New URL Window

3. In the **Name** box, enter the name of the Web page as you want it to appear on the **Documents** page.
4. In the **URL** field, enter or paste the appropriate Web address.

*Tip: For accuracy, open the Web page and copy-and-paste the Web address into this field.*

5. If necessary, in the **Category** drop-down list, select a category.
6. Click **Create URL**.

The link appears on the selected **Documents** page next to its URL icon.

### 1.8.5.9 Downloading Documents

When you download a document, you save it to a location other than TeamConnect, such as your personal computer. You can also download email messages.

**Note:** If you need to make changes to a document, you must first check it out of TeamConnect. See [Editing Documents](#).

#### To download a document

1. [Locate the document](#) that you want to download and click its link in the **Name** column.  
The document opens in its related application. If an additional window opens asking how you want to handle the file, select the desired options.  
If the document does not open, you may not have its application installed. For example, if you want to open a .doc file, you must have Microsoft Word installed on your computer.
2. To save the document, use the save function that applies to the associated application.

### To download an email message

1. Locate the email message that you want to download and click its link in the **Name** column.
2. Click **Download Email**, and then click the appropriate options to save the email message to the preferred location.

#### 1.8.5.10 Generating Documents Using TeamConnect Data

The Document Generator lets you use a template to generate a document that contains data from TeamConnect records. For example, your organization may have already set up templates for generating letters to your clients. Information in the generated letter, such as contact names, addresses, and record numbers, are automatically filled in when you generate the document.

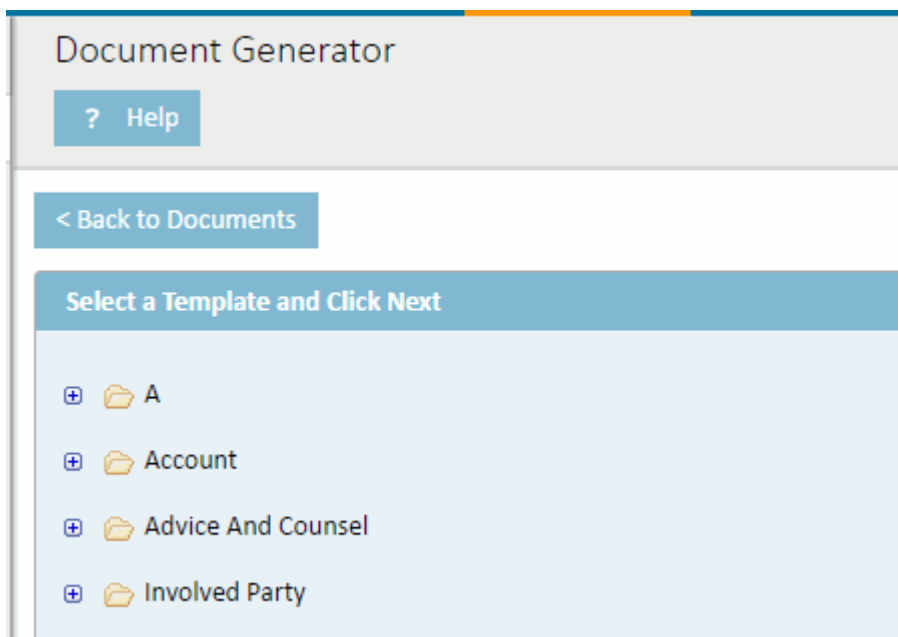
Templates are usually pre-defined by your system administrator. You must have the appropriate rights to generate a form document.

You can generate a document using the **Documents** tab or the **Documents** page for a record.

### To generate a document from the Documents tab

1. Click the **Documents** tab, and then click **More Actions**.
2. Select **Generate Document**.

The first page of the Document Generator opens.



Documents Tab - Document Generator First Page

3. Expand the tree to locate the appropriate template.
4. Check the box of the desired template, and then click **Next**.



The object record selection page opens.



Object Record Selection Page

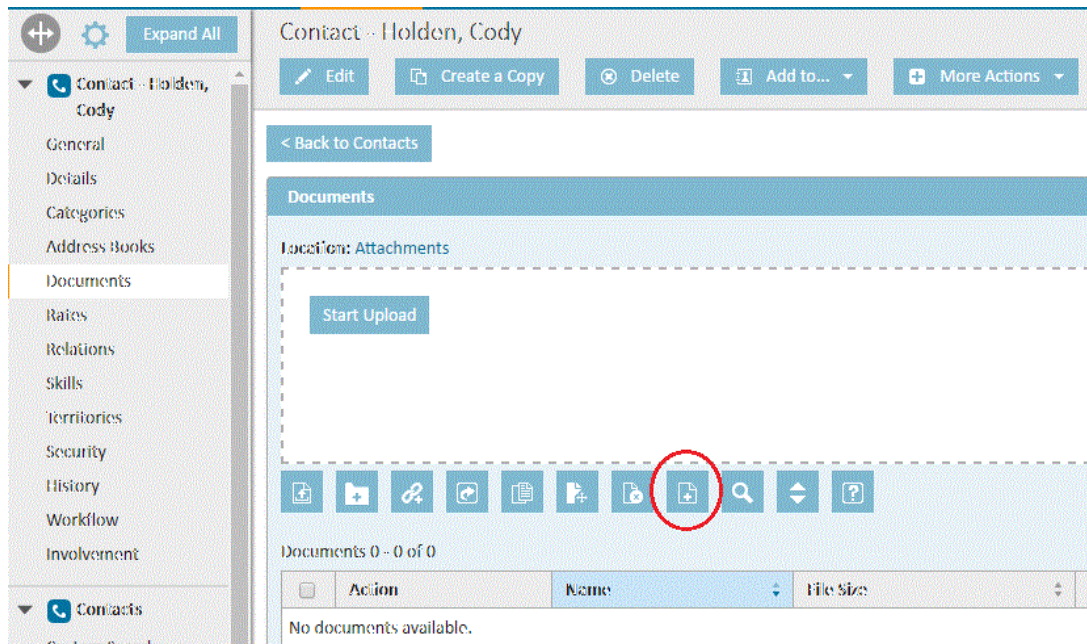
5. Click the Select button of the object record that will be used during document generation.
6. On each page that follows, enter or select the appropriate information, and then click **Next** until you reach the final page.
7. To view the document before it is generated, click **Preview Document**.
8. If necessary, in the **Name** field, enter the name of the document. This is the name of the final generated document.
9. In the **Default Category** drop-down, select the appropriate category.
10. Click **Save Document** to create the document. The generated document is saved to the selected object record.

You can leave the document in this location until you are ready to use it, or you can print it. See [Printing Documents](#) for more information.

#### To generate a document from the Documents page for a record

1. [Open](#) the **Documents** page for the appropriate record type.
2. Click the Document Generator icon.

The first page of the Document Generator opens.



Documents Page for a Record - Document Generator First Page

3. Expand the tree to locate the appropriate template.
4. Check the box of the desired template, and then click **Next**.
5. On each page that follows, enter or select the appropriate information, and then click **Next** until you reach the final page.
6. If necessary, in the **Name** field, enter the name of the document. This is the name of the final generated document.
7. To view the document before it is generated, click **Preview Document**.
8. In the **Default Category** drop-down, select the appropriate category.
9. Click **Save Document** to create the document. The generated document is saved to the selected object record.

You can leave the document in this location until you are ready to use it, or you can print it. See [Printing Documents](#) for more information.

#### 1.8.5.11 Using Windows Explorer with Documents (WebDAV)

If your system administrator enabled it, once you open the **Documents** page for a record, you may see an **Open In Explorer** link.

WebDAV enables Windows Explorer to connect to your TeamConnect documents.

##### To use WebDAV

1. Click the **Open in Explorer** link to open a Windows Explorer folder that links to the Attachment Root directory of the current record.

2. If WebDav is enabled, you can use Windows Explorer to perform the following document tasks:
  - Open, view, and save documents to your computer
  - Open, edit, and save documents
  - Copy documents
  - Move documents

For more information, see your system administrator.

## 1.8.6 Viewing or Editing General Information of a Document

Use the Document **General** page to view information about a document, including the author of the document, when the document was last checked out and back in, when it was created and modified, and the file type and size. If necessary, you can change the name of the document, the author of the document, and the date the document was authored.

### Custom Details

This page may also contain a **Details** section that displays custom fields that appear or disappear based on the selections you make on the Documents **Categories** page. See [Adding and Removing Document Categories](#) for more information.

#### To view or edit documents general information

1. Open the **Documents** page with the document you want to view or edit.
2. In the **Action** column, click the **Properties** icon.
3. To edit the document name, author, or date, click **Edit**, and then make the appropriate changes. See the [Document General Page Field Descriptions](#) for specific details.
4. [Select a save option.](#)

### Document General Page Field Descriptions

The tables below contain the default sections and fields. The information that you see may vary based on your company policies and access rights. The fields are listed in alphabetical order.

#### Document Information Section

Field Name (* = required)	Description
<b>Author</b>	The name of the user who authored the document. The default author is the person who uploaded the document.

	<ul style="list-style-type: none"> <li>• To change the name of the author, click the search contacts icon.</li> <li>• To view the contact record for this author, click the view contact icon.</li> </ul>
<b>Checked Out By</b>	The name of the user who currently has the document checked out.
<b>Checked Out On</b>	The date that the user checked out the document.
<b>Date Authored</b>	The date that the document was uploaded to TeamConnect. To change this date, click the calendar icon, or enter the date manually.
<b>File Size</b>	The size of the document.
<b>File Type</b>	The name of the application that applies to this document or the format of the document.
<b>Folder Path</b>	The path that leads to location where the document is stored. If you click the link in the field, you will open the document General page for the named document.
<b>*Name</b>	The name of the document. The default is the current name of the document.
<b>Record</b>	The name of the user that added the document to TeamConnect. If you click the link in this field, you will open the contact <b>General</b> page for the named user.

## Notes Section

Notes that apply to the selected document.

### Access Information Section

Field Name	Description
<b>Created By</b>	The name of the user who added the document to TeamConnect. To view the associated contact record, click the link in this field.
<b>Created On</b>	The date that the document was added to TeamConnect.
<b>Modified By</b>	The name of the user who last modified the document. To view the associated contact record, click the link in this field.

<b>Modified On</b>	The date on which the document was last modified.
--------------------	---

### Details Section (optional)

The information in this section varies depending on the selections you make on the **Categories** page. If there are no custom fields associated with the assigned categories, or if you do not have rights to view custom fields, this section does not appear.

See [Adding and Removing Document Categories](#) for more information.

## 1.8.7 Viewing Document Versions

Use the Document **Versions** page to view the new versions of a document that are created each time you check a document out, and then back in to TeamConnect. Each version is listed with the checked in date and time, the name of the user who modified the document, and any necessary notes.

The link to the **Versions** page only appears if you have the appropriate rights.

### To view documents versions

1. [Locate the document](#) whose versions you want to view.
2. In the **Action** column, click the **Properties** icon.
3. In the left pane, click the **Versions** link.

You can also use the **Versions** page to revert to a previous version of a document. See [Reverting to a Previous Version](#) for more information.

***Tip:** If you created a copy of this document to a different location in TeamConnect, the copy is not reverted. You must re-copy this document if you want the latest version.*

### 1.8.7.1 Reverting to a Previous Version

Reverting lets you delete the current version of a document and replace it with an older version. You may need to revert to a previous version of a document if the current version is corrupt or the updates to the current version are no longer necessary.

### To revert to a previous version of a document

1. Open the **Versions** page for the appropriate document.
2. Click **Edit**.
3. In the **Versions** section, select the radio button of the version to which you want to revert.
4. Click **Revert**.

The version that you selected is now listed in the **Current Version** area and any listed versions that were created after the new current version are permanently removed from the **Versions** section.

## 1.8.8 Adding and Removing Document Categories

Use the document **Categories** page to assign a document to a grouping that may describe the type of document or its purpose. Categories are usually pre-defined by your system administrator and each category may include one or more subcategories.

Additionally, when you select a category, custom fields may appear on the General page in the Details section. These fields are related to the categories that you select and may have related fields that must be completed before you can save the record.

### How Categories Affect the Details Page

For example, you may have a document that you assigned to a category named *Service of Process*. Once you assign the document to the Service of Process category, the **Details** section on the **General** page may display custom fields that allow you to enter the date of the Service of Process, the name of the court that is handling the case, and other information related to the Service of Process in this matter.

Based on your access rights, you can view, add, change, or delete categories.

#### 1.8.8.1 Viewing or Creating Documents History

Use the **History** page to view and write historical details that are related to the selected document, such as the content of telephone calls or historical notes that relate to the selected document.

See [Working with Record History](#).

## 1.8.9 Printing Documents

You can print a list of documents on any documents page in TeamConnect.

You can also open a document and print it using the related application's print function.

### To open a document and print it

1. In TeamConnect, locate the document that you want to print.
2. In the **Name** column, click the document to open it.  
  
If the document does not open, you may not have its application installed on your computer. For example, to open a .doc file, you must have Microsoft Word installed on your computer.
3. If an additional window opens asking how you want to handle the file, select the desired option, and then click **OK**.
4. When the document opens, use the print function that applies to the application.

You can also print a list of documents on any page.

### 1.8.10 Deleting Documents and Document Folders

Deleting a document removes it from TeamConnect and you cannot reverse the deletion. If you delete a document folder, the folder and all of its contents are also deleted.

You can only delete documents and folders that you own or to which you have the appropriate rights.

#### What is Deleted

When you delete a document, the following items are deleted:

- **Documents:**
  - Associated approvals
  - All shortcuts that lead to this document
  - Associated history records
  - Categories and the related custom fields:
    - If the category that you selected for the document caused custom fields to appear in the **Details** section on the **General** tab, when you delete the category, the custom fields are deleted.
    - If the deleted category was set as the default, the Root category is automatically set as the default category.
- **Document folders:**
  - All associated subfolders and their contents

**Note:** You cannot delete a checked out document or a document within a checked out folder.

**Tip:** The top-level folder is the page that opens when you click the **Documents** link for a record. If necessary, you can void the transaction first and then delete the record.

#### To delete a document or document folder

1. Locate the document or folder that you want to delete.
2. In the **Action** column, click the properties icon.
3. Click **Delete**, and then click **Yes** in the confirmation window.

The document is deleted from the **Documents** page and from TeamConnect.

### 1.8.11 Updating Security Rights of Document Folders

Document folders have security settings that determine whether subfolders and files receive the same rights as a folder. By default, system administrators set the default security settings of all folders.

You can update the settings of each folder on the **Security** page of a folder:

- **Replace Rights on subfolders**—With a checkmark, if you change the rights of the folder, its subfolders and files also receive the same rights. Without a checkmark, if you change the rights of the folder, the rights of its subfolders and files do not change.

If you add a folder or file to the folder with this setting, the folder or file does not update with the folder's rights. To receive security updates, the folder or file must be in the folder when the rights change.

- **Replace Rights on existing files**—With a checkmark, if you change the rights of the folder, its files also receive the same rights. Without a checkmark, if you change the rights of the folder, the rights of its files do not change.

If you add a file to the folder with this setting, the file does not update with those rights. To receive security updates, the file must be in the folder when the rights change.

## 1.9 Workflow Processes

Approval processes in TeamConnect provide structure and enforcement for your workflow. For example, a request to post an invoice might automatically route from staff members to a manager, and then to a director for approval, before it is posted to the appropriate budgets. At each stage in the workflow, one or more users are responsible for specific activities.

A workflow process has the following different types of users.

Workflow User	Description of User	Page with User's Requests
Requestor	Users who submit the request by <a href="#">completing an action that requires an approval</a> .	<b>Active Requests</b> page of <b>My Requests</b>
Approvers	Users who <a href="#">approve or reject the request</a> .	<b>Requests to Approve</b> page of <b>My Approvals</b>
Reviewers	Users who <a href="#">evaluate the request</a> .	<b>Requests to Review</b> page of <b>My Approvals</b>
Process Manager	Users who see all requests in workflow processes and have the authority to <a href="#">make workflow decisions</a> .	All pages of <b>My Workflow Processes</b>

When your system has workflow processes in place, a user (as the requestor) triggers an approval process by performing an action. Examples of this type of action include posting an invoice,



changing the phase of a project, or deleting a record. This trigger creates an approval request, causing the record to become part of a workflow process.

A workflow process can have one or more stops. Each stop consists of one or more users who must approve the approval request. Users may also reassign the request or add additional stops prior to their approval. Workflow routes can also be subject to [route conditions](#) or [hold conditions](#) set in place.

If the process continues with successful approval and the request does not expire or get rejected, the original action finally executes.

If no active user is part of a stop, the following situations determines the outcome.

Situation When No User is Part of the Stop	Situation Outcome
The stop only requires one member's approval.	<p>The workflow process skips the stop and continues to the next stop.</p> <p>The skipped stop is not listed on the approval schedule, but you can <a href="#">view the details on the approval history</a>.</p>
The stop requires all members for approval.	<p>An error message appears.</p> <p>Depending on how the process is defined, if an approver rejects the request anywhere along the approval route, one of the following outcomes occur:</p> <ul style="list-style-type: none"> <li>• The request is rejected and the requestor is notified.</li> <li>• The request is sent back to the approvers in the first stop.</li> <li>• The request is sent to the process manager, who determines the action to take.</li> </ul>

In addition to the pages and processes described in this topic, each individual record is associated with a **Workflow** page. Users with rights to open a record can [view the workflow process history](#). The **Workflow** page is not editable.

### 1.9.1 Using the Approval Schedule and Approval History

Within records that require an approval process, TeamConnect provides two different views of the approval process: approval schedule and approval history.

#### Approval Schedule

The **Approval Schedule** displays as a status of all assigned approvers for the entire process. The status information is useful for checking how much progress a request has made, in the context of the whole route. By reviewing the approval status, you can learn how many stops are in the process, who the approvers are in each stop, and whether each approver has approved or rejected the request.

If the Process Manager made any reassignments, you can also view this information in the approval status.

**Note:** If it appears that a number is skipped in the **Stop** column, it may mean that there are no assignees with the specified on the selected record. The details of the skipped stop appear in the approval history.

For each active request, the approval status displays in the following locations:

- On the **Workflow** page, in the **Active Requests** section while a request is still pending approval.
- On the **General** page, the record displays the **Pending Approval** status on its toolbar, where it can be seen by the requestor while the request is pending approval.

**Note:** Once a request is completed—approved, rejected, or canceled—the approval schedule no longer displays in the record. Instead, you can view the approval history.

The following table lists the fields in the **Approval Schedule** section. Fields are listed in alphabetical order.

#### Approval Schedule Field Descriptions

Field/Button	Description
<b>Request Section</b>	
<b>Expiration Date</b>	(optional) The expiration date of the request. The expiration action is based on options set by your system administrator.
<b>Request Date</b>	The date that the request was triggered.
<b>Request Name</b>	The type of action being requested.
<b>Requestor</b>	The name of the user who is responsible for sending a record into an approval process.
<b>Status</b>	The current status of the entire request. Options are as follows: <ul style="list-style-type: none"> <li>• <b>Pending</b>—The request remains in this status until all stops are complete.</li> <li>• Other possible statuses include Rejected, Error, or Canceled.</li> </ul>
<b>Stop Section</b>	

<b>Action Date</b>	The date on which the action occurred.
<b>Expiration Date</b>	The date by which the approver must submit a decision.
<b>Internal Comment</b>	The reason that the request is approved or rejected, if necessary. This information is originally entered in the Internal Comment field when the action is approved or rejected.
<b>Performer</b>	The members of this stop who may approve or reject the request. Clicking this link opens this user's contact record.
<b>Status</b>	<p>The current status of the approval on the stop route. Possible options are as follows:</p> <ul style="list-style-type: none"> <li>• <b>Pending</b>—The request remains in this status until all stops are complete.</li> <li>• <b>Approved</b>—This stop is approved by all members and moves on to the next approval stop on the route.</li> <li>• <b>Rejected</b>—Someone on this stop did not approve this action.</li> <li>• <b>Error</b>—The stop contains an error, such as a member in this stop is assigned a role that does not exist. You can review the history to view the error, and then make the appropriate corrections.</li> <li>• <b>Canceled</b>—This stop on the route is canceled.</li> <li>• None or empty field—This request has not reached this stop.</li> </ul> <p><i>Note: Your system administrator may provide additional statuses.</i></p>
<b>Stop</b>	The approval rule check points where members and groups are assigned to handle a request, such as wanting to post an invoice. The route follows the order in which you organize the stops, starting with Stop 1.

## Approval History

The **Approval History** section records each event as it occurs during the approval process on the **Workflow** page of records. Once a request is completed—approved, rejected, or canceled—the complete approval history for that process displays in the **Completed Requests** section. Events that are tracked in the approval history include the following:

- Requestor cancels the request.
- Approver approves or rejects the request.
- Approver sends for review.
- Reviewer submits review.

- Approval time expires.
- Approval request is sent back to first stop by Process Manager (restarted).
- Process Manager assigns another user or reassigns the same user.
- Process Manager returns request to requestor (rejects).
- Request is automatically approved or rejected upon expiration.
- Error that a user is not found for a certain role or path in the system.
- Error that a user group assigned to approve is empty.
- Error that a user assigned to approve is inactive.

## 1.9.2 Setting Up Email Approvals

Users are able to receive and reply to email notifications in order to approve and move items through workflow. The details below outline the settings necessary for *TeamConnect Admins* to enable this feature.

**Important:** Email approvals may not function as some versions of Outlook include line breaks on any line with more than 76 characters. Users with Outlook 2010, 2013 and later versions will need to make the following adjustments to their Outlook configuration for email approvals to function:

1. Click on the File tab and select Options.
2. Click on "Advanced" in the left column of the Options window.
3. Scroll down to the "International Options" section.
4. Change "Preferred encoding for outgoing messages" to "Unicode (UTF-8)".
5. Click "OK" at the bottom of the Options window.
6. Restart Outlook.

### Admin Settings to Enable Email Approvals

Click the **Admin** tab and select **Email** from the left side navigation pane.

#### Outgoing Mail Server Settings

- **Outgoing Mail Server (SMTP):** Your company's mail server, ex. 10.50.5.9
- **Reply-to Address:** The email address listed here will be the "from" address users see when receiving email notifications. For example, if it's set to John.Doe@gmail.com and Betty receives an approve/reject email, it will appear as if it was sent from John.Doe@gmail.com.

You'll need to make sure that mailto: links are enabled for the mail server and that the approve/reject emails are formatted correctly. The "Use SSL" and "Outgoing Server Requires Authentication" options are not required for email approvals to function.

**Note:** *This has only been tested on Gmail, Outlook and hMail. Do not use Hotmail or Yahoo Mail.*

## Incoming Mail Server Settings

This block signals TeamConnect where to acquire and read the responses to approve/reject emails.

- **Account Type:** IMAP (Internet Message Access Protocol) This allows email clients to access email on a remote mail server, which is ideal for cases where the mailbox is managed by multiple users.
- **Incoming Mail Server:** The incoming mail server is the server associated with your email address you'll use to read the emails from. You'll enter "imap.(insert your mail server here).com".

For example, Gmail accounts will say "imap.gmail.com".

- **Username:** This is the email account from which the email approve/reject email responses will be read. An example address format could be "Jane.Doe@gmail.com". TeamConnect reads the emails sent to this account about once a minute, so it is important to not manually open any emails at this account. For this reason, it is suggested to enter an account that is not used. If this email is used for other purposes than email approvals, make sure to not open the approval responses. After TeamConnect processes the email approval responses, they will appear as "read" messages.

**IMPORTANT: Do not use Hotmail or Yahoo! mail. You'll need to make sure that mailto: links are enabled for the mail server and that the approve/reject emails are formatted correctly. This has only been tested on Gmail, Outlook and Hmail.**

- **Password:** This needs to be the actual password for the email account used in the above username field. If it's not correct TeamConnect will be unable to log-in remotely to the IMAP email account, i.e. cannot read the approval response emails.
- The checkbox next to 'Use SSL/TLS' must be checked.

## Additional Settings Needed Outside of TeamConnect

1. The Incoming Mail Server needs to have IMAP enabled. If you go into the settings within the email account you want to use for the incoming mail server, they should have a button/option/tab/etc that will allow you to enable IMAP. For example, Gmail users can:
  - i. Sign in to Gmail.
  - ii. Click the gear in the top right .
  - iii. Select Settings.
  - iv. Click Forwarding and POP/IMAP.
  - v. Select Enable IMAP.
  - vi. Click Save Changes.
2. Two ports need to be opened on your computer - the port numbers needed are dependent on the mail servers you use for outgoing and incoming. This can be found by a simple google search of your mail server and IMAP settings. For Mitratech employees, it's port 25 (outgoing) and port 993 (incoming gmail).
3. Make sure that at the very bottom of the Admin Settings > Email 'Status:' you have selected **Start**.

4. Stop and restart the app server after making all changes.

### 1.9.3 Using Approval Process Email Notifications

TeamConnect sends email to requestors, approvers, reviewers, and process managers to notify them of certain events.

If your administrator has given you access to set your email notification preferences, you can select which emails you want to receive on the **Email Notifications** page of your Preferences. This page provides you with options for receiving emails related to your approval requests, in addition to other types of notifications.

There are certain email notifications that are set by administrators on a per-process basis. These emails are automatically sent when those events occur. You do not have the option to receive them or not. For example, if you need to be involved in certain approval processes but you are not necessarily an approver, the process may be set up to keep you notified of approvals and rejections.

To receive email notifications, your email preferences must be set appropriately. For more details, see [Setting Up Email Notifications](#).

#### Workflow Email Notification Types

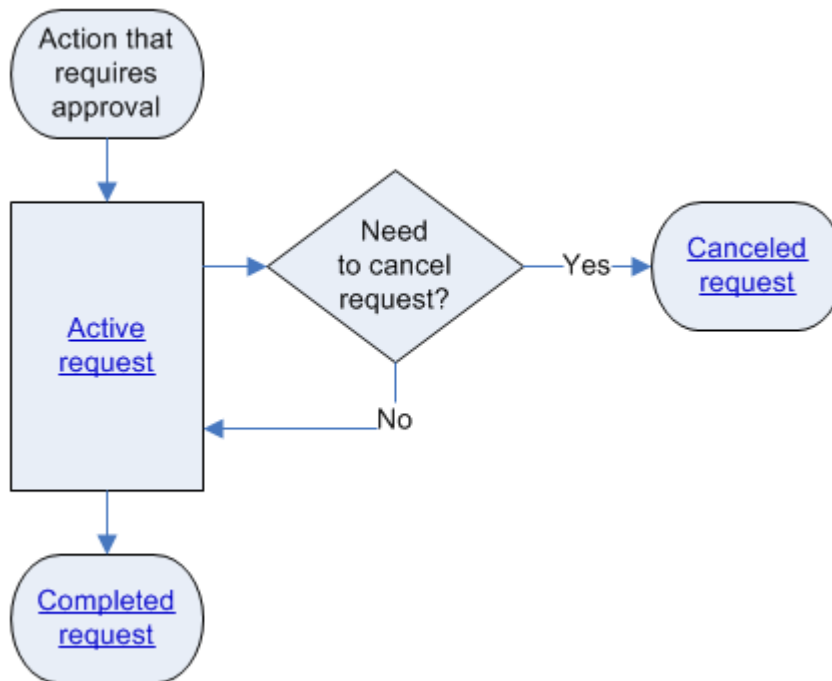
Your role in workflow	Events that can trigger email notifications	Optional email notification setting	Set by administrator for each process
Any user in the system, such as a supervisor	A request is rejected		<b>x</b>
	A request is approved		<b>x</b>
	An error occurs in a process		<b>x</b>
Requestor	Your request is approved	<b>x</b>	
	Your request is rejected	<b>x</b>	
Approver	You receive a request for approval	<b>x</b>	
	Your review request is returned to you	<b>x</b>	
	A request for approval is going to expire	<b>x</b>	
	A request for approval in your list is canceled	<b>x</b>	

Reviewer	You receive a request for review	x	
Process Manager (all members of the group assigned to Process Manager role)	An error occurs in a process	x	
	A route or stop expires	x	
	A request is rejected	x	

### 1.9.4 Submitting Requests for Actions that Require Approval

If you are responsible for sending a record into an approval process, you are considered the requestor in the approval process. For example, if you click **Post** in an invoice record, and the invoice must be approved before it posts, then you are the requestor who triggered an approval process for that invoice.

The following flowchart shows the life cycle of a request from the requestor's point of view. Click areas of the flowchart for more information.



As a requestor, you can easily track both your [active requests](#) and [the requests that are completed](#) (approved or rejected). You also have the ability to [cancel an approval request](#) once it is sent. To receive emails about the status of your requests for approval, you can [set approval process email notifications](#).

You can view the status of the records for which you have requested some kind of approval in the following ways:

- Select **Requests** from the **All** drop-down list on the tab bar to open **My Requests**. If your home page has a **My Requests** portal pane, you can also click one of the links that summarizes the status of your requests to open **My Requests**.
- Open the record of a request, which displays the status on its toolbar. From that record, you can click the **Workflow** page for [a list of active requests and completed requests](#).

#### 1.9.4.1 Viewing Your Active Requests

The requests for approval that you submitted are summarized on the **My Requests** page. Requests that are not approved or rejected yet appear in the **Active Requests** section.

##### To view your active requests

1. From the **All** tab, open the **Requests** page.

The **Active Requests** page provides you with a summary of the status of each request.

2. If necessary, click **Show Filter** to enter search criteria, and click **Search** to display only certain types of requests for a record type.

**Note:** You can also open a record and click the **Workflow** page to track [the approval schedule of a request](#).

##### Active Requests Section of My Requests Page

Field/Button		Description
<b>Search</b>		This drop-down list filters the list of results on this page. Choose a specific record type from the drop-down list so that only records of that type appear in the list of results.  After making your selections, click <b>Search</b> to filter the results. If a search filter appears, enter search criteria and click <b>Search</b> again.
<b>Show Filter/Hide Filter</b>		This link shows or hides the search criteria that you can use to find a certain type of request for a record type.
Search Filter  <i><b>Note:</b> The search filter only appears if one is</i>	<b>Field</b>	The name of the field you want to search.



<i>specified for the record type you select.</i>	<b>Operator</b>	The option that describes the criteria that you want to use to further narrow your results. See Operators for more information.
	<b>Value</b>	The value you enter that the search uses to return specific results.
	<b>Search</b>	The button that you click to execute the search.
	<b>Clear Values</b>	The link that you click to remove all values from the search
	<b>Reset to Default</b>	The link that you click to reset all operators and values to the default fields.
<b>Active Requests</b> table  <i>Note: Different fields may appear depending on your system settings.</i>	<b>Record</b>	Click to open the record for which you submitted an approval request. Examples of records include an invoice, project, document, or expense.  In the record page, you can view the approval status. Depending on how the approval rule is defined, you may not be able to edit the record while it is pending approval.
	<b>Request</b>	The name of the approval request.
	<b>Submitted</b>	The date on which you submitted the request.
	<b>Expected Completion Date</b>	The date by which the approval process should be completed.  Depending on how the approval process is defined in the approval rule by administrators, and what happens during the approval process, the expected completion date could change to a later date.
	<b>Days Remaining</b>	The number of days remaining in the approval process, based on the expected completion date.
	<b>Stop</b>	The current stop of the approval process.  For example, if there are three stops in the approval route, and the approvers in the first stop already approved the request, then the approval is in stop 2 of 3.

**Cancel**

Click to cancel the selected request or requests. For details, see [Canceling Approval Requests](#).

**Caution:** Be careful not to click this button when you want to close the **My Requests** page. You may accidentally cancel any selected approval requests.

### 1.9.4.2 Viewing Your Completed Requests

The requests for approval that are completed are summarized on the **My Requests** page. You can view the approved or rejected requests in the **Completed Requests** section. This section also lists requests that you canceled.

The **Completed Requests** section (the [Completed Requests Section of My Requests Page](#) image) provides you with a summary of each approved, rejected, and canceled request. The **Action** column indicates the final status of the request.



#### To view your completed requests

1. From the **Tools** tab, open the **My Requests** page.  
The **My Requests** page opens.
2. Click the **Completed Requests** page.
3. In the **Search** drop-down list, select which requests you want to view, depending on how they were completed or resolved:
  - **All Requests Completed**—Requests that have been approved, canceled, or rejected.
  - **Requests Approved**—Requests that were fully approved by the approvers in the route as well as requests that were automatically approved when the route expired.
  - **Requests Canceled**—Requests that you canceled.
  - **Requests Rejected**—Requests that were rejected by an approver in the route, returned to you by the Process Manager, or automatically rejected when the route expired.
4. Using the **Completed Date From** and **Completed Date To** date fields, select the date range for completion that you want to view.
5. Click **Search**.

The list of your completed requests displays.

**Note:** You can also open a record and click the **Workflow** page to track [the approval history of a request](#).

#### Completed Requests Section of My Requests Page

Field/Button		Description
<b>Search</b>		This drop-down list filters the list of results on this page by the reasons for request completion. After making your selections, click <b>Search</b> to filter the results.
<b>Completed Date From</b>		The date that specifies the earliest completion date to appear in the search results. Select the  to specify a date.
<b>Completed Date To</b>		The date that specifies the latest completion date to appear in the search results. Select the  to specify a date.
<b>Completed Requests</b> table	<b>Record</b>	Click to open the record for which you submitted an approval request.
	<b>Request</b>	The name of the approval request.
	<b>Submitted</b>	The date on which you submitted the request.
	<b>Completed</b>	The date that the approval process completed.
	<b>Duration</b>	The number of days the request was in the approval process before completion.
	<b>Action</b>	A description of the action that caused the approval process to complete.

### 1.9.4.3 Canceling Approval Requests

You can cancel an approval request if you need to make a modification in the record before sending it for approval. You can cancel an approval request only if you submitted the request. You can cancel a request from these locations:

- In the record itself (such as an invoice), from the record toolbar.

An approval status section is displayed in a record only when you have submitted an approval request. Approvers, reviewers, and Process Managers also see this section in the record, but it is displayed with slightly different information.

- On the **My Requests** page.

**To cancel an approval request in the individual record page**

1. Select **Requests** from the **All** drop-down list on the tab bar.  
The **My Requests** page is displayed.
2. Click the **Active Requests** link (see the [Active Requests Section of My Requests Page](#) image).
3. Click the **Record** hyperlink of the request that you want to cancel.  
The record opens with the **General** page displayed.
4. In the record toolbar, click **Cancel Request**. (Note that only you, the requestor, will see this button in the toolbar. Other users cannot see it.)
5. In the confirmation message that is displayed, click **Yes**.  
The system automatically makes a date and time entry when the request was canceled and updates the status of the request to "canceled" on the **Workflow** page.  
The approval request is removed from approvers' work lists.  
**Note:** A cancellation email will only be sent to the user who put the record into workflow.
6. If necessary, make any needed modifications and resubmit your request for approval.

#### To cancel approval requests on the My Requests page

1. Go to the **My Requests** page using the hyperlinks in the **My Requests** portal pane, or by selecting **My Requests** from the **All** drop-down list on the tab bar.
2. The **My Requests** page is displayed.
3. Click the **Active Requests** link (see the [Active Requests Section of My Requests Page](#) image).
4. Select the check-box next to each request that you want to cancel.
5. Click **Cancel**.  
The selected requests are canceled and removed from approvers' work lists.

### 1.9.5 Approving, Rejecting, and Reviewing Requests

After a requestor completes an action that sends a record into an approval process, one or multiple approvers become assignees of the approval request. In addition to approving or rejecting a request, the approver can send the request to someone else to review. Users may also add an additional approval before and/or after approving the request.

### 1.9.5.1 Approving or Rejecting a Request

As an approver for certain processes defined in TeamConnect, you are responsible for approving or rejecting requests from other users to complete certain operations. For example, you may be an approver for invoice, task, or expense posting, or deletion of records.

Depending on how the approval rule is defined, you may be able to edit the record while it is pending approval. If the stop requires only one approver, you become the only approver for the stop for this request. You cannot edit the request if all members of the stop must approve.

If you and the other approvers at the current stop do not make a decision before the expiration date of the request, the request will be automatically approved, rejected, or sent to the Process Manager, depending on the settings made on the administrative side of TeamConnect.

The **My Approvals** page provides you with a central location where you can manage your tasks as an approver for approval processes.

You can access the **My Approvals** page in one of the following ways:

- Click one of the hyperlinks in the **My Pending Approvals** portal pane on your home page.
- Select **Approvals** from the **All** drop-down list on the **Main** menu, which opens the Requests to Approve page.

You can approve or reject requests routed to you for authorization in the following ways:

- From [the My Approvals page](#).
- From [the record for which the request](#) has been made.
- From [an email about the request](#) if your system allows approval through email.

The following events occur after you approve and reject a request.

#### What Happens after You Approve or Reject

Approver's Action	On the My Approvals pages	On the Workflow page of the record	To the event that created the request	To people who receive notifications
<b>Approve</b>	TeamConnect removes the request from your <b>Requests to Approve</b> list.  If the approval stop requires only one approval, TeamConnect also removes	Your approval displays in the <b>Approval History</b> of the record with your internal comments. Other approvers, reviewers, the Process Manager, and the requestor can view the Approval History in the <b>Approval</b>	<b>If the request requires no other approvers.</b>	

	the request from other approvers in the same stop and sends requests to approvers in the next stop.	<b>Requests</b> section of the record.	The requested action (such as invoice posting) takes place.	The requestor is notified by email (if email notifications are set up accordingly).  Depending on the route settings, other users may also receive notification emails.
<b>Rejection</b>	TeamConnect removes the request from approvers in the same stop.	Your rejection displays in the <b>Approval History</b> of the record with your internal comments. Other approvers, reviewers, the Process Manager, and the requestor can view the Approval History in the <b>Approval Requests</b> section of the record.  The comments that you wrote for the requestor displays in the <b>Comments to Requestor</b> field under the <b>Approval Requests</b> section of the record.  If you rejected the posting of an invoice, your <b>Reason for Rejection</b> and <b>Comments to Requestor</b> display in the invoice's <b>Comments to Vendor</b> field	The requested action (such as posting an invoice) does not take place.	Depending on your system settings, the request is sent to the Process Manager, the requestor, or approvers in the first stop of the approval route.

For details about setting your email notifications to receive emails about your requests for approval, see [Setting Up Email Notifications](#).

#### 1.9.5.1.1 Approving or Rejecting from My Approvals

The **My Approvals** page allows you to approve or reject multiple requests at once without opening each request individually. If necessary, you can also [approve or rejecting requests within individual records](#).

##### To approve requests from the My Approvals page

1. Open the **My Approvals** page from the link in the **My Approvals** portal pane or by selecting **Approvals** from the **All** drop-down list on the **Main** menu.

The **Requests to Approve** page displays with a list of requests. If necessary, select the **Record** drop-down and click **Search** to display only certain record types.

2. Click the **Approve** button in the **Action** column of the request(s) you want to approve.

If batch approval is enabled for your system, you can click **Approve** below the table to approve all the selected requests.

3. The **Confirm Approval** popup appears. Enter the reason for approving the request(s) in the **Internal Comment** field. You can enter up to 2000 characters, including spaces.
4. Click **Confirm Approval**.

Several things occur after you approve a request, as described in [the What Happens after You Approve or Reject table](#).

##### To reject requests from the My Approvals page

1. Go to the **My Approvals** page using the link in the **My Approvals** portal pane on your home page or by selecting **Approvals** from the **All** drop-down list on the **Main** menu.

The **Requests to Approve** page displays with a list of requests. If necessary, select the **Record** drop-down and click **Search** to display only certain record types.

2. Click the **Reject** button in the **Action** column of the request(s) you want to approve.

If batch approval is enabled for your system, you can click the **Reject Selected** button below the table to reject all the selected requests.

The **Rejection Confirmation** window displays. Depending on whether this is the final rejection of the request and whether the request is for an invoice record, you may see up to three fields: the **Internal Comment** field, a **Rejection Reason** drop-down list, and a **Comments to Requestor** field.

3. Enter the reason for rejecting the request(s) in the **Internal Comment** field. These comments display in the **Approval History**.
4. If the **Rejection Reason** field displays, select a rejection reason.
5. If the **Comments to Requestor** field displays, enter comments. You can enter up to 2000 characters, including spaces.
6. Click **Confirm Rejection**.

Several things occur after you reject a request, as described in [the What Happens after You Approve or Reject table](#).

#### Requests to Approve Page of My Approvals

Column or Control		Description
<b>Record</b>		<p>This drop-down list filters the list of results on this page. Choose a specific record type from the drop-down list so that only records of that type appear in the list of results.</p> <p>After making your selection, click <b>Search</b> to filter the results. If a search filter appears, enter search criteria and click <b>Search</b> again.</p>
<b>Default</b>		<p>The field that you use to select a record type as the default records that appear when you open the <b>Requests to Approve</b> page. Click <b>Select</b> to save the default record type.</p>
<b>Show Filter/Hide Filter</b>		<p>This link shows or hides the search criteria that you can use to find a certain type of request for a record type.</p>
<b>Search Filter</b>  <i>Note: The search filter only appears if one is specified for the record type you select.</i>	<b>Field</b>	The name of the field you want to search.
	<b>Operator</b>	The option that describes the criteria that you want to use to further narrow your results. See Operators for more information.
	<b>Value</b>	The value you enter that the search uses to return specific results.
	<b>Search</b>	The button that you click to execute the search.
	<b>Clear Values</b>	The link that you click to remove all values from the search
	<b>Reset to Default</b>	The link that you click to reset all operators and values to the default fields.
<b>Requests to Approve</b> table  <i>Note: Different fields may appear</i>	<b>Record</b>	<p>The record for which a user has submitted an approval request. For example, the record could be an invoice or some other type of record. Click the record's hyperlink to view it.</p>



<i>depending on your system settings.</i>	<b>Request</b>	The name of the request that has been submitted. The request name indicates the type of action being requested. If the record is an invoice, the invoice amount is part of the request name.
	<b>Type</b>	The type of record on which a request was submitted (such as an invoice).  This column does not appear if you select a record from the <b>Record</b> drop-down.
	<b>Requestor</b>	The name of the user who triggered the approval request. Click to open the user's contact record.
	<b>Due Date</b>	The date by which the approval request expires. If the approver performs no action before this date, TeamConnect automatically approves, rejects, or sends the request to the Process Manager, depending on how the system's approval rule.  Click the column heading to sort your approval list according to the expiration date.  This column does not appear if the <b>Days Pending Action</b> column is present.
	<b>Days Pending Action</b>	The number of days before the approval request expires.  This column does not appear if the <b>Due Date</b> column is present.
	<b>Last Approver</b>	The name of the last approver that manually approved the request.
	<b>Action</b>	Click the <b>Approve</b> button to approve a request.  Click the <b>Reject</b> button to reject a request.
<b>Approve Selected</b>		The button you use to approve multiple requests at the same time. Select the check-boxes of one or more requests before clicking this button.  This button only appears if your system administrator enables batch approval.
<b>Reject Selected</b>		The button you use to reject multiple requests at the same time. Select the check-boxes of one or more requests before clicking this button.

	This button only appears if your system administrator enables batch rejection and you select one type of record from the <b>Record</b> drop-down list.
<b>Send for Review</b>	Click this button to send the approval request for review to one or more other users. You can send more than one request for review at one time by selecting multiple check-boxes.  See <a href="#">Sending Approval Requests for Review</a> .

#### 1.9.5.1.2 Approving or Rejecting Within Records

Instead of approving or rejecting requests from the **My Approvals** page, you can open records that are pending your approval, review them, and make your decision directly on the record page.

Depending on how the approval rule is defined, you may be able to send a request to another user to approve before making your decision.

##### To send the request to another user before you approve

1. Click the **Send for Approval** button. Your system administrator can enable this button to display.

The **Send for Approval** popup appears.

2. From **Recipient Type**, select whether you want to send the request for review to a **User** or **Group**.
3. From **Recipient**, select the user or group to whom you want to send the approval request.
4. In the **Internal Comment** field, type any comments that you want the approver(s) to receive regarding this request. You can enter up to 2000 characters, including spaces.
5. Click **Send**.

If the new approver(s) approve the request, the request returns to you and you can approve or reject. If the new approver rejects the request, the reject action occurs.

##### To approve a request from within the record

1. Go to the **My Approvals** page using the hyperlink in the **My Approvals** portal pane or by selecting **Approvals** from the **All** drop-down list on the **Main** menu.

The **Requests to Approve** page displays.

2. Click the **Record** hyperlink of the record that you want to approve or reject.

The record opens in Read-only mode and displays the **General** page. The **Approve** button is available on the record toolbar.

**Note:** Depending on how the approval rule is defined, you may be able to send the record to someone else to review before you make your decision. For details, see [Sending Approval Requests for Review](#).

3. Click **Approve** from the record toolbar.

The **Confirm Approval** window opens.

4. Enter your comments in the **Internal Comment** field. These comments will be recorded in the **Approval History**.

You can enter up to 2000 characters, including spaces, in this field.

5. If **Display next approval after my response** displays, click its check-box if you want the next request in your queue to open automatically when you click the **Confirm Approval** button.

6. If **Add a new approver after approval** displays and you want to send the approval request to another user or group, select this check-box.

After you select the check-box, specify the **User** or **Group**.

7. Click **Confirm Approval**.

If you clicked the **Display next approval after my response** check-box, the next request in your queue opens for your action.

If not, you may click the **Previous Record** link or the **Next Record** link to move to another record in your queue.

For additional details about approving a request, see [the What Happens after You Approve or Reject table](#).

### To reject a request from within a record

1. Go to the **My Approvals** page using the hyperlink in the **My Approvals** portal pane or by selecting **Approvals** from the **All** drop-down list on the **Main** menu.

The **Requests to Approve** page displays.

2. Click the **Record** hyperlink of the record that you want to approve or reject.

The record opens in Read-only mode and displays the **General** page. The **Reject** button is available on the record toolbar.

**Note:** Depending on how the approval rule is defined, you may be able to send the record to someone else to review before you make your decision. For details, see [Sending Approval Requests for Review](#).

3. Click **Reject** from the record toolbar.  
The **Confirm Rejection** window opens.
4. Enter your comments in the **Internal Comment** field. These comments will be recorded in the **Approval History**.  
You can enter up to 2000 characters, including spaces, in this field.
5. If the **Rejection Reason** field displays, select a rejection reason for the invoice.
6. If the **Comment to Requestor** field displays, enter comments about the rejection.  
You can enter up to 2000 characters, including spaces, in this field.
7. Click **Confirm Rejection**.

For additional details about rejecting a request, see [the What Happens after You Approve or Reject table](#).

#### 1.9.5.1.3 Approving or Rejecting through Email

If your system administrator has allowed approval and rejection through email, you receive an email when you have a request to approve.

##### To approve or reject a request from an email

**Note:** Clicking the **Approve** or **Reject** buttons in the email you receive does not automatically approve the request. You must send the second email that opens to complete the action.

1. When you receive an approval request email, review the contents of the email to determine what action to take next.  
You should not respond to this email.
2. Click the **Approve** or **Reject** button in the email.

**Note:** *Mailto links must be enabled for your email client for these buttons to work when you click them. You cannot enable mailto links for hotmail accounts.*

3. A new email opens with **[Approved]** or **[Rejected]** and the name of the request in the subject of the email.

Enter **Internal Comments** and **Comments to the Requestor** in the spaces provided if necessary.

If your system requires internal comments and you do not enter comments, you receive an error in the form of another email.

4. Click the **Send** button on the email.

The request is not approved or rejected until you send the email.

After you approve or reject a request, it no longer appears in your list on the **Requests to Approve** page of **My Approvals**.

If a workflow stop requires only one approver, another approver might have already approved or rejected the request when you try to approve it through email. If you approve or reject a request after another approver has completed an action, your action and comments do not go through and you receive an error email.

#### 1.9.5.1.4 Sending Requests for Review

As an approver, you may want to request feedback from one or more TeamConnect users about whether to approve or reject a request.

**Note:** *The approval rule that defines the approval process must be configured for you to have the ability to request reviews.*

After the reviewer submits the review request, you receive their comments in an email (if you are able to receive email notifications). The reviewer's comments also display in [the record's Approval History](#).

If the stop has more than one possible approver but requires only one approval to move to the next stop, you become the only approver for the stop in this request when you send the request for review. You cannot send the request for review if all members of the stop must approve.

**Note:** *If you send a request for review, you must still approve the request by the original expiration date.*

#### To request a review of a single record pending your approval

1. Open the **My Approvals** page by clicking its link in the **My Approvals** portal pane or by selecting **My Approvals** from the **All** drop-down list on the **Main** menu.

The **Requests to Approve** page displays.

2. Click the **Record** link of the request that you want to send for review.

The **General** page of the record is displayed.

3. Click **Send for Review** in the record toolbar.

The **Send for Review** window appears.

4. In the **Send for Review** window, in the **Internal Comment** field, type any comments that you want the reviewer(s) to receive regarding this request. You can enter up to 2000 characters, including spaces.

5. Select whether you want to send the request for review to a user or group.
6. Select the user or group to whom you want to send the review request.
7. Click **Send**.

The following events occur:

- A message appears at the top of the browser window: "Review Request Sent".
- The reviewer receives an email, if email notifications are set accordingly.
- The request displays in the reviewer's **My Approvals** portal pane and **Requests to Review** section on the **My Approvals** page.
- The request for review is recorded in the record's Approval History.

#### To send multiple approval requests for review

1. Go to the **My Approvals** page using the hyperlink in the **My Approvals** portal pane or by selecting **Approvals** from the **All** drop-down list on the **Main** menu.

The **Requests to Approve** page displays.

2. Select the check-boxes next to the approval requests that you want to send for review. As you click check-boxes, the **Internal Comment** field appears for each clicked request.
3. For each request, type any comments that you would like the reviewer(s) to receive in the **Internal Comment** field. You can enter up to 2000 characters, including spaces.
4. Click **Send For Review**.

The **Send for Review** window opens.

5. Select whether you want to send the request for review to a user or group.
6. Select the user or group to whom you want to send the review request.
7. Click **Yes**.

The following events occur:

- The reviewer receives an email, if email notifications are set accordingly.
  - The request is displayed in the reviewer's **My Approvals** portal pane and **Requests to Review** section on the **My Approvals** page.
  - The request for review is recorded in the record's **Approval History**, including your comments.
  - If not all members of your approval stop are required to approve, then the approval request is removed from the other approvers' work lists.
8. If you want to send the records to any other users, repeat steps 2 through 7.

### 1.9.5.2 Reviewing a Request

The approver of a request may want someone else to review the request first. If an approver asks you to review a request, you can provide comments, but you cannot actually approve or reject the request unless you are an approver in the process.

All your review requests appear on the **Requests to Review** page of **My Approvals**, which you can access in one of the following ways:

- Click the **Pending Review** hyperlink in the **My Pending Approvals** portal pane on your home page.
- Select **Approvals** from the **All** tab, which opens the **Requests to Review** page.

If your email notifications are set, you can be notified through email when you receive a request for a review. For details about setting your email notifications to receive emails about review requests, see [Setting Up Email Notifications](#).

You can submit your reviews either [from the My Approvals page](#) or [within the individual records](#) while you are viewing them.

After you submit a review, you can find a record of the review in the [Approval History of the record](#). As the reviewer, if the record is still pending approval after you review it, you may not be able to see the **Approval History**.

Before you submit a review, a review may no longer appear in your reviews request list in the following cases:

- The user who made the request (the requestor) cancels it.
- The request is approved, rejected, or expired, and is therefore no longer in the current approval stop.
- The approver who sent the request is reassigned by the Process Manager.

#### 1.9.5.2.1 Reviewing from My Approvals

The **My Approvals** page allows you to submit multiple reviews at once, to quickly move reviews out of your work list. Or, you can submit them one at a time.

You can also submit reviews directly within the requested records, as described in [Submitting Reviews Through Individual Records](#).

#### To submit reviews from the My Approvals page

1. Go to the **My Approvals** page using the hyperlink to your pending reviews in the **My Approvals** portal pane or by selecting **Approvals** from the **All** drop-down list on the **Main** menu.
2. Click the **Requests to Review** link, if that section is not already displayed.

The list of reviews that have been requested from you is displayed.

3. Type your review comments in the **Comment to Requestor** field for each selected request. You can enter up to 2000 characters, including spaces.
4. Click **submit review**.

Your reviews are sent to the approver who requested it in an email, if emails are set accordingly, and is recorded in the record's **Approval History**, where the approver can view it directly in the record.

#### Requests to Review Section of My Approvals Page

Field/Button	Description
<b>Record</b>	Indicates the record on which there is a pending action for which an approver has requested your review. For example, the record could be an invoice, document, or some other type of record. Click the record's hyperlink to view it.
<b>Type</b>	The type of record for which the request was submitted.
<b>Request</b>	The name of the request that has been submitted. The request name indicates the type of action being requested. If the record is an invoice, the invoice amount is part of the request name.
<b>Requestor</b>	Displays the name of the user who triggered the approval request. Click to open the user's contact record.
<b>Expiration Date</b>	The date by which the approver must submit his or her decision. In order to give the approver time to consider your feedback, submit your review some time before this date.
<b>Approver</b>	The approver who requested a review from you.
<b>Message</b>	Comments from the approver who is requesting a review from you.
<b>Submit Review</b>	Click this button to submit a review for a request. A window will open, allowing you to enter comments.
<b>Internal Comment</b>	Type your review feedback before clicking submit review. You can enter up to 2000 characters, including spaces.



## 1.9.5.2.2 Reviewing Within Records

It may be easier for you to submit a review while you are viewing the record for which a request has been made, such as an invoice.

### To submit your review through a requested record

1. Go to the My Approvals page using the hyperlink to your pending reviews in the **My Approvals** portal pane or by selecting **Approvals** from the **All** drop-down list on the **Main** menu.

2. Click the **Requests to Review** link, if that section is not already displayed.

The list of reviews that have been requested from you is displayed.

3. Click the **Record** hyperlink of the record that you want to review.

The record opens with its **General** page displayed.

4. Click **Submit Review** in the record toolbar.

The **Submit Review** window opens.

Submit Review Window

5. Type your feedback in the **Comment to Requestor** field. You can enter up to 2000 characters, including spaces.
6. Click **Submit Review**.

Your review is sent to the approver who requested it in an email, if emails are set accordingly, and is recorded in the record's **Approval History**, where the approver can view it directly in the record.

## 1.9.6 Making Workflow Decisions

As a Process Manager, you are responsible for keeping approval processes flowing smoothly through the organization.

A Process Manager only needs to take action when the situations in the following table occur:

Situation	How Process Managers Handle the Situation
-----------	---

Requests have errors	<ul style="list-style-type: none"> <li>• <a href="#">Reassign an approver</a> in the workflow process</li> </ul>
Requests expire	<ul style="list-style-type: none"> <li>• <a href="#">Restart the workflow process</a></li> </ul>
Requests have been rejected	<ul style="list-style-type: none"> <li>• <a href="#">Return the workflow process to requestor</a></li> <li>• <a href="#">Return requests with errors to workflow</a></li> </ul>
Actions that create requests fail	<ul style="list-style-type: none"> <li>• <a href="#">Restart the workflow process</a></li> <li>• <a href="#">Re-execute the action of the workflow process</a> that failed</li> </ul>
Requests require replacing an approver	<ul style="list-style-type: none"> <li>• <a href="#">Temporarily reassign an approver</a> in the workflow process</li> <li>• <a href="#">Permanently reassign an approver</a> in the workflow process</li> </ul>
Requests need to move in workflow without being approved or rejected	<ul style="list-style-type: none"> <li>• <a href="#">Moving Requests Forward in a Process</a></li> <li>• <a href="#">Reassigning a Request</a></li> </ul>

The **My Workflow Processes** page provides a central location where you can manage your tasks as a Process Manager for approval processes, also called workflow processes.

To access the **My Workflow Processes** page, do one of the following:

- Click one of the hyperlinks in the **My Workflow Processes** portal pane on your home page.
- Select **Workflow Processes** from the **All** drop-down list on the tab bar.


The **My Workflow Processes** portal pane on your home page displays a summary of the processes in progress for which you are responsible. For example, if you have four processes in progress, the second hyperlink in the portal pane may say, **2 Workflow Processes In Progress (0 rejected to first stop, 2 on track)**.

#### 1.9.6.1 Monitoring Current Requests

The list of current processes that you are responsible for as a Process Manager is available on the **In Progress** and **Attention Required** pages of **My Workflow Processes**. The **In Progress** page lists processes with no issues. The **Attention Required** page lists processes with errors, expirations, or rejections.

From the **In Progress** and **Attention Required** pages, you can redirect the approval route of a request in one of the following ways:

- [Reassign](#) an individual request.
- Restart the approval process.
- [Return](#) the approval request to the requestor.

You can determine the issue for requests on the **Attention Required** page from the **View Log**  button. The table below describes your options for resolving each type of issue.

#### Resolving Process Issues


Route Status	Log text	Description of issue	Options for resolving these issues
<b>Error</b>	Error: User <name> inactive	Approver's user account is inactive but their approval is required in the route.	<ul style="list-style-type: none"> <li>If the request must continue through the specified approval route, contact an administrator for help in resolving the issue with the user or group in the approval route. Then, you can restart the process.</li> </ul>
<b>Error</b>	Error: User with .lineItemList.pro ject.mainAssignee .user not found	<p>A user with a particular role or involvement with the record must approve, but there is no such user.</p> <p>For example, a request must be approved by the main assignee of a matter, but there is no main assignee.</p>	<ul style="list-style-type: none"> <li><i>Reassign</i> the approval to another user or group. See <a href="#">Sending Approval Requests for Review</a>.</li> </ul>
<b>Error</b>	Error: Group <user group name> empty	No users in a group that is part of the approval route.	<ul style="list-style-type: none"> <li><i>Return the request to the requestor</i>. This is a final rejection. Administrator intervention may still be needed if you resubmit the request. See <a href="#">Returning Requests to Requestors</a>.</li> </ul>
<b>Stop Expired</b>	Expired	<p>You receive notification that a stop has expired.</p> <p>Stop expirations are only sent to you if specified in the approval rule.</p>	<ul style="list-style-type: none"> <li><i>Reassign</i> the approval to another user or group. See <a href="#">Sending Approval Requests for Review</a>.</li> <li><i>Restart</i> the process by sending it to the first stop in the approval route. See <a href="#">Restarting Processes</a>.</li> <li><i>Return the request to the requestor</i>. This is considered a final reject, so the user</li> </ul>




<b>Route Expired</b>	Expired	You receive notification that an entire route has expired.  Route expirations are only sent if specified in the approval rule.	would have to re-submit the request to start a new approval process. See <a href="#">Returning Requests to Requestors</a> .
<b>Rejected</b>	Rejected	Rejections are sent if specified in the definition of the process.  The user who rejects the request may provide comments to help you resolve the issue.	

#### Attention Required and In Progress Pages Field Descriptions

Field/Button	Description
<b>Search</b> (Attention Required page only)	Select whether you want to view <b>All Processes, Errors, Expirations, or Rejections</b> .
<b>Record Drop-Down List</b>	<p>In the drop-down box, select a specific system, project, involved, or milestone approval type that you want to display, and then click <b>Search</b>.</p> <p>To further refine the search results, in the adjoining text box, enter identifying text that will help you locate the appropriate approvals, and then click <b>Search</b>. You can enter all or part of the identifying text as follows:</p> <ul style="list-style-type: none"> <li>• <b>Accounts or Documents</b>—The name of the account or document</li> <li>• <b>Appointments</b>—The subject of the appointment</li> <li>• <b>Contacts or Involveds</b>—The first or last name of the contact or involved party</li> <li>• <b>Expenses, History, Tasks, Line Items, or Milestones</b>—A short description of the expense, history, task, line item, or milestone</li> <li>• <b>Invoice</b>—An invoice number</li> <li>• <b>Projects</b>—The name or number of a project</li> </ul> <p>If you leave this box blank, the returned results will be all of the approvals that match the selected record type.</p>

<b>Show Filter/Hide Filter</b>		This link shows or hides the search criteria that you can use to find a certain type of request for a record type.
<b>Search Filter</b>  <i>Note: The search filter only appears if one is specified for the record type you select.</i>	<b>Field</b>	The name of the field you want to search.
	<b>Operator</b>	The option that describes the criteria that you want to use to further narrow your results. See Operators for more information.
	<b>Value</b>	The value you enter that the search uses to return specific results.
	<b>Search</b>	The button that you click to execute the search.
	<b>Clear Values</b>	The link that you click to remove all values from the search
	<b>Reset to Default</b>	The link that you click to reset all operators and values to the default fields.
<b>Attentions Required/In Progress</b> table  <i>Note: Different fields may appear depending on your system settings.</i>	<b>Record</b>	Lists the record for which a user has submitted an approval request. For example, the record could be an invoice, matter, document, or some other type of record. Click the record's hyperlink to view it.
	<b>Type</b>	The type of record for which a request was submitted (such as an invoice or matter).
	<b>Request</b>	The name of the request that has been submitted. The request name indicates the type of action being requested. If the record is an invoice, the invoice amount is part of the request name.
	<b>Requestor</b>	Displays the name of the user who triggered the approval request. Click to open the user's contact record.
	<b>Approval Rule</b>	<p>Displays the name of the approval rule that defines the approval process.</p> <p>If you have the appropriate administrative rights, you can view the details of the rule by clicking the rule's hyperlink. This can help you understand the approval process in order to troubleshoot an issue.</p>

	<b>Route Status</b>	<p>Indicates the status of the process.</p> <p>On your <b>Attention Required</b> page, the status could be:</p> <ul style="list-style-type: none"> <li>• <b>Rejected</b></li> <li>• <b>Stop Expired</b></li> <li>• <b>Route Expired</b></li> <li>• <b>Error</b></li> </ul> <p>On your <b>In Progress</b> page, all processes listed are <b>On Track</b> or <b>Rejected to First Stop</b>. In either case, this means that the process does not currently require you to take any action as the Process Manager. However, processes that were rejected to the first stop may need to be reviewed by you because an issue caused the approval process to begin again.</p>
	<b>Days in Route</b>	Indicates how many days have passed since the beginning of the process.
	<b>Days Remaining</b>	Indicates how many days are left in the approval process, based on the approval rule. Depending on how the approval rule is defined, this could be based on a total number of days for the route, or the total number of days allotted for each stop.
	<b>Action</b>	<ul style="list-style-type: none"> <li>• <b>View Log</b> —Click this button to view the log of events or activities that have occurred as part of the approval process. The log lists the following information for each event: <ul style="list-style-type: none"> <li>○ <b>Stop</b>—The stop in the approval route at which the event occurred.</li> <li>○ <b>Performer</b>—The user who performed the activity.</li> <li>○ <b>Activity</b>—For example, Sent to Process Manager, Expired, Reassigned, or Sent for Review.</li> <li>○ <b>Date/Time</b>—The date and time of the event.</li> <li>○ <b>Internal Comment</b>—Comments entered by reviewers, approvers, or the Process Manager for events. You can enter up to 2000 characters, including spaces.</li> </ul> </li> </ul>

		<ul style="list-style-type: none"> <li>• <b>Reassign</b> —Click this button to open the <b>Reassign</b> page and reassign the approvals to other users. Reassignment is one option for resolving process issues. For details, see <a href="#">Reassigning Processes</a>.</li> </ul>
		<p>Select a process and click <b>Restart</b> to start the approvals over again at the first stop in the route.</p> <p>Restarting is one option for resolving process issues. For details, see <a href="#">Restarting Process</a>.</p>
		<p>Select a process and click <b>Return to Requestor</b> to reject the request and discontinue the approval process.</p> <p>Returning the request to the requestor is one option for resolving process issues. For details, see <a href="#">Returning Requests to Requestors</a>.</p>

### 1.9.6.2 Reviewing Completed Requests

The **Completed Requests** page of **My Workflow Processes** provides you with a list of all completed requests that you can view as a Process Manager.

A process is considered completed when it is approved, rejected, or canceled. You cannot restart, reassign, or return completed requests.



#### To review completed processes

1. Go to the **My Workflow Processes** page using the hyperlinks in the **My Workflow Processes** portal pane or by selecting **Workflow Processes** from the **All** drop-down list on the tab bar.
2. Click the **Completed Requests** page.
3. In the **View** drop-down list, select which processes you want to view, depending on how they were completed or resolved:
  - **All Requests Completed**—All requests that are approved, canceled, or rejected.
  - **Requests Approved**—Requests that approvers in the route approved by the approvers in the route and requests that automatically approved when the route expired.
  - **Requests Canceled**—Requests that the requestor canceled.
  - **Requests Rejected**—Requests that an approver in the route rejected, requests that the process manager returned to the requestor, or requests that TeamConnect rejected when the route expired.
4. Using the **Completed Date From** and **Completed Date To** date fields, select the date range for completion that you want to view.

5. Click **Search**.

The list of processes refreshes to provide you with the filtered list of completed processes.

#### Completed Requests Section of Workflow Processes

Field/Button		Description
<b>Search</b>		This drop-down list filters the list of results on this page by the reasons for request completion. After making your selections, click <b>Search</b> to filter the results.
<b>Completed Date From</b>		The date that specifies the earliest completion date to appear in the search results. Select the  to specify a date.
<b>Completed Date To</b>		The date that specifies the latest completion date to appear in the search results. Select the  to specify a date.
<b>Completed Requests</b> table	<b>Record</b>	Click to open the record for the approval request.
	<b>Request</b>	The name of the approval request.
	<b>Requester</b>	The name of the user who trigger the action that created the request.
	<b>Rule</b>	The record and rule type that triggered the action that created the request.
	<b>Start Date</b>	The start date of the approval process.
	<b>Completed</b>	The date that the approval process completed.
	<b>Action</b>	A description of the action that caused the approval process to complete.

#### 1.9.6.3 Returning Requests to Requestors

As a Process Manager, you may need to return a request for approval to the requestor. Returning the request to the requestor is considered a rejection of the request. If the requestor submits the request again after you have sent it back, a new approval request process begins.

##### To return a request to the requestor



1. Go to the **My Workflow Processes** page using the hyperlinks in the **My Workflow Processes** portal pane or by selecting **Workflow Processes** from the **All** drop-down list on the tab bar.
2. Locate the process that you need to return to the requestor on the **Attention Required** page or the **In Progress** page.
3. Select the process that you want to restart. You can select more than one.  
The **Internal Comment** field displays.
4. Enter a message to the requestor regarding your cancellation of the request. You can enter up to 2000 characters, including spaces.
5. Click **Return to Requestor**.  
The **Return Request** dialog box opens, confirming that you want to return the request to the requestor.
6. Click **Yes**.

Returning a request has the following results:

- The request is not complete. For example, if the request was to post an invoice, the invoice is not posted.
- The request appears on the **Completed** page of the **My Workflow Processes** page with the status **Rejected**.
- The request for approval no longer appears on the list of the approvers' pending requests in their **My Approvals** page.
- Depending on email settings, the requestor may be notified that their request was rejected.

#### 1.9.6.4 Reassigning Individual Requests

As a process manager on the **My Workflow Processes** pages, you can reassign a request from one approver to another in the following situations:

- A request is not yet completed (**In Progress** page).
- A request has an error (**Failed Actions** page).
- A request expired (**Attention Required** page).
- A request has a rejection (**Attention Required** page).

##### To reassign a process

1. Go to the **My Workflow Processes** page using the hyperlinks in the **My Workflow Processes** portal pane or by selecting **Workflow Processes** from the **All** drop-down list on the tab bar.
2. Click **Attention Required** or **In Progress**, depending on whether the process has an issue or not.

The list of processes is displayed.

3. Click the **Reassignment** icon for the process that you want to reassign.

The **Reassignment** page displays for that process.

4. For each stop where reassignment is necessary, select the **Reassignment Action** and then select the user to whom you want to reassign the approval request.

If you are reassigning an expired process or stop to the same user, select **Assign again**.

If necessary, click **Show Approval History** to review the previous events in the process.

5. When you finish selecting assignees, click **OK**.

The new approvers are notified of the request in their **My Approvals** page, and receive an email if their notifications are set accordingly.

The reassignment now appears in the request log when you click the **View Log**  icon.

#### 1.9.6.5 Reassigning All Requests for a User

Use the **Reassignment** page to reassign approval requests from one user to another. When a user is unable to approve for a short period of time, you may want to reassign requests temporarily from the [Temporary Reassignment](#) section. If a user is no longer able to approve at all, you can assign requests permanently from the [Permanent Reassignment](#) section.

**Note:** *If you try to reassign Request A from User 1 to User 2 when both users are required in the same stop, the reassignment does not happen until User 2 has approved the request. When User 2 approves Request A, TeamConnect reassigns the same request from User 1 to User 2, and User 2 must approve again.*

**Note:** *Avoid including reassignment entries that create a circular logic and assign the same request back to the original assignee. For example, if you have a reassignment entry that reassigns Request A from User 1 to User 2 and another entry that reassigns Request A from User 2 to User 3, an entry that reassigns Request A from User 3 to User 1 creates a circular logic that does not end until you delete one of the entries.*

#### To reassign approval requests temporarily

1. Click the **My Workflow Processes** page using the hyperlinks in the **My Workflow Processes** portal pane or by selecting **Workflow Processes** from the **All** drop-down list on the tab bar.
2. Click the **Reassignment** link in the left pane.
3. From the **Temporary Reassignment** section, enter information for a reassignment entry. Refer to [the Temporary Reassignment table](#) for more information about these fields.

4. Click **Add** to save the request.

### To reassign approval requests permanently

1. Click the **My Workflow Processes** page using the hyperlinks in the **My Workflow Processes** portal pane or by selecting **Workflow Processes** from the **All** drop-down list on the tab bar.
2. Click the **Reassignment** link in the left pane.
3. From the **Permanent Reassignment** section, enter information for a reassignment entry. Refer to [the Permanent Reassignment table](#) for more information about these fields.
4. Click **Add** to save the request.

### Conflicting Reassignments

If two reassignment entries conflict, TeamConnect applies the first entry added, which is the first entry in the table.

According to the previous image, TeamConnect reassigns all requests to User 5. If you want TeamConnect to reassign invoices until 11/15/2013 to User 6, delete the first entry and enter it again to change the order.



According the previous image, TeamConnect reassigns in the following way:

- Invoices reassign to User 6 until 11/15/2013.
- Invoices reassign to User 5 from 11/16/2013 to 11/22/2013.
- All other objects reassign to User 5 until 11/22/2013.

**Note:** *If the entries that conflict are in the two different reassignment tables, you cannot tell which entry you added first from the **Reassignment** page. Check the **Workflow** page of a record to determine the first reassignment entry.*



#### Temporary Reassignment Section of the Reassignment Page

Field	Description
<b>Assign From</b>	The name of the user whose approval requests are being assigned.
<b>Assign To</b>	The name of the user who is being assigned to the approval requests.

<b>Record Type</b>	The type of records that you want to reassign. Only record types with active approval requests appear.
<b>Start Date</b>	The date when you want the requests to start reassigning. If you want all requests to reassign immediately, you do not need to select a date.
<b>End Date</b>	The date that you want TeamConnect to stop reassigning requests.
<b>Reassign</b>	The field that specifies whether reassigned requests return to the original assignee after the <b>End Date</b> . Place a checkmark in the check-box to specify that the requests return.
<b>Add</b>	The button you click to add the reassignment entry.
<b>Cancel</b>	The button you click to clear the reassignment fields or cancel your changes to a reassignment entry.
<b>Action</b>	<p>The column that includes the following actions:</p> <ul style="list-style-type: none"> <li>Click the  button to edit the reassignment entry. The fields above the table populate with the reassignment information.</li> </ul> <p>Click <b>Add</b> to save your changes. All workflow requests update with changes you make.</p> <p>Click <b>Cancel</b> to cancel any updates you have made.</p> <ul style="list-style-type: none"> <li>Click the  button to delete the reassignment entry. If the <b>Start Date</b> of the rule has already passed, the reassigned requests return to the original assignee.</li> </ul>

#### Permanent Reassignment Section of the Reassignment Page

Field	Description
<b>Assign From</b>	The name of the user whose approval requests are being assigned.
<b>Assign To</b>	The name of the user who is being assigned to the approval requests.
<b>Record Type</b>	The type of records that you want to reassign. Only record types with active approval requests appear.
<b>Start Date</b>	The date when you want the requests to start reassigning. If you want all requests to reassign immediately, you do not need to

	select a date.
<b>Add</b>	The button you click to add the reassignment entry.
<b>Cancel</b>	The button you click to clear the reassignment fields or cancel your changes to a reassignment entry.
<b>Action</b>	<p>The column that includes the following actions:</p> <ul style="list-style-type: none"> <li>Click the  button to edit the reassignment entry. The fields above the table populate with the reassignment information. Click <b>Add</b> to save your changes. Click <b>Cancel</b> to cancel any updates you have made.</li> </ul> <p>Click <b>Add</b> to save your changes. All workflow requests update with changes you make.</p> <p>Click <b>Cancel</b> to cancel any updates you have made.</p> <ul style="list-style-type: none"> <li>Click the  button to delete the reassignment entry. If the <b>Start Date</b> of the rule has already passed, the reassigned requests return to the original assignee.</li> </ul>

#### 1.9.6.6 Returning Requests with Failed Actions to Workflow

Use the **Failed Actions** page to review errors that occur when the action that triggers a workflow process fails. For example, a prerequisite that prevents an invoice from posting could cause the request to fail and appear on this page after an invoice is approved.

You have two options when an action fails:

- [Execute the failed action](#) again without sending the request to more approvers.
- [Restart the workflow process](#) from the beginning with the first approver.

##### To re-execute the failed action

1. Resolve the issue that caused the error.
2. You can re-execute the failed action in one of the following ways:
  - Click the **Execute** button in the **Action** column of a request.
  - Select the check-box(es) for the approval request(s) that you want to restart and click the **Execute Action** button.
3. From the **Execute Action** popup, click **Yes**.

### To restart the workflow process

1. Resolve the issue that caused the error.
2. You can restart the workflow process in one of the following ways:
  - Click the **Restart** button in the **Action** column of a request.
  - Select the check-box(es) for the approval request(s) that you want to restart and click the **Restart Workflow** button.
3. From the **Restart Request** popup, click **Yes**.

### Failed Actions Fields

Field	Description
<b>Record</b>	The name of the record that is part of the request.
<b>Type</b>	The type of the record that is part of the request.
<b>Request</b>	The action that triggers a workflow process. For example, when TeamConnect creates the request after a user posts an invoice, <b>Post</b> appears in this field.
<b>Error Message</b>	A description of the error that causes the request action to fail. For example, the message might be for an error that occurs when TeamConnect tries to post an invoice, as shown in the <a href="#">Failed Actions Page image</a> .
<b>Action</b>	You can click one of the following buttons to complete an action on this page: <ul style="list-style-type: none"> <li>• <b>Execute</b>—Click this button to execute the failed action again without sending the request to anymore approvers.</li> <li>• <b>Restart</b>—Click this button to restart the workflow process from the beginning with the first approver.</li> </ul>
<b>Execute Action</b>	The button you click to execute the failed action(s) again for all selected requests.
<b>Restart Workflow</b>	The button you click to restart the workflow process(es) for all selected requests.

### 1.9.6.7 Returning Requests With Errors to Workflow

As a Process Manager, you may need to restart a workflow process in the following situations:

- The process expires at its current stop in the approval route.
- The request has been rejected and you believe that the approval process needs to be restarted.
- The process is in progress and you believe that it should restart for some other reason.
- The process encounters an error that one or more users in the approval route are not available.

In most of these circumstances, you can restart the process or from a point in the process without taking additional action.

However, if the process has encountered an error, the system may require some troubleshooting before the process can be restarted. For example, a system administrator may need to modify the approval route, user group, or user account in the system. The process cannot be restarted and removed from your **Attention Required** page until such issues are resolved at the administrative level.

#### To restart a process

1. Go to the **My Workflow Processes** page using the hyperlinks in the **My Workflow Processes** portal pane or by selecting **Workflow Processes** from the **All** drop-down list on the tab bar.
2. Locate the process that you need to restart on the **Attention Required** page or the **In Progress** page.

***Important:** If you are restarting a process with an **Error** status on the **Attention Required** page, you may not be able to restart the process without assistance from an administrator.*

3. Select the process(es) that you want to restart.
4. Click **Restart**.

A popup displays to confirm that you want to restart the request(s).

If you are restarting a process on the **Attention Required** page, the **Start with latest stop** check-box also appears. Select this check-box to send the request to the approver(s) part of the workflow process when the error, expiration, or rejection occurred.

5. Click **Yes**.

The approvers in the first stop of the approval route receive notification of the request. The **Approval Status** on the **Workflow** page of the record updates.

### 1.9.6.8 Moving Requests Forward in A Process

As an administrator, you may want to forward the approval process to another user from a particular stop in the workflow. This option is available for processes in both the **Attention Required** and **In Progress**.

1. Navigate to the Workflow Processes page.
2. Click the Forward icon on the right side of the records.
3. Select which stop you want to forward to from the dropdown menu.

Note: If a workflow is on the final stop, TeamConnect will display a message stating that there are no more stops left to forward to.

## 1.9.7 Using Hold Conditions

*Hold condition functionality is available for users with TeamConnect 4.0.8+.*

Hold conditions provide admins with a measure to ensure that approvals do not go past a given stop if conditions are not met. Used in tandem with universal edit rights for approvers at a stop, hold conditions can also allow an admin to automatically inform users how to resolve the hold and allow workflow users to resolve it themselves. The **Approve** button will be grayed-out until the condition is resolved.

For more details, see [Defining Actions for Approval Rules](#).

### Hold Conditions in Practice

#### ▣ Setup

1. Navigate to the desired object from TeamConnect by clicking **Setup** followed by clicking **Object Definitions** from the Go to... dropdown menu.
2. Click the object. Click the **Rules** tab once it opens.
3. Select the preset rule or create a new one. *More details on rules can be found [here](#).*
4. Click the **Action** tab.
5. For **Stop-level parameters**, users will be able to select a preset [condition](#) via the Hold Condition dropdown. **Note:** *Only conditions on the selected object will be available.*
6. Write a clear, concise message to direct users what the hold condition is, giving them enough information to resolve the issue themselves. **Note:** *There is a 2000 character limit.*
7. Save and close.





## 1.10 Accessibility

TeamConnect is accessible to users with disabilities and complies with Section 508 of the Federal Rehabilitation Act, including accessibility mode and keyboard access.

### 1.10.1 Enabling Accessibility Mode

Use the **General** page of user preferences to provide additional accessibility options for those with vision impairment or difficulty using the mouse.

#### The Go Button

In some drop-down boxes in TeamConnect, selecting an option automatically performs a page refresh, which makes it impossible for keyboard users to select an option. When you enable accessibility mode, a **Go** button appears next to certain drop-down lists. This allows you to press the Up and Down keys to navigate the drop-down list, then press the **Go** button to refresh the page.

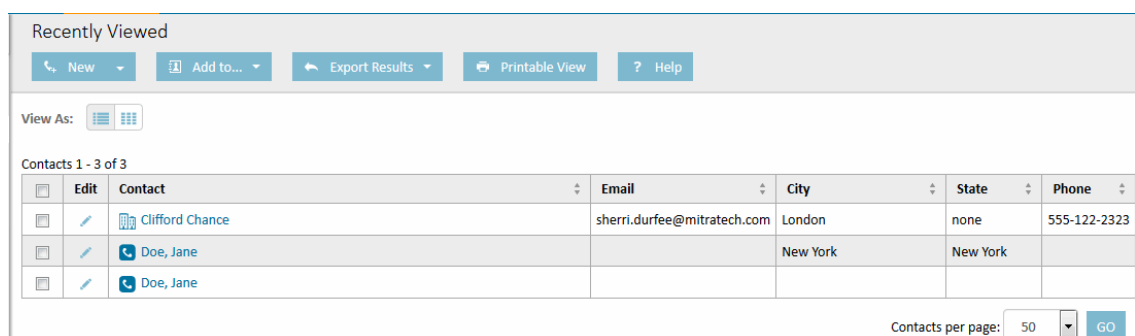
#### To enable keyboard accessibility mode

1. On the global toolbar, click **Preferences**.  
The Preferences **General** page opens.
2. In the **Accessibility** section, click the **Enable accessibility mode** check-box.



General Page, Accessibility Setting

A **Go** button appears next to all drop-down lists where you make selections that change the way information appears on a screen. You can tab to the drop-down list, use the arrow keys to select the appropriate option, and then press the Tab key followed by the Enter key on your keyboard to start the action.



Contacts List View with Accessibility Go Button

## 1.10.2 Using Assistive Technology

Various types of assistive software, such as JAWS® for Windows, are available to make web-based applications like TeamConnect more accessible to users with disabilities. These products achieve accessibility in various ways, such as audibly reading text that appears on the screen.

Depending on your assistive software program, there may be some settings that you must make to work properly in TeamConnect. For example, in JAWS, you must select the **Use Virtual Labels in Forms Mode** option on the **HTML Options** screen.

Make sure that you set all of the appropriate options in your assistive software application. For details on setting HTML options, refer to your assistive software program documentation.

Currently, TeamConnect supports JAWS® 11 or later.

### 1.10.2.1 Using JAWS® with TeamConnect

When using JAWS® with TeamConnect, you should be aware of the following:

#### Pages/Screens

- You can TAB to all clickable elements on a page.
- You can trigger actions or open a link using the Spacebar or Enter keys on your computer.
- All required fields are marked with an asterisk (\*).
- Most pages are readable without style sheets and it should be easy to distinguish page headings and the information hierarchy. Note that some hidden dialog boxes and menus may appear on a page.
- You can navigate TeamConnect's basic functionality without style sheets.
- WAI-ARIA Landmarks are defined so that users can skip repetitive navigation links.
- When a form element is in focus, JAWS® will read the form label.

#### Browsers

Depending on your browser version, JAWS® may announce certain page elements and labels more than once.

#### Limitations

Users with or without disabilities can access TeamConnect with widely varying sets of capabilities, software, and hardware. While every effort has been made to comply with Section 508 Standards, TeamConnect might not satisfy every accessibility requirement.

TeamConnect is not certified for the following:

- Braille access
- Synthesized speech
- Voice input

Please be aware of the following:

- The TeamConnect auto-suggest feature may not work with JAWS®.
- Depending on your browser version, JAWS® may announce certain page elements and labels more than once.

### 1.10.3 Using Keyboard Shortcuts

Keyboard shortcuts provide an alternative to a mouse for navigating on a page.

In most fields in TeamConnect, you can still use the common shortcut keys that you normally use in other applications. TeamConnect also provides its own access keys.

**Note:** The plus (+) signs indicate that the three keys should be pressed at the same time. You do not have to press the plus sign (+) key on your keyboard.

Be aware of the following limitations:

- On the Home Page, you cannot move portal panes around using the keyboard.
- The **Recent Items** drop-down menu is not keyboard accessible.
- The Screen Designer Tool uses drag and drop functionality and is not keyboard accessible.
- Record snapshots are not currently accessible.

#### 1.10.3.1 Commonly Used Access Keys

The table below describes some commonly used access keys on the keyboard and in a browser:

##### Keyboard Shortcuts

Shortcut keys	Functionality
CTRL+A	Selects everything on the screen or in the field.  <i>Tip: To select all in a field, first place the focus on the field that contains the desired information.</i>
CTRL+C	Copies the selected data.
CTRL+F	Opens the <b>Find</b> dialog box where you can enter the text you want to find. This is useful when you are not familiar with the layout of a screen or tab.  <b>Points to Remember:</b> <ul style="list-style-type: none"> <li>• When you are looking for the value of a field, make sure that the screen is in Read-only mode. The command is unable to find values in most fields when the screen is in Edit mode.</li> </ul>

	<ul style="list-style-type: none"> <li>When you are looking for the label of a field, the screen can be in either Read-only or Edit mode.</li> <li>The system searches only the currently displayed tab or screen. To find a value in a record, you must open each page or screen separately and search within it.</li> </ul>
CTRL+V	Pastes the selected data.
CTRL+X	Cuts the selected data.
TAB	Moves between fields on the screen, typically from left to right, according to the pre-determined tab order.
SHIFT+TAB	Moves backward to the field or button that, according to the Tab order, comes before the currently selected field or button.
Up and Down arrows	Selects the desired option in drop-down boxes, multiple-select fields, and radio buttons.
Enter key	Submits the form or triggers the desired action.
Spacebar	Select/Deselect check-boxes.

## Using Firefox

If you are using Firefox, the following applies:

- When TeamConnect provides a **Browse** button to locate a file, users are unable to use the Tab key to place the focus in the blank browser results box. This is because this box is populated with a filename that you select while browsing; users are unable to type in the box.
- Use the ALT + Shift + <key> sequence to move around.

### Web Browser Shortcut Keys

Shortcut keys	Functionality
Back	Move back to a previous page. Typically appears as a back arrow in your browser's toolbar.
Forward	Move forward to a previous page. Typically appears as a forward arrow in your browser's toolbar.
Refresh	Update the contents of the current page. Typically listed under the <b>View</b> menu as <b>Refresh</b> or <b>Reload</b> .

Stop	Stops the page from loading content or halts an action. Typically appears as an <b>X</b> in your browser's toolbar.
------	---

**Note:** If you are using Internet Explorer, there are specific instructions to access functions associated with buttons on the toolbar, such as the **New** or **Save** buttons. See [Using Internet Explorer](#) for more information.

### 1.10.3.2 Using Keyboard Features

TeamConnect provides additional features to provide full keyboard accessibility and better interaction with assistive technology. Specifically, these features let you use the keyboard to make selections from drop-down lists and list boxes.

You do not need to enable accessibility to use these features.

#### Using the Safari Browser

If you are using the Safari browser, to use the TAB key to set focus, it is recommended that you change your preferences as follows: Open your Safari browser, select Preferences, and then select **Press tab to highlight each item**. Restart the Safari browser for the changes to take effect.

#### Making Selections in Drop-down Lists

In all drop-down lists, you can use the UP and DOWN ARROW keys to scroll through the items.

##### To make a selection in a drop-down list

1. Press the TAB key until the focus is on the appropriate drop-down list, for example, the **per page** area in the bottom right corner of a list view.
2. Press the UP and DOWN ARROW keys to select the desired item in the list.
3. When the selected item is in the box, select one of the following actions:
  - If there is no action attached to the selection, click the TAB key to continue to the next field.
  - If an action is attached to the selection, press the TAB key to advance to the **Go** button, and then click ENTER to start the desired action.

#### Making Multiple Selections in List Boxes

You can make multiple consecutive or non-consecutive selections in list fields without using a mouse.

##### To select consecutive items from a list box

1. Press the TAB key until the focus is on the appropriate list box.

2. Use the UP or DOWN ARROW key to move within the list to the first item.
3. Press and hold SHIFT.
4. Use the UP or DOWN ARROW key to move to the last item you want from the list.
5. Release SHIFT.

All of the consecutive items, including the first and last items, are selected.

#### To select non-consecutive items from a list box (IE browsers only)

1. Press the TAB key until the focus is on the appropriate list box.
2. Press SHIFT+F8 to activate the multiple select capability.
3. Use the UP or DOWN ARROW key to move within the list to the desired item.
4. Press SPACEBAR to select the item.
5. Repeat these two previous steps to select additional items.

All of the desired items are selected.

#### 1.10.3.3 TeamConnect-Specific Access Keys

**Note:** These keys do not translate to the Apple/Mac operating system. Certain browsers may have their own shortcut keys that will override these functions. For more information, review the shortcuts preset by your browser.

Shortcut combination keys allow you to access certain commands in TeamConnect. The action is usually executed when you press a key on the keyboard, such as the ENTER key.

The key combinations may vary based on the browser and operating system that you use. Typically, shortcut keys combine the ALT key or CTRL key with other keys. The use of access keys depends on the browser. For example, in Internet Explorer, access keys are used with the ALT key to give focus to the control, and the ENTER key usually activates the control.

**Tip:** Shortcut keys appear in small tool-tips when you place your mouse cursor over the control. For users of assistive technology, your software will announce this information.

The tables below describe the commonly used TeamConnect shortcut key combinations based on the way that you are viewing the screen. If necessary, substitute the ALT key with the appropriate key for your browser and operating system.

##### Record List Screen

Shortcut keys	Functionality
ALT+N	Create a new record.
ALT+P	Printable View.

**Record Edit Screen**

Shortcut keys	Functionality
ALT+C	Save and close.
ALT+E	Edit.
ALT+H	Open the online help.
ALT+M	Hides and collapses the left pane.
ALT+N	Creates a new record on the list screen.
ALT+O	Create a group appointment or create a company contact.  <i><b>Note:</b> This shortcut key is context-sensitive and works based on the type of record that you have open, for example, Appointments or Contacts.</i>
ALT+P	Printable View.
ALT+S	Save.
ALT+V	Save and view.
ALT+W	Cancel and view.
ALT+X	Cancel and close.
ALT+Y	Create a copy.
ALT+Z	Cancel.

**Record View Screen**

Shortcut keys	Functionality
ALT+Z	Delete.

## Using Internet Explorer

If you are using Internet Explorer, use the following information to access the drop-down multiple-function lists in the tool bar, such as the drop-down lists associated with the **New** or **Save** buttons.

### To access drop-down lists on the tool bar

1. Press the shortcut keys that take you to the desired function key. For example, press Alt + S to go to the **Save** button.
2. Press the TAB key to activate the associated drop-down arrow, and then click the ENTER key to open the drop-down menu.
3. Press the shortcut keys that move you down the list to the desired menu item, such as Alt + C to select **Save and Close**.
4. Press the ENTER key to complete the action.

Additionally, you can use the Internet Explorer browser options to change the size of the text in TeamConnect.

## 1.11 TeamConnect Mobile

TeamConnect Mobile is a mobile application that you can use to approve invoices on your phone. You can find the application in your mobile application store.

**Note:** To use TeamConnect Mobile, you must have the rights to approve invoices.

The following types of phones support the mobile application:

- iPhone
- Android
- BlackBerry/BlackBerry Keypad

After you have installed the application on your phone, set up the application to work with your instance of TeamConnect.

When you launch the application, the login screen appears.

### To login to the TeamConnect mobile application

1. If you are not logging in to the application for the first time, enter the URL to the TeamConnect instance.



**Note:** Make sure to begin the URL with `http://` or `https://`.

2. Enter your username and password.

Single sign-on does not work with TeamConnect Mobile. If you do not have TeamConnect credentials, contact your TeamConnect administrator.

3. Select **Login**.

### 1.11.1 Invoices Pending for Approval

When you press the **Approvals** button from the **Home** screen, the **Pending For Approval** screen appears.

From the **Pending For Approval** screen, you can approve, reject, and view general information about invoices. From this screen, you can view up to 100 invoices at one time.

**Note:** From this screen on a BlackBerry Keypad, you can view up to 25 invoices at one time.

The screenshot shows the 'Pending For Approval' screen. At the top, there is a dark blue header with a back arrow on the left, the title 'Pending For Approval' in the center, and a home icon on the right. Below the header is a 'Sort By' dropdown menu set to 'Due Date'. The main content area displays two invoice entries. Each entry includes the following details:

- Invoice Number:** Cong3 of INVOICE1 (for the first entry) and cong3 of INVOICE2 (for the second entry).
- Invoice Amount (USD):** 1265.00
- Matter:** DISP-000019-Gratz vs. Bollinger
- Invoice Date:** 01/13/2012 (264 Days Old)
- Vendor:** [Creator 0094 , Invoice](#)
- Due Date:** 10/05/2012 (Due in 2 Days)

Below the details for each invoice are two buttons: 'Reject' and 'Approve'. A right arrow icon is visible next to the invoice number for each entry.

Pending for Approval Screen


**To approve invoices**

1. From the **Pending For Approval** screen, select **Approve** under the invoice you want to approve.
2. (Optional) Enter a comment for other TeamConnect users in the **Internal Comment** field.
3. Select **Approve**.


#### To reject invoices

1. From the **Pending For Approval** screen, select **Reject** for the invoice you want to reject.
2. (Optional) Enter comments about why you are rejecting the invoice in the **Comment to Requestor** field.
3. (Optional) Enter a comment for other TeamConnect users in the **Internal Comment** field.
4. Select **Reject**.

#### To sort invoices from the Pending For Approval screen

1. From the **Pending For Approval** screen, select the **Sort By**
drop-down field.
2. Select a field for sorting.
3. Select **Done**.
4. Select the  icon to reverse the sorted order.

### 1.11.2 Viewing and Adjusting Invoice Details

From the **Pending For Approval** screen, select the  icon for details of the invoice.

From the **Invoice Details** screen, you see the following tabs:

#### Invoice Details screen

Tab Name	Description
<b>General</b>	<p>General information about the invoice. Refer to <a href="#">Invoice Header Information Section</a> for invoice field descriptions.</p> <p>From this tab, select the link for the vendor to view vendor contact information. Refer to <a href="#">Contacting a Vendor or Approver</a> for more information.</p>
<b>Summary</b>	<p>Invoice summary information separated by fees and expenses.</p> <p>Refer to <a href="#">Adjusting by Total Amount</a> for descriptions of each field.</p>

<b>Line Items</b>	<p>Details for each line items of the invoice. Refer to <a href="#">(Detail Adjustment) Line Items Field Descriptions</a> for descriptions of each field.</p> <p>From this tab, select the link in the line item area to view approver contact information. Refer to <a href="#">Contacting a Vendor or Approver</a> for more information.</p>
<b>History</b>	History of approvals for the invoice.

The screenshot shows the 'Invoice Details' screen with a dark blue header and a home icon. Below the header are four tabs: 'General', 'Summary', 'Line Items', and 'History'. The 'Line Items' tab is selected. Below the tabs, there is a dropdown menu set to 'All Line Items' and a 'Warnings Only' toggle switch set to 'OFF'. The main content area displays a list of three line items:

Item #	Date	Day	Description	Amount 1	Amount 2	Amount 3	Action
1)	07/05/2012	F	B180 Avoidance Actio...	11.00	9.00	99.00	<a href="#">Approver 0100, Invoice</a> >>
2)	04/01/2012	E	E106 Online research	22.00	7.00	154.00	>>
3)	07/05/2012	F	B180 Avoidance Actio...	11.00	9.00	99.00	<a href="#">Approver 0100, Invoice</a> >>

At the bottom of the screen, there are two buttons: 'Reject' and 'Approve'.

Invoice Details Screen

### To adjust summary amounts


1. From the **Summary** tab, select a link for one of the amounts.
2. Select the **Adjust**  drop-down to determine how you want to adjust the amount.
3. Enter the new amount, reduced by amount, or reduced by percentage in the text box.
4. (Optional) Enter comments about the reason for the adjustment in the **Adjustment Comments** field.

5. (Optional) Enter comments for other TeamConnect users in the **Internal Comments** field.
6. Select **Save**.

### To adjust line items

1. From the **Line Items** tab, find the line item you want to adjust.

You can specify which line items appear in the list in both of the following ways:

- Select the  drop-down at the top and select an option.
  - Change the **Warnings Only**  **OFF** field to **ON** to specify that only the line items with warnings appear.
2. Select the  icon for the line item you want to adjust.
  3. To adjust the **Rate**, **Units**, or **Amount** fields, select the link for the field.
  4. Select the Adjust  drop-down to determine how you want to adjust the amount.
  5. Enter the new amount, reduced by amount, or reduced by percentage in the text box.
  6. (Optional) Enter comments about the reason for the adjustment in the **Adjustment Comments** field.
  7. (Optional) Enter comments for other TeamConnect users in the **Internal Comments** field.
  8. Select **Save**.

### 1.11.3 Contacting a Vendor or Approver

When you click the link for a vendor or approver, the **Contact Details** screen appears. From this screen, you can do the following:

- Call a contact—Select the phone number in the **Phone** field.
- Email a contact—Select the email address in the **Email** field.
- View a contact's website—Select the website in the **Web** field.

## 1.12 Troubleshooting and Frequently Asked Questions

This section provides additional frequently asked questions and troubleshooting information about the TeamConnect end user interface.

## 1.12.1 Frequently Asked Questions

This section includes additional information about application features, such as tips or helpful facts.

### 1.12.1.1 Managing Records

**Q: How do I close a record?**

From an open record, click the **Cancel** button. If the record is in edit-mode, the record will be closed without saving any changes. If the record is in read-only mode, you can click a Back to records link instead (for example, from an expense record, you could click the Back to Expenses link at the upper right area of the record to close the record and return to the Expense list page).

**Q: I copied a record but, in the new record, some of the old record's values were replaced by blanks.**

If any record's field displays SECURED as the value then, if you create a copy of that record, the new record will not contain a value for that field.

**Q: I'm trying to delete a record. I have it open on the Edit page, but I don't see a Delete button.**

The delete button appears when you open a record - not in **Edit** mode. To delete a record, click its link in a list view. The delete button appears in the toolbar.

**Q: I edited a record, then saved it, but it still appears in my Unsaved Items list.**

There are four options associated with the **Save** button: **Save and Close**, **Save and Preview**, **Save and New**, and **Save**. The first three options will remove the record from the Unsaved Items list, but the basic "Save" option will save the record yet still leave it open for further editing. Thus the record is still considered an Unsaved Item in this case.

**Q: I'm entering a negative amount in a numeric field, and I'm getting an error.**

In read-only mode, negative numbers are displayed in parentheses. In edit mode, negative numbers are preceded by a minus sign. You should use the minus sign, not the parentheses, when editing a negative number.

**Q: I'm clicking on the Contact search icon, and I'm getting an error.**

If the error is related to some other field on your edit page, you will need to fix the error first. The Contact search tries to jump to a new page, but you are not allowed to navigate away from an edit page that has an error, except by clicking the Cancel button to cancel your edits.

**Q: The required fields on my web browser page look different than the optional fields.**

It's possible that you see a different appearance for some fields on a page because you have a browser add-in (for example, Google Toolbar) that changes screen appearance. TeamConnect itself does not use different visual styles for required fields.

**Q: I'm clicking on a column header, but my list isn't re-sorting.**

In most parts of the TeamConnect user interface, when you see a tabular list of results, you can sort that list by clicking on any column header. However, in a few places the tabular list is pre-sorted and cannot be altered by clicking on column headers. Such places include the workflow displays (Active Requests, Completed Requests, etc.) for various record types.

**Q: How do I save a record?**

Once you finish creating or editing a record, you can click the **Save** button to save the record. You must be in edit mode to access the Save button.

**Save**—Saves the record and you remain in your current mode.

**Save and Close**—Saves the record and returns you to the list view.

**Save and View**—Saves the records and returns you to view mode.

**Save and New**—Saves the current record and opens a new blank record of the same type.

**Q: I entered Custom Search, then requested a search view that had auto-search turned on. However, nothing happened until I manually clicked the Search button.**

The auto-search feature only applies to search views that are designated as collections. When you click on links in the left pane that represent collections, auto-search is used, but it is never used in the Custom Search page.

**Q: I only worked on two records today, but the Recently Viewed collection is showing me many more.**

Recently Viewed presents up to 25 records, in reverse chronological sequence, covering the most recent 90 days, so records from previous days may also appear.

**Q: There are no records in my Recently Viewed list, and I don't see the "Try another search" link.**

"Try another search" only appears when Custom Search finds no records. If you are working with a record for the very first time, the Recently Viewed collection is empty until you actually view or edit one or more records.

### 1.12.1.2 Contacts/Address Books

**Q: When I click the plus sign (+) in the Address Information block to add an additional address, the original address is automatically assigned the Primary status. When I click the plus sign to add an additional phone or fax number or Email or web address, the original information is not automatically selected as the default.**

If you add an additional phone or fax number, Email or web address, you must manually select the primary information.

**Q: I added a contact to an address book and decided to cancel the operation. The contact is still added to the address book.**

Once you add a contact to an address book, you can remove it manually by opening the address book and deleting the contact.

**Q: I subscribed to several address books and only some of them now appear in the collections pane. Where are the rest of them?**

When you subscribe to address books, you can only select and subscribe to the address books that appear in the list that you are currently viewing.

For example, if you have four pages of address books from which to choose, and you want to subscribe to address books on pages one and three, select the address books that you want on the first page, and then click **Subscribe**. Next, go to page three and repeat the process.

*Tip: To see a larger number of address books per page, select a larger number in the **Subscribed Address Books per page** drop-down list.*

**Q: I created a Private address book and I want to change it to a Public address book, but I get an error when I try to do this.**

Once you create an address book, you cannot change its designation. You can delete the address book, and then create it again and give it the appropriate **Public** or **Private** designation.

### 1.12.1.3 Reports

**Q: I refreshed my browser and now the data in my dashboards is distorted.**

Please contact your TeamConnect Certified Professional.

**Q: The reports in my dashboard appear truncated.**

Change your screen resolution. If the problem persists, contact your TeamConnect Certified Professional.

**Q: I ran a report on Documents that filters on the document size, but got "no data to display".**

Be sure that you are supplying the correct range of sizes. Document sizes are in bytes, not kilobytes or megabytes.

**Q: After I add a report to a portal pane, the report does not show up.**

You must be using Flash 10.1 and above to display reports in portal panes.

**Q: I'm editing a report specification and I can't drill down to some fields in the Fields list hierarchy.**

The hierarchy in the Fields list is very extensive, but there are some limitations. Sometimes a field in the list represents an entire entity of its own. For example, the **Vendor** field in an Invoice record represents an entire Contact record of its own. You can usually drill down into entities to see all their fields, but here are some exceptions:

- If the field that you're looking at is a Project field of record type **Any**, or if you're looking at the Project field in the Task record type, the Project field can be expanded to show the system fields associated with the Project, but none of the Project's custom fields are available. Also, the **Categories**, **Current Phase**, **Default Category**, and **Assignee Role** fields of the project are not available for use as search criteria. They are available as result fields.
- A Project field can be navigated to see the Project record's own fields, but the related objects for that Project record aren't available.
- When you drill down into a field of type User (for example, the **Created By** or **Modified By** fields found in most record types) you see information from the Contact record associated with the user, not from the User record itself.

**Q: I've made changes to a report and saved the changes, but I still see the old data.**

When a report is run, the data is automatically stored in your computer's memory. If you navigate away from the report to do other work, when you return to the report, you will still see the old data - even if you changed and saved the report.

For report changes to take effect, one of the following must occur:

- The Data Warehouse is refreshed since the last time you ran the report.
- The report properties are edited and saved (by you, or another user) since the last time you ran the report.
- Your computer runs out of memory, so the oldest report data is deleted to make room for new, incoming report data.



**Note:** This answer not apply to Tabular Only reports because Tabular report data is not saved in memory.

#### 1.12.1.4 Other Questions

**Q: I clicked the browser Back button and, when I went Forward again, my work was gone.**

Incomplete data entry is lost when you use the Back button. Finish and save your data entry before going back.

**Q: After clicking my web browser Stop button, the links on my home page are not responding. Why?**

After you click your browser **Stop** button, you need to click the browser **Refresh** button and wait until the page is completely refreshed (i.e. the browser activity icon stops moving). Afterward, TeamConnect links will become active.

### 1.12.2 Troubleshooting

#### Calendar

**Issue: The appointments in the calendar display appear to be in the wrong time zone.**

Resolution: In order for the calendar to display appointments in the appropriate time zone, the machine on which the browser is running must be set to the same time zone as the time zone set in User Preferences in TeamConnect. Contact your system administrator to set these preferences.

#### Task or Invoice Rates

**Issue: On a task or invoice, the default rate that is defined on the contact Rate page is not appearing or is incorrect.**

Resolution: If you defined more than one default rate on a contacts' Rate page, a start or end date may be incorrect or blank. For specific details, see [Notes for Rates](#).

#### Browser Hangs

**Issue: I clicked the browser's Stop button, and some of the buttons and links on the page don't work now.**

Resolution: If you click the Stop button while a page is loading, you will leave the page in an inconsistent state. You cannot rely on items on the page working correctly unless you click Refresh to force the page to finish loading.

## Document Reports

**Issue:** I am creating a report that uses Document records as data, but I am seeing duplicate document names in my report.

Resolution: If a Document has been checked in more than once - in other words, has multiple versions - each version acts like a separate Document record in the report results. If you'd like the report to return only the current versions of the documents, then apply appropriate filter criteria to the report.

## 1.13 Glossary

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#)

### A

#### **Accounts**

Records used for tracking various amounts of money allocated for specific purposes. There are two kinds of accounts: [Budgets](#) and [Reserves](#).

#### **Address Books**

Groups of [Contacts](#) created by each user as desired. Each user can organize person and company contacts into logical groups. A user could create a group of contacts with whom he or she frequently interacts, so that their phone numbers are easily accessible when needed. Members of address books who are also [Users](#) can be added as attendees to appointments.

#### **Application**

A software program specialized for a particular purpose, for example word processing or email.

#### **Appointments**

Events, engagements, or meetings scheduled for a particular purpose at an assigned place and time with specific [Resources](#). Appointments can be made only for [Users](#) who become the appointment's [Attendees](#). Appointments can be easily associated with specific [Projects](#) if necessary.

#### **Approval**

Authorization from one or more users (known as [approvers](#)) which is required for certain actions that are attempted by users. For example, deleting a matter, posting an [invoice](#), or changing the [phase](#) of a matter could require approval and therefore trigger an approval process. When you trigger an approval process, you are considered the [requestor](#) in the approval process. The attempted action is completed upon the final approval.

Approvals are the major component of TeamConnect [Workflow](#).

#### **Approvers**

Designated [Users](#) who have the authority to approve or reject certain actions attempted by users of TeamConnect. There can be several approvers at each level of the [Approval](#) process. This approval level is also known as an approval [Stop](#). During an approval process, approvers may be able to send

the approval request to [reviewers](#) for review before they make their decision. Approvers can view their approval requests on the **My Approvals** screen.

### **Assignees**

TeamConnect [Users](#) who are assigned to [Projects](#) or [Tasks](#). There can be one or several users assigned to a project. The main assignee is in charge of the whole project, whereas the rest of the assigned users are responsible for certain parts of the project. In tasks, there can be only one assignee at a time. All assignees can be re-assigned, if necessary.

Contacts who are [Involved](#) in the project are different than assignees. For example, you select outside counsel to work on a matter; you do not assign outside counsel to a matter because they are not users.

### **Attendee**

TeamConnect [Users](#) who are listed as taking part in an event, such as a meeting or conference call, for which there is an [Appointment](#) scheduled. Only users, not external contacts, can be attendees.

## **B**

### **Billable Tasks**

[Tasks](#) that have a financial cost and therefore can be posted against (debited from) accounts, for example, buying a new computer is a task that has a specified cost and needs to be posted against an account. Alternatively, as task such as calling a meeting has no cost and is considered non-billable.

Billable tasks can be tracked by their rates on an hourly basis. All billable tasks can be posted against certain accounts according to the posting criteria you set for each individual account.

### **Budget**

Budget is a type of [Account](#) that represents a certain amount of money set aside by the organization or its individual departments to do business for a specific period of time, such as the fiscal year.

## **C**

### **Category**

Items in the user interface that are associated with specific groups of fields displayed in a record.

By adding or deleting categories in a record, you can display or hide the associated sections or fields in that record. In some cases, you might need the user rights to view specific categories and the custom fields belonging to them.

### **Check In or Out**

A way to manage files that are uploaded to the Documents screens so that only one person at a time can edit them. If a file is checked out, no other users can edit the file. When the user is done editing and checks the file in, it becomes available for check out.

### **Child Record**

A record that has a parent record. Records can have hierarchical relationships between records. One (parent) record can have multiple (child) records, whereas each child record can have only one parent.

Child records might be required to have a parent. However, parent records are not required to have child records.

[Categories](#) can also have parent-child relations.

For more details, see "Parent and Child Accounts" on page 87 and "Parent and Child Projects" on page 110.

### **Contact**

Individuals or organizations that conduct business with your organization and in one way or another are involved in the records with which you work. For example, staff, claimants, opposing parties, witnesses, agencies, vendors, outside counsels, and so on can all be described by contact records. You specify [Categories](#) to indicate the type of contact.

### **Contact-Centric Projects**

Projects that are associated with and revolve around a specific contact without which they cannot exist. For example, an insurance claim record cannot be created without specifying an insured contact.

Within a contact-centric project, the contact record is displayed as a single hyperlink. It is created through a search module field and displayed in the general information section of the project.

### **Custom Objects**

Objects created for your organization. Unlike System Objects, they do not come standard with TeamConnect. They are created by your TeamConnect developer. Custom objects can have names used within your organization for records or files, such as matters, and policies. Within this documentation, custom objects are referred to as [Projects](#).

## **D**

### **Default**

A value or setting that TeamConnect automatically selects if you do not specify a substitute. For example, TeamConnect may select certain [Categories](#) as the default for your [Records](#), or set a contact's address as the default.

The default directory in document screens is the directory the operating system searches unless you specify a different directory.

The default can also be an action that the program will take. For example, a user who creates an appointment is automatically (or by default) added as an attendee.

### **Details**

Additional, custom fields that may be created for various records. These fields are created to meet the specific needs of your organization. They are usually organized into groups on the screen, known as [Detail Forms](#) or blocks.

### **Detail Forms**

Certain groups of [Details](#) created for a specific object [Category](#). They can be displayed on a separate tab or as a section on the **General** tab of an object record.

### **Document**

A file uploaded to the Documents area of TeamConnect. These can be plain text, MS Word, MS Excel, PDF, image (JPEG, GIF, BMP), HTML, XML, and so on. They are usually associated with TeamConnect records, for example, a picture taken of a car damaged in an auto accident.

## **E**

### ***Edit mode***

A state of a record in which you can modify it. In most screens, you can switch a record from [Read-only](#) mode to Edit mode by clicking the **Edit** button, if you have the appropriate [Rights](#).

### ***Email Account***

An account for email messaging setup on an email server and accessed by the email client. A default email account is one that has already been set up for you.

### ***Email Client***

An email application for sending and receiving email messages, for example, Microsoft Outlook.

### ***Embedded projects***

Child projects that are embedded in their parent project and cannot be created or accessed outside it. They also have a more simplified screen layout and functionality than child records.

Within the parent record, its embedded records are displayed as multiple hyperlinks in a separate tab or section with a search or batch screen.

### ***Entity***

This term is sometimes used to refer to Objects in TeamConnect.

### ***Expenses***

Costs of doing business in your organization, such as the cost of merchandise bought and the cost of services received. For example, office supplies, transportation, phone calls, meals, and employee training, could be considered expenses. Expenses can be project-specific (associated with a particular [Project](#)), or global (not associated with projects).

Expenses can be posted against [Accounts](#) according to the posting criteria you set for each individual account.

## **F**

### ***Floating Tasks***

[Tasks](#) that do not have a specific due date.

### ***Forums***

Areas in TeamConnect where discussions are held. Forums contain Topics, wherein messages are posted discussing the topic.

## **G**

### ***Global Search***

Global Search allows you to search for specific record types or across **All Records** or **All Projects**. You can enter a few consecutive characters from any record's name or number, select the type of record, and the system will figure out what to search by, name or number, and whether the characters come from the beginning, middle, or end.

### ***Global Tasks***

[Tasks](#) that are not associated with a specific project.

### ***Group***

[User](#) groups contain users who have the same or similar responsibilities within the organization. Each user group has its own set of access rights assigned to it. Your TeamConnect system administrator can provide you with further information about the groups that exist in your organization.

## **H**

### ***History***

Records that reflect chronology or history of the record to which they are related. History entries may contain facts about payments made by adverse carriers, automated assignments, and so on. Within different companies these entries are referred to as docketing or calendaring, logs, diaries, notes, journals, and so on.

### ***Home Page***

A screen which is displayed when you first log into TeamConnect, containing [portal panes](#) of useful links and information specific to your role in your organization. You may have access to more than one home page.

### ***Hyperlink***

An element in an electronic document or on the screen that links to another place in the same document (screen) or to a different location. You click the hyperlink to follow the link. You can create hyperlinks to certain documents in the Documents area.

## **I**

### ***IMAP***

Short for Internet Message Access Protocol, an Internet-specific protocol that allows email clients to retrieve email messages from servers. This protocol cannot send email.

### ***Invoices***

Bills sent by [Vendors](#) who provided your organization with their goods or services. Within TeamConnect, there are two distinguished types of invoices: Invoice Expenses and Invoice Tasks. Invoices can be posted against certain [Accounts](#) according to the posting criteria set for each individual account.

### ***Involved***

[Contacts](#) involved in [Projects](#), such as claimants, witnesses, policemen, plaintiffs, defendants, and so on. Unlike project [Assignees](#), involved parties are not TeamConnect users. The information about the contact's involvement is displayed both in the contact record and the project record.

### ***IP Address***

TCP/IP (Transmission Control Protocol/Internet Protocol) are the two main communication protocols used to access Internet addresses. A computer must have an IP address before it can connect to the Internet. An IP "data packet" (like an email) must have a "send-to" IP address before it can be sent to another computer, just like postal mail needs an envelope with a real address. An IP address is several numbers separated by periods, for example, 192.68.20.50.

## **L**

### ***Line Items***

Items in the list of goods or services specified in the vendor [Invoice](#). The details of each line item, such as the type (task or expense), price, quantity, and dates must be entered on the **Line Items** tab of the corresponding invoice.

## **M**

### ***Main Record***

A record that may have multiple related records that are not hierarchically subordinate to it, but were created for it and hence, are dependent on it.

For example, a Dispute record can have multiple tasks and appointments, an invoice can have multiple history records, a contact can have multiple documents, and so on. Typically, main records are projects. However, all record types can have dependent records of type history and document.

Within a main record, its dependent records are displayed as multiple hyperlinks in a search screen, or in a separate tab or section of the record.

### ***Member***

This term can refer to the following:

[Group](#) members are [Users](#) who are included in a group, from which they can obtain rights to various objects and tools in TeamConnect.

[Route](#) members are users who are included in the Approval process in an Approval Rule.

[Address Book](#) members are contacts that are included in a user's personal address book.

### ***Memo Text***

A type of text field in which you can type free form text of a length up to what your database permits.

### ***Matters***

Types of [records](#) that are specifically designed to meet the legal needs of an organization. The standard TeamConnect matter types are advice & counsel, disputes, and transactions. They are sometimes referred to as "projects," such as in the **Project** search module.

## **N**

### ***Network***

A group of computers that all share the same applications. The applications may reside on a single main computer. The main computer is the server, and the computers (including applications) being served are the clients. Keep in mind that an application may also act as a client.

## O

### OK

A command used to complete a task.

### Other

Describes a different alternative that does not fit under an appropriate category or grouping.

## P

### **Parent-child relationships**

Hierarchical relationships, where one "parent" record can have multiple "child" records, whereas each child record can have only one parent. For example, one dispute record can have multiple account records related to it. Within a parent record, child records are always displayed as hyperlinks on a search screen, on a separate tab or section of the record.

In the **Category** drop-down list on the **Categories** tab, the parent-child relationships between categories are indicated by indentations that reflect the hierarchical relationships between them.

For more details on the parent-child relations between projects, see "Parent and Child Projects" on page 110.

### **Parent Record**

A record of a higher "rank" in the record hierarchy, which may have multiple related records of a lower "rank," often known as [Child](#) Records. In most cases, the parent record is a matter. The only other record type that may have parent-child relations is Account.

Within each child record, the parent record is always displayed as a single hyperlink in the **General Information** section on the **General** tab of the record. You can select the parent record by using the [Search Module field](#) (or a drop-down list, depending on your settings).

### **Phases**

Specific stages each [Project](#) must go through to be completed. Together these phases constitute the project's life cycle.

### **Port**

An input-output point where data can enter or leave a computer. If an application is set up on a computer, for example, Outlook, this computer is the host for this application. A computer port allows the application to communicate outside the computer.

### **Portal Pane**

A component of a [home page](#) that could display items such as your current [projects](#), [tasks](#), and [appointments](#), hyperlinks to the Web, or hyperlinks to create new records. Portal panes in your home page are tailored to your specific needs by your system administrator, to give you quick access to tasks you need to perform frequently.

### **Preferences**



System options that you can use to meet your individual needs and personal preferences. They are listed in the **Options** drop-down list on the **Main** menu bar.

### ***Process Manager***

A TeamConnect user who is responsible for monitoring [approval](#) processes that are sent to approvers by requestors. Process Managers receive notifications of approval requests when there is an error. They may also be notified when a request is rejected or expired. They can restart an approval process to resolve errors, rejections, or expirations, or reassign the task of approving a request to other users when necessary. They can also reject approval requests.

The Process Manager role is assigned to a user group rather than an individual user. However, each user within that group has equal authority to perform Process Manager activities.

### ***Projects***

Types of [records](#) that are specifically designed to meet the business needs of your organization, also known as [Custom Objects](#). All available types of projects for your organization are displayed at the end of the **Go to** and **Create a new** drop-down lists on the **Main** menu bar.

### ***Project-specific Task***

[Tasks](#) that are not associated with a specified [project](#).

### ***Protocol***

A method or type of communication that a computer network uses.

## **R**

### ***Read-only Access***

Record-level security state in which the record cannot be modified as a result of the rights assigned to it or for workflow reasons, for example, when the record is pending approval, and the approval rule prevents anyone from modifying the record. This relates to Record Security and not to the Read-only Mode display state.

### ***Read-only Mode***

A state of a record in which you cannot modify it. Records in Read-only mode can be switched to [Edit mode](#) if you have the appropriate [Rights](#).

### ***Record***

A record is a collection of data about something or someone with reference to a particular matter, such as a lawsuit, meeting, or financial transaction, saved in the database. Records have a number and a name assigned to identify the record in the database and help you search for it.

### ***Rejection***

Denial of an [Approval](#) request to do a user-attempted operation by members of a [Route](#). If the operation is rejected, this is noted on the **Workflow** tab of the [Record](#).

### ***Relation***

An association between one record and another. They are typically displayed as hyperlinks on the screen, and can be established through [Search Module fields](#), or the **Relations** tab.

### ***Related Records***

Associated records that can be accessed from within other records (often collectively referred to as 'parent records') by clicking hyperlinks. Often they may be represented by a separate tab or section in the parent record, which has the structure of a [Search Screen](#). For example, appointments, tasks, histories, expenses, documents, milestones, involved, and accounts are related records within [Projects](#).

### ***Request***

An attempted operation by a user that is not completed until it has been approved by the approvers in a specified approval route. A request can be tracked by the requestor, the approvers who receive the request, and possibly a Process Manager, if one is specified for that approval process.

### ***Requestor***

A user who performs an action that triggers an [approval](#) process, such as posting an [invoice](#), voiding an invoice, or changing the [phase](#) of a [project](#) (such as a matter). The [request](#) is sent to [approvers](#) according to the approval route that is defined for that workflow process. The requestor's request is approved or rejected by the approvers. Requestors can view the progress of their requests on the **My Requests** screen. They can also cancel a request if necessary.

### ***Required Fields***

Fields that must have a value entered or selected in them to save the record. TeamConnect typically indicates which fields are required whenever you attempt to save a record with no values in the required fields. The labels of required fields, by default, are displayed in red with an asterisk next to them. You can change the display settings for required fields through the **Appearance** options accessible from the **Options** drop-down list on the **Main** menu bar.

### ***Reserve***

A type of [Account](#) where the money is allocated from the [Budget](#) for the settlement of matters, such as paying outside counsel fees and other costs accrued while working on a matter, or an amount set aside for claim payments.

### ***Resources***

Means and facilities necessary for conducting an event, for which an appointment has been scheduled. For example, you may need to book a conference room or a projector.

### ***Reviewer***

A user who is asked by an [approver](#) to review an [approval](#) request. Reviewers are not approvers, so they do not approve or reject the request. They can provide feedback to the approver who makes a decision in the process. Reviewers can view their requests for review on the **My Approvals** screen.

### ***Rights***

Permissions to perform operations in TeamConnect, such as accessing, creating, updating (editing), and deleting whole records or parts of them. Rights are assigned to you by your TeamConnect system administrator based on your role within the organization.

### ***Root***

The top [Category](#) in the categories hierarchy tree. It is always automatically added to every record in TeamConnect. In [Projects](#), this category has the same name as the object itself.

### ***Route***

A defined path of users whose [Approval](#) is required to authorize an operation in TeamConnect, such as posting an invoice. The path can consist of one or more [Stops](#), that can each contain one or more members who can approve or reject the operation. Routes are used for [Workflow](#) purposes, to help define and control your business processes.

### **Rules**

Your organization's business [Workflow](#) requirements defined by your TeamConnect system administrator. Rules allow the system to automatically check and validate operations that users attempt, prompt them if values are missing, route requests for [Approvals](#) if necessary, and so on.

## **S**

### **Search Module Fields**

A special utility that allows you to quickly search for a specific record and link it to the record with which you are currently working without leaving its screen. The Search Module is always represented by a field with two icons, **Find** and **Open**, next to it.

### **Search Screen**

A type of screen that is displayed whenever you are looking for a specific record. In the user interface, search screens can be displayed as separate screens, tabs, or sections of a tab. They consist of either two tabs (**Filter** and **Results**) or on the same page as a Search Criteria section and Displayed Record section. You can access these screens if you select an option in the **Go to** drop-down list on the **Main** menu bar or if you click the **Find** icon in record screens.

### **Search Views**

Different sets of search criteria represented by different fields on the **Filter** tab of a [Search Screen](#). All available Search Views are displayed in the **Current View** drop-down list on the toolbar of a [Search Screen](#).

### **Security**

A level of access protection you can set for the records you create. For example, you can make a record Private or Public or you can grant or deny the rights to read, update and delete a record to other [Users](#). All record security details can be viewed through the **Security** tab of the record.

### **Server**

A main computer that services a network or part of a network. For example, a server that manages the flow of email through a network is called an email server. A computer or application that utilizes a server for email messaging is called an email client. In the case of email, using it in a network is done by an email client (application), such as Microsoft Outlook.

### **Shortcut Key**

A special key combination, using the keys on your keyboard, that causes a specific command to be executed. For example, CTRL+C will copy the selected item. Typically, shortcut keys combine the CTRL or ALT keys with some other keys.

### **SMTP**

Short for Simple Mail Transfer Protocol, a protocol for sending email messages between servers.

### **Stop**

An [Approval](#) node, or check-point in a [Route](#) within the TeamConnect [Workflow](#). Each stop consists of a certain number of [Approvers](#), whose authorization is required for a pending action. Each approval route may have several stops. If you are an approver, you receive the approval [request](#) when it reaches your stop.

### **Super User**

A user who is given the rights to access TeamConnect records marked as **Private**.

## **T**

### **Tab**

A user interface element that is displayed at the top of record screens. When clicked, each tab opens a new set of information from the record.

### **Task**

Internal assignments for TeamConnect [Users](#). For example, preparing reports, reviewing documentation, and other activities could be considered tasks. Tasks can be project-specific (associated with a particular [Project](#)) or global (not associated with projects). They can also be billable (posted against an account) or non-billable (a task without a cost).

### **Template**

A computer file with a pre-designed format and structure, for example, for a fax, letter, or expense report, which can be copied to serve as a pattern or model for additional files of the same type.

### **Territory**

A geographic area with which a [Contact](#) is associated. You can associate several territories to a contact.

### **Transaction**

Events in your business that change its financial position, such as buying supplies, paying bills, withdrawing money, or buying equipment. For example, [Expenses](#), [Tasks](#), and [Invoices](#) in addition to regular money withdrawals from and transfers between [Accounts](#) are transactions. Transactions are posted or completed against accounts.

## **U**

### **User**

Persons who have access to TeamConnect. Unlike [Contacts](#), each user has his or her own username and password, which is recorded in his or her user account (accessible only by the Administrator). Users have [Rights](#) assigned to them according to their roles within the organization, which allow them to use the appropriate areas of TeamConnect.

Unlike contacts, users can be assigned to tasks and matters and can be attendees of appointments.

## **V**

### **Vendor**

Outside [Contacts](#) with whom your organization conducts business. These can be suppliers of goods or services, hotels, travel agents, outside counsels, independent consultants, and so on.

### **Version Control**

A method of tracking the edits made to a document that is uploaded to TeamConnect. Version control allows users to access previous versions of a document, and even revert the current version back to a previous version if needed. Version control must be activated by the TeamConnect Administrator.

## **W**

### **Wizard**

A utility within TeamConnect that defines information that needs to be captured to create valid [Records](#). A wizard guides the user through the entire process of capturing the record for the first time. It automatically determines the sequence of steps in which the user should collect data by displaying the appropriate pages on the screen and prompting the user what information needs to be entered. For example, the Appointment Scheduler is a wizard. Other wizards may be defined specifically for your organization.









### **Workflow**

A defined sequence of activities within an organization to produce a final outcome. For example, a [request](#) to post an [invoice](#) might be automatically routed from staff members to a manager to a director for [approval](#), before it is posted to the appropriate budgets. At each stage in the workflow, one or more users are responsible for specific activities. Once the operation is complete (for example, the approval for posting the invoice is obtained), TeamConnect ensures that the users responsible for the next operation receive the data they need to execute their stage of the process.

TeamConnect's standard workflow consists only of approval processes. However, your workflow process could be designed to extend beyond the approval processes to include other activities that help to move information and tasks through your organization.

## **2 Enterprise Administrator Help**

Welcome to the *TeamConnect® Enterprise Administrator Help*.

 <a href="#">Groups</a>	 <a href="#">Users</a>
 <a href="#">Admin Settings</a>	 <a href="#">Logging</a>
 <a href="#">End-user Rights</a>	 <a href="#">Administrator Rights</a>
 <a href="#">Solution Developer Rights</a>	 <a href="#">Documents and Folders</a>

### **2.1 Getting Started**

This chapter provides an overview of common page elements and terminology that are unique to the TeamConnect Admin area. See [Records](#) for general descriptions of application pages.

## 2.1.1 Admin Area User Interface

This section describes the common elements that are found on application pages under the **Admin** tab.

The [Account Administration](#) section includes descriptions of how to manage user and group accounts. From the **Admin** tab, open the Users list page by clicking the **Users** link, or open the Groups list page by clicking the **Groups** link.

The [System Administration](#) section includes descriptions of how to manage system settings and how to manage system and audit logs. From the **Admin** tab, open the Admin Settings **General** page by clicking the **Admin Settings** link. Open the Logging area by clicking the **Logging** link.

### 2.1.1.1 Group Rights Pages

From a group record, the **System Rights** and **Category Rights** pages use a **View** drop-down list to filter the Permissions you will see on the page's body. Details on Group Rights functionality can be found on the [Hierarchy Overview of Group Rights](#).

Action	Permission	Group
Select Unselect	View all records	Some rights selected
Select Unselect	Edit all records	Some rights selected
Select Unselect	Create all records	Some rights selected
Select Unselect	Delete all records	Some rights selected
Select Unselect	Masse dit all records	No rights selected
Select Unselect	View printable view for all records	No rights selected
Select Unselect	View export results for all records	No rights selected
Select Unselect	Perform other actions	Some rights selected

Group Record - System Rights Page, All User Rights View

For example, if you select the **User Rights** option from the **View** drop-down list, the resulting list of rights will apply globally to records and areas that are typically accessed by end users.

Otherwise, if you select Account from the View drop-down list, the Permission list will refresh with a list of rights specific to account records.

The screenshot displays the 'System Rights' configuration page for the 'Account' group. The interface includes a navigation menu on the left with options like 'General', 'Users', 'System Rights', 'Category Rights', 'Tool Rights', 'Object Views', and 'History'. The main content area shows a table of permissions for the 'Account' group, with a 'View' dropdown set to 'Account'. The table lists 16 permissions, all of which are currently granted (indicated by green checkmarks).

Group	Permission
	<input checked="" type="checkbox"/> View Account records
	<input checked="" type="checkbox"/> Edit Account records
	<input checked="" type="checkbox"/> Create Account records
	<input checked="" type="checkbox"/> Delete Account records
	<input type="checkbox"/> Mass edit Account records
	<input type="checkbox"/> View printable view for Account records
	<input type="checkbox"/> View export results for Account records
	<input checked="" type="checkbox"/> View Account all categories
	<input checked="" type="checkbox"/> Create Account all categories
	<input checked="" type="checkbox"/> Delete Account all categories
	<input checked="" type="checkbox"/> View Account all custom fields
	<input checked="" type="checkbox"/> Edit Account all custom fields
	<input checked="" type="checkbox"/> View Account security
	<input checked="" type="checkbox"/> Edit Account security
	<input checked="" type="checkbox"/> View Account all transactions
	<input checked="" type="checkbox"/> Create Account all transactions

**Group Record - System Rights Page for Accounts**

For more information, see [Understanding Rights](#) and [Assigning or Viewing Group System Rights](#).

### 2.1.1.2 Admin Settings General Page

From the **Admin** tab, **Admin Settings** link, you can configure certain system settings. There are multiple pages from the Admin Settings area that are labeled by function. For example, you can configure general settings, email-related settings, search results settings, etc. In addition, you can view the application version number (**About** page) and record notes about changes you make to Admin Settings (**History** page).

Admin Settings - General

**General**

Replace Microsoft Office Special Characters

**Record Locking**

Disable record locking in edit mode

Enable record locking in edit mode

Idle time limit (minutes):  (Enter 0 if you do not want an idle time limit)

Allow users to update Relations for locked or read-only records

**Business Objects**

Version: Business ObjectsXI

Business Objects Single Sign-On Application URL:

Central Management Server Name:

Cookie Domain:

User's TeamConnect and WebIntelligence username and/or password may be different

**Session**

Session timeout (minutes):  Value must be greater than 0 but not greater than 480 (8 hours)

**TeamConnect URL**

The Login URL can be used by different TeamConnect features. The URL can be provided in the body of email notifications. The URL is also used for accessing TeamConnect documents through WebDAV.

Login URL:

Admin Settings - General Page

### 2.1.1.3 Save Options

When working with user or group records, the **Save** button on the upper left area of the page has a drop-down option that allows selection among four save options. The default setting for this option can be determined by administrators as outlined in [Setting up the User Interface](#).

If you click or hover over the drop-down arrow, as shown at the right, the following choices for saving will appear:

- **Save & Close** will save the record, then close the record's page and return you to the previous page you were working on.
- **Save** will save the record but leave the record in edit mode.
- **Save & View** will save the record, then present the record in view-only mode.
- **Save & New** will save the record, then open a new record.

If the label on the Save button is already the choice you want, simply click the button. If not, click the drop-down arrow on the right edge of the button and make your choice from the list that is shown there.



## 2.1.2 Terminology

Between TeamConnect 2.5 and the current TeamConnect version there has been significant redesign of the user's application look and feel. Accordingly, there have been changes to the terms used to refer to application pages and elements on each page. Although the user-facing application area and User Help use the TeamConnect 3.x terminology, there are some remaining TeamConnect 2.5 screen elements in the Designer application area and [Customization Help](#). The Administration Help uses a combination of terminology.

The following table includes common terms used to reference pages or screens and the elements within. Please be aware that the terms may be used interchangeably.

**Synonymous Terms in TeamConnect Guides**

User Help Term	Customization Help Term
Page	Screen or Tab
Section	Block
Collection	Search View, with criteria
Record type	Object

## 2.2 Account Administration

This chapter describes how to plan and set up users and groups of users for TeamConnect. A major consideration for organizing groups is determining which rights should be assigned. This chapter is organized by concepts you should understand before you can start the tasks of creating and configuring users and groups.

**Note:** *Account administration in this chapter refers to setting up users and user groups. [Account records](#) related to finances are different.*

Support staff might use some procedures under the [Users](#) section. For example, [Activating and Inactivating a User](#).

### Before You Begin

The **TeamConnectAdmin** user account is provided for you. Log into TeamConnect as this user for initial system and accounts setup. The TeamConnectAdmin user can create and administer contacts, users, groups, and system settings. For example, create contacts and user accounts for yourself and the solution developer who is going to customize TeamConnect for your organization.

The TeamConnectAdmin user has predefined rights that cannot be changed. The associated contact records do not represent real people, and cannot be changed or deleted. You cannot delete the TeamConnectAdmin user account.

## 2.2.1 Planning and Best Practices

The following are recommendations for planning your accounts setup.

Many of the tasks below will have already been handled by your TeamConnect Certified Professional. For example, all tasks pertaining to assigning rights to groups are handled by the TeamConnect Certified Professional only. Your role as a System Administrator is to create user accounts and assign those users to the appropriate groups. You may also create new groups that have no rights, and use those groups as routes in workflows, such as the flow that approves invoices.

Information about assigning rights to groups is provided here to help you better understand the nature of security in TeamConnect.

### Recommended Practice

You can choose different ways to set up group and user accounts. The following is suggested to save time.

1. Gather business requirements for different groups you need to support. For example, which rights or records access each group will require and which users belong to which group(s).
2. Plan group accounts.  
Organize each group's rights to records, categories/custom fields, and tools.
3. Create group accounts and assign rights to groups.
4. Create user accounts.
5. From each group account, add users to groups.

### Information to Gather

Before assigning rights to a group of users, obtain answers to the following questions:

- Which record types should the group access?
- For each record type, decide the following:
  - Which access rights should be assigned?  
For example, view records, edit records, create records, or delete records.  
For a record type you'll usually also need to assign access rights to sections or pages that are associated with the record type. For example, view record security or edit record security.
  - Which categories should be accessible to the group? Which access rights should be assigned? For example, view record category, add record category, or remove record category.
  - Which custom fields should be available to the group? Which access rights should be assigned? For example, view category custom fields, or edit category custom fields.
- For TeamConnect features or areas, such as User Preferences, Admin Settings, Designer System Settings, object definitions, etc., decide on the following:

- Which access rights should be assigned? For example, view object definition records, edit object definition records, create object definition records, or delete object definition records.

For a TeamConnect feature or area you may also need to assign access rights to sections or pages that are associated with the record type. For example, view object definition rules, edit object definition rules, create object definition rules, or delete object definition rules.

- Which tools should the group be able to use?

Before creating users, gather the following information:

- Basic information about your users. See [Creating or Editing Users](#) for additional details.
- Required authentication mechanisms

### Tips

- If an individual requires custom rights, create a separate group, assign required rights, and add the user to that group.
- During group account planning, consider your users' job roles and organize group rights accordingly. For example, users that share job titles can be grouped with the same rights requirements.

## 2.2.2 Understanding Rights

There are three general ways to provide access to information and functionality in TeamConnect:

- Assigning rights
- Setting user type
- Assigning object views

This section includes conceptual information for understanding common traits of TeamConnect rights.

### Using this Section

- Read Understanding Rights to get an overview of how rights work in TeamConnect. Review the examples provided to apply the concepts to scenarios you might use.
- Use specific procedures to navigate to application pages for assigning rights. For example:
  - [Assigning or Viewing Group System Rights](#)
  - [Assigning Group Category and Custom Field Rights](#)
  - [Assigning Group Tool Rights](#)
- Refer to the appendixes for detailed descriptions of available rights. For example:
  - [Rights for End-users](#)
  - [Rights for Administrators](#)
  - [Rights for Solution Developers](#)

- For more information, see [Creating or Editing Users](#), [Changing Group Default Object Views](#).

## Types of Rights

Rights are used to assign access to view and work with:

- Records  
Categories, custom fields, custom blocks, forms
- TeamConnect features or areas
- Tools

Rights are also divided into three groups:

- **Rights for End-users**—Majority of these rights affect viewing and working with records, categories, and custom fields. This group also includes rights to areas, such as Preferences.
- **Rights for Administrators**—These rights affect viewing and working with user and group accounts, the Admin Settings area, and logs.
- **Rights for Solution Developers**—These rights affect customizing and managing Custom Tools, Global Navigation toolbar, Home Pages, Lookup Tables, Miscellaneous Settings, Object Definitions, Portal Panes, Routes, and viewing and working with System Settings (from the Designer area of the application).

### 2.2.2.1 Rights for End-Users

The rights grouped under the User Rights list view (from a user record's **System Rights** page) refer to rights that are expected to be assigned to a group of end-users.

- Records:
  - Account
  - Appointment
  - Contact
  - Document
  - Expense
  - History
  - Invoice
  - Task
  - Project
- Categories, custom fields, custom blocks, or forms for the record types listed above
- TeamConnect features or areas:
  - User Preferences
  - Scheduled Actions

- Tools (for more information, see [Assigning Group Tool Rights](#))

**Note:** Users can also set access [rights for individual records](#).

TeamConnect Legal Matter Management Records

- Advice & Counsel
- Cost Center
- Dispute
- Transaction

### 2.2.2.2 Rights for Administrators

- Records:
  - Groups
  - Users
- TeamConnect features or areas:
  - **Admin Settings**—System settings that are available from the **Admin** tab.
  - **Logs**—The Logging area includes system loggers, system appenders, audit loggers, and audit appenders.

### 2.2.2.3 Rights for Solution Developers

When assigning rights to a group of solution developers, you will typically assign rights to view or work with TeamConnect features or areas:

- **Administer Custom Tools**—Allows management of custom tools.
- **Global Navigation**—Allows customization of the global navigation toolbar and links that display under the toolbar.
- **Home Pages**—Allows management and customization of home pages in TeamConnect after logging in.
- **Lookup Tables**—Allows management and customization of system lookup tables, custom lookup tables, and multi-currency lookup tables in TeamConnect.
- **Miscellaneous Settings**—Allows management of custom fields that should be considered part of a user's system preferences. Custom fields added to the Designer application, **Miscellaneous Settings** page correspond to fields that display on a user's **Preferences** area, **Other Settings** page. Also includes rights to override record edit locks and to override system search results limits.
- **Object Definitions**—Allows management of system and custom object definitions (record types).
- **Portal Panes**—Allows management and customization of applets that are available for user home pages. Portal Panes can be information-driven, for example, containing TeamConnect

search views or RSS feeds. In addition, Portal Panes can be functionality-driven, for example, containing links to frequently used areas, such as workflow approval or record creation wizards.

- **Routes**—Allows management and customization of the part of a workflow approval rule where approvers, reviewers, process managers, and email notifications are defined.
- **System Settings**—From the Designer application, these are TeamConnect system settings that apply to all users, including Rule Execution, Default Object Views, Authentication Plug-In (default and custom plug-ins), and Clear Cache Properties. These rights also allow access to manage system appenders and audit appenders.

## Access Rights to TeamConnect Designer

If you assign any of the following rights to a group (from a group record's **System Rights** page), its members will be able to access the Designer area of TeamConnect by clicking a **Setup** link from the upper right corner of TeamConnect:

- Any rights resulting from the Setup Rights view.
- Any rights resulting from the Administer Custom Tools view.
- Any rights resulting from the Global Navigation view.
- Any rights resulting from the Home Pages view.
- Any rights resulting from the Lookup Tables view.
- Any rights resulting from the Custom Lookup Tables view.
- Any rights resulting from the Multi-Currency view.
- Any rights resulting from the Miscellaneous Settings view.
- Any rights resulting from the Object Definitions view.
- Any rights resulting from the Portal Panes view.
- Any rights resulting from the Routes view.
- Any rights resulting from the Global Navigation view.
- Any rights resulting from the System Settings view (under the Setup rights list).

A second browser window will open with the Designer area of the application. The initial browser for the User area of the application will remain open.

The screenshot shows the TeamConnect home page. At the top, there is a navigation bar with tabs for Home, Finance, Contacts, Calendar, Documents, Admin, and All Services. Below the tabs, there is a search bar with the text 'Contacts' and a 'GO' button. The main content area is divided into three panels:

- All Services:** A sidebar menu with icons and links for Accounts, Advice & Counsel, Appointments, Approvals, and Calendar.
- My Assigned Disputes:** A table with columns for Name, Category, Phase, and Significant. It lists three disputes:
 

Name	Category	Phase	Significant
<a href="#">DISP-000454-Gravin, Alexander v ABC Inc</a>	Claim > Premise	Intake	No
<a href="#">DISP-000455-Kenneth L. Winters alleges</a>	Employment > Discrimination	Intake	No
- Recently Viewed Contacts:** A table with columns for Name, City, State, and Phone. It lists one contact:
 

Name	City	State	Phone
<a href="#">Aetna Life and Casualty</a>	Springfield	MA	(800) 272-2216

Home Page with Setup Link to Designer Application Area

Rights related to Solution Developers are described in the appendix, [Rights for Solution Developers](#).

## Tips

- If a user is a member of multiple groups, the rights that the user inherits are cumulative. The user inherits all rights assigned to all groups.
- Categories only apply to rights for record types that end-users work with, but not to rights for administrators. For example, when assigning rights to work with user records, you do not need to assign rights to associated categories.
- There are additional rights dependencies that may result in users not able to view expected data. Such data may display as "SECURED". See each record type's rights in the appendix, [Rights for End-users](#), for more information.
  - For example, if you do not give the user or group at least the Read right to Contact, the user or group is unable to add or see appointment attendees, vendors in invoices, contacts associated with tasks and expenses, task and project assignees, and so on, even if the group or user has all the appropriate rights to Appointment, Task, Invoice, and project records.
  - Similarly, if you do not give at least the Read right to a custom object, the user or group is unable to see or select the records of this custom object as parents in other custom object records, as posting criteria in accounts, or see custom fields of type Custom Object with this object, and so on.
- Pre-population rules do not check the current user's rights when they trigger the creation or update of object records. For example, a user without rights to create an account could create a project and automatically trigger a pre-population rule that creates an associated account. You should test and review your templates and pre-population rules to make sure that they comply with your organization's security policies.

#### 2.2.2.4 Rights for System Record Types

This section describes commonalities among rights to view and work with system record types (records that are provided standard with TeamConnect).

From a group's **System Rights** page, if you select a record type from the **View** drop-down list, the page will be updated with a list of rights for that record type.

Each record type's rights can be broken down into two groups:

- **Basic rights**—Include the minimal rights required to view or work with a group of records. These rights include:
  - **View records**—Allows searching, viewing, and printing records.  
  
Allows access to the list page for a record type, advanced search page for a record type. Allows access to **General** and **Workflow** pages for contact records.  
  
For example, if you assign the **View Contact Records** right to a group, group members can view the **Contact List** page, **Advanced Contact Search** page. In addition, group members can open contact records and view the records' **General** and **Workflow** pages. For example, see [Example for Assigning Rights to View Contacts](#).
  - **Edit records**—Allows editing, searching, viewing, and printing records.  
  
Allows access to the list page for a record type, advanced search page for a record type. Allows access to **General** and **Workflow** pages for expense records. For example, see [Example for Assigning Rights to Edit Contacts](#).
  - **Create records**—Allows creating and printing records.  
  
Allows access to the list page for a record type, advanced search page for a record type. Allows access to **General** and **Workflow** pages for expense records. For example, see [Example for Assigning Rights to Create Contacts](#).
  - **Delete records**—Allows deleting, searching, viewing, and printing records.  
  
Allows access to the list page for a record type, advanced search page for a record type. Allows access to **General** and **Workflow** pages for expense records. For example, see [Example for Assigning Rights to Delete Contacts](#).  
  
For Edit, Create, and Delete record rights, a group must also have View record rights. The corresponding ability to search, view, and print records results.
- **Additional rights**—Allow access to view or work with additional pages that belong to the record type, for example, the **Security** page. These rights typically vary per record type. In addition, these rights require that the View records and/or Edit records rights for the record type have also been assigned.

For example, if the **Expense Post** right is assigned to a group, group members can post expenses. This right applies to certain record types but not all.

**Note:** *There may be exceptions for each record type's rights to **General** pages or other page. Refer to the appendix, [Rights for End-users](#), for details.*



## Records Linked From Other Records

Certain records, such as Documents and Histories, appear as pages that belong globally to other records. For example, if a group has rights to work with Documents, then from any system object record (Contact, Expense, Account, Appointment, etc.), a group member will see a page link for **Documents** associated with that record.

Certain records appear as pages that belong to specific other records. For example, the following list displays records that appear as page links for other records:

- **Document**—If rights are assigned for Document records, a **Documents** page link will be available from all system object records.
- **History**—If rights are assigned for History records, a **History** page link will be available from all system object records.
- **Account**—If rights are assigned for Account records, an **Accounts** page link will be available from Project (custom object) records.
- **Expense**—If rights are assigned for Expense records, an **Expenses** page link will be available from Project (custom object) records.
- **Task**—If rights are assigned for Task records, a **Tasks** page link will be available from Project (custom object) records.
- **Attendee**—If rights are assigned for User, Groups, or Contact records, an **Attendees** page link will be available from Appointment records.
- **Appointment**—If rights are assigned for Appointment records, an **Appointments** page link will be available from Project (custom object) records.

## Tips and Warnings

- Administrator rights, such as rights to view or work with User Accounts, are assigned at the record type level but do not have associated categories and custom fields.
- Assigning rights to the Documents record type affects user access to the Documents sub-page of all records. Also note that assigning the **Create Document records** right (to upload documents to a record) requires assigning the **View Document records** right. For additional information on operations that are available with the Create Document records right, see [Document Rights](#).
- If rights are assigned for a group, including Create and Edit record rights, it is likely that Create, Edit, and Delete rights to sub-pages and sections that are part of that record type will also be assigned.
- Certain records, such as invoices, tasks, and expenses, require additional operations such as:
  - Post
  - Void
  - Assign Task
- You can also set [rights for individual records](#).

#### 2.2.2.4.1 Prerequisites for Rights to System Record Types

There are commonly repeating prerequisites to assigning rights. Descriptions follow for basic record rights and well as rights to sections or pages associated with a record type.

### Requirements for Basic Record Rights

The following list describes the dependencies between basic rights for records. Rights like **View Contact records** and **Create Contact records** do not have any prerequisite rights.

- **View records**—No additional rights required.
- **Edit records**—Requires **View records**.

For example, if **Edit Contact records** rights are assigned, **View Contact records** rights must also be assigned.

- **Create records**—Requires **View records**.
- **Delete records**—Requires **View records**.

For example, if **Delete Contact records** rights are assigned, **View Contact records** rights must also be assigned.

### Requirements for Additional Rights to Records

The following list describes the dependencies between rights for additional record pages, such as the **Security** page. Generally rights for additional record pages require the corresponding View record right. For example, to assign **View Expense Security**, **View Expense records** must also be assigned.

- **View Record section/page**—Requires the corresponding View records right.  
For example, if **View Contact security** rights are assigned, **View Contact records** rights must also be assigned.
- **Edit Record section/page**—Requires the corresponding View records right, Edit records right, and View Record section/page right.  
For example, if **Edit Contact security** rights are assigned, **View Contact records**, **Edit Contact records**, and **View Contact security** rights must also be assigned..
- **Create Record section/page**—Requires the corresponding View records right, Edit records right, and View Record section/page right.
- **Delete Record section/page**—Requires the corresponding View records right, Edit records right, and View Record section/page right.

#### 2.2.2.4.2 Rights Impacts on Related Records

For system record types, user access to certain record pages (such as Documents, History) depends on rights set at the Document record rights and History record rights level.

For custom objects (projects/matters), user access to record pages (such as Involved) depend on rights set within the custom object record rights.

### 2.2.2.5 Rights for Categories and Custom Fields

Generally rights are assigned to a category as a prerequisite to assigning rights to its associated custom fields, sections, and forms. Access to categories and custom fields also require access to the related record type. Users looking to set or configure Group Rights should view the following pages for further detail: [Hierarchy Overview of Group Rights](#), [Assigning Group Category and Custom Field Rights](#), [Example for Assigning Rights to Contact Categories and Fields](#).

## Types of Category Rights

By default, each record type has two categories, the base category and a Default category. Typically, a TeamConnect Certified Professional will add additional categories for your company. These custom categories will have corresponding rights.

Each category for a record type has the corresponding rights:

### View Category

- For the base category, allows viewing the Categories page or section of the record.
- For the Default category or custom categories, allows viewing the selected category from the **Categories** page or section.

**Add Category**—Allows adding the category for the record.

For tasks, expenses: allows selecting a default category. This requires both add and remove rights.

**Remove Category**—Allows Removing the category for the record.

For tasks, expenses—allows changing a default category. This requires both add and remove rights.

### View All Custom Fields

Allows user complete view rights for all subfields on the category. If this is not checked, users will not be able to add **Edit All Custom Fields**, as View rights are a precursor to editing.

### Edit All Custom Fields

Allows user complete edit rights for all subfields on the category.

### Individual Fields

For example, for contacts, the corresponding base category rights are View Contact category, Add Contact category, and Remove Contact category.

If a custom category, Contractor, were added for contacts, the corresponding category rights would be: View Contractor category, Add Contractor category, and Remove Contractor category. See [Example for Assigning All Rights to Contacts Pages and Fields](#) and [Example for Assigning Rights to Contact Categories and Fields](#) for examples on assigning rights to categories and custom fields.

## Tips

- For accounts, contacts, invoices, projects, appointments, documents, involved's, and histories—the base View Category right allows viewing the selected category from the **Categories** page or section.

- For tasks, expenses—the base View Category right allows viewing the selected category from the Category drop-down list on the **General** page.
- The list of categories in the **Categories** drop-down list in the **Categories** block displays only those categories to which the user has the Create rights.

## Types of Custom Field Rights

Each custom field that is associated with a record type's category has the corresponding right:

- View Custom Field—Allows viewing the custom field, form, or section.
- Edit Custom Field—Allows entering, changing, deleting field values in the custom field, form, or section.

For example, for expenses, the corresponding base category's custom field rights are View Contact custom fields and Edit Contact custom fields.

If a custom category, Contractor, was added for contacts, and a custom field, Private Investigator, was added, then the corresponding custom field rights could be: View Contractor custom fields and Edit Contractor custom fields.

## Differences in Category Functionality

There are several groups of objects that have different functionality on their record screens in the user interface. These groups of objects are:

- **Account, Contact, Invoice, Project** (custom object)—Record screens of these objects have a **Categories** system block where the user, depending on his or her rights, can work with the appropriate categories.

In the **Categories** drop-down list of the **Categories** block, the user can see all categories for which he or she has Create rights; these are the categories that the user can add. Below the data entry fields, the user can see all the existing categories that are added to each record. Depending on his or her rights, the user might be able to delete these categories.

- **Task and Expense**—Record screens of these objects do not have a separate **Categories** block. Instead, they have a **Default Category** field in the **General** block where the user, depending on his or her rights, can work with the appropriate categories.

By selecting a category in this field, the user does the following:

- a. Adds this category to the record.
- b. Deletes the previously selected category.

***Important:** If users lack Delete rights for a category, they cannot select a new one—even if they have Create rights for the category.*

- **Appointment, Document, History, and Involved**—The record screens of these four objects do have a separate **Categories** block like most system objects. Yet they also have the **Default Category** field in the General block, which results in the following functionality:
  - Depending on their rights, users can add the appropriate category and set it as the default by selecting it in the **Default Category** field.

- However, by selecting a new default category, the user does not delete the previous default category from the record. It can be deleted only by using the **Categories** block.

## Tips

- Rights for adding categories allow users to select or activate a category for the specified record type. Generally the active categories act like filters to provide or restrict access to related custom fields. Categories are created by solution developers with the exception of the primary and default categories, which are provided for every record type.
- Rights for removing categories allow users the ability to deactivate a category for the specified record type. Afterward, the result for the user is that any related custom fields for that category will be hidden from the **Details** page or **Details** section. Categories can only be deleted from the system by solution developers.
- Solution Developer Rights to work with Categories and Custom Fields only apply to rights for records (except users and group).

### 2.2.2.5.1 Prerequisites for Rights to All Categories and Custom Fields

There are commonly repeating prerequisites to assigning rights. Descriptions follow for rights to all categories and well as rights to custom fields for a record type.

## Requirements for Rights to All Categories

The following table describes the dependencies between rights to access all categories for a record type.

Dependencies for Rights to all Categories for a Record Type

Rights	Also Requires ___ Right	Example
<b>View Record all categories</b>	<b>View Records</b>	For example, if the <b>View Contact all categories</b> right is assigned, the <b>View Contact Records</b> right must also be assigned.
<b>Add Record all categories</b>	<b>View Records</b> <b>Edit Records</b> <b>View Record all categories</b>	For example, if the <b>Create Contact all categories</b> right is assigned, the following must also be assigned: <ul style="list-style-type: none"> <li>• <b>View Contact Records</b> right</li> <li>• <b>Edit Contact Records</b> right</li> <li>• <b>View Contact all categories</b> right</li> </ul>
<b>Remove Record all categories</b>	<b>View Records</b> <b>Edit Records</b> <b>View Record all categories</b>	

## Requirements for Rights to All Custom Fields

The following table describes the dependencies between rights to access all custom fields for a record type.

Dependencies for Rights to all Custom Fields for a Record Type

Rights	Also Requires ___ Right	Example
<b>View Record all custom fields</b>	<b>View Records</b> <b>View Record all categories</b>	For example, if the <b>View Contact all custom fields</b> right is assigned, the following must also be assigned: <ul style="list-style-type: none"> <li>• <b>View Contact Records</b> right</li> <li>• <b>View Contact all categories</b> right</li> </ul>
<b>Edit Record all categories</b>	<b>View Records</b> <b>Edit Records</b> <b>View Record all categories</b>	

### 2.2.2.5.2 Prerequisites for Rights to Single Categories and Custom Fields

There are commonly repeating prerequisites to assigning rights. Descriptions follow for basic record rights and well as rights to sections or pages associated with a record type.

## Requirements for Rights to Individual Categories and Custom Fields

The following generally apply to all record types that are standard with TeamConnect:

- When assigning view or edit rights to a custom field, the corresponding rights to the related category are also required.
- Update rights to custom field requires view rights to custom field and view rights to related category.

The following table describes the dependencies between rights to access an individual category for Account, Contact, Invoice, Project (custom object).

Dependencies for Rights to a Category for a Record Type

Rights	Also Requires ___ Right	Example
<b>View Record category</b>	<b>View Records</b>	For example, if the <b>View Contact Employee category</b> right is assigned, the <b>View Contact Records</b> right must also be assigned.

<b>Add Record category</b>	<b>View Records</b> <b>Edit Records</b> <b>View Record category</b>	For example, if the <b>Create Contact Employee category</b> right is assigned, the following must also be assigned: <ul style="list-style-type: none"> <li>• <b>View Contact Records</b> right</li> <li>• <b>Edit Contact Records</b> right</li> <li>• <b>View Contact Employee category</b> right</li> </ul>
<b>Remove Record category</b>	<b>View Records</b> <b>Edit Records</b> <b>View Record category</b>	

The following table describes the dependencies between rights to access a custom field for a record type.

**Dependencies for Rights to a Custom Field for a Record Type**

<b>Rights</b>	<b>Also Requires ___ Right</b>	<b>Example</b>
<b>View Record custom field</b>	<b>View Records</b> <b>View Record category</b>	For example, if the <b>View Contact Full-time custom field</b> right is assigned (associated with the <b>Employee</b> category), the following must also be assigned: <ul style="list-style-type: none"> <li>• <b>View Contact Records</b> right</li> <li>• <b>View Contact Employee category</b> right</li> </ul>
<b>Edit Record custom field</b>	<b>View Records</b> <b>Edit Records</b> <b>View Record category</b>	

Categories can be created with parent-child hierarchies.

The following table describes the dependencies between rights for categories with parent-child relationships for a record type. For the example column, assume that categories have been added to the Contact object definition.

**Dependencies for Rights to Categories with parent-child relationships for a Record Type**

<b>Rights</b>	<b>Also Requires ___ Right</b>	<b>Example</b>

<b>View Record category</b> (child)	<b>View Records</b> <b>View Record category</b> (parent) <b>Add Record category</b> (parent) <b>Remove Record category</b> (parent)	For example, if the <b>View Contact all categories</b> right is assigned, the <b>View Contact Records</b> right must also be assigned.
<b>Add Record category</b> (child)	<b>View Records</b> <b>Edit Records</b> <b>View Record category</b> (child) <b>View Record category</b> (parent) <b>Add Record category</b> (parent) <b>Remove Record category</b> (parent)	For example, assume that a solution developer has created a parent category, External, for Contacts. Also assume that a child category, External Court, was created for Contacts. Assigning the <b>Add External &gt; Court category</b> right requires assigning the <b>View External &gt; Court category</b> , <b>Add External category</b> , and <b>View External category</b> rights.
<b>Remove Record category</b> (child)	<b>View Records</b> <b>Edit Records</b> <b>View Record category</b> (child) <b>View Record category</b> (parent) <b>Add Record category</b> (parent) <b>Remove Record category</b> (parent)	

### 2.2.2.6 Rights for Custom Record Types

Individual records of custom objects are collectively called Projects in the standard TeamConnect documentation. Depending on the number of custom objects defined for your organization, there are several sets of these rights, each group listed under the object's name in the **View** drop-down list on the **Rights** and **Category Rights** pages in user and group accounts, for example Claim, Policy, and Matter. Every time a new custom object, and embedded custom object, is created, a new set of rights are automatically added to the list.

Each set of rights for custom object records includes the following object-level and category-level rights:



- To the custom object itself—Refer to [Project/Matter Rights](#).
- To the custom object's Involved object—Refer to [Project/Matter Involved Rights](#).

**Note:** Even though embedded objects can be accessed only from within custom objects, all rights to them must be set separately like to a regular custom object.

### 2.2.2.7 Rights for TeamConnect Features or Areas

TeamConnect areas include Admin Settings and System Settings. Rights to areas generally include view rights to the related pages and edit rights to change values from related pages. The table below provides a summary of rights to TeamConnect areas that can be assigned and the resulting user features.

**Rights to TeamConnect Areas**

Rights	Related Functionality
<b>View</b>	Viewing related pages
<b>Edit</b>	Entering, changing, or deleting values from related pages

For example, if the **View Admin Settings records** right is assigned to a group, group members can view the **Admin Settings** pages and configured values. In addition, group members can view the **Logging** pages for system logs and audit logs.

For example, if the **Edit Admin Settings records** right is assigned, group members can edit the **Admin Settings** pages. In addition, group members can edit system logger levels, clear system and audit logs.

## 2.2.3 Groups

Group accounts are used to organize users who need similar security rights. In this version of TeamConnect, rights are configured for groups and not for users.

### Group Record Pages

There are six group-specific pages:

- The **General** page lets you enter information about the group, such as Description and ID.
- The **Users** page lets you add users to a group or remove users from a group.
- The **System Rights** page lets you view rights to work with records, application areas, and application features
- The **Category Rights** page lets you view rights to categories and custom fields by record type.
- The **Tool Rights** page lets you view rights to system tools and custom tools.
- The **Object Views** page lets you view the default object views for a group.

Additional pages associated with a group include:

- The **History** page lets you add notes about changes to the group or review previous history notes.

## Collaborati Spend Management (CSM) Default Group

CSM comes with a default CSM Admin group:

User group	Rights
<b>CSM Admin</b>	<p>Members of the <b>CSM Admin</b> group have no Admin (access to the Admin tab), User, category, or Setup (access to the Designer area) rights. They have rights only to the following:</p> <ul style="list-style-type: none"> <li>• <b>CSM Settings</b> object and its embedded objects</li> <li>• <b>Activity Code Authorization</b> object</li> <li>• <b>Expense Code Authorization</b> object</li> <li>• <b>Invoice Payment</b> object</li> <li>• <b>Payment Field Mapping</b> object</li> <li>• <b>Synchronization History</b> object</li> <li>• <b>Task Code Authorization</b> object</li> <li>• <b>Timekeeper</b> object</li> <li>• <b>Vendor</b> object and its embedded objects</li> <li>• <b>Vendor Timekeeper Classification Contact Mgmt</b> object</li> <li>• <b>Invoice Validation Rule Setting</b> tool</li> <li>• <b>Xml Worksheet</b> tool</li> </ul>

## TeamConnect Legal Matter Management Default Groups

Matter Management comes with five user groups with the following sets of pre-configured rights based on their job responsibilities within a legal department:

User group	Rights
<b>Dispute Attorney</b>	<p>Members of the <b>Dispute Attorney</b> group have no Admin rights (access to the Admin tab) or Setup rights (access to the Designer area). They have rights only to the following:</p> <ul style="list-style-type: none"> <li>• All User Rights</li> </ul> <p><i>Caution: "All User Rights" includes the Account object.</i></p> <ul style="list-style-type: none"> <li>• <b>Dispute</b> object, and its embedded objects</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Advice &amp; Counsel</b> object</li> <li>• <b>Cost Center</b> object and its embedded object</li> <li>• Category rights to some Line Item categories</li> <li>• <b>Time Entry Tool</b> tool</li> </ul> <p><i>Note: The contact's default <b>Category</b> needs to be set to Internal (or the current Time Entry Setting, <b>General</b> tab, <b>Timekeeper Category</b> value).</i></p>
<b>Legal Administrator</b>	<p>Members of the <b>Legal Administrator</b> group have no Admin rights (access to the Admin tab) or Setup rights (access to the Designer area). They have rights only to the Dispute object, its related and embedded objects, and to the following:</p> <ul style="list-style-type: none"> <li>• <b>All User Rights</b></li> </ul> <p><i>Caution: "All User Rights" includes the Account object.</i></p> <ul style="list-style-type: none"> <li>• <b>Advice &amp; Counsel</b> object</li> <li>• <b>Cost Center</b> object and its embedded object</li> <li>• <b>Dispute</b> object and its embedded objects</li> <li>• <b>Time Entry Setting</b> object and its embedded objects</li> <li>• <b>Time Period</b> object</li> <li>• <b>Transaction</b> object and its embedded objects</li> <li>• Category rights to some Line Item categories</li> <li>• <b>DeleteAllRecords</b> tool</li> <li>• <b>Import Design</b> tool</li> <li>• <b>Legal Preferences</b> tool</li> <li>• <b>Reassign Work</b> tool</li> <li>• <b>Time Entry Tool</b> tool</li> </ul> <p><i>Note: The contact's default Category needs to be set to Internal (or the current Time Entry Setting, <b>General</b> tab, <b>Timekeeper Category</b> value).</i></p> <ul style="list-style-type: none"> <li>• <b>Xml Worksheet</b> tool</li> </ul>
<b>Paralegal</b>	<p>Members of the <b>Paralegal</b> group have no Admin rights (access to the Admin tab) or Setup rights (access to the Designer area). They have rights only to the following:</p>

	<ul style="list-style-type: none"> <li>• <b>All User Rights</b></li> </ul> <p><i>Caution: "All User Rights" includes the Account object.</i></p> <ul style="list-style-type: none"> <li>• <b>Dispute</b> object and its embedded objects</li> <li>• <b>Advice &amp; Counsel</b> object</li> <li>• <b>Cost Center</b> object and its embedded object</li> <li>• <b>Transaction</b> object, and its embedded objects</li> <li>• Category rights to some Line Item categories</li> <li>• <b>Time Entry Tool</b> tool</li> </ul> <p><i>Note: The contact's default <b>Category</b> needs to be set to Internal (or the current Time Entry Setting, <b>General</b> tab, <b>Timekeeper Category</b> value).</i></p>
<b>Reporting</b>	<p>Members of this group can view the <b>Reports</b> home page.</p> <p>This user group has no Admin, User, Category, Tool, or Setup rights.</p>
<b>Transaction Attorney</b>	<p>Members of the <b>Transaction Attorney</b> group have no Admin rights (access to the Admin tab) or Setup rights (access to the Designer area). They have rights only to the following:</p> <ul style="list-style-type: none"> <li>• <b>All User Rights</b></li> </ul> <p><i>Caution: "All User Rights" includes the Account object.</i></p> <ul style="list-style-type: none"> <li>• <b>Advice &amp; Counsel</b> object</li> <li>• <b>Cost Center</b> object and its embedded object</li> <li>• <b>Transaction</b> object, and its embedded objects</li> <li>• Category rights to some Line Item categories</li> <li>• <b>Time Entry Tool</b> tool</li> </ul> <p><i>Note: The contact's default <b>Category</b> needs to be set to Internal (or the current Time Entry Setting, <b>General</b> tab, <b>Timekeeper Category</b> value).</i></p>

For more information about configuring group accounts, see [Creating or Editing Groups](#).

Because users should not alter TeamConnect's automated financial accounts, you should restrict user access to the Account object. Many rules and wizards depend on the standard configuration.

### 2.2.3.1 Working with Groups

You can perform the following tasks when you are working with groups.

#### Groups List

- [Searching for Groups](#)—Search for a group to view or edit.
- [Viewing Groups](#)—View a list of groups or open an individual group.
- [Creating or Editing Groups](#)—Add or update a group.
- [Deleting a Group](#)—Delete a group.

#### Group General Page

- [Creating or Editing Groups](#)—Add or update a group.

#### Group Users Page

- [Changing Group Members](#)—Add users to a group or remove users from a group.

#### Group System Rights Page

- [Assigning or Viewing Group System Rights](#)—Assign or view rights to work with records, application areas, and application features.

#### Group Category Rights Page

- [Assigning Group Category and Custom Field Rights](#)—Assign category and custom field rights to a group.

#### Group Tool Rights Page

- [Assigning Group Tool Rights](#)—Assign rights to work with system and custom tools.

#### Group Object Views Page

- [Changing Group Default Object Views](#)—Set a group's default object views.

#### Group History Page

- [Viewing or Creating Group History](#)—Create or view notes about changes to a group.

### 2.2.3.2 Searching for Groups

You can search groups using the Advanced Group Search page. Search using multiple field values as criteria. For example, you can search for expenses by Name, Description, and creation date.

**To perform an advanced search for a group**

1. Select the **Admin** tab.
2. Click the **Groups** link under the tab bar.
3. Click the **Advanced Search** link.
4. (optional) To search for a group within a specific collection, in the **Search Within** drop-down list, select the collection name.
5. (optional) To limit the fields used to search, in the **Filter** drop-down list, select an appropriate option.
6. Enter search criteria in the resulting fields. See [Group General Page](#) for descriptions of search criteria fields.
7. Click **Search**.
8. To open a group account, from the right pane, click a group name link.

### 2.2.3.3 Viewing Groups

You can view groups by browsing group collections. Alternatively, you can search for a group.

#### To view groups

1. Click the **Admin** tab.
2. Click the **Groups** link under the tab bar.
3. (optional) From the left navigation pane, click a Group collection link. For example, **Recently Viewed**, **New Groups**, or **All Groups**.
4. To open a group, from the right pane, click a group name link.

### 2.2.3.4 Creating or Editing Groups

Before creating a group, it is recommended to search existing groups to make sure you will not create a duplicate.


Each group of users requires:

- Configuration of rights for record types
- Configuration of rights for categories and related custom fields
- Configuration of rights for tools
- Addition of group members

The following steps describe the general results from assigning rights but each record type might have specific exceptions or rights dependencies. See [Rights for End-users](#), [Rights for Administrators](#), and [Rights for Solution Developers](#) for more information.

**Tip:** You can create a group by making a copy of an existing group and updating its properties, such as Name, members, and rights. See [Creating a Group Using a Template](#).

### To create or edit a group account

1. Select the **Admin** tab.
2. Click the **Groups** link under the tab bar.
3. Select one of the following options:
  - To create a group, click **New**.
  - To edit a group, click the **Edit Record** icon  to the left of a group.
4. From the **General** page, complete the following fields:
  - (required) **Group Name**—Enter the name of this group. Maximum length: 50 characters.
  - (required) **Unique Name**—Enter a unique name to identify this group. Enter letters and numbers in default English only.
  - (optional) **Description**—Enter a description of this group. Maximum length: 250 characters.
5. To set up record rights, click **System Rights** from the left navigation pane.

For more information, see [Assigning or Viewing Group System Rights](#), [Rights for End-users](#), [Rights for Administrators](#), and [Rights for Solution Developers](#).

For an overview on rights, see [Understanding Rights](#).
6. To set up rights for categories and custom fields, click **Category Rights** from the left navigation pane.

For more information, see [Assigning Group Category and Custom Field Rights](#).

For an overview on rights, see [Understanding Rights](#).
7. To enable tools, click **Tool Rights** from the left navigation pane.

Check the tools to enable.

For more information, see [Assigning Group Tool Rights](#).
8. To add users to the group, click **Users** from the left navigation pane. Do the following for each user to add:
  - Click **Add New** Item.
  - From the **Users** drop-down list, select a user to add to the group.

Set a user's primary group from the user's account.

For more information, see [Changing Group Members](#).
9. [Click a save option](#).

#### 2.2.3.4.1 Creating a Group Using a Template

You can use an existing group like a template. You can create a copy of a group and then edit some fields to create a new group.

Field values from the following original group's pages are copied to the new record:

- **General** page
- **Users** page
- **System Rights** page
- **Category Rights** page
- **Tool Rights** page

#### To create a group using a template

1. [Search for a group to copy and open it.](#)
2. Click **Create a Copy**.
3. Edit the group as required. For example, change the **Group Name**.
4. [Click a save option.](#)

Field values from the original group's **History** page are not copied to the new group.

#### 2.2.3.5 Changing Group Members

From a group's **Users** page, you can add or remove users from a group. Afterward, the group members will inherit any additional rights assigned to the group.

#### To change members of a group

1. [Search for a group and open it.](#)
2. Click **Edit**.
3. From the left navigation pane, click **Users**.
4. To add a user to the group:
  - a. Click **Add New Item**.
  - b. From the **Users** drop-down list, select a user to add to the group.  
Set a user's primary group from the user's account.
5. To remove a user from the group:
  - a. Select the check-box to the left of the user to remove membership.
  - b. Click **Remove**.
6. [Click a save option.](#)



### 2.2.3.6 Assigning or Viewing Group System Rights

From a group's **System Rights** page, you can assign or view rights to work with records, application areas, and application features.

You can assign or view rights that apply:

- Globally—If you expect to enable rights across multiple record types or multiple application areas, work with these types of rights to save time. Global rights are organized into three groups: User rights, Admin rights, and Setup rights.

For example, if you select the **View** drop-down list, **User Rights** option then the resulting page you can assign rights to view all contact records, view all document records, view all project records, etc., with one button click.

- Individually—If you expect to enable rights for a limited number of record types (for example, allow restricted access to a couple record types), work with these types of rights.

For example, if you select the **View** drop-down list, **Contacts** option then from the resulting page you can assign more granular level rights such as View Contact records, Edit Contact records, Create Contact records.

System Rights		
Action	Permission	Group
Select Unselect	View all records	Some rights selected
Select Unselect	Edit all records	Some rights selected
Select Unselect	Create all records	Some rights selected
Select Unselect	Delete all records	Some rights selected
Select Unselect	Mass edit all records	Some rights selected
Select Unselect	View printable view for all records	Some rights selected
Select Unselect	View export results for all records	Some rights selected
Select Unselect	Perform other actions	✓ All rights selected

Sample Screen for Assigning Contact (Individual) Record Rights to a Group

Additional information about groups:

- New groups do not have any rights (record rights, category or custom fields rights, or tool rights) assigned by default.
- Rights are cumulative. If a user is a member of multiple groups then the user inherits all rights assigned to all groups.
- If you assign global rights to a group and later add a custom object (record type), category, or custom fields, it is recommended to manually grant access rights to the new object, category, or custom fields.

#### To assign or view individual record rights

1. [Search for a group and open it.](#)
2. To change record rights, from the left navigation pane, click **System Rights**.

From the **View** drop-down list, select a record type (for example, Contacts).

- To assign a specific record or area right, select the check-box for the permission under the **Group** column.

- To unassign a specific record or area right, clear the check-box for the permission under the **Group** column.
3. [Click a save option.](#)

## Group System Rights Page Field Descriptions

Field descriptions for specific records or areas are described in:

- [Rights for End-users](#)
- [Rights for Administrators](#)
- [Rights for Solution Developers](#)

### Group System Rights Page Fields

Label	Description
<b>View drop-down list</b>	<p>A list of record types and application areas to which you can assign access rights, for example, system settings, object definitions, etc.</p> <p>After selecting a list option, the workspace is updated with a list of rights specific to that option. For example, if you select Contacts, the rights list displays only rights for working with contact records.</p>
<b>Action</b>	<p>(Available only in Edit view when a global rights option is selected from the <b>View</b> menu, such as <b>User Rights</b>, <b>Admin Rights</b>, or <b>Setup Rights</b>)</p> <p>Displays both of the following per permission:</p> <ul style="list-style-type: none"> <li>• <b>Select</b>—Enables all permissions for all child records under this rights group.</li> <li>• <b>Unselect</b>—Disables all permissions for all child records under this rights group.</li> </ul>
<b>Permission</b>	<p>The available Permissions list items depends on the type of option selected from the <b>View</b> drop-down list.</p> <ul style="list-style-type: none"> <li>• If a global rights option is selected from the <b>View</b> drop-down list (for example, <b>User Rights</b>, <b>Admin Rights</b>, <b>Setup Rights</b>), permissions include the following: <ul style="list-style-type: none"> <li><b>View all record types</b>—To grant search and read rights for records of this type, select the <b>View all record types</b> check-box.</li> <li><b>Edit all record types</b>—To grant rights to edit records of this type, select the <b>Edit all record types</b> check-box.</li> <li><b>Create all record types</b>—To grant rights to create records of this type, select the <b>Create all record types</b> check-box.</li> </ul> </li> </ul>

	<p><b>Delete all record types</b>—To grant rights to delete records of this type, select the <b>Delete all record types</b> check-box.</p> <p><b>Perform all other actions</b></p> <p><i>Note:</i> See <a href="#">Understanding Rights</a>, <a href="#">Rights for End-users</a>, <a href="#">Rights for Administrators</a>, and <a href="#">Rights for Solution Developers</a>.</p> <ul style="list-style-type: none"> <li>• If an individual rights option is selected from the <b>View</b> drop-down list (for example, Contacts, Documents, etc.), permissions like the following are available: <ul style="list-style-type: none"> <li>View <i>Contact</i> records</li> <li>Edit <i>Contact</i> records</li> <li>Create <i>Contact</i> records</li> <li>Delete <i>Contact</i> records</li> <li>View <i>Contact</i> all categories</li> <li>Create <i>Contact</i> all categories</li> <li>Delete <i>Contact</i> all categories</li> <li>View <i>Contact</i> all custom fields</li> <li>Edit <i>Contact</i> all custom fields</li> </ul> </li> </ul> <p><i>Note:</i> The list above provides examples of permissions related to individual record types but is not all-inclusive.</p>
<b>Group</b>	<p>(Available only when a global rights option is selected from the <b>View</b> menu, such as <b>User Rights</b>, <b>Admin Rights</b>, or <b>Setup Rights</b>)</p> <p>Displays one of the following per permission:</p> <ul style="list-style-type: none"> <li>• <b>All rights selected</b>—Indicates the permission was enabled globally.</li> <li>• <b>Some rights selected</b>—Indicates that permissions for the child record types were enabled individually, resulting in some permissions being disabled.</li> <li>• <b>No rights selected</b>—Indicates that no permissions for the child record types were enabled.</li> </ul>

#### 2.2.3.6.1 Assigning or Viewing Global User Rights for a Group

From a group's **System Rights** page, you can assign or view rights typically associated with end-user functions using a global User Rights view. This option allows for quick assignment of access rights to multiple record types or application areas that end-users typically use.

**Note:** In this topic, global User Rights refers to Permissions available when you select the **User Rights** option from the **View** drop-down list of a group account's **System Rights** page.

System Rights		
System Rights 1 - 8 of 8		View: User Rights
Action	Permission	Group
Select Unselect	View all records	Some rights selected
Select Unselect	Edit all records	Some rights selected
Select Unselect	Create all records	Some rights selected
Select Unselect	Delete all records	Some rights selected
Select Unselect	Mass edit all records	Some rights selected
Select Unselect	View printable view for all records	Some rights selected
Select Unselect	View export results for all records	Some rights selected
Select Unselect	Perform other actions	✓ All rights selected

**Group - System Rights Page for Global User Rights (Edit View)**

### To assign or view global User Rights for a group

1. [Search for a group and open it.](#)
2. From the left navigation pane, click **System Rights**.
3. From the **View** drop-down list, select **User Rights**. To assign rights, do one of more of the following:
  - To assign view rights to all records from the User application area, associated categories and custom fields, associated pages for the records (such as Security), click **Select** next to the **View All Records Permission**.
  - To assign edit rights to all records from the User application area, associated categories and custom fields, associated pages for the records (such as Security), click **Select** next to the **Edit All Records Permission**.
  - To assign create rights to all records from the User application area, associated categories and custom fields, associated pages for the records (such as Security), click **Select** next to the **Create All Records Permission**.
  - To assign delete rights to all records from the User application area, associated categories and custom fields, associated pages for the records (such as Security), click **Select** next to the **Delete All Records Permission**.
  - To assign mass edit rights to all records from the User application area, associated categories and custom fields, associated pages for the records (such as Security), click **Select** next to the **Mass Edit All Records Permission**.
  - To assign any remaining rights to all records from the User application area, associated categories and custom fields, associated pages for the records (such as Security), click **Select** next to the **Perform other actions Permission**.
4. If you are viewing currently assigned rights, the status displays under the Group column. For more information, see [Group System Rights Page Field Descriptions](#).
5. [Click a save option.](#)

2.2.3.6.2 Results from Assigning Global User Rights

The following tables list the granular level rights that are automatically assigned when you assign one or more global User Rights. In this topic, global User Rights refers to Permissions available when you select the **User Rights** option from the **View** drop-down list of a group account's **System Rights** page.

System Rights		
Action	Permission	Group
Select Unselect	View all records	Some rights selected
Select Unselect	Edit all records	Some rights selected
Select Unselect	Create all records	Some rights selected
Select Unselect	Delete all records	Some rights selected
Select Unselect	Mass edit all records	Some rights selected
Select Unselect	View printable view for all records	Some rights selected
Select Unselect	View export results for all records	Some rights selected
Select Unselect	Perform other actions	✔ All rights selected

Group - System Rights Page with User Rights View (Edit View)

The following table displays rights assigned from the group **System Rights** page when the **View** drop-down list **User Rights** option is selected and the **View All Records** permission displays the status **All Rights Selected**.

**Rights Assigned with View all records Permission (for global User Rights)**

Record Type	Individual Right
Account	<ul style="list-style-type: none"> <li>View Account records</li> <li>View Account all categories</li> <li>View Account all custom fields</li> <li>View Account security</li> <li>View Account all transactions</li> </ul>
Appointment	<ul style="list-style-type: none"> <li>View Appointment records</li> <li>View Appointment all categories</li> <li>View Appointment all custom fields</li> <li>View Appointment security</li> <li>View Appointment Resources</li> </ul>
Contact	<ul style="list-style-type: none"> <li>View Contact records</li> <li>View Contact all categories</li> <li>View Contact all custom fields</li> <li>View Contact involvement</li> <li>View Contact rates</li> </ul>

	<ul style="list-style-type: none"> <li>• View Contact relations</li> <li>• View Contact security</li> <li>• View Contact skills</li> <li>• View Contact territories</li> </ul>
Documents	<ul style="list-style-type: none"> <li>• View Document records</li> <li>• View Document all categories</li> <li>• View Document all custom fields</li> <li>• View Document security</li> </ul>
Expenses	<ul style="list-style-type: none"> <li>• View Expense records</li> <li>• View Expense all categories</li> <li>• View Expense all custom fields</li> <li>• View Expense security</li> </ul>
History	<ul style="list-style-type: none"> <li>• View History records</li> <li>• View History all categories</li> <li>• View History all custom fields</li> <li>• View History security</li> </ul>
Invoice	<ul style="list-style-type: none"> <li>• View Invoice records</li> <li>• View Invoice all categories</li> <li>• View Invoice all custom fields</li> <li>• View Line Item records</li> <li>• View Line Item all categories</li> <li>• View Line Item all custom fields</li> <li>• View Invoice security</li> </ul>
Preferences	<ul style="list-style-type: none"> <li>• View Preferences</li> </ul>
Report	<ul style="list-style-type: none"> <li>• View Report records</li> </ul>
Task	<ul style="list-style-type: none"> <li>• View Task records</li> <li>• View Task assignees</li> <li>• View Task all categories</li> </ul>

	<ul style="list-style-type: none"> <li>• View Task all custom fields</li> <li>• View Task security</li> </ul>
Scheduled Actions	<ul style="list-style-type: none"> <li>• View Scheduled Actions records</li> </ul>

The following table displays rights assigned from the group System Rights page when the View drop-down list User Rights option is selected and the Edit All Records permission displays the status All Rights Selected.

**Rights Assigned with Edit all records Permission (from global User Rights)**

Record Type	Individual Right
Account	<ul style="list-style-type: none"> <li>• Edit Account records</li> <li>• Edit Account all custom fields</li> <li>• Edit Account security</li> </ul>
Appointment	<ul style="list-style-type: none"> <li>• Edit Appointment records</li> <li>• Edit Appointment all custom fields</li> <li>• Edit Appointment security</li> <li>• Edit Appointment Attendees</li> <li>• Edit Appointment Resources</li> </ul>
Contact	<ul style="list-style-type: none"> <li>• Edit Contact records</li> <li>• Edit Contact all custom fields</li> <li>• Edit Contact rates</li> <li>• Edit Contact security</li> <li>• Edit Contact skills</li> </ul>
Documents	<ul style="list-style-type: none"> <li>• Edit Document records</li> <li>• Edit Document all custom fields</li> <li>• Edit Document security</li> </ul>
Expenses	<ul style="list-style-type: none"> <li>• Edit Expense records</li> <li>• Edit Expense all custom fields</li> <li>• Edit Expense security</li> </ul>
History	<ul style="list-style-type: none"> <li>• Edit History records</li> </ul>

	<ul style="list-style-type: none"> <li>Edit History all custom fields</li> <li>Edit History security</li> </ul>
Invoice	<ul style="list-style-type: none"> <li>Edit Invoice records</li> <li>Edit Invoice all custom fields</li> <li>Edit Line Item records</li> <li>Edit Line Item all custom fields</li> <li>Edit Invoice security</li> </ul>
Preferences	<ul style="list-style-type: none"> <li>Edit Preferences</li> </ul>
Report	<ul style="list-style-type: none"> <li>Edit Report records</li> <li>Edit Report security</li> </ul>
Task	<ul style="list-style-type: none"> <li>Edit Task records</li> <li>Edit Task assignees</li> <li>Edit Task all categories</li> <li>Edit Task all custom fields</li> <li>Edit Task security</li> </ul>

The following table displays rights assigned from the group System Rights page when the View drop-down list User Rights option is selected and the Create All Records permission displays the status All Rights Selected.

**Rights Assigned with Create all records Permission (from global User Rights)**

Record Type	Individual Right
Account	<ul style="list-style-type: none"> <li>Create Account records</li> <li>Create Account all categories</li> <li>Create Account all transactions</li> </ul>
Appointment	<ul style="list-style-type: none"> <li>Create Appointment records</li> <li>Create Appointment all categories</li> <li>Create Appointment Attendees</li> <li>Create Appointment Resources</li> </ul>
Contact	<ul style="list-style-type: none"> <li>Create Contact records</li> </ul>



	<ul style="list-style-type: none"> <li>• Create Contact all categories</li> <li>• Create Contact rates</li> <li>• Create Contact relations</li> <li>• Create Contact skills</li> <li>• Create Contact territories</li> </ul>
Documents	<ul style="list-style-type: none"> <li>• Create Document records</li> <li>• Create Document all categories</li> </ul>
Expenses	<ul style="list-style-type: none"> <li>• Create Expense records</li> <li>• Create Expense all categories</li> </ul>
History	<ul style="list-style-type: none"> <li>• Create History records</li> <li>• Create History all categories</li> </ul>
Invoice	<ul style="list-style-type: none"> <li>• Create Invoice records</li> <li>• Create Invoice all categories</li> <li>• Create Line Item records</li> <li>• Create Line Item all categories</li> </ul>
Task	<ul style="list-style-type: none"> <li>• Create Task records</li> <li>• Create Task all categories</li> </ul>

The following table displays rights assigned from the group System Rights page when the View drop-down list User Rights option is selected and the Delete All Records permission displays the status All Rights Selected.

**Rights Assigned with Delete all records Permission (from global User Rights)**

Record Type	Individual Right
Account	<ul style="list-style-type: none"> <li>• Delete Account records</li> <li>• Delete Account all categories</li> </ul>
Appointment	<ul style="list-style-type: none"> <li>• Delete Appointment records</li> <li>• Delete Appointment all categories</li> <li>• Delete Appointment Attendees</li> <li>• Delete Appointment Resources</li> </ul>

Contact	<ul style="list-style-type: none"> <li>Delete Contact records</li> <li>Delete Contact all categories</li> <li>Delete Contact rates</li> <li>Delete Contact relations</li> <li>Delete Contact skills</li> <li>Delete Contact territories</li> </ul>
Documents	<ul style="list-style-type: none"> <li>Delete Document records</li> <li>Delete Document all categories</li> </ul>
Expenses	<ul style="list-style-type: none"> <li>Delete Expense records</li> <li>Delete Expense all categories</li> </ul>
History	<ul style="list-style-type: none"> <li>Delete History records</li> <li>Delete History all categories</li> </ul>
Invoice	<ul style="list-style-type: none"> <li>Delete Invoice records</li> <li>Delete Invoice all categories</li> <li>Delete Line Item records</li> <li>Delete Line Item all categories</li> </ul>
Task	<ul style="list-style-type: none"> <li>Delete Task records</li> <li>Delete Task all categories</li> </ul>
Scheduled Actions	<ul style="list-style-type: none"> <li>Delete Scheduled Actions records</li> </ul>

The following table displays rights assigned from the group System Rights page when the View drop-down list User Rights option is selected and the Perform other actions permission displays the status All Rights Selected.

**Rights Assigned with Perform other actions Permission (from global User Rights)**

Record Type	Individual Right
Appointment	<ul style="list-style-type: none"> <li>Scheduler Tool</li> </ul>
Documents	<ul style="list-style-type: none"> <li>View letter generation</li> <li>View XML generation</li> <li>View template data mapping</li> </ul>

	<ul style="list-style-type: none"> <li>• View version</li> <li>• Revert version</li> </ul>
Expenses	<ul style="list-style-type: none"> <li>• Post Expense</li> <li>• Void Expense</li> </ul>
Invoice	<ul style="list-style-type: none"> <li>• Post Invoice</li> <li>• Void Invoice</li> </ul>
Task	<ul style="list-style-type: none"> <li>• Post Task</li> <li>• Void Task</li> <li>• Assign Task</li> </ul>

#### 2.2.3.6.3 Assigning Global Admin Rights to a Group

From a group's **System Rights** page, you can assign rights typically associated with system administrator functions using a global Admin Rights view. This option allows for quick assignment of access rights to multiple record types or application areas that system or business administrators typically use.

*Note: In this topic, global Admin Rights refers to Permissions available when you select the **Admin Rights** option from the **View** drop-down list of a group account's **System Rights** page.*

#### To assign global Admin Rights

1. [Search for a group and open it.](#)
2. From the left navigation pane, click **System Rights**.

From the **View** drop-down list, select **Admin Rights**.

- To assign view rights to all records from the Admin application area, associated pages for the records (such as Security), click **Select** next to the **All Admin View Rights Permission**.
- To assign edit rights to all records from the Admin application area, associated pages for the records (such as Security), click **Select** next to the **All Admin Edit Rights Permission**.
- To assign create rights to all records from the Admin application area, associated pages for the records (such as Security), click **Select** next to the **All Admin Create Rights Permission**.
- To assign delete rights to all records from the Admin application area, associated pages for the records (such as Security), click **Select** next to the **All Admin Delete Rights Permission**.

See [Assigning or Viewing Group System Rights](#) for information about assigning groups of rights to end-users, solution developers, or administrators.

3. [Click a save option.](#)

#### 2.2.3.6.4 Results from Assigning Global Admin Rights

The following tables list the granular level rights that are automatically assigned when you assign one or more global Admin Rights. In this topic, global Admin Rights refers to Permissions available when you select the **Admin Rights** option from the **View** drop-down list of a group account's **System Rights** page.

The following table displays rights assigned from the group **System Rights** page when the **View** drop-down list **Admin Rights** option is selected and the **All admin view rights** permission displays the status All Rights Enabled.

**Rights Assigned with All admin view rights Permission (from global Admin Rights)**

Record Type	Individual Right
Group	<ul style="list-style-type: none"> <li>• View Group records</li> <li>• View Group members</li> </ul>
User	<ul style="list-style-type: none"> <li>• View User records</li> <li>• View User group memberships</li> </ul>
Admin Settings	<ul style="list-style-type: none"> <li>• View Admin Settings records</li> </ul>

The following table displays rights assigned from the group **System Rights** page when the **View** drop-down list **Admin Rights** option is selected and the **All admin edit rights** permission displays the status **All Rights Enabled**.

**Rights Assigned with All admin edit rights Permission (from global Admin Rights)**

Record Type	Individual Right
Group	<ul style="list-style-type: none"> <li>• Edit Group records</li> </ul>
User	<ul style="list-style-type: none"> <li>• Edit User records</li> </ul>
Admin Settings	<ul style="list-style-type: none"> <li>• Edit Admin Settings records</li> </ul>

The following table displays rights assigned from the group **System Rights** page when the **View** drop-down list **Admin Rights** option is selected and the **All admin create rights** permission displays the status **All Rights Enabled**.

**Rights Assigned with All admin create rights Permission (from global Admin Rights)**

Record Type	Individual Right
Group	<ul style="list-style-type: none"> <li>• Create Group records</li> <li>• Create Group members</li> </ul>
User	<ul style="list-style-type: none"> <li>• Create User records</li> <li>• Create User group memberships</li> </ul>

The following table displays rights assigned from the group **System Rights** page when the **View** drop-down list **Admin Rights** option is selected and the **All admin delete rights** permission displays the status **All Rights Enabled**.

**Rights Assigned with All admin delete rights Permission (from global Admin Rights)**

Record Type	Individual Right
Group	<ul style="list-style-type: none"> <li>• Delete Group records</li> <li>• Delete Group members</li> </ul>
User	<ul style="list-style-type: none"> <li>• Delete User records</li> <li>• Delete User group memberships</li> </ul>

#### 2.2.3.6.5 Assigning Global Setup Rights to a Group

From a group's **System Rights** page, you can assign rights typically associated with solution developer functions using a global Setup Rights view. This option allows for quick assignment of access rights to multiple record types or application areas that solution developers typically use.

**Note:** In this topic, *global Setup Rights* refers to Permissions available when you select the **Setup Rights** option from the **View** drop-down list of a group account's **System Rights** page.

#### To assign global Setup Rights

1. [Search for a group and open it.](#)
2. From the left navigation pane, click **System Rights**.  
From the **View** drop-down list, select **Setup Rights**.
  - To assign view rights to all records from the Designer (Setup) application area, associated pages for the records (such as Security), click **Select** next to the **All setup view rights Permission**.

- To assign edit rights to all records from the Designer (Setup) application area, associated pages for the records (such as Security), click **Select** next to the **All setup edit rights Permission**.
- To assign create rights to all records from the Designer (Setup) application area, associated pages for the records (such as Security), click **Select** next to the **All setup create rights Permission**.
- To assign delete rights to all records from the Designer (Setup) application area, associated pages for the records (such as Security), click **Select** next to the **All setup delete rights Permission**.

See [Assigning or Viewing Group System Rights](#) for information about assigning groups of rights to end-users, solution developers, or administrators,.

3. [Click a save option.](#)

#### 2.2.3.6.6 Results from Assigning Global Setup Rights

The following tables list the granular level rights that are automatically assigned when you assign one or more global Setup Rights. In this topic, global Setup Rights refers to Permissions available when you select the **Setup Rights** option from the **View** drop-down list of a group account's **System Rights** page.

The following table displays rights assigned from the group **System Rights** page when the **View** drop-down list **Setup Rights** option is selected and the **All setup view rights** permission displays the status **All Rights Enabled**.

**Rights Assigned with All setup view rights Permission (from global Setup Rights)**

Record Type	Individual Right
Administer Custom Tools	<ul style="list-style-type: none"> <li>• View Administer Custom Tools records</li> </ul>
Global Navigation	<ul style="list-style-type: none"> <li>• View Global Navigation records</li> </ul>
Home Pages	<ul style="list-style-type: none"> <li>• View Home Pages records</li> </ul>
Lookup Tables (System)	<ul style="list-style-type: none"> <li>• View Lookup Tables records</li> </ul>
Custom Lookup Tables	<ul style="list-style-type: none"> <li>• View Custom Lookup Tables records</li> </ul>
Multi-currency	<ul style="list-style-type: none"> <li>• View Multi-currency records</li> </ul>
Miscellaneous Settings	<ul style="list-style-type: none"> <li>• View Miscellaneous Settings records</li> </ul>
Object Definitions	<ul style="list-style-type: none"> <li>• View Object Definitions records</li> <li>• View Object Definitions custom fields</li> </ul>

	<ul style="list-style-type: none"> <li>• View Object Definitions forms</li> <li>• View Object Definitions object views</li> <li>• View Object Definitions blocks</li> <li>• View Object Definitions rules</li> <li>• View Object Definitions rule qualifiers</li> <li>• View Object Definitions search views</li> <li>• View Object Definitions templates</li> <li>• View Object Definitions wizards</li> </ul>
Portal Panes	<ul style="list-style-type: none"> <li>• View Portal Panes records</li> </ul>
Record Locks	<ul style="list-style-type: none"> <li>• Release Edit Locks</li> </ul>
Routes	<ul style="list-style-type: none"> <li>• View Routes records</li> </ul>
Route Stops	<ul style="list-style-type: none"> <li>• View Route Stops records</li> </ul>
Search Results	<ul style="list-style-type: none"> <li>• Bypass Search Limits</li> </ul>
System Settings	<ul style="list-style-type: none"> <li>• View System Settings records</li> </ul>

The following table displays rights assigned from the group **System Rights** page when the **View** drop-down list **Setup Rights** option is selected and the **All setup edit rights** permission displays the status **All Rights Enabled**.

**Rights Assigned with All setup edit rights Permission (from global Setup Rights)**

Record Type	Individual Right
Administer Custom Tools	<ul style="list-style-type: none"> <li>• Edit Administer Custom Tools records</li> </ul>
Global Navigation	<ul style="list-style-type: none"> <li>• Edit Global Navigation records</li> </ul>
Home Pages	<ul style="list-style-type: none"> <li>• Edit Home Pages records</li> </ul>
Lookup Tables (System)	<ul style="list-style-type: none"> <li>• Edit Lookup Tables records</li> </ul>
Custom Lookup Tables	<ul style="list-style-type: none"> <li>• View Custom Lookup Tables records</li> </ul>
Multi-currency	<ul style="list-style-type: none"> <li>• Edit Multi-currency records</li> </ul>

Miscellaneous Settings	<ul style="list-style-type: none"> <li>Edit Miscellaneous Settings records</li> </ul>
Object Definitions	<ul style="list-style-type: none"> <li>Edit Object Definitions records</li> <li>Edit Object Definitions custom fields</li> <li>Edit Object Definitions forms</li> <li>Edit Object Definitions object views</li> <li>Edit Object Definitions blocks</li> <li>Edit Object Definitions rules</li> <li>Edit Object Definitions rule qualifiers</li> <li>Edit Object Definitions search views</li> <li>Edit Object Definitions templates</li> <li>Edit Object Definitions wizards</li> </ul>
Portal Panes	<ul style="list-style-type: none"> <li>Edit Portal Panes records</li> </ul>
Routes	<ul style="list-style-type: none"> <li>Edit Routes records</li> </ul>
Route Stops	<ul style="list-style-type: none"> <li>Edit Route Stops records</li> </ul>
System Settings	<ul style="list-style-type: none"> <li>Edit System Settings records</li> </ul>

The following table displays rights assigned from the group **System Rights** page when the **View** drop-down list **Setup Rights** option is selected and the **All setup create rights** permission displays the status **All Rights Enabled**.

**Rights Assigned with All setup create rights Permission (from global Setup Rights)**

Record Type	Individual Right
Administer Custom Tools	<ul style="list-style-type: none"> <li>Create Administer Custom Tools records</li> </ul>
Global Navigation	<ul style="list-style-type: none"> <li>Create Global Navigation records</li> </ul>
Home Pages	<ul style="list-style-type: none"> <li>Create Home Pages records</li> </ul>
Lookup Tables (System)	<ul style="list-style-type: none"> <li>Create Lookup Tables records</li> </ul>
Custom Lookup Tables	<ul style="list-style-type: none"> <li>Create Custom Lookup Tables records</li> </ul>
Multi-currency	<ul style="list-style-type: none"> <li>Create Multi-currency records</li> </ul>



Miscellaneous Settings	<ul style="list-style-type: none"> <li>• Create Miscellaneous Settings records</li> </ul>
Object Definitions	<ul style="list-style-type: none"> <li>• Edit Object Definitions records</li> <li>• Edit Object Definitions custom fields</li> <li>• Edit Object Definitions forms</li> <li>• Edit Object Definitions object views</li> <li>• Edit Object Definitions blocks</li> <li>• Edit Object Definitions rules</li> <li>• Edit Object Definitions rule qualifiers</li> <li>• Edit Object Definitions search views</li> <li>• Edit Object Definitions templates</li> <li>• Edit Object Definitions wizards</li> </ul>
Portal Panes	<ul style="list-style-type: none"> <li>• Edit Portal Panes records</li> </ul>
Routes	<ul style="list-style-type: none"> <li>• Edit Routes records</li> </ul>
Route Stops	<ul style="list-style-type: none"> <li>• Edit Route Stops records</li> </ul>
System Settings	<ul style="list-style-type: none"> <li>• Edit System Settings records</li> </ul>

The following table displays rights assigned from the group **System Rights** page when the **View** drop-down list **Setup Rights** option is selected and the **All setup delete rights** permission displays the status **All Rights Enabled**.

**Rights Assigned with All setup delete rights Permission (from global Setup Rights)**

Record Type	Individual Right
Administer Custom Tools	<ul style="list-style-type: none"> <li>• Delete Administer Custom Tools records</li> </ul>
Global Navigation	<ul style="list-style-type: none"> <li>• Delete Global Navigation records</li> </ul>
Home Pages	<ul style="list-style-type: none"> <li>• Delete Home Pages records</li> </ul>
Lookup Tables (System)	<ul style="list-style-type: none"> <li>• Delete Lookup Tables records</li> </ul>
Custom Lookup Tables	<ul style="list-style-type: none"> <li>• Delete Custom Lookup Tables records</li> </ul>
Multi-currency	<ul style="list-style-type: none"> <li>• Delete Multi-currency records</li> </ul>

Miscellaneous Settings	<ul style="list-style-type: none"> <li>Delete Miscellaneous Settings records</li> </ul>
Object Definitions	<ul style="list-style-type: none"> <li>Delete Object Definitions records</li> <li>Delete Object Definitions custom fields</li> <li>Delete Object Definitions object views</li> <li>Delete Object Definitions blocks</li> <li>Delete Object Definitions rules</li> <li>Delete Object Definitions rule qualifiers</li> <li>Delete Object Definitions search views</li> <li>Delete Object Definitions templates</li> <li>Delete Object Definitions wizards</li> </ul>
Portal Panes	<ul style="list-style-type: none"> <li>Delete Portal Panes records</li> </ul>
Routes	<ul style="list-style-type: none"> <li>Delete Routes records</li> </ul>
Route Stops	<ul style="list-style-type: none"> <li>Delete Route Stops records</li> </ul>

#### 2.2.3.6.7 Example for Assigning Rights to View Contacts

In order for a group member to access contacts, you must assign minimal view rights. Afterward, the group member will be able to view the Contact list page, the **General** page for contact records, the **Workflow** page for contact records. In addition, the group member will be able to search contacts from the **Advanced Contact Search** page.

#### To assign rights to view contacts

1. [Search for a group and open it.](#)
2. Click **System Rights** from the left pane.
3. Select **Contact** from the **View** drop-down list (right side).
4. Select the **View Contact records** check-box.
5. [Click a save option.](#)

The View Contact records right results in the display of the **Contacts** tab, **Contacts List** page, **Manage Contact Collections** page, **Contact Collections Subscription** page, **Advanced Contact Search** page. From a contact record, the View Contact records right results in the display of the **General** page and **Workflow** page, including read-access to sections and fields on those pages.

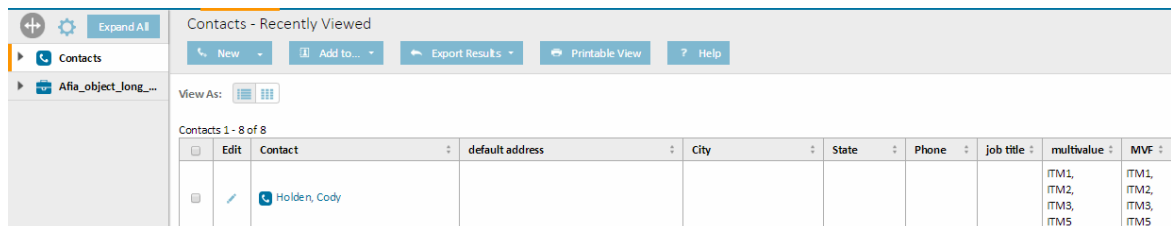
## 2.2.3.6.8 Example for Assigning Rights to Create Contacts

In order for a group member to create contacts, you must assign the basic right, Create Contact records. In addition, this right requires the assignment of View Contact records.

**To assign rights to create contacts**

1. [Search for a group and open it.](#)
2. Click **System Rights** from the left pane.
3. Select **Contact** from the **View** drop-down list (right side).
4. Select the **Create Contact records** check-box.
5. Also select the **View Contact records** check-box.
6. [Click a save option.](#)

The Create Contact records right requires the assignment of the View Contact records right. As a result, a group member with Create Contact records right will have access to all pages that result from the View Contact record right. In addition, the Create Contact record right will result in a **New** button that is available on the Contact List page.



The screenshot shows the 'Contacts - Recently Viewed' page. The top navigation bar includes buttons for 'New', 'Add to...', 'Export Results', 'Printable View', and 'Help'. The left sidebar shows a tree view with 'Contacts' and 'Afia\_object\_Long...'. The main content area displays a table of contacts. The table has columns for 'Edit', 'Contact', 'default address', 'City', 'State', 'Phone', 'job title', 'multivalued', and 'MVF'. A single contact is listed with the name 'Holden, Cody' and five multivalued items (ITM1 through ITM5).

	Edit	Contact	default address	City	State	Phone	job title	multivalued	MVF
<input type="checkbox"/>		Holden, Cody						ITM1, ITM2, ITM3, ITM5	ITM1, ITM2, ITM3, ITM5

**Contacts List Page (New Button)**

## 2.2.3.6.9 Example for Assigning Rights to Edit Contacts

**To assign rights to edit contacts**

1. [Search for a group and open it.](#)
2. Click **System Rights** from the left pane.
3. Select **Contact** from the **View** drop-down list (right side).
4. Select the **Edit Contact records** check-box.
5. Also select the **View Contact records** check-box.
6. [Click a save option.](#)

The Edit Contact records right requires the assignment of the View Contact records right. As a result, a group member with Edit Contact records right will have access to all pages that result from the View Contact record right. In addition, the Edit Contact record right will result in an Edit Record icon that is available on the Contact List page. From contact records, an **Edit** button will be available.

#### 2.2.3.6.10 Example for Assigning Rights to Delete Contacts

##### To assign rights to delete contacts

1. [Search for a group and open it.](#)
2. Click **System Rights** from the left pane.
3. Select **Contact** from the **View** drop-down list (right side).
4. Select the **Delete Contact records** check-box.
5. Also select the **View Contact records** check-box.
6. [Click a save option.](#)

The Delete Contact records right requires the assignment of the View Contact records right. As a result, a group member with Delete Contact records right will have access to all pages that result from the View Contact record right. In addition, the Delete Contact record right will result in a **Delete** button from an open contact's pages.

#### 2.2.3.6.11 Example for Assigning All Rights to Contacts Pages and Fields

For records that contain information that should be commonly shared, you may choose to assign access to all related pages, categories, custom fields, sections, and forms. If you assign rights to work with all categories and all custom fields from the **System Rights** page, those rights will propagate to categories and custom fields that are created in the future.

##### To assign rights to all pages and fields of contacts

1. [Search for a group and open it.](#)
2. Click **System Rights** from the left pane.  
The **System Rights** page will open.
3. Select **Contact** from the **View** drop-down list (right side).
4. Select the check-box to the left of Permission (table heading). Afterward, all rights related to contact records should be selected.

**Note:** *The following rights are selected and give global access to the current Contact categories and custom fields:*

- **View Contact all categories**
- **Create Contact all categories**

- **Delete Contact all categories**

If you assign the **View all categories**, **Create all categories**, or **Delete all categories** rights, the group will automatically have rights to access and work with categories that are created in the future for the record type.

- **View Contact all custom fields**

- **Edit Contact all custom fields**

If you assign the **View all custom fields** or **Edit all custom fields rights**, the group will automatically have rights to access and work with custom fields that are created in the future for the record type.

5. [Click a save option.](#)

After assigning rights above, the following screen views are available to group members:

**Note:** All basic contact pages are available (**General, Workflow**). All additional contact pages are available (**Details, Categories, Rates, Relations, Skills, Territories, Security**).

#### 2.2.3.6.12 Examples for Assigning Projects Rights

The custom object, Dispute, will be used to show how rights appear for custom objects (projects). The following screens display rights for Dispute records. An Involved child object was added to the Dispute object.

**Note:** Dispute is used as an example and it is automatically included with TeamConnect Legal Matter Management but the Dispute object is not included with TeamConnect Enterprise.

Group - Paralegal

Save Cancel & View Create a Copy Help Add Bookmark

**System Rights**

System Rights 1 - 140 of 140

<input type="checkbox"/>	Permission
<input checked="" type="checkbox"/>	View Dispute records
<input checked="" type="checkbox"/>	Edit Dispute records
<input checked="" type="checkbox"/>	Create Dispute records
<input checked="" type="checkbox"/>	Delete Dispute records
<input type="checkbox"/>	Mass edit Dispute records
<input type="checkbox"/>	View printable view for Dispute records
<input type="checkbox"/>	View export results for Dispute records
<input checked="" type="checkbox"/>	Create a copy
<input checked="" type="checkbox"/>	View Dispute assignees
<input checked="" type="checkbox"/>	Edit Dispute assignees
<input checked="" type="checkbox"/>	Create Dispute assignees
<input checked="" type="checkbox"/>	Delete Dispute assignees
<input checked="" type="checkbox"/>	View Dispute all categories
<input checked="" type="checkbox"/>	Create Dispute all categories

**Group System Rights for Dispute (Custom Object)**

**Note:** Rights for the child Involved object include Permission list items, View Involved Party Records through Edit Involved Party Security.

The **Create a Copy** right is unique to custom objects. If it is assigned the resulting application page will display the **Copy** button from an open dispute record.

#### 2.2.3.6.12.1 Assigning All Rights to Dispute Pages and Fields

The following section describes how to assign access to all dispute record related pages, categories, custom fields, sections, and forms.

**Note:** Dispute is used as an example and it is automatically included with TeamConnect Legal Matter Management but the dispute object is not included with TeamConnect.

#### To assign all rights to dispute record pages and fields

1. [Search for a group and open it.](#)
2. Click **System Rights** from the left pane.  
The **System Rights** page will open.
3. Select **Dispute** from the **View** drop-down list (right side).
4. Select the check-box to the left of Permission (table heading). Afterward, all rights related to dispute records should be selected.
5. [Click a save option.](#)

After assigning rights above, the respective screen views are available to group members.

**Note:** The links in the left pane above represent record pages that the group member can access.

### 2.2.3.7 Assigning Group Category and Custom Field Rights

From a group's **Category Rights** page, you can assign rights to specific categories and custom fields per record type. Additionally, you can assign high level rights to all categories and custom fields across all record types. For more details on how field-level security works for TeamConnect 4.2+, see [Rights for Categories and Custom Fields](#).

#### To assign category and custom field rights

1. [Search for a group and open it.](#)
2. To change rights for categories and custom fields, from the left navigation pane, click **Category Rights**. See the appendix, [Rights for End-users](#) for additional information about general rights for categories and custom fields for each record type.
3. To assign global category or custom field rights for all record types, do the following:
  - From the **View** drop-down list, select **User Rights**.
  - From the resulting rights list, click the **Select** button for the following permission rows:
    - **View all categories**—Allows access to the **Categories** pages of all records.
    - **Add all categories**—Allows addition or selection of all available categories from the **Categories** pages of all records.
    - **Remove all categories**—Allows removal or clearing of all categories from the **Categories** pages of all records.
    - **View all custom fields**—Allows access to custom fields associated with any categories from the **Details** pages of all records.
    - **Edit all custom fields**—Allows data entry or changes for custom fields values associated with any categories from the **Details** pages of all records.

Afterward, all rights for categories and custom fields of all record types will be assigned to the group.


4. To assign specific category or custom field rights for a specific record type, do the following:
  - From the **View** drop-down list, select a record type. For example, Contacts.
  - To assign the following base category rights, select the check-box next to each right:
    - **View Contact category**—Allows access to the **Categories** page of the contact record.
    - **Add Contact category**—Allows addition or selection of all available categories from the **Categories** page of the contact record.
    - **Remove Contact category**—Allows removal or clearing of all categories from the **Categories** page of the contact record.
    - **View Contact custom fields**—Allows access to custom fields associated with any contact categories from the **Details** page of the contact record.
    - **Edit Contact custom fields**—Allows data entry or changes for custom fields values associated with any contact categories from the **Details** page of the contact record.
  - To assign additional category or custom field rights, select the respective check-box.
  - **Note:** Further detail on this functionality is highlighted in [Hierarchy Overview of Group Rights](#).
5. [Click a save option](#).

#### Group Category Rights Page Field Descriptions

Label	Description
<b>View drop-down list</b>	<p>A list of record types for which you can assign rights to work with associated categories and custom fields.</p> <p>After selecting a list option, the workspace is updated with a list of rights specific to categories and custom fields for that option. For example, if you select Contacts, the rights list displays only rights for working with contact categories and custom fields.</p>
<b>Permission</b>	<p>A general list of rights follows. For more information, see <a href="#">Understanding Rights</a> and <a href="#">Rights for Categories and Custom Fields</a>.</p> <ul style="list-style-type: none"> <li>• <b>View category</b>—To grant rights to view categories for the selected record type and rights to set a category as default, select the <b>View category</b> check-box.</li> <li>• <b>Add category</b>—To grant rights to add a category to the selected record type, select the <b>Add category</b> check-box.</li> <li>• <b>Remove category</b>—To grant rights to remove a category from the selected record type, select the <b>Remove category</b> check-box.</li> <li>• <b>View all custom fields</b>—To grant rights to see custom fields associated with the selected record type, select the <b>View all custom fields</b> check-</li> </ul>



box. Additional requirements include: Rights to read the associated category and its parent categories; and rights to read associated custom objects.

- **Edit all custom fields**—To grant rights to edit custom field values associated with the selected record type and categories, select the **Edit all custom fields** check-box. Additional requirements include: Rights to view the custom fields on that category.
- **Individual Fields**—To manage view and edit rights for custom fields on a given object. Click the dropdown arrow on the respective row () to open the view and edit options.

**Important:** Adjustments made to the category rights at-large will affect custom field rights. For example, if the **View all custom fields** check-box is selected, all custom field **View** check-boxes will become read-only. If the individual fields were not selected prior to clicking the parent category **View All** button, they will appear to be read-only and unchecked. Despite this, view rights will be on for all custom fields in the category. Likewise, the same conditions apply for the **Edit all custom fields** check-boxes and the respective custom fields on a category. Similar to how the category rights function, **Edit** rights will not be available for any custom field that does not have **View** rights enabled.

#### 2.2.3.7.1 Example for Assigning Rights to Contact Categories and Fields


There may be a group that you'd like to assign access to work with some custom fields but not all custom fields of a record type. It is recommended to assign rights to view and add the related category. Afterward, you can assign rights to one or more custom fields associated with that category.

The following example will serve to demonstrate how to add **View** rights to all of the custom fields under the External->Attorney category. **Edit** rights will also be activated for two of the three custom fields ("Minority" and "Overall Performance").

For more details on how field-level security works for TeamConnect 4.2+, see [Hierarchy Overview of Group Rights](#).

#### To assign group rights to specific custom fields

1. [Search for a group and open it.](#)
2. Click **System Rights** from the left pane.
3. Select **Contact** from the **View** drop-down list (right side).
4. Select the **View Contact records** check-box.
5. Select the **Edit Contact records** check-box.

6. Save.
7. Click **Category Rights** from the left pane.
8. Select **Contact** from the **View** drop-down list (right side).
9. Click the **Edit** button.
10. Enable **View Category** for External > Attorney.
11. Click the **Individual Fields** drop-down arrow () for External > Attorney.
12. Click the parent **View** check-box for the **Fields** under External->Attorney.
13. Click the Minority and Overall Performance check-boxes underneath **Edit** for **Fields**.
14. [Click a save option.](#)

After assigning rights above to a group, the following screen views can be selected under the **Categories** subsection of the **Contact** profile. Make sure to **Save** after making changes to the contact record.

**Categories**







Check the categories you wish to select for this record. To set a primary category, click the button next to the category name.






Selecting a category may reveal additional fields for this record. After adding and removing categories, you may want to [refresh this page](#) or click the links on the left to review any changes.

- Contact
- External
  - Agency
  - Arbitrator
  - Attorney
    - Managing Partner
    - Partner
    - Sr Associate
    - Associate

**Assigning Category to Contact**

After assigning the following category, the following custom fields are visible via the **Custom Fields** link on the sidebar navigation:

 Edit
 Create a Copy
 Delete
 Share
 Printable View
 Help

-  Profile
-  Notes & Docs
-  Rates
-  Involvement
-  Relations
-  Custom Fields

**Contact Details** [Edit](#)

**Search Tags:**

**Incorporation Locale:**

**Attorney Details** [Edit](#)

**Woman: Yes**

**Minority: No**

**Overall Performance: 1.00**

Version 4.2.0 Build 67

Copyright © 1999-2016 Mitrtech Holdings, Inc. All Rights Reserved.

**Note:** If rights are turned off on system blocks, those blocks will not be visible. Rights turned off on custom blocks will appear as SECURED.

### 2.2.3.8 Hierarchy Overview of Group Rights

Improved granular group rights (as of TeamConnect 4.2) allows for more robust control over user privileges. In setting up these rights, it is important to understand which settings take precedence. Some sections, as detailed below, will change the read-only/read-write status of others. An example of setting up rights can be found on the [Example for Assigning Rights to Contact Categories and Fields](#).

### Hierarchy Overview of Rights

The following sections describe how group rights can be configured in TeamConnect 4.2+.

It should be noted that the far left checkbox on the top row will give all view and edit rights on all object categories and fields. When this box is checked, all subsidiary custom fields will be unable to be checked, but will retain view and edit permissions.



Chart Number	Section Type	Details
1	Full Parent-level rights management	<p>These checkboxes will enable or disable all items within the column.</p> <p>Users should note that:</p> <ol style="list-style-type: none"> <li>Edit rights cannot be enabled for any category lacking the respective View rights.</li> <li>Category rights take precedence over custom field-level rights.</li> </ol>
2	Individual Category Item Rights	<p>Each of these checkboxes enables or disables the respective right for that category and column (e.g., selected example provides <b>View</b> rights for <b>External</b> category on Contact object).</p> <p>Selecting the <b>View all custom fields</b> or <b>Edit all custom fields</b> box will "gray-out" the boxes for custom fields (see Section 4). If you have not individually checked the custom field-level boxes prior to the View/Edit all custom fields box, they will be grayed out as empty boxes. <b>However, you will still have View or Edit rights if the Section 2 View all custom fields/Edit all custom fields boxes are checked, regardless of whether the</b></p>

Chart Number	Section Type	Details
		<b>Section 4 boxes have check-marks.</b> For more detail, see Section 4 of this table, or the Addendum at the end of this document.
3	Full Category-level rights management	Clicking any box in this column will apply full rights on the category to the group. As this will enable the <b>View all custom fields/Edit all custom fields</b> boxes, the Section 4 custom field-level rights boxes will be grayed out. If you have not individually checked the custom field-level boxes prior to the View/Edit all custom fields box, they will be grayed out as empty boxes. <b>However, you will still have View or Edit rights if the Section 2 View all custom fields/Edit all custom fields boxes are checked, regardless of whether the Section 4 boxes have check-marks.</b>
4	Custom Field-level rights management	Click the green caret symbol to view the custom fields for the respective category. If the <b>View all custom fields/Edit all custom fields</b> boxes are checked for the row, these custom field-level boxes will be grayed out. In this case, even if they are grayed out without checks present.

### 2.2.3.9 Assigning Group Tool Rights

From a group's **Tool Rights** page, you can assign rights to tools. Afterward, the group members will be able to view and work with the tools.

**Note:** Some tools are available from the User application area and other tools are available from the Designer application area.

#### To assign group tool rights

1. [Search for a group and open it.](#)
2. To make tools available, from the left navigation pane, click **Tool Rights**.
  - To enable a tool for the group, select the tool's check-box.
  - To disable a tool, clear the tool's check-box.

For more information, see [the Groups Tool Rights fields table](#).

3. [Click a save option](#).

#### Groups Tool Rights Field Descriptions

Label	Description
<b>Permission</b>	<p>Select the check-box for each tool to enable:</p> <ul style="list-style-type: none"> <li>• <b>Contact Sweeper</b>—Administrator's tool to perform a duplicate contacts check across the system. Recommend restricting access to this tool to system administrators.</li> <li>• <b>Design Snapshot Tool</b>—Solution developer's tool to capture your latest design snapshot, provide a summary of archived design snapshots, generate design upgrade files. After the right is assigned, access this tool from the Designer area.</li> <li>• <b>Design Import Tool</b>—Solution developer's tool to apply a design upgrade file to upgrade a QA or production TeamConnect instance from an older design version to a newer design version. After the right is assigned, access this tool from the Designer area.</li> <li>• <b>Reassignment Tool</b>—Reassigns multiple appointments, projects, and tasks.</li> <li>• <b>Exchange Rates</b>—Administrator's tool to update current exchange rates. For details see <a href="#">Updating Exchange Rates</a>.</li> <li>• <b>Invoice Validation Rule Setting</b>—Solution developer's tool to activate or deactivate invoice validation rules. After the right is assigned, access this tool from the Designer area. Members of the CSM Admin group have default access rights to the Invoice Validation Rule Setting tool.</li> <li>• <b>Global Search Index Tool</b>—The Global Search Index Tool is utilized to update Global Search after objects have been enabled or disabled and clear search server information.</li> <li>• <b>Map Line Item Codes</b>—Solution developer's tool to map invoice line item codes (fee/task codes, expense codes, or activity codes). This tool only appears on the Tool Rights page if it has been loaded in TeamConnect.</li> <li>• <b>Expense Unit Prices</b>—Administrator's tool. For details, see <a href="#">Updating Expense Unit Prices</a>.</li> <li>• <b>Data Import Wizard</b>—Administrator's tool to import data from an Excel spreadsheet into TeamConnect fields. For details, see <a href="#">Importing Data to Create Records</a>.</li> <li>• <b>Screen Designer</b>—Solution developer's tool to build a screen block from a WYSIWYG layout. For details, see Using Screen Designer.</li> </ul>

- **Locale Manager**—Solution developer's tool to control which locales are available to use as system default, or to choose as an end user's preference.
- (TeamConnect Legal Matter Management and TeamConnect® Express) **Create Next Fiscal Budget**—Business manager's tool to define the fiscal year for matter budgets.
- (TeamConnect Legal Matter Management and TeamConnect® Express) **Legal Preferences Tool**—Business manager's tool to set preferences for budget definition. For example, defining the budget year start, setting up budget detail levels.
- (TeamConnect Legal Matter Management only) **Time Entry Tool**—Business manager's tool to manage timekeepers' entries.

*Note: Additional tools may be available depending on whether custom tools have been added to your system.*

#### 2.2.3.10 Changing Group Default Object Views

From a group's **Object Views** page, you can change the default set of [object views](#) for the group. If a solution developer has designed custom object views (created custom pages, sections, fields, and/or forms for an object), you can select a custom object view.

Object views can also be assigned at the system level and this is performed from the Designer area of TeamConnect. Object views assigned at the group level override system level object views.

Users that belong to multiple groups will see the object views assigned to their default group as specified in their user account. [Assigning different views of an object](#) to different groups of users is one way of controlling access to object record information. No matter which categories or custom fields individual members of a user group can access, they cannot see them if these categories or custom fields are not included in the object view assigned to their default group.

##### To change default object views for a group

1. [Search for a group and open it.](#)
2. Click **Object Views** from the left pane.
3. (optional) For an object (record type) click the corresponding **Object View** drop-down list and select a different object view.
4. [Click a save option.](#)

*Note: The selected settings take effect immediately, and users who do not have a default group set in their user accounts see the object views assigned to the system.*

### 2.2.3.11 Viewing or Creating Group History

From a group's **History** page, you can create a history (containing notes about changes to the group), view a list of related history records, or perform an advanced search for a history.

#### To create a history for a group

1. [Search for a group and open it.](#)
2. Click **History** from the left pane.  
The group's History list page will open. You can click a history **Description** to view its details.
3. Click **New History**.
4. Enter a change **Description**.
5. [Click a save option.](#)

### 2.2.3.12 Deleting a Group

When you delete a group, the users that are group members will not be deleted but they will lose security rights that were granted through the group.

#### Scenarios that Prevent Deletion

When you try to delete a group, the following scenarios prevent the deletion:

- If any record's **Security** page grants rights to the group (record-level security).
- If any User accounts specify the group as their Primary group.
- If the group is assigned to the Process Manager role of any active workflow rule.
- If the group is linked to any active workflow request (even if the group is not assigned to a Process Manager role).

#### To delete a group

1. Select the **Admin** tab.
2. Click the **Groups** sub-tab link.
3. (optional) From the left navigation pane, select a Group view.
4. Click the name of the group to delete.  
The group record opens.
5. Click **Delete**.

#### What is Deleted

When you delete a group, the following associated data gets deleted:

- All rights assigned in the **System Rights**, **Category Rights**, and **Tool Rights** pages of the group account.
- Member list (not the user accounts).
- Default object view selections.
- Approval route stop members where this group was selected.
- Home Page copies created for the members of this group.
- Associated History records.

## 2.2.4 Users

You need to create a user account for anyone who needs to log into TeamConnect. Each user account also requires an associated contact record. Some contacts, such as external vendors, will be created in TeamConnect but will not have corresponding user accounts.

Generally, user accounts will not be used to assign security rights unless the user was migrated from TeamConnect 2.x. For new user accounts, you can view

For more information on setting up rights, see [Assigning or Viewing Group System Rights](#).

### User Record Pages

The following are user-specific pages:

- The **General** page lets you enter information about the user, such as User Name and Password. You can also manage a user's account from the **General** page, for example, activate or inactive an account and set an account expiration date.
- The **Group Memberships** page lets you add a user to a group or remove a user from a group.
- The **System Rights** page lets you assign or view rights to work with records, application areas, and application features.
- The **Category Rights** page lets you assign or view rights to categories and custom fields by record type.
- The **Tool Rights** page lets you assign or view rights to system tools and custom tools.
- The **History** page is also associated with a user. The **History** page lets you add notes about changes to the user or review previous history notes.

#### 2.2.4.1 Working with Users

You can perform the following tasks when you are working with users.

### Users List

- [Searching for Users](#)—Search for a user to view or edit.
- [Viewing Users](#)—View a list of users or open an individual user.
- [Creating or Editing Users](#)—Add a user or edit a user's information.



- [Deleting a User](#)—Delete a user.

## User General Page

- [Creating or Editing Users](#)—Add a user or edit a user's information.
- [Activating and Inactivating a User](#)—Activate or inactivate a user.
- [Prompting Users to Change Their Passwords](#)—Set a prompt for a users to change their password on application login.
- [Editing or Viewing a User's Preferences](#)—Edit or view a user's preferences.
- [Unlocking User Accounts](#)—Unlock user accounts that have been locked due to unauthorized access attempts or incorrect passwords.

## User Group Memberships Page

- [Changing a User's Group Membership](#)—Add a user to a group or remove a user from a group.

## User System Rights Page

- [Assigning or Viewing User System Rights](#)—Assign or view rights to work with records, application areas, and application features.

## User Category Rights Page

- [Assigning or Viewing User Category Rights](#)—Assign category and custom field rights to a user.

## User Tool Rights Page

- [Assigning or Viewing User Tool Rights](#)—Assign rights to work with system and custom tools.

## User History Page

- [Viewing or Creating User History](#)—Create or view notes about changes to a user.

### 2.2.4.2 Searching for Users

From the **Users - Search** page, search for users using multiple field values as criteria. For example, you can search for users by **Name**, **User Type**, and associated **Contact**.

#### To perform an advanced search for a user


1. Click the **Admin** tab.
2. Click the **Users** link under the tab bar.
3. Click the **Custom Search** link.

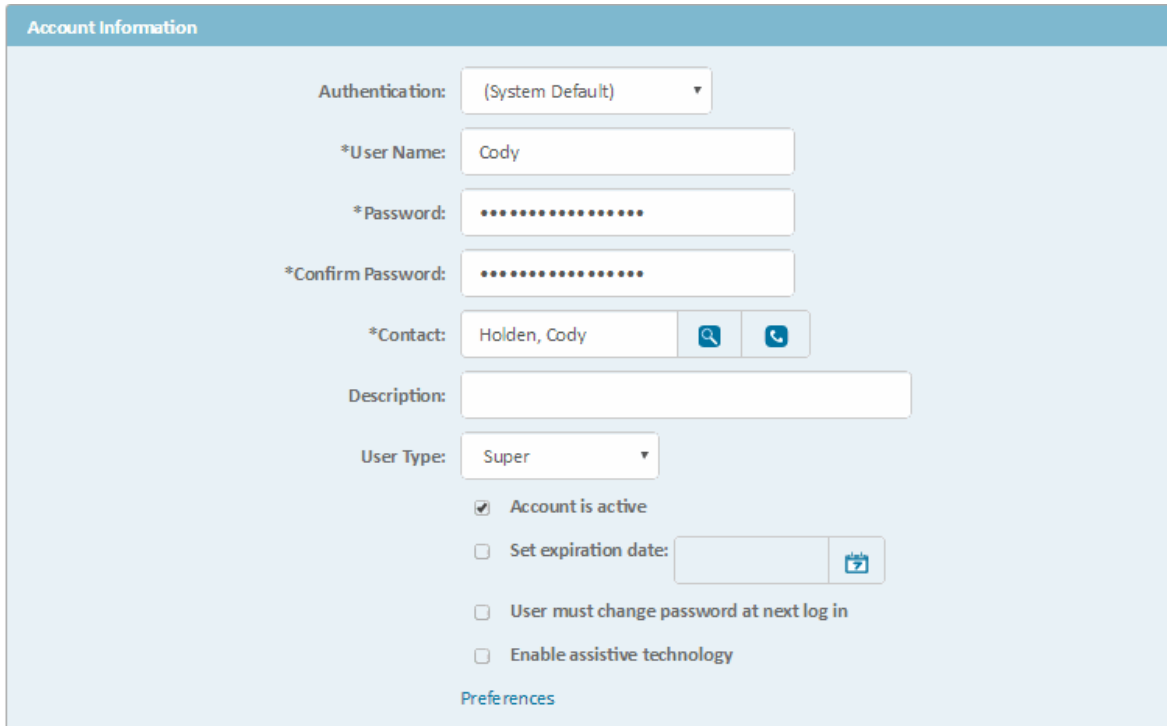
The **Users - Search** page opens.

4. (optional) To limit the fields used to search, in the **Field** drop-down list of the **Filter Criteria** section, select an appropriate option.
5. Enter search criteria in the resulting fields. See [User General Page Field Descriptions table](#) for descriptions of search criteria fields.
6. Click **Search**.
7. To open a user account, from the right pane, click a user's associated contact name link.



### 2.2.4.3 Viewing Users

You can view users by browsing user collections. Alternatively, you can search for a user (see [Searching for Users](#)).

From a user record's **General** page, you must search for a contact to associate with the user. Afterward in edit-mode, a contact icon  will display, allowing you to click through to the associated contact record.



The screenshot displays the 'Account Information' section of a user record's general page. The fields and their values are as follows:

- Authentication:** (System Default) [dropdown arrow]
- \*User Name:** Cody
- \*Password:** [masked with dots]
- \*Confirm Password:** [masked with dots]
- \*Contact:** Holden, Cody [text input]  
- Description:** [empty text input]
- User Type:** Super [dropdown arrow]
- Account is active
- Set expiration date: [calendar icon]
- User must change password at next log in
- Enable assistive technology

At the bottom of the form, there is a link labeled 'Preferences'.

User Record - General Page

#### To browse a list of users by view

1. Click the **Admin** tab.
2. Click the **Users** link under the tab bar.
3. (optional) From the left pane, select a User collection:

- **Recently Viewed**—Displays the last 50 users that you opened. You cannot delete this collection link, change its order. It is always the first collection listed in the left pane.
  - **Super Users**—Displays users whose **User Type** is **Super**, which typically means they have more security rights than Normal users.
  - **Limited-Privilege Users**—Displays users whose **User Type** is **Limited-Privilege**.
  - All Users
  - **Normal Users**—Displays users whose **User Type** is **Normal**.
  - **Inactive Users**—Displays users with inactive accounts.
  - **Active Users**—Displays users with active accounts.
  - **New Users**—Displays users that have been created within the current session. The results of this collection are sorted alphabetically.
4. To open a user account, from the right pane, click a user's associated contact name link.

#### 2.2.4.4 Creating or Editing Users

Before creating a user, search existing users to make sure you do not create a duplicate. Users intending to create user accounts must have **View** and **Edit** rights for the Contact, User, and Preferences object to successfully complete the task.

***Note:** If you are editing an existing user in a way that affects their access rights to records or other features, and that user is presently logged in, some of your changes might not take effect until that user ends their session and then logs back in.*


To create a new user, you enter information similar to the following image.

The screenshot shows the 'Account Information' form for creating a new user. The form includes the following fields and options:

- Authentication:** (System Default) [dropdown]
- \*User Name:** CMHolden [text input]
- \*Password:** [password input]
- \*Confirm Password:** [password input]
- \*Contact:** Holden, Cody [text input with search icon]
- Description:** [text input]
- User Type:** Normal [dropdown]
- Account is active
- Set expiration date: [calendar icon]
- User must change password at next log in
- Enable assistive technology

#### Creating a New User

#### To create or edit a user

1. Click the **Admin** tab.
2. Click the **Users** link under the tab bar.
3. Click the **New** button.
4. Select an **Authentication Type**.
5. Enter a **User Name**. Maximum length: 50 characters. User names cannot contain spaces or more than one @ character.
6. Enter the user's **Password**. Maximum length: 250 characters. Restrictions are based on the **Password Parameters** on the **Security** page of **Admin Settings**.
7. Retype or copy the password into the **Confirm Password** field.
8. Click the  **Search Contacts** icon and search for a contact to associate with the user. Create a contact record for the user if one does not exist.
9. Select a **User Type** from the drop-down list.
  - **Super**—Can access all public and private records, as long as the user has group rights to those records.
  - **Normal**—Can access all public records. Can access private records only if record-level security is granted to the user.

- **Limited-privilege**—Can access only records created by the user and records to which they have been granted explicit access by another user. Limited-privilege users will only see records to which they are assigned unless they are specifically assigned rights at the record level.

If you need to allow limited-privilege users to perform tasks not normally available to them, such as the ability to approve an invoice, there must be a rule written that gives them record level rights to the invoice so that they can see and approve it.

10. Select the **Account is active** check-box.

11. Click a save option.

**Note:** If your licensing scheme specifies a limit on the number of named active users, and the user you are saving is active and would exceed the limit, you will see an error message. You can retain the user's information by clearing the **Account is Active** check-box, then trying the save again. For details about licensing schemes, see [Viewing Information About Your Product License](#).

**Note:** Newly added users will be temporarily unable to see data on reports that are run from the Reports tab. This is because Data Warehouse does not yet have the new user's information and cannot determine their security rights. This situation will resolve itself when the next Data Warehouse refresh job runs.

#### User General Page Field Descriptions

Label	Description
<b>User Name</b>	(required) Type a name. Maximum length: 50 characters.
<b>Password</b>	(required) Type the password. Maximum length: 250 characters.
<b>Confirm Password</b>	(required) Retype or copy the password.
<b>Contact</b>	(required) Search for the contact (person or company) for which you are creating the account. If the contact does not yet exist, you can add a new contact from the <b>Contacts Search</b> window of the Search Module.
<b>Description</b>	Enter a brief description for the user, if necessary. Maximum length: 250 characters.
<b>User Type</b>	Select one of the following: <ul style="list-style-type: none"> <li>• <b>Super</b>—Can access all public and private records. This setting overrides access restrictions based on record-level security settings. Recommended for system administrators and solution developers.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Normal</b>—Can access all public records. Can access private records only if created by the user or if record-level security is granted to the user. Recommended for internal staff.</li> <li>• <b>Limited-Privilege</b>—Can access only records created by the user and records where record-level security has been explicitly granted to the user. Recommended for Outside Counsel.</li> </ul> <p>If you need to allow limited-privilege users to perform tasks not normally available to them, such as the ability to approve an invoice, there must be a rule written that gives them record level rights to the invoice so that they can see and approve it.</p> <p><b>Note:</b> Access to a record automatically grants access to its embedded records but not to its related records. For example, if a limited-privilege user has the right to access a dispute record, they cannot automatically access the dispute's related involved parties, tasks, histories, or expenses, but they can access its allegation (embedded object) records.</p>
<b>Authenticati on Type</b>	<p>Select the authentication mechanism from the drop-down list to use when authenticating a user during login. This setting determines which TeamConnect login URL to give a user. Select one of the following:</p> <ul style="list-style-type: none"> <li>• <b>System Default</b>—This option maps to the default authentication mechanism selected in the Admin Settings. See <a href="#">Authentication</a> for more information.</li> <li>• <b>Standard Authenticator</b>—If this user does not require custom authentication, select this option.</li> <li>• <b>Additional options</b>—Authentication options for installed plug-ins can also be selected. See <a href="#">Authentication</a> for more information.</li> </ul> <p>If you installed a 3rd party user authentication plug-in (for example, LDAP), this option will display. If this user should use custom authentication, select the plug-in option. See <a href="#">Authentication</a> for more information.</p> <p><b>Note:</b> For the TeamConnectAdmin and System user accounts, the only option is Standard TeamConnect and cannot be changed.</p> <p><b>Note:</b> If a custom authentication type, such as NTLM authentication, is selected in the Authentication drop-down list, the <b>Password</b> field will not display. In addition, if the system's default authentication is set to a custom authentication type, such as NTLM, and the user's Authentication is set to (System Default), the <b>Password</b> field will not display.</p>

<b>Account is active</b>	<p>Selecting this check-box enables the account. If you remove the checkmark to disable a user account, the record remains in the system. Recommended for staff on vacation or on leave of absence.</p> <p>Users with disabled user accounts:</p> <ul style="list-style-type: none"> <li>• Cannot log into TeamConnect.</li> <li>• Do not display in the list of available users in drop-down list and Search Module fields (for example, for assignees, attendees).</li> <li>• Do not receive any approval requests if they are stop members in other routes.</li> </ul>
<b>Set expiration date</b>	<p>To disable the account on a specific date, select this check-box and enter a date. Recommended for temporary employees, as they are not be able to log into TeamConnect after the expiration date.</p>
<b>User must change password at next log in</b>	<p>Select this check-box to prompt the user to change his or her password on the next application login.</p> <p><i><b>Note:</b> Make sure that the date range between the current date and date set in this field is larger than the <b>System Settings, Password Policy</b> page's <b>Maximum Age</b> field value. Otherwise the user will not be able to change his/her password.</i></p> <p>See <a href="#">Defining the User Password Policy</a>.</p>
<b>Enable assistive technology</b>	<p>Select this check-box to <a href="#">enable Accessible Mode</a> for the vision impaired. This setting currently applies to the Designer application that is available if the user has rights to any rights for solution developers. Changes take effect the next time the user logs into the system.</p> <p>For more information on Accessibility in TeamConnect, see the User Guide.</p>

#### 2.2.4.4.1 Creating a User Using a Template

You can use an existing user like a template. You can create a copy of a user record and then edit some fields to create a new user.

Field values from the following original user's pages are copied to the new record:

- **General** page
- **Group Memberships** page
- **System Rights** page
- **Category Rights** page
- **Tool Rights** page

**To create a user using a template**

1. Search for a user to copy, and open it.
2. Click **Create a Copy**.
3. Edit the user as required. For example, change the **User Name** and **Password**.
4. [Click a save option](#).

Field values from the original user's **History** page are not copied to the new user.

**2.2.4.4.2 Example for Creating a Limited-Privilege User**

There may be individuals for whom you want to create user accounts but only with restricted access to a select number or type of records. For example, if your company has external vendors, you may want to provide the vendors with access to the specific project records with which they are involved.

Rights to work with either specific record types or all record types are set at the group level. At the user level, there is an additional setting (User type) that determines whether the user has access to all public records or limited access to records.

***Note:** If you need to allow limited-privilege users to perform tasks not normally available to them, such as the ability to approve an invoice, there must be a rule written that gives them record level rights to the invoice so that they can see it and approve it.*

**To create a limited-privilege user**

1. Create users. See [Creating or Editing Users](#).
2. For each user, from the **General** page, **User** type drop-down list, select **Limited-Privilege**.
3. Open the project record to grant access rights to (for this group).
4. Click **Security** from the left pane of the project.
5. To assign the user rights to search, view, or open the project, from the **User Rights** section, click **Edit**.
6. Click **Add User Rights**.
7. From the **Allow** drop-down list, select **Allow**.
8. From the **Users** drop-down list, select the user.
9. [Click a save option](#).

**2.2.4.5 Activating and Inactivating a User**

From a user's **General** page, you can activate or inactivate the user's account.

**To activate or inactivate a user**



1. [Search for a user and open it.](#)
2. To activate a user, select the **Account is active** check-box.
3. To inactivate a user, clear the **Account is active** check-box.
4. To set a future date when the user's account will be disabled, select the **Set expiration date** check-box. Click the Calendar icon to select a date.
5. [Click a save option.](#)

#### 2.2.4.6 Prompting Users to Change Their Passwords

From a user's **General** page, you can enable a prompt that requires the user to change their password to enforce security standards, for example, if you assign a generic password to new users.

##### To prompt users to change their passwords

1. [Search for a user and open the user's account.](#)
2. Select the **User must change password at next log in** check-box to prompt the user to change his or her password on the next TeamConnect login.

Make sure that the time span between the current date and date set in this field is larger than the **Maximum Age** field value on the **Security** page of the **Admin Settings**. Otherwise the user will not be able to change his/her password. See [Defining the User Password Policy](#).

3. [Click a save option.](#)

#### 2.2.4.7 Editing or Viewing a User's Preferences

A user can set individual application preferences. In addition, an administrator with appropriate rights to user records can view or edit a user's preferences from the user record's **General** page.

##### To edit or view a user's Preferences

1. [Search for a user and open the user's account.](#)
2. From the **General** page, click **Preferences**.

The user's **Preferences** page opens.

For information about field descriptions, from the User Guide or help, search for [Personalizing Your Settings](#).

#### 2.2.4.8 Unlocking User Accounts

To guard against unauthorized access, you can limit the number of failed login attempts on the **Password Policy** page of the Admin Settings area. For details, see [Security Settings](#).

When the **Enable User Lockout** check-box is selected, users have a specific number of attempts within a time interval to successfully enter a password for their user name. If users exhaust their number of attempts, TeamConnect locks the user account, preventing login attempts.

Whether user accounts are locked due to unauthorized individuals attempting to access the system or legitimate users who enter incorrect passwords, you can unlock user accounts if you have Update rights to user accounts.

#### To unlock a user account

1. [Search for a user and open the user's account.](#)
2. Clear the **Account is Locked** check-box.

This check-box only appears on the **General** tab when a user account is locked.

3. [Click a save option.](#)

The user account is unlocked and user access is restored. The **Account is Locked** check-box is removed from the **General** tab of the user account.

When an account is locked, TeamConnect does not attempt to assign a new password. If desired, you can assign one (for example, if a user cannot remember the current password) or prompt the user to change the password on his or her next login. See [Creating or Editing Users](#).

#### 2.2.4.9 Changing a User's Group Membership

From the **Group Membership** page, add users to groups or remove users from groups.

##### To change a user's group membership

1. [Search for a user and open the user's account.](#)
2. Click the **Group Memberships** link from the left pane.
3. To add the user to a group:
  - a. (optional) Click **Add New Item**.
  - b. From the corresponding **Group** drop-down list, select the group to add membership.
  - c. (optional) To apply the group's object views as this user's default, select the corresponding **Primary** button.
4. To remove the user from a group:
  - a. Select the check-box to the left of the group to remove membership.
  - b. Click **Remove**.
5. [Click a save option.](#)

##### Users Group Membership Page Field Descriptions

Label	Description
-------	-------------

<b>Group Membership Row</b>	<p>For each group, the following fields apply:</p> <ul style="list-style-type: none"> <li>• <b>Line</b>—Read-only. Displays row number.</li> <li>• <b>Group</b>—Select a group from the drop-down list to add or remove the current user to/from the member list.</li> <li>• <b>Primary</b>—Select one group's Primary button for the user. The user's default object view is determined by the primary group assigned. If the user belongs to multiple groups, the first group will be selected as Primary by default.</li> <li>• <b>Action</b>—Click the icon to edit an existing user's account.</li> </ul>
<b>Remove</b>	To remove a user from the current group, select the check-box to the left of the user's row and click the <b>Remove</b> button.
<b>Add Another Item</b>	To add a blank Group Membership row, click the <b>Add Another Item</b> button and then select a group (see the <b>Group Membership row</b> description above).

#### 2.2.4.10 Assigning or Viewing User System Rights

From a user account's **System Rights** page, you can do the following:

- (For users created in TeamConnect version 2.x or earlier) Assign and view rights to records, application areas, and application features at the user account level.

In addition, view rights to records, application areas, and application features inherited from the users' member group(s).

- (For users created in TeamConnect 3.x) View rights to records, application areas, and application features inherited from the users' member group(s). For these users, you cannot assign or manage record rights at the user level. You must edit a member group's record rights. For more information, see [Assigning or Viewing Group System Rights](#).

(For users created in TeamConnect 2.5 SP4 or earlier) You can assign rights that apply:

- **Globally**—If you expect to enable rights across multiple record types or multiple application areas, work with these types of rights to save time. Global rights are organized into three groups: User rights, Admin rights, and Setup rights.

For example, if you select the View drop-down list, User Rights option then the resulting page you can assign rights to view all contact records, view all document records, view all project records, etc., with one button click.

- **Individually**—If you expect to enable rights for a limited number of record types (for example, allow access to a couple record types), work with these types of rights.

For example, if you select the View drop-down list, Contacts option then from the resulting page you can assign more granular level rights such as View Contact records, Edit Contact records, Create Contact records.

Holden, Cody

Save Cancel & View Create a Copy Help Add Bookmark

### System Rights

System Rights 1 - 16 of 16

Group	Permission
<input checked="" type="checkbox"/>	View Account records
<input checked="" type="checkbox"/>	Edit Account records
<input checked="" type="checkbox"/>	Create Account records
<input checked="" type="checkbox"/>	Delete Account records
<input checked="" type="checkbox"/>	Mass edit Account records
<input checked="" type="checkbox"/>	View printable view for Account records
<input checked="" type="checkbox"/>	View export results for Account records
<input checked="" type="checkbox"/>	View Account all categories
<input checked="" type="checkbox"/>	Create Account all categories
<input checked="" type="checkbox"/>	Delete Account all categories
<input checked="" type="checkbox"/>	View Account all custom fields
<input checked="" type="checkbox"/>	Edit Account all custom fields
<input checked="" type="checkbox"/>	View Account security
<input checked="" type="checkbox"/>	Edit Account security
<input checked="" type="checkbox"/>	View Account all transactions
<input checked="" type="checkbox"/>	Create Account all transactions

**User Record - System Rights Page for Contact Rights (Edit Mode for Users Created in TeamConnect 2.5 SP4 or Earlier)**

**Note:** In the screenshot above, the check-boxes under the **Group** column are read-only.

Rights are cumulative. If a user has rights assigned at the user-account level and rights inherited from member groups, the user's rights are the sum of user rights and member group rights. If a user is a member of multiple groups then the user inherits all rights assigned to all groups.

If you assign global rights to a user and later add a custom object (record type), category, or custom fields, it is recommended to manually grant access rights to the new object, category, or custom fields.

#### To assign or view individual record rights

1. Search for a user account and open it.
2. To change record rights, from the left navigation pane, click **System Rights**.
3. From the **View** drop-down list, select an option (for example, Contacts).
4. (For users created in TeamConnect 2.5 SP4 or earlier only) You can do one or more of the following:
  - To assign a specific record or area right, select the check-box for the permission under the **User** column.

- To unassign a specific record or area right, clear the check-box for the permission under the **User** column.
  - If you are viewing existing rights, the **User** column displays rights assigned at the user account level. The **Group** column displays rights inherited from member groups.
5. (For users created in TeamConnect 3.x) The **Group** column displays rights currently inherited from member groups.
- The **System Rights** page for a user created in TeamConnect 3.x does not display a **User** column.
6. Click a save option.

## User System Rights Page Field Descriptions

Field descriptions for specific records or areas are described in:

- [Rights for End-users](#)
- [Rights for Administrators](#)
- [Rights for Solution Developers](#)

### User System Right Page Fields

Label	Description
<b>View drop-down list</b>	<p>A list of record types and application areas to which you can assign access rights, for example, system settings, object definitions, etc.</p> <p>After selecting a list option, the workspace is updated with a list of rights specific to that option. For example, if you select Contacts, the rights list displays only rights for working with contact records.</p>
<b>Action</b>	<p>(Available only for users migrated from TeamConnect 2.5 SP4 or earlier. Also only available in Edit view when a global rights option is selected from the <b>View</b> menu, such as <b>User Rights</b>, <b>Admin Rights</b>, or <b>Setup Rights</b>)</p> <p>Displays both of the following per permission:</p> <ul style="list-style-type: none"> <li>• <b>Select</b>—Enables all permissions for all child records under this rights group. The corresponding rights will be applied at the user account level.</li> <li>• <b>Unselect</b>—Disables all permissions for all child records under this rights group. The corresponding rights will be applied at the user account level.</li> </ul>
<b>Permission</b>	<p>The available Permissions list items depends on the type of option selected from the <b>View</b> drop-down list.</p> <ul style="list-style-type: none"> <li>• If a global rights option is selected from the <b>View</b> drop-down list (for example, User Rights, Admin Rights, Setup Rights), permissions include the following:</li> </ul>

	<p><b>View all record types</b>—To grant search and read rights for records of this type, select the <b>View all record types</b> check-box.</p> <p><b>Edit all record types</b>—To grant rights to edit records of this type, select the <b>Edit all record types</b> check-box.</p> <p><b>Create all record types</b>—To grant rights to create records of this type, select the <b>Create all record types</b> check-box.</p> <p><b>Delete all record types</b>—To grant rights to delete records of this type, select the <b>Delete all record types</b> check-box.</p> <p><b>Perform all other actions</b></p> <p><i>Note:</i> See <a href="#">Understanding Rights</a>, <a href="#">Rights for End-users</a>, <a href="#">Rights for Administrators</a>, and <a href="#">Rights for Solution Developers</a>.</p> <ul style="list-style-type: none"> <li>• If an individual rights option is selected from the <b>View</b> drop-down list (for example, Contacts, Documents, etc.), permissions like the following are available: <ul style="list-style-type: none"> <li><b>View Contact records</b></li> <li><b>Edit Contact records</b></li> <li><b>Create Contact records</b></li> <li><b>Delete Contact records</b></li> <li><b>View Contact all categories</b></li> <li><b>Create Contact all categories</b></li> <li><b>Delete Contact all categories</b></li> <li><b>View Contact all custom fields</b></li> <li><b>Edit Contact all custom fields</b></li> </ul> </li> </ul> <p><i>Note:</i> The list above provides examples of permissions related to individual record types but is not all-inclusive. For comprehensive lists of permissions according to record type, see <a href="#">System Record Rights</a>.</p>
<b>Group</b>	<p>Display values indicate which rights are currently assigned from member group(s). The available Permissions list items depends on the type of option selected from the <b>View</b> drop-down list.</p> <ul style="list-style-type: none"> <li>• If a global rights option is selected from the <b>View</b> drop-down list (for example, User Rights, Admin Rights, Setup Rights), one of the following may display: <ul style="list-style-type: none"> <li>○ <b>All rights selected</b>—Indicates the permission was enabled globally.</li> </ul> </li> </ul>

	<ul style="list-style-type: none"><li>○ <b>Some rights selected</b>—Indicates that permissions for the child record types were enabled individually, resulting in some permissions being disabled.</li><li>○ <b>No rights selected</b>—Indicates that no permissions for the child record types were enabled.</li><li>• If an individual rights option is selected from the <b>View</b> drop-down list (for example, Contacts, Documents, etc.), one of the following displays:<ul style="list-style-type: none"><li>○ A read-only green check-box <input checked="" type="checkbox"/> for individual record rights that are currently assigned to a member group.</li><li>○ A read-only empty check-box <input type="checkbox"/> for individual record rights that are currently unassigned to all member groups.</li></ul></li></ul>
--	--

#### 2.2.4.10.1 Assigning or Viewing Global User Rights for a User

From a user's **System Rights** page, you can assign or view rights typically associated with end-user functions using a global User Rights view. This option allows for quick assignment of access rights to multiple record types or application areas that end-users typically use.

**Note:** In this topic, *global User Rights* refers to Permissions available when you select the **User Rights** option from the **View** drop-down list of a user account's **System Rights** page.

**Note:** Assigning global User rights is only supported for users migrated from TeamConnect 2.5 SP4 or earlier.

#### To assign or view global User Rights for a user

1. Search for a user account and open it.
2. From the left navigation pane, click **System Rights**.
3. From the **View** drop-down list, select **User Rights**. To assign rights, do one or more of the following:
  - To assign view rights to all records from the User application area, associated categories and custom fields, associated pages for the records (such as Security), click **Select** next to the **View All Records Permission**.
  - To assign edit rights to all records from the User application area, associated categories and custom fields, associated pages for the records (such as Security), click **Select** next to the **Edit All Records Permission**.
  - To assign create rights to all records from the User application area, associated categories and custom fields, associated pages for the records (such as Security), click **Select** next to the **Create All Records Permission**.

- To assign delete rights to all records from the User application area, associated categories and custom fields, associated pages for the records (such as Security), click **Select** next to the **Delete All Records Permission**.
  - To assign any remaining rights to all records from the User application area, associated categories and custom fields, associated pages for the records (such as Security), click **Select** next to the **Perform other actions Permission**.
4. If you are viewing currently assigned rights, the status of enabled rights per permission displays under a column:
- **User**—Displays status of rights assigned at the user account level.
  - **Group**—Displays status of rights assigned at the member groups level.
- For more information, see [User System Rights Page Field Descriptions](#).
5. [Click a save option](#).
- For more information about the granular record-level rights bundled under a global User Rights permission, see [Results from Assigning Global User Rights](#). Information presented in the data tables apply to users and groups.

#### 2.2.4.10.2 Assigning or Viewing Global Admin Rights for a User

From a user's **System Rights** page, you can assign rights typically associated with system administrator functions using a global Admin Rights view. This option allows for quick assignment of access rights to multiple record types or application areas that system or business administrators typically use.

**Note:** In this topic, *global Admin Rights* refers to *Permissions* available when you select the **Admin Rights** option from the **View** drop-down list of a user account's **System Rights** page.

**Note:** Assigning global Admin rights is only supported for users migrated from TeamConnect 2.5 SP4 or earlier.

#### To assign or view global Admin Rights for a user

1. Search for a user account and open it.
2. From the left navigation pane, click **System Rights**.  
From the **View** drop-down list, select **Admin Rights**.
  - To assign view rights to all records from the Admin application area, associated pages for the records (such as Security), click **Select** next to the **All Admin View Rights Permission**.
  - To assign edit rights to all records from the Admin application area, associated pages for the records (such as Security), click **Select** next to the **All Admin Edit Rights Permission**.



- To assign create rights to all records from the Admin application area, associated pages for the records (such as Security), click **Select** next to the **All Admin Create Rights Permission**.
  - To assign delete rights to all records from the Admin application area, associated pages for the records (such as Security), click **Select** next to the **All Admin Delete Rights Permission**.  
  
See [Assigning or Viewing Group System Rights](#) for information about assigning groups of rights to end-users, administrators, or solution developers.
3. If you are viewing currently assigned rights, the status of enabled rights per permission displays under a column:
- **User**—Displays status of rights assigned at the user account level.
  - **Group**—Displays status of rights assigned at the member groups level.
- For more information, see [User System Rights Page Field Descriptions](#).
4. [Click a save option](#).
- For more information about the granular record-level rights bundled under a global Admin Rights permission, see [Results from Assigning Global Admin Rights](#). Information presented in the data tables apply to users and groups.

#### 2.2.4.10.3 Assigning or Viewing Global Setup Rights for a User

From a user's **System Rights** page, you can assign rights typically associated with solution developer functions using a global Setup Rights view. This option allows for quick assignment of access rights to multiple record types or application areas that solution developers typically use.

**Note:** In this topic, *global Setup Rights* refers to Permissions available when you select the **Setup Rights** option from the **View** drop-down list of a group account's **System Rights** page.

**Note:** Assigning global Setup rights is only supported for users migrated from TeamConnect 2.5 SP4 or earlier.

#### To assign or view global Setup Rights for a user

1. Search for a user account and open it.
2. From the left navigation pane, click **System Rights**.  
  
From the **View** drop-down list, select **Setup Rights**.
  - To assign view rights to all records from the Designer (Setup) application area, associated pages for the records (such as Security), click **Select** next to the **All setup view rights Permission**.

- To assign edit rights to all records from the Designer (Setup) application area, associated pages for the records (such as Security), click **Select** next to the **All setup edit rights Permission**.
- To assign create rights to all records from the Designer (Setup) application area, associated pages for the records (such as Security), click **Select** next to the **All setup create rights Permission**.
- To assign delete rights to all records from the Designer (Setup) application area, associated pages for the records (such as Security), click **Select** next to the **All setup delete rights Permission**.

See [Assigning or Viewing Group System Rights](#) for information about assigning groups of rights to end-users, solution developers, or administrators.

3. If you are viewing currently assigned rights, the status of enabled rights per permission displays under a column:
  - **User**—Displays status of rights assigned at the user account level.
  - **Group**—Displays status of rights assigned at the member groups level.

For more information, see [User System Rights Page Field Descriptions](#).

4. [Click a save option](#).

For more information about the granular record-level rights bundled under a global Setup Rights permission, see [Results from Assigning Global Setup Rights](#). Information presented in the data tables apply to users and groups.

#### 2.2.4.11 Assigning or Viewing User Category Rights

From a user account's **Category Rights** page, you can do the following:

- (For users created in TeamConnect 2.5 SP4 or earlier) You can assign or view rights to categories and custom fields.

Afterward, the user will be able to view and work with the categories (for example, add and remove categories from a record type) and custom fields (for example, populate or edit the field value).

- (For users created in TeamConnect 3.x) You can view rights to categories and custom fields inherited from the users' member group(s). For these users, you cannot assign or manage category or custom field rights at the user account level. You must edit a member group's category or custom field rights. For more information, see [Assigning Group Category and Custom Field Rights](#).

(For users created in TeamConnect 2.5 SP4 or earlier) You can assign rights that apply:

- Globally—If you expect to enable rights to work with most categories and custom fields across multiple record types, work with these types of rights to save time.

For example, if you select the **View** drop-down list, **User Rights** option then on the resulting page you can assign rights to view all categories (for all contact records, all document records, all project records, etc.) with one button click.

**Note:** In the screenshot above, the permission status messages under the **Group** column are read-only and indicate which permissions are inherited from a member group.

- Individually—If you expect to enable more restricted access to categories and custom fields, you can set more granular level access per record type and per specific category/custom field.

**Note:** In the screenshot above, the check-boxes under the **Group** column are read-only and indicate which permissions are inherited from a member group.

### To assign or view category and custom field rights

1. Search for a user account and open it.
2. To change rights for categories and custom fields, from the left navigation pane, click **Category Rights**. See the appendix, [Rights for End-users](#) for additional information about general rights for categories and custom fields for each record type.
3. (For users created in TeamConnect 2.5 SP4 or earlier only) To assign global category or custom field rights for all record types, do the following:
  - From the View drop-down list, select **User Rights**.
  - From the resulting rights list, click the **Select** button for the following permission rows:
    - **View all categories**—Allows access to all records', **Categories** pages.
    - **Add all categories**—Allows addition or selection of all available categories from all records', **Categories** pages.
    - **Remove all categories**—Allows removal or clearing of all categories from all records', **Categories** pages.
    - **View all custom fields**—Allows access to custom fields associated with any categories from all records', **Details** pages.
    - **Edit all custom fields**—Allows data entry or changes for custom fields values associated with any categories from all records', **Details** pages.

Afterward, all rights for categories and custom fields of all record types will be assigned to the user.
4. (For users created in TeamConnect 2.5 SP4 or earlier only) To assign specific category or custom field rights for a specific record type, do the following:
  - From the **View** drop-down list, select a record type. For example, Contacts.
  - To assign the following base category rights, select the check-box next to each right:
    - **View Contact category**—Allows access to contact record, **Categories** page.
    - **Add Contact category**—Allows addition or selection of all available categories from the contact record, **Categories** page.
    - **Remove Contact category**—Allows removal or clearing of all categories from the contact record, **Categories** page.

- **View Contact custom fields**—Allows access to custom fields associated with any contact categories from the contact record, **Details** page.
- **Edit Contact custom fields**—Allows data entry or changes for custom fields values associated with any contact categories from the contact record, **Details** page.

To assign additional category or custom field rights, select the check-box next to the right.

5. If you are viewing currently assigned rights, the status of enabled rights per permission displays under a column:

- **User** column—Displays status of rights assigned at the user account level.
- **Group** column—Displays status of rights assigned at the member groups level.

For more information, see [User System Rights Page Field Descriptions](#).

6. [Click a save option](#).

#### 2.2.4.12 Assigning or Viewing User Tool Rights

From a user account's **Tool Rights** page, you can do the following:

- (For users created in TeamConnect 2.5 SP4 or earlier) Assign rights to tools. Afterward, the user will be able to view and work with the tools.
- (For users created in TeamConnect 3.x) View tool rights inherited from member group(s). For these users, you cannot assign or manage tool rights at the user account level. You must edit a member group's tool rights. For more information, see [Assigning Group Tool Rights](#).

**Note:** In the screenshot above, the check-boxes under the **Group** column are read-only and indicate which permissions are inherited from a member group.

Some tools are available from the User application area and other tools are available from the Designer application area.

##### To assign user tool rights

1. Search for a user and open it.
2. To make tools available, from the left navigation pane, click **Tool Rights**.
3. (For users created in TeamConnect 2.5 SP4 or earlier) You can do the following:
  - To enable a tool for the user, select the tool's check-box.
  - To disable a tool, clear the tool's check-box.
4. View tool rights currently inherited from member groups under the **Group** column. A green check-box indicates the permission is enabled.
5. [Click a save option](#).

### 2.2.4.13 Viewing or Creating User History

From a user's **History** page, you can create a history (containing notes about changes to the user), view a list of related history records, or perform an advanced search for a history.

#### To create a history for a user

1. [Search for a user and open the user's account.](#)
2. Click **History** from the left pane.  
The user's History list page will open. You can click a history **Description** to view its details.
3. Click **New History**.
4. Enter a change **Description**.
5. [Click a save option.](#)

### 2.2.4.14 Deleting a User

When you delete a user, the groups that the user belongs to will not be deleted.

#### Scenarios that Prevent Deletion

When you try to delete a user, the following scenarios prevent the deletion:

- You are deleting your own user account.
- The user is a Process Manager in a route or an approval process in any status (pending, completed, and so on).
- The user is an approver in an approval process in any status (pending, completed, and so on).
- The user is the last member of a group that has a Process Manager role in an approval rule.

#### To delete a user account

1. Select the **Admin** tab.
2. Click the **Users** sub-tab link.
3. (optional) From the left navigation pane, select a User list view.
4. Click the link with the user's name.
5. Click **Delete**.

#### What is Deleted

When you delete a user, the following associated data gets deleted:

- All rights assigned in the Rights, Categories, and Tools tabs of the account.
- Group Membership information.

- Record-level Security tab settings for the user.
- Project assignment entries where this user was selected.
- Password history.
- Home Page copies created for this user.
- History records.

#### 2.2.4.15 Using the User General Page

For a list of pages that belong to a user, see [Working with Users](#).

From the General page, you can perform the following:

- [Creating or Editing Users](#)
- [Example for Creating a Limited-Privilege User](#)
- [Activating and Inactivating a User](#)
- [Prompting Users to Change Their Passwords](#)
- [Editing or Viewing a User's Preferences](#)

## 2.3 System Administration

TeamConnect system administration includes the initial configuration of Admin Settings and management of TeamConnect logs.

Admin Settings refers to TeamConnect system settings that are configured by Administrators. Additional system settings are located in the TeamConnect Designer area under System Settings. Both Admin Settings and System Settings are global and apply to all users by default. Certain settings can be customized by users for their individual application sessions.

Managing TeamConnect logs includes viewing TeamConnect log files, changing logging levels (which types of events are logged), and clearing log files. See [Appendixes](#) for more information about creating appenders and additional log management tasks.

### 2.3.1 Working with Admin Settings

The Admin Settings area allows you to define or enable global settings according to the business needs of your organization. Admin Settings generally define system behavior for all users.

Admin settings include the following:

- [General Admin Settings](#)
- [User Interface Settings](#)
- [Region Settings](#)
- [Calendar Settings](#)
- [Documents Settings](#)

- [Search Results Settings](#)
- [Reporting Settings](#)
- [Security Settings](#)
- [Email Settings](#)
- [Notification Settings](#)
- [Invoices Settings](#)
- [Maintenance Settings](#)
- [Connections Settings](#)
- [Admin Settings History](#)
- [About TeamConnect](#)

## Admin Settings vs. User Preferences

Users can override the following Admin Settings from their [Preferences pages](#):

- Date and time formats on the Admin Settings **General** page:
  - Date and Time formats in Read-only mode
  - Date and Time formats in Edit mode
  - Time zone
- Search result settings on the Admin Settings **General** page:
  - Number of results per page
  - Number of columns of contact results per page
  - What to do when there is only one record returned by a search
- **User Interface** page:

All appearance settings can be set by each user. For example, screen colors and fonts.

### 2.3.1.1 General Admin Settings

You can configure the following Admin Settings from the General page:

- (For the TeamConnect Designer area only) Working with documents using applets.
- Record locking—Configure the record locking behavior for scenarios where multiple users may try to update the same TeamConnect file.
- WebIntelligence Single sign-on—Allows TeamConnect users to click reports views from TeamConnect without having to enter Business Objects WebIntelligence login information.

A conceptual overview of the settings follows. If you prefer to skip to the procedure for configuring the settings, see [Setting Up General Settings](#).

Admin Settings - General

**General**

Replace Microsoft Office Special Characters

**Record Locking**

Disable record locking in edit mode

Enable record locking in edit mode

Idle time limit (minutes):  (Enter 0 if you do not want an idle time limit)

Allow users to update Relations for locked or read-only records

**Business Objects**

Version: Business ObjectsXI

Business Objects Single Sign-On Application URL:

Central Management Server Name:

Cookie Domain:

User's TeamConnect and WebIntelligence username and/or password may be different

**Session**

Session timeout (minutes):  Value must be greater than 0 but not greater than 480 (8 hours)

**TeamConnect URL**

The Login URL can be used by different TeamConnect features. The URL can be provided in the body of email notifications. The URL is also used for accessing TeamConnect documents through WebDAV.

Login URL:

Admin Settings - General Page

#### 2.3.1.1.1 Record Locking

The Record Locking settings determine how an open record can be updated. Options include:

- **Not locking records**—Allows multiple users to view a record that is already open, but only the first user who saves the record will have record changes saved.
- **Locking records**—Prevents other users from updating an open record. Record locking only applies to system objects and custom objects for end-users (see [Release Edit Locks Right](#)). record locking does not apply to user or group records.

**Setting a lock time-out**—Prevents other users from updating an open record. After the current record owner has been idle for a specified time duration, the record will be unlocked.

In addition, you can enable or disable rights to update the **Relations** tab of a locked contact or project record.

Optionally, you can assign rights to unlock records from Group rights.

#### 2.3.1.1.2 BusinessObjects Web Intelligence Single Sign-on

If your company uses BusinessObjects Web Intelligence to generate reports based on TeamConnect data, you can enable BusinessObjects Web Intelligence single sign-on (BOSS) from TeamConnect. Afterward, when a user tries to view a report from TeamConnect, they do not need to type their WebIntelligence username and password.



Currently, the single sign-on capability can be configured for BusinessObjects Web Intelligence XI.

## When TeamConnect and Web Intelligence User Names and Passwords Differ

By default, TeamConnect passwords and user names are sent to WebIntelligence for authentication purposes. If the username and password combinations do not match, when the user attempts to run a report, an error page is displayed with several options, one of which is the option to log in to Business Objects manually.

To avoid the manual Web Intelligence login, from Admin Settings, for the General page, select the **User's TeamConnect and WebIntelligence username and/or password may be different** check-box. Also let the users know they must enter their Web Intelligence login information from the **Preferences** area, **Other Settings** page.

## Setting up Business Objects WebIntelligence Support

The following is an overview of steps to set up Business Objects WebIntelligence support:

1. Deploy and configure the Business Objects Single Sign-on (BOSS) web application. See the BusinessObjects documentation and Business Objects Single Sign-on in the Installation Help for more information.

**Note:** *The BusinessObjects Web Intelligence administrator should configure the single sign-on capability.*

2. A solution developer must enable user access to Business Objects reports from TeamConnect home pages.
3. Configure BusinessObjects Web Intelligence settings from TeamConnect Admin Settings.
4. If any user's TeamConnect user and password combination is different from their BusinessObjects Web Intelligence user and password combination, notify users that they need to update their User **Preferences**, **Other Settings** page.

## Points to Remember

- If you upgrade from Business Objects 5.x or 6.x to Business Objects XI, make sure all WebIntelligence URLs in portal panes are updated accordingly.
- Sessions on the WebIntelligence server are created only when users click the WebIntelligence URLs in their home pages, not during the TeamConnect login.

### 2.3.1.1.3 Setting Up General Settings

The following steps describe how to configure the **General** page from Admin Settings. Field descriptions are provided in a table below.

After you update and save the **General** settings page, the changes take effect for users automatically.

### To set up General settings

1. Select the **Admin** tab.
2. Click the **Admin Settings** link under the tab bar.
3. The **General** page should display. Otherwise click the **General** link in the left pane.
4. Edit the fields based on descriptions in [the Admin Settings - General Page fields table](#).
5. Click **Update**.

#### Admin Settings - General Page Field Descriptions

Section	Field Names and Descriptions	Default Value
<b>General</b>	<p><b>Replace Microsoft Office Special Characters</b></p> <p>To enable pasting special characters from Microsoft Office documents, select the check-box.</p> <p>If the check-box is cleared, field values containing Microsoft office special characters may not display properly in the user interface but the value stored in the database will be accurate.</p> <p><i><b>Note:</b> For existing records containing Microsoft Office special characters in field values, you need to create a script to update values manually.</i></p>	<p><b>Replace Microsoft Office Special Characters</b>—Disabled (check-box cleared)</p>
<b>Record Locking</b>	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Disable record locking in edit mode</b>— Allows multiple users to edit the same record simultaneously. The first changes checked in by a user will be saved. Subsequent attempts to check in record changes will trigger warnings that the Document has changed.</li> <li>• <b>Enable record locking in edit mode</b>— Prevents multiple users from editing the same Document simultaneously.</li> </ul> <p>If you selected <b>Enable record locking in edit mode</b>, you can type the <b>Idle time limit (minutes)</b>, which is the maximum time length after which a record that is locked without any changes saved will become unlocked.</p>	<p>The following are enabled:</p> <ul style="list-style-type: none"> <li>• Enable record locking in edit mode</li> <li>• The <b>Idle time limit (minutes)</b> is active and set to <b>15</b>.</li> <li>• Allow users to update Relations for locked or read-only records</li> </ul>

	<p>To allow users to update the Relations tab for locked records, check <b>Allow users to update Relations for locked or read-only records</b>.</p> <p>For more information, see <a href="#">Record Locking</a>.</p>	
<p><b>Business Objects WebIntelligence Settings for version XI</b></p>	<p>The following descriptions apply if you are using Business Objects XI:</p> <ul style="list-style-type: none"> <li>• <b>Business Objects Single Sign-on Application URL</b>—The fully qualified URL of the Business Objects Single Sign-on web application that was deployed and configured as part of the TeamConnect installation. The URL must include the domain name.</li> <li>• <b>Central Management Server Name</b>—The name of the central management server. You must provide the name, not the IP address.</li> <li>• <b>Cookie Domain</b>—If BOSS is deployed on a different server node than the WebIntelligence node, you must provide the name of the cookie domain.</li> <li>• <b>User's TeamConnect and WebIntelligence username and/or password may be different</b>—See <a href="#">BusinessObjects Web Intelligence Single Sign-on</a>.</li> <li>• <b>Version</b>—Select Business Objects XI</li> </ul> <p>For more information, see <a href="#">BusinessObjects Web Intelligence Single Sign-on</a>.</p>	
<p><b>Session</b></p>	<p>Enter a numeric value (minutes) after which an inactive user's session will time out. Afterward, the user will need to log in to TeamConnect again. If a negative value is entered, user sessions will not time out.</p> <p><i>Note: If your application server has an application session timeout configuration, the TeamConnect <b>Session Timeout</b> value will override the application server timeout setting.</i></p>	<p>60 minutes</p>

	<p>If a user is running both a User application area browser and a Designer application area browser, the Designer browser session starts when the browser is opened. As a result, it is possible for the User browser session to time out before the Designer browser session.</p>	
<p><b>TeamConnect URL</b></p>	<p><b>Login URL</b>—Enter the TeamConnect login URL. For example:</p> <p><i>http://host_name:8001/teamconnect3/login</i></p> <p>(where you need to replace host_name with the IP address or application server name; replace 8001 with the port running the TeamConnect application; replace teamconnect3 with the TeamConnect application name)</p> <p><b>Important:</b> <i>If you use the WebDAV feature to associate Document folders with email messages, you must put a valid URL in the <b>Login URL</b> field.</i></p> <p>For email notifications, this URL is sent in the body of the email. This field is also required if your company uses HTTPS for the login URL, if you use a proxy server with TeamConnect, or if you use a different application server to log in to TeamConnect.</p> <p><b>Note:</b> <i>You must include <b>/login</b> at the end of the URL. It is also recommended to type the public IP address, not the local IP address of the host server.</i></p>	<p>Configure the TeamConnect URL in the TeamConnect system settings</p>

### 2.3.1.2 User Interface Settings

You can configure the following Admin Settings from the **User Interface** page:

- **Company Settings**—Enable custom logos and alternate texts and select color profiles for company-wide UI customization. **Note:** *The Color Profile option will determine the **Use System Color** preference for all users.*
- **Button Defaults**—Set the default Save Button and default Cancel button options (e.g., Save vs. Save & View).
- **Custom Color Profile**—Select a designed custom color profile.
- **Batch Display Settings**—Move the **Action** column and its buttons from the right of the display to the left of the display.

- User selection—For certain fields that are populated by user or contact names, you can choose the selection format (drop-down list or search).
- Login screen Message—Text that displays on the login page. For example, a welcome message.

A conceptual overview of the settings follows. If you prefer to skip to the procedure for configuring the settings, see [Setting up the User Interface](#).

#### 2.3.1.2.1 Custom Logo

If you choose to replace the default company logo with a custom logo, the graphic in the upper left of application pages (after login) will be changed.

**Note:** For best appearance, use custom logos that are 50 pixels or less in height.

#### 2.3.1.2.2 Color Profile Administration

Beginning with TeamConnect 5.0, team administrators are able to further customize TeamConnect for their company through color profile administration.

Custom profile color schemes can be designed by an administrator and made available to all users. While regular TeamConnect user preferences on color profiles will trump color profiles set by administrators, administrators retain control over what selections are made available.

### ▣ Developing a Custom Color Profile

The full process for developing custom profiles is as follows:

1. Right-click the LESS Template File Example provided on step 2 and click **Save Link As..** or **Save Target As..**
2. Save the file locally with your desired template name. Ensure that it maintains the ".less" extension.

**Note:** You may need to select 'All Files' from the Save as Type dropdown and manually enter the '.less' extension after the file name in the File name entry field before saving.

LESS Template Link	TCE Version Introduced	Additions from Initial Template
<a href="#">Initial LESS Template File Example</a>	TeamConnect 5.0	N/A
<a href="#">LESS Template File Example - 5.0.1</a>	TeamConnect 5.0.1	Now supports custom colors for top navigation bar, block titles, and left navigation bar.

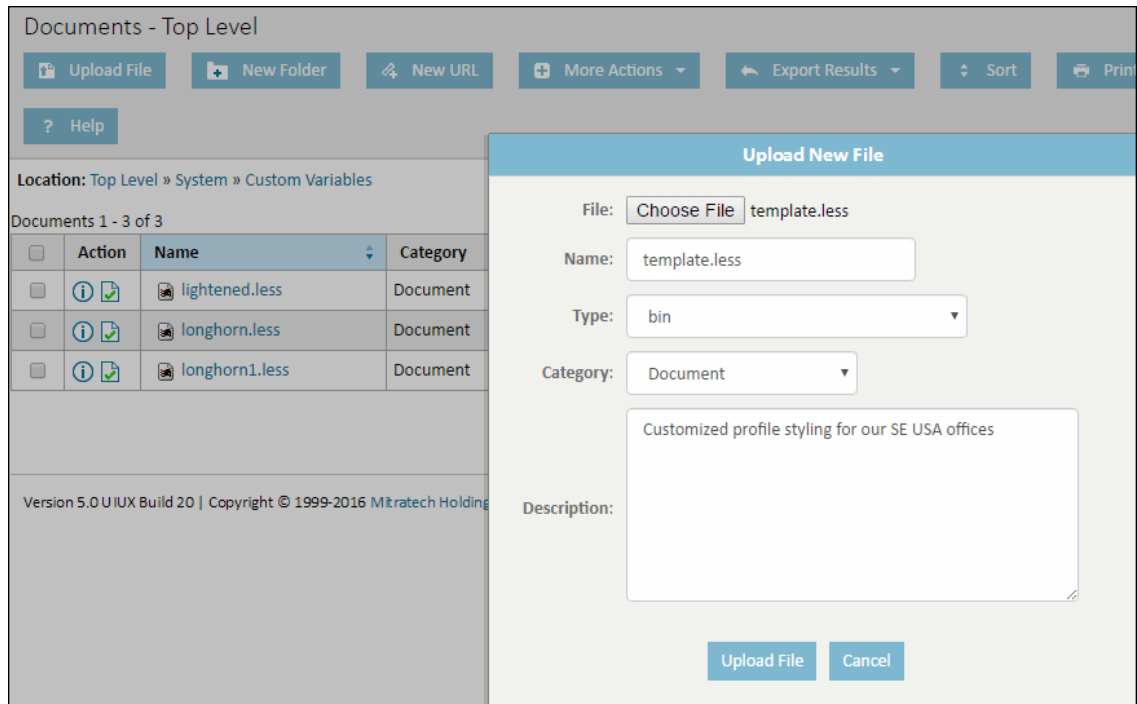
3. Edit the LESS file to your desired color specifications with your favorite text editor (e.g. Notepad++, Sublime, etc.). Colors are determined via hex codes. Hex codes color pickers can be found on several web sites, including [the w3schools color picker](#). **Important:** Please review the *Best Practices for Custom Color Profile Editing* below to ensure compatibility with TeamConnect.
4. Save the new LESS file, making sure that you do not have any spaces or special characters in the name.

### ■ Uploading a LESS file

Administrators can upload a new LESS file two ways. One method is using the **Upload a new Color Profile** button on the **Company Settings** block of **Admin Settings>User Interface**.

Alternatively, users can upload LESS files by following these steps:

1. Select the **Documents** tab from the top navigation bar.
2. Click **Top Level** from the sidebar to view document folder structure.
3. Click **System**.
4. Click **Custom Variables**. This Custom Variables folder is only for uploading LESS files. **Note:** *The Custom Styles folder is also new as of TeamConnect 5.0. For more information on the Custom Styles folder, see step 10 of this procedural.*
5. Once inside the **Custom Variables** folder, click the **Upload File** button. Select your LESS file. The type field can be set to **bin**. See the image below for more detail.



Uploading a LESS file to Custom Variables

## Implementing a new Custom Profile

1. To implement a new color profile, go to the **General** tab of your Preferences. The Preferences link is located under the user icon with your name in the upper-right corner of the application.
2. From the **User Interface** section of the General preferences page, select the **Color Profile** you uploaded.
3. Click the **Update** button to apply these changes.

The screenshot shows the TeamConnect user interface. At the top, there is a navigation bar with 'Home', 'Finance', 'Legal', 'Contacts', 'Calendar', 'Documents', 'Reports', 'Search', and 'Admin'. Below this is a sidebar with a 'Preferences' section containing 'General', 'Reset Password', 'Email Notifications', and 'Other Settings'. The main content area is titled 'Holden, Cody's Preferences - General' and includes an 'Update' button and a 'Help' icon. The settings are organized into sections:
 

- General:** 'Number of card columns in contact card view' is set to 3. There are three checkboxes: 'Do not update Global Search drop down with correspond...' (unchecked), 'Automatically populate Contact field if search returns o...' (checked), and 'Automatically populate Project field if search returns o...' (checked).
- Calendar:** 'Time interval' is set to 30.
- User Interface:** 'Color Profile' is set to 'test3'.
- Button Defaults:** 'Default Save Button' is 'System Save Button' and 'Default Cancel Button' is 'System Cancel Button'.
- Accessibility:** 'Enable accessibility mode' is unchecked.

#### New LESS File Implementation

Users are able to click the **Use System Color** option within their **Preferences** as a color profile. The choice selected from the **Color Profile** field on the **Company Settings** block of **Admin Settings > User Interface** determines which file is used as this option.

The screenshot shows the 'Company Settings' page. It features a dark purple background with white text. The settings include:
 


- Enable custom logo:** A checked checkbox.
- Logo File:** A dropdown menu showing 'teamconnect\_legal\_logo.gif' and an 'Upload a new logo' button.
- Logo Alt Text:** A text input field containing 'TC Logo'.
- Color Profile:** A dropdown menu showing 'test3' and an 'Upload a new Color Profile' button with a help icon.

#### Setting the Use System Color option

### ☰ Removing Color Profiles



To remove a color profile as a choice for their users, administrators must follow these steps in order for best results:

1. From the **Custom Variables** folder in your **Documents**, select the check-box for the LESS file you would like to remove. **Note:** See the steps for **Uploading a LESS file** for help locating this folder, if necessary.
2. Click the  icon to go to the respective page for that file.
3. Click the **Delete** button to remove the LESS file. **Note:** *After deleting the LESS file, users will no longer see that option as a choice in their Color Profile selection. However, users who already had this option selected will still see their same profile design.*
4. To entirely remove the color profile, go to the **Custom Styles** folder within the **System** document directory and repeat the same delete process as outlined in steps 2 and 3 of this section. Users with this style in use will be moved to the default color scheme.

If the CSS (**Custom Styles**) file is deleted and the LESS file (**Custom Variables**) remains, the option will persist for users, and the stylesheet will be re-created upon selection.

## ■ Best Practices for Custom Color Profile Editing

Color profile designers are encouraged to make edits to their profile, test it, and return to make further edits in lieu of completing it all at once without viewing it in the application. **Important:** *If you intend on using the same file name for these revisions and edits, you will need to delete the pre-existing CSS and LESS files for these changes to be reflected. For more information, see [Removing Color Profiles within this Color Profile Administration section](#).*

- @field-text should have good contrast with:
  - @block-header
  - @block-body
  - @table-highlight
  - @button-background
  - @button-background-hover
- @label-color should have good contrast with:
  - @block-body
  - @block-header
  - @table-highlight
- @selected-border should have good contrast with:
  - the white (#FFFFFF) background
- @hover-border should have good contrast with:

- the white (#FFFFFF) background
- the gray (#949494) main navigation bar background
- @main-border should have good contrast with:
  - the white (#FFFFFF) background & text in Calendar
  - the gray (#949494) main navigation bar background
  - the gray (#EDEDED) on a contact card
- @anchor-hover should have good contrast with:
  - the white (#FFFFFF) background
- @anchor-text should have good contrast with:
  - the white (#FFFFFF) background
- @block-body should have good contrast with:
  - @field-text
  - @label-color
  - the white (#FFFFFF) background
- @block-edit-icon should have good contrast with:
  - @block-header
- @table-highlight should have good contrast with:
  - @field-text
  - @table-border
- @button-background should have good contrast with:
  - the gray (#EDEDED) on the left navigation & button bar
  - the white (#FFFFFF) background
- @button-background-hover should have good contrast with:
  - the gray (#EDEDED) on the left navigation & button bar
  - the white (#FFFFFF) background
- @block-border
  - determines the color of the box shadow, so consider the color against the white (#FFFFFF) background
- @table-border should have good contrast with:
  - @table-highlight
  - @block-body

- the white (#FFFFFF) background
- @left-nav-icon should have good contrast with:
  - the gray (#EDED) on the left navigation
  - the white (#FFFFFF) background
- @block-header should have good contrast with:
  - the white (#FFFFFF) background & text

### ▣ Sample values for a High Contrast Theme

- @selected-border #AAA
- @hover-border #6C6C6C
- @mani-border #333
- @anchor-text #333
- @anchor-hover #666
- @block-header #666
- @block-body #EEE
- @block-edit-icon #000
- @table-highlight #BCBCBC
- @button-background #666
- @button-background-hover #333
- @field-text #333
- @block-border #ADADAD
- @table-border #ADADAD
- @label-color #6C6C6C
- @left-nav-icon #333

#### 2.3.1.2.3 Moving Action Buttons

The Action column and buttons appear in areas where you want to copy or edit information. By default, this column appears as the last column on the right of a display and contains action icons, such as edit or copy.

If you prefer, you can move this column and icons to the far right of the display for all users.

#### 2.3.1.2.4 Setting User Selection Options

For certain TeamConnect records, a field may require selection of a TeamConnect user (to populate the field). For example, when adding an assignee to a project or when adding a user to a group account. The **User Selection** block of Admin Settings>User Interface determines whether user records are selected from a drop-down list or from a search module.

- **Select users from drop-down list**—Select this option if your organization does not have too many user names to display in a drop-down list. If your organization has a large number of users, performance can be affected if the drop-down list setting is selected. It may also be inefficient for users to sift through the list of names to find the appropriate user or users to select.

- **Select users from User Search Module**—This option is recommended if there are too many users to effectively display in a drop-down list.

**Item Selection**

Select items from drop-down list

Select items from Lookup Module when there are more than 100 items.

**User Selection Options**

#### 2.3.1.2.5 Item Selection

This setting determines whether lookup table records are selected from a drop-down list or from a search module.

- **Select items from a drop-down list**—Select this option if your organization always wants to display lookup table items using a drop-down list, no matter how many items there are.
- **Select items from Lookup Module when there are more than 100 items**—This option is recommended if there may be 100 or more items to display. If there are fewer than 100 items to display, the items will show in a drop-down list. If there are 100 or more, a search module with auto-suggest capability is displayed instead.

**Note:** *The setting that you choose is the default selection for look up items. The default item selection could still be overridden by developers for specific fields in the design.*

#### 2.3.1.2.6 Setting up the User Interface

The following steps describe how to configure the **User Interface** page from Admin Settings. Field descriptions are provided in a table below.

After you update and save **Admin Settings** from the **User Interface** page, the changes take effect for users automatically.

#### To set up the User Interface

1. Select the **Admin Settings** link from the **All** button..
2. Click the **User Interface** link in the left pane.
3. Edit the fields based on descriptions in [the Admin Settings - User Interface Page fields table](#).
4. Click **Update**.

#### Admin Settings - User Interface Page Field Descriptions

Section	Field Names and Descriptions	Default Value
---------	------------------------------	---------------

<p><b>Company Settings</b></p>	<ul style="list-style-type: none"> <li>• <b>Enable custom logo</b>—Select this check-box to change the logo that appears on the upper left of application pages. If this check-box is cleared, skip the remaining fields under the Company Settings section.</li> <li>• <b>Logo File</b>—To upload a logo file, click the <b>Upload a new logo</b> link. Browse for the local file and select it. Click OK.</li> </ul> <p><i>Note: It is recommended to upload logo files with a maximum height of 50 pixels.</i></p> <p>If you have already uploaded a new logo file, you can select it from the Logo File drop-down list.</p> <ul style="list-style-type: none"> <li>• <b>Logo Alt Text</b>—Enter text to display while the browser is loading the image, or if the browser has images turned off. For example, "Company logo". For more information, see <a href="#">Custom Logo</a>.</li> <li>• <b>Color Profile Administration</b>—Upload and manage user color profiles. For more information, it is strongly recommended that you view <a href="#">Color Profile Administration</a>.</li> </ul>	<ul style="list-style-type: none"> <li>• Disabled (<b>Enable custom logo</b> check-box cleared)</li> <li>• <b>Logo File—Default</b> (Mitratesh logo)</li> </ul>
<p><b>Button Defaults</b></p>	<p>Select the default Save and default Cancel buttons from the following options:</p> <ul style="list-style-type: none"> <li>• Save Changes</li> <li>• Save &amp; Close</li> <li>• Save &amp; View</li> <li>• Save &amp; New</li> <li>• Cancel &amp; Close</li> <li>• Cancel &amp; View</li> </ul>	<p><b>Save Changes</b> <b>Cancel &amp; Close</b></p>
<p><b>Batch Display Settings</b></p>	<ul style="list-style-type: none"> <li>• <b>Position action buttons on the left</b>—Place a checkmark in this checkbox for the <b>Action</b> column in a table of batch entries to display on the left side of the table. Remove the checkmark for the <b>Action</b> column to display on right side of the table.</li> </ul>	<p>Disabled</p>

<b>User Selection</b>	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Select users from drop-down list</b>—Displays TeamConnect users in a drop-down list on pages requiring user name input. Recommended when the number of user accounts is not large.</li> <li>• <b>Select users from User Search Module</b>—Displays a search module on pages requiring user name input. Recommended when the number of user accounts is large.</li> </ul> <p>For more information, see <a href="#">Setting User Selection Options</a>.</p>	<b>Select users from drop-down list</b>
<b>Item Selection</b>	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Select items from a drop-down list</b>—Displays lookup table items using a drop-down list, no matter how many items there are.</li> <li>• <b>Select items from Lookup Module when there are more than 100 items</b>—Displays a drop-down list if there are fewer than 100 items to display. If there are 100 or more, a search module with auto-suggest capability is displayed instead.</li> </ul>	<b>Select users from drop-down list</b>
<b>All Services Menu Settings</b>	<ul style="list-style-type: none"> <li>• <b>Disable Documents in All Services Menu</b>—Select this check-box to remove the <b>Documents</b> link from the All Services Menu and the All Services Portal Pane. You must log back in to TeamConnect after changing this setting for it to take effect.</li> <li>• As of TeamConnect 5.0.1, enabling this checkbox will remove the Documents search view from the left navigation panel.</li> <li>• <b>Disable History in All Services Menu</b>—Select this check-box to remove the <b>History</b> link from the All Services Menu and the All Services Portal Pane. You must log back in to TeamConnect after changing this setting for it to take effect.</li> </ul>	<ul style="list-style-type: none"> <li>• Disabled (<b>Disable Documents in All Services Menu</b> check-box cleared)</li> <li>• Disabled (<b>Disable History in All Services Menu</b> check-box cleared)</li> </ul>
<b>Login Settings</b>	<b>Message</b> —Type a message to display on the login screen. For example, "Welcome to TeamConnect".	

--	--	--

### 2.3.1.3 Regions and Localization

TeamConnect provides options that let you customize system and custom data for multiple languages and regions. Localized areas include the user interface, and features such as the time and date display. You can localize one or more languages and character sets. Localization key values have been updated in TeamConnect 5.0. Detail on these updates can be found in the online help on the Localization Key Changes page.

**Note:** Numbers are not localized and are displayed as Western Arabic numerals, such as 0, 1, 2, 3, 4.

TeamConnect provides the following options to help you localize data:

- **The Admin Settings Region page**—Lets you select a default locale, and select the way that you want to display the system date, time zone, and year. You can also select your local currency type on this page.
- **Locale Manager tool**—Lets you select, change, and delete one or more locales. Additionally, you can export TeamConnect messages and user interface fields to an Excel spreadsheet, enter translations, and then upload the translated data back into TeamConnect. This process allows international users to view information in their preferred language.

#### 2.3.1.3.1 Region Settings

You can configure the following from the **Region** page:

- **Default locale information**—The system default locale and how it is applied to individual end users.
- **System date format**—Choose the default date format for users.
- **System time zone and format**—Choose the default time zone and time format for users.
- **Two-digit year threshold**—Choose a Year marker that differentiates when two-digit dates are considered 20th century versus 21st century.
- **Default currency**—Choose a system currency.

##### 2.3.1.3.1.1 Setting Regional Settings

Use the Admin Settings **Region** page to set your default locale, date and time formats, year threshold, and default currency.

Once you update and save settings on the **Region** page, the changes are visible to users immediately.

#### To set regional settings

1. Click the **All** button.

2. Click **Admin Settings** under the dropdown bar.
3. In the left pane, click the **Region** link.
4. Select or edit the fields based on descriptions in [the Admin Settings - Region Page field table](#).
5. Click **Update**.

#### Admin Settings - Region Page Field Descriptions

Section	Field Names and Descriptions
<b>Locale</b>	<ul style="list-style-type: none"> <li>• <b>Default Locale</b>—Select the default locale that controls the language used in the TeamConnect session, as well as the country format options for dates, times and numbers. You can add or remove locale choices for this drop-down list by running the Locale Manager tool. See <a href="#">Working with Locales</a>. Default Locale: English - United States   <i>Note: A change in default locale will force a page refresh, and will change the values of Date Format and Time Format to values consistent with the new locale.</i></li> <li>• <b>Default users to their browser locale if user locale not set</b>—Set how the system default locale is applied. <ul style="list-style-type: none"> <li>○ If the end user has set a locale in their own Preferences page, that locale will be used during their TeamConnect session.</li> <li>○ If the end user has not set a locale preference, and this check-box is selected, the locale associated with the user's web browser will be used during their TeamConnect session.</li> <li>○ If the end user has not set a locale preference, and this check-box is not selected, the system default locale will be used during their TeamConnect session.</li> <li>○ Default to browser locale: unselected</li> </ul> </li> </ul>
<b>Date Formats</b>	<ul style="list-style-type: none"> <li>• <b>Edit mode date format</b>—Selecting one of the following (where mm is month, dd is day, and yyyy is year): <ul style="list-style-type: none"> <li>○ <b>mm/dd/yyyy</b>—For example, 12/25/2000 is December 25, 2000 (default); or</li> <li>○ <b>yyyy-mm-dd</b>—For example, 2000-12-25</li> </ul> </li> <li>• <b>Read-only mode date format</b>—Select one of the following (where mm is month, dd is day, and yyyy is year):   <i>Note: Year values are always displayed in 4 digits.</i> <ul style="list-style-type: none"> <li>○ <b>mm/dd/yy</b> (default)—For example, 12/25/05 is December 25, 2000; or</li> </ul> </li> </ul>



	<p><b>Note:</b> <i>In edit-mode, this date will display with the year in 4-digit format (mm/dd/yyyy)</i></p> <ul style="list-style-type: none"> <li>○ <b>Month dd, yyyy</b> (abbreviated)—For example, Dec. 25, 2000; or</li> <li>○ <b>Month dd, yyyy</b>—For example, December 25, 2000; or</li> <li>○ <b>Day, Month dd, yyyy</b>—For example, Sunday, December 25, 2000; or</li> <li>○ <b>yyyy-mm-dd</b>—For example, 2000-12-25</li> </ul> <p><b>Note:</b> <i>These values are based on the default locale, and may be overridden at runtime if the end user has selected a different locale in their Preferences.</i></p>
<b>Time Formats</b>	<ul style="list-style-type: none"> <li>• <b>Edit mode time format</b>—Select one of the following: <ul style="list-style-type: none"> <li>○ <b>HH:MM AM/PM</b> (default)—Where HH is hour, MM is minutes. For example, 05:10 PM; or</li> <li>○ <b>Military time</b>—For example, 17:10.</li> </ul> </li> <li>• <b>Read-only mode time format</b>—Select one of the following: <ul style="list-style-type: none"> <li>○ <b>HH:MM AM/PM</b> (default)—Where HH is hour, MM is minutes. For example, 05:10 PM; or</li> <li>○ <b>Military time</b>—For example, 17:10.</li> <li>○ <b>Time Zone</b>—Click the drop-down list to change the system Time Zone. Default is Greenwich Mean Time.</li> </ul> </li> </ul> <p>These values are based on the default locale, and may be overridden at runtime if the end user has selected a different locale in their Preferences.</p>
<b>Year Threshold</b>	<p><b>Two-digit Year Threshold</b>—Type a two-digit year value for the system to interpret which two-digit year values fall in the 20th century or 21st century. Default is 25.</p> <p>For example, a threshold of 25 would mean that values from 00 to 25 are interpreted as 21st century dates (2000 to 2025). Consequently, values from 26 to 99 are interpreted as 20th century dates (1926 to 1999).</p>
<b>Multi-currency</b>	<p><b>Default Currency</b>—Select the local currency. Default is United States, Dollars.</p> <p>The options within the list are <a href="#">set through the Designer application area</a>.</p>

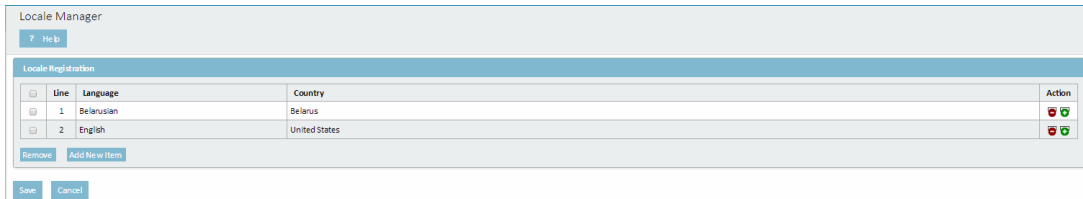
## 2.3.1.3.2 Working with Locales

Use the **Locale Manager** tool to add, change, or delete locales. Only locales that you add in this tool can be selected by administrators or end users.

### To access the Locale Manager tool

1. Click the **All** tab.
2. In the **Tools** section, select **Locale Manager**.

The Locale Manager tool page opens.



Locale Manager Tool

### To add a locale

1. Click **Add New Item**.  
A new row appears in the table of locales.
2. (optional) Click the **Language** drop-down list, and then select a language.
3. Click the **Country** drop-down list, and then select a country.
4. Repeat steps 1 through 3 for as many locales as you need to add.
5. Click **Save**.

TeamConnect checks for duplicate additions and displays error messages if any are found. If there are no errors, your locales are saved.

### To remove a locale

1. Select the check-box next to the locale that you want to remove.
2. Click **Remove**.  
That locale's row disappears from the table of locales.
3. Click **Save**.

If you attempt to remove the locale that is the current Default Locale for the system, you will see an error message.

The table below contains the supported locale languages and the associated country.

#### Supported Locales/Languages

Language	Country
Albanian	Albania
Belarusian	Belarus
Bulgarian	Bulgaria
Catalan	Spain
Chinese (Simplified)	China
Chinese (Simplified)	Singapore
Chinese (Traditional)	Hong Kong
Chinese (Traditional)	Taiwan
Croatian	Croatia
Czech	Czech Republic
Danish	Denmark
Dutch	Belgium
Dutch	Netherlands
English	Australia
English	Canada
English	India
English	Ireland
English	Malta
English	New Zealand
English	Philippines

<b>English</b>	Singapore
<b>English</b>	South Africa
<b>English</b>	United Kingdom
<b>English</b>	United States
<b>Estonian</b>	Estonia
<b>Finnish</b>	Finland
<b>French</b>	Belgium
<b>French</b>	Canada
<b>French</b>	France
<b>French</b>	Luxembourg
<b>French</b>	Switzerland
<b>German</b>	Austria
<b>German</b>	Germany
<b>German</b>	Luxembourg
<b>German</b>	Switzerland
<b>Greek</b>	Cyprus
<b>Greek</b>	Greece
<b>Hebrew</b>	Israel
<b>Hindi</b>	India
<b>Hungarian</b>	Hungary
<b>Icelandic</b>	Iceland

<b>Indonesian</b>	Indonesia
<b>Irish</b>	Ireland
<b>Italian</b>	Italy
<b>Italian</b>	Switzerland
<b>Japanese (Gregorian calendar)</b>	Japan
<b>Korean</b>	South Korea
<b>Latvian</b>	Latvia
<b>Lithuanian</b>	Lithuania
<b>Macedonian</b>	Macedonia
<b>Malay</b>	Malaysia
<b>Maltese</b>	Malta
<b>Norwegian (Bokmål)</b>	Norway
<b>Polish</b>	Poland
<b>Portuguese</b>	Brazil
<b>Portuguese</b>	Portugal
<b>Romanian</b>	Romania
<b>Russian</b>	Russia
<b>Serbian (Cyrillic)</b>	Bosnia and Herzegovina
<b>Serbian (Cyrillic)</b>	Montenegro
<b>Serbian (Cyrillic)</b>	Serbia
<b>Slovak</b>	Slovakia

<b>Slovenian</b>	Slovenia
<b>Spanish</b>	Argentina
<b>Spanish</b>	Bolivia
<b>Spanish</b>	Chile
<b>Spanish</b>	Colombia
<b>Spanish</b>	Costa Rica
<b>Spanish</b>	Dominican Republic
<b>Spanish</b>	Ecuador
<b>Spanish</b>	El Salvador
<b>Spanish</b>	Guatemala
<b>Spanish</b>	Honduras
<b>Spanish</b>	Mexico
<b>Spanish</b>	Nicaragua
<b>Spanish</b>	Panama
<b>Spanish</b>	Paraguay
<b>Spanish</b>	Peru
<b>Spanish</b>	Puerto Rico
<b>Spanish</b>	Spain
<b>Spanish</b>	United States
<b>Spanish</b>	Uruguay
<b>Spanish</b>	Venezuela

<b>Swedish</b>	Sweden
<b>Thai (Western digits)</b>	Thailand
<b>Turkish</b>	Turkey
<b>Ukrainian</b>	Ukraine
<b>Vietnamese</b>	Vietnam

#### 2.3.1.3.3 Localizing Data in TeamConnect

By default, all of the information that is visible in the TeamConnect user interface and all error/information messages appear in English. If you wish to view this information in a different language, you can export the English data to an Excel spreadsheet, enter the translated data into corresponding fields on the spreadsheet, and then import the translated data back into TeamConnect.

Once the translated data is imported back into TeamConnect, the translated elements appear in your preferred language. For more information about setting your locale languages, see [Working with Locales](#).

##### 2.3.1.3.3.1 Exporting Data for Translation

TeamConnect lets you export localization data so that you can translate it into your preferred locale language. Exported localization data is saved to an Excel workbook named *TeamConnect\_MessageResources.xls*.

#### To export data for translation

1. Click the **All** tab.
2. In the **Tools** section, select **Locale Manager**.  
The Locale Manager opens.
3. In the left pane, click the **Export** link.
4. Click the **Export to file** button.  
A dialog box opens.
5. In the dialog box, select one of the following options:
  - To immediately open the spreadsheet, click **Open with**, and then click **OK**.
  - To save the spreadsheet without opening it, click **Save File**, select the appropriate location, and then click **OK**.

## Using the Exported Worksheets

*TeamConnect\_MessageResources.xls* contains all of the TeamConnect data that is available to translate.

The workbook contains the following spreadsheets:

- **Static Resources**—Contains all of the system error messages and default user interface fields and labels.
- **Design Resources**—Contains fields and labels for all of the following :
  - Object definitions
  - Related objects
  - System lookup tables
  - Custom lookup tables
  - Home pages
  - Portal panes
  - Routes
  - Wizards

This page also contains the text for the following:

- The custom logo
- The welcome message on the login page
- The maintenance notification message
- Additionally, this page may also contain fields and labels for other designs, such as CSM and TeamConnect Legal if installed.
- **Custom Resources**—Contains custom message keys and values. May also contain [custom message keys](#) and values for other designs, such as CSM and TeamConnect Legal if installed.

The first two columns of each spreadsheet are in default English and cannot be modified or deleted:

- **Message Key** column—Contains the identification keys for corresponding default values.
- **Default Value** column—Contains the actual text for the default data. For example, for error or informational messages, the **Default Value** is the actual text of the message.



	A	B	C
1	Message Key	Default Value	Chinese
	404.message	The page you are looking for cannot be found. The link may be outdated or temporarily unavailable.   Please try the following:   <ul><li>If you typed the page address in the Address bar, make sure it is spelled correctly. </li><li>Go back to the previous page and try again. </li></ul>	
3			
4	404.title	Page Not Found	
5	accessDeniedPage.goHome	Go to Home Page	
6	accessDeniedPage.li.1	You do not have permission to view this page.	
7	accessDeniedPage.li.2	If you believe this to be an error, please contact your system administrator for more details.	
8	accessDeniedPage.title	Access Denied.	
9	accessDeniedPage.windowTitle	Access Denied	
10	account.accessControlList	Access Control	
11	account.accessControlList.directive	Option	
12	account.accessControlList.permission	Rights Selection	
13	account.accountInvolvedType	Post Account to this Involved	
14	account.accountOverdraftType	Overdraft Type	
15	account.accountProjectType	Posting Project to this Account	
16	account.accountVendorType	Post Account to this Vendor	
17	account.accrued	Total Accruals	
18	account.active	Active	
19	account.allocated	Total Allocated	
20	account.allocationLimit	Allocation Limit	
21	account.allowAccrualInvoice	Accrual	
22	account.allowCreditNoteInvoice	Credit Note	
23	account.allowExpense	Post Expenses to this Account	
24	account.allowInvoiceExpense	Post Invoice Expenses to this Account	
25	account.allowInvoiceNonUSTax	Post Non-US Tax to this Account	
26	account.allowInvoiceTask	Post Invoice Tasks to this Account	

Exported Excel Spreadsheet with Message Keys and Default Value Text

Additional columns correspond to the locales that you registered in TeamConnect using the Locale Manage tool. See [Working with Locales](#) for more information.

You cannot add or remove any columns in this workbook.

### Entering or Changing Localized Data

You can enter or change translated data in the columns that correspond to the locales that you registered in TeamConnect.

	A	B	C
1	Message Key	Default Value	Chinese
	404.message	The page you are looking for cannot be found. The link may be outdated or temporarily unavailable.   Please try the following:   <ul><li>If you typed the page address in the Address bar, make sure it is spelled correctly. </li><li>Go back to the previous page and try again. </li></ul>	该网页您正在寻找无法找到。该链接可能是过时的或暂时无法使用。   请尝试以下方法：   <ul><li>如果您键入的地址栏的网页地址，请确认其拼写正确。 </li><li>返回到上一页并再试一次。 </li></ul>
3			
4	404.title	Page Not Found	找不到网页
5	accessDeniedPage.goHome	Go to Home Page	进入主页
6	accessDeniedPage.li.1	You do not have permission to view this page.	您没有权限查看此页面。
7	accessDeniedPage.li.2	If you believe this to be an error, please contact your system administrator for more details.	如果您认为这是一个错误，请联系您的系统管理员更多细节。
8	accessDeniedPage.title	Access Denied.	访问被拒绝。
9	accessDeniedPage.windowTitle	Access Denied	访问被拒绝
10	account.accessControlList	Access Control	访问控制
11	account.accessControlList.directive	Option	选项
12	account.accessControlList.permission	Rights Selection	选择权
13	account.accountInvolvedType	Post Account to this Involved	这涉及到邮政帐户
14	account.accountOverdraftType	Overdraft Type	透支型
15	account.accountProjectType	Posting Project to this Account	此帐户发布项目
16	account.accountVendorType	Post Account to this Vendor	邮政帐户此供应商
17	account.accrued	Total Accruals	总应计
18	account.active	Active	主动
19	account.allocated	Total Allocated	已分配
20	account.allocationLimit	Allocation Limit	分配限制
21	account.allowAccrualInvoice	Accrual	应计
22	account.allowCreditNoteInvoice	Credit Note	信用证
23	account.allowExpense	Post Expenses to this Account	此帐户后费用
24	account.allowInvoiceExpense	Post Invoice Expenses to this Account	发布此帐户开支发票
25	account.allowInvoiceNonUSTax	Post Non-US Tax to this Account	后非美国税务此帐户

Excel Spreadsheet with Localized Data

#### To enter or change localized data

Enter or change translated data in the appropriate columns and corresponding fields on the spreadsheet.

To make changes to localized data after the file is imported, see [Changing Localized Data After Import](#).

Once you enter the localized data, you can upload the translations. See [Importing Translated Data](#) for more information.

**Note:** *If you copy and paste translated data from third-party translation software, such as Google Translate, you may have difficulty deleting or making changes to the text once you paste it in the spreadsheet. To avoid this, after you copy the text, go to the appropriate cell on the spreadsheet, right-click, select **Paste Special**, and then select **Text**. This removes any hidden characters and allows you to make changes to the pasted text when necessary.*

## Tags, Punctuation, and Symbols

If the information in the **Default Value** column contains html tags, such as <li> </li> or %20, or symbols, punctuation marks, and/or numbers, such as {0} or "{1}", the corresponding translation must also include this information.

For example, if you are translating the following into German:

```
<b>Results Display</b>
{1}: Multiple matches were found for "{0}".
(ANY)
```

The translated text must appear as follows:

```
<b>Ergebnisse anzeige </b>
{1}: Mehrere Spiele wurden fur "{0}" gefuden.
(Beliebig)
```

**Important:** *If you are translating into a language that may have very long words, to improve readability, it is recommended that you use recognized abbreviations in place of the long version of words whenever possible.*

## Adding new locales

You cannot add new locales to the spreadsheet. You must first register the new locale in TeamConnect, and then export the *TeamConnect\_MessageResources.xls* file again. A new blank column appears on the spreadsheet, along with the previously localized data. See [To add a locale](#) for more information.

### 2.3.1.3.3.2 Importing Translated Data

After you export and then translate the data, you can import the spreadsheets into TeamConnect. The default data is replaced and you can view TeamConnect in your preferred language.

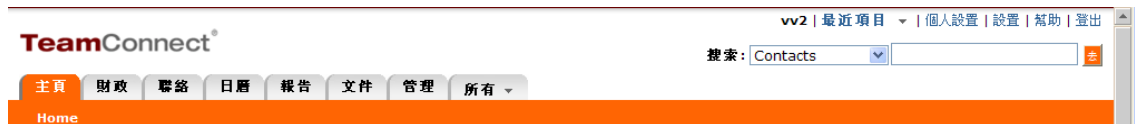
For information about exporting, see [Exporting Data for Translation](#).

**Important:** *You can only import Excel files that end in .xls or .xlsx.*

## To import translated localized data

1. Click the **All** tab.
2. In the **Tools** section, select **Locale Manager**.
3. In the left pane, click the **Import** link.
4. Click **Browse** to locate and select the appropriate .xls or .xlsx file.
5. Click **Upload**.

The file is added to TeamConnect and the data is displayed in your preferred language.



TeamConnect Tabs and Global Menu Translated to Chinese

## Changing Localized Data After Import

To change localized data after you import the spreadsheet, you must export the *TeamConnect\_MessageResources.xls* file again, make changes, and then import the spreadsheet back to TeamConnect. Your changes appear once the spreadsheet is imported.

### 2.3.1.4 Calendar Settings

**Note:** For TeamConnect, always choose the option **Use standard appointment records**. Use of other calendaring applications is not presently supported in TeamConnect.

### 2.3.1.5 Documents Settings

You can configure the following Admin Settings from the **Documents** page:

- **Enable document version control**—Enable management of files uploaded to TeamConnect. For example, when one user checks out a file, other users can view that file but cannot upload changes to that file.
- **Enable WebDAV**—Allows users to manage TeamConnect Documents area files using Internet Explorer.
- **Maximum size of uploaded file (megabytes)**—Defines the maximum document size that can be uploaded to TeamConnect. The default is 50 MB and the minimum value allowed is 1 MB.

A conceptual overview of the settings follows. If you prefer to skip to the procedure for configuring the settings, see [Managing Documents](#).

### 2.3.1.5.1 Document Version Control

Enabling Document Version Control provides document version management in TeamConnect. When a document has been updated, a user can check out the existing document, and check in the latest document, resulting in an additional document version.

If you enable Document Version Control, users who need rights to view different document versions will also need the Document Read rights and Document Version Read rights. Users who need rights to revert to previous document versions will also need the Document Update rights and Document Version Revert rights. For more information, see the Documents section of the appendix, [Rights for End-users](#).

**Important:** *When Document Version Control is enabled, the TeamConnect database stores all versions of all documents, requiring additional storage space.*

### 2.3.1.5.2 Network File Systems for Documents (WebDAV)

The WebDAV feature allows users to perform the following from Internet Explorer:

- Upload local files to TeamConnect
- Download TeamConnect files locally
- Copy files to other TeamConnect folders
- Move files to other TeamConnect folders

**Important:** *The TeamConnect WebDAV feature is certified for Internet Explorer versions 6 and 7.*

**Important:** *If you use the WebDAV feature, you must put a valid URL in the Login URL field of Email Settings. See [the Admin Settings, Email Page Field Descriptions table](#) for details.*

If the Enable WebDAV setting is enabled, TeamConnect Documents pages will display a Open In Explorer link in the upper right. Clicking this hyperlink opens the current TeamConnect Documents folder in Internet Explorer. You can also refer to 3rd party documentation for using Internet Explorer 6 as a WebDAV client.

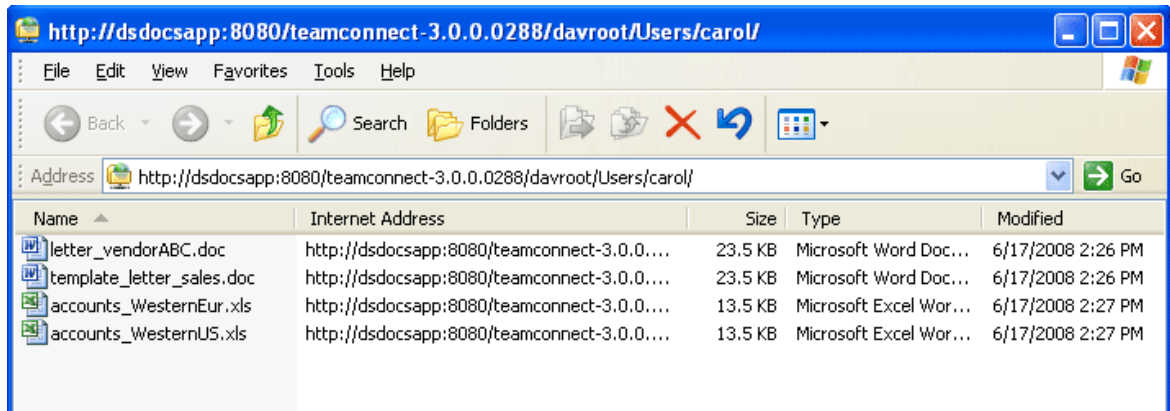
The following overview includes prerequisites and steps to use WebDAV for Documents folders:

1. Enable WebDAV from TeamConnect Admin Settings.
2. Send users instructions for setting up Internet Explorer to automatically link to TeamConnect Documents folders. In addition, you must provide the following URLs to users:
  - User's Documents folder URL:  
`http://Hostname:PortNumber/application_name/davroot/Users/user_name`  
For example,  
`http://10.7.7.3:8005/teamconnect/davroot/Users/John`
  - TeamConnect Root Documents folder URL:  
`http://Hostname:PortNumber/application_name/davroot/`

For example,

`http://10.7.7.3:8005/teamconnect/davroot/`

After WebDAV is enabled from TeamConnect and configured from Internet Explorer, a screen like the following becomes available:



Explorer Screen Networked to TeamConnect User's Documents Folder (WebDAV)

## Points to Remember

- Internet Explorer only supports WebDAV URLs that are less than 100 characters long. See the following Microsoft Knowledge Base note for more information:  
<http://support.microsoft.com/kb/325355>
- TeamConnect includes a WebDAV level 2 server capability.
- TeamConnect supports the WebDAV functionality specified in RFC 2518.
- The WebDAV server uses the same authentication set up for the user in TeamConnect. If single sign-on is not set up, the server requests a username and password; this is the username and password of the TeamConnect user.
- TeamConnect will automatically check out and check in a file when it receives a put (copy) request if the file is not locked by the user or checked out to another user.
- Checking out a file in TeamConnect will not lock the file in Internet Explorer.
- Microsoft refers to WebDAV as "Web Folders".

### 2.3.1.5.3 Managing Documents

Use the Admin Settings, **Documents** page to perform the following tasks:

- Enable document versioning
- Enable WebDAV
- Set the maximum size of documents that can be added to TeamConnect

#### To set up Documents

1. Select the **Admin** tab.
2. Click **Admin Settings** under the tab bar.
3. Click the **Documents** link in the left pane.
4. Edit the fields based on descriptions in [the Admin Settings - Documents Page field table](#).
5. Click **Update**.

#### Admin Settings - Documents Page Field Descriptions

Section	Field Names and Descriptions
<b>Document Security Default Settings</b>	<ul style="list-style-type: none"> <li>• <b>Replace rights on subfolders</b>—The default setting for the <b>Replace rights on subfolders</b> setting on the <b>Security</b> page of a record. If this setting has a checkmark, if you change the rights of a folder, its subfolders and files also receive the same rights. Without a checkmark, if you change the rights of a folder, the rights of its subfolders and files do not change.  If you add a folder or file to a folder with this setting, the folder or file does not update with the folder's rights. To receive security updates, the folder or file must be in the folder when the rights change.</li> <li>• <b>Replace rights on existing files</b>—The default setting for the <b>Replace rights on existing files</b> setting on the <b>Security</b> page of a record. If this setting has a checkmark, if you change the rights of a folder, its files also receive the same rights. Without a checkmark, if you change the rights of a folder, the rights of its files do not change.  If you add a file to a folder with this setting, the file does not update with those rights. To receive security updates, the file must be in the folder when the rights change.</li> </ul>
<b>Document Settings</b>	<ul style="list-style-type: none"> <li>• <b>Enable Document Version Control</b>—To enable access to previous Document versions, select the check-box.  For more information, see <a href="#">Document Version Control</a>.</li> <li>• <b>Enable WebDAV</b>—To enable use of the WebDAV protocol to view the TeamConnect Documents area folders from Internet Explorer, select the check-box.  To restrict the TeamConnect Documents area folders viewing to the TeamConnect web application, clear the check-box.  For more information, see <a href="#">Network File Systems for Documents (WebDAV)</a>.</li> <li>• <b>Maximum size of uploaded file (megabytes)</b>—Defines the maximum size of documents that can be uploaded to TeamConnect. The default is 50 MB and the minimum value allowed is 1 MB.</li> </ul>

**Note:** *The size of the documents that users can add also depends on the amount of memory that is currently allocated to the application server. Users may have difficulty adding documents if your allocated memory is low.*

### 2.3.1.6 Search Results Settings

You can configure the following Admin Settings from the **Search Results** page:

- Default Global Search object and Global Search parameters—For global searches, select the default record type to search. Set global search term requirements.
- Default Search results display and parameters—Set the number of search results that display per application page. Set the maximum search results that display.
- Search timeout threshold—Set a maximum time (seconds) that the system performs a search.
- Search across Projects—For certain search fields, you can set searches to find results across different projects (or all custom objects).
- Enable Custom Search—Turn the Custom Search feature on or off.
- Disable Global Search—Turn off Global Search and remove the Global Search bar.
- Custom search will respect Custom Field Rights—Users who do not have rights to a custom field cannot search on those fields. This setting will default to 'on' for new implementations and will default to 'off' for upgraded TeamConnect instances.

A conceptual overview of the settings follows. If you prefer to skip to the procedure for configuring the settings, see [Setting Up Search Results](#).

Search Settings	
<input checked="" type="checkbox"/>	Enable Custom Search
<input type="checkbox"/>	Disable Global Search

Search Parameters	
Default option in Global Search:	All
*Minimum characters in Global Search:	3
*Records per result page:	100
*Card columns in contact card view:	3
*Maximum Search results:	200
*Maximum Search time (seconds):	10
*Maximum document size(MB) for search:	50

Other Settings	
<input checked="" type="checkbox"/>	Show number of secure results in Global Search results
<input type="checkbox"/>	Disallow Global Search on Related Records
<input checked="" type="checkbox"/>	Automatically populate Contact field if search returns one result
<input checked="" type="checkbox"/>	Automatically populate Project field if search returns one result
<input type="checkbox"/>	Enable search across All Projects for Project field
<input type="checkbox"/>	Custom search will respect Custom Field rights

#### Admin Settings - Search Results Page

##### 2.3.1.6.1 Maximum Search Results

When TeamConnect contains many records it is recommended that you limit the number of search results that display and set a search timeout. This will ensure reasonable system performance times for users.

To limit the number of search results that display, you can decrease the **Maximum Search results** field value (default 1000).

To limit the time allowed to perform a search, you can decrease the **Maximum Search time (seconds)** field value (minimum value, and default, is 10 seconds). If you are using TeamConnect's built-in reporting feature, put a higher value in this field.



## Records Not Displayed

In addition to the search criteria that the user specifies on the search screen, other criteria also determines which records are retrieved and displayed. The following records are NOT displayed:

- Records that do not match invisible qualifiers that may be defined in the search view.
- Records that are filtered out by the system because of where the user is accessing the search screen. For example, when a user searches for child projects from within a parent project, only the projects of the appropriate child object are displayed. Similar filtering applies for other types of related records.
- Records to which the current user lacks access due to the security settings on the record's Security tab. In certain cases, these records can be retrieved but cannot be accessed from a search screen because the hyperlink to the record is replaced with the word "secured."
- Records that are not retrieved before the Maximum number of search results is reached, which discontinues any further searching.
- Records that are not retrieved before the Maximum search time is reached, which discontinues any further searching.

### 2.3.1.6.2 Setting Up Search Results

The following steps describe how to configure the **Search Results** page from Admin Settings. Field descriptions are provided in a table below.

#### To set up Search Results settings

1. Select the **Admin** tab.
2. Click **Admin Settings** under the tab bar.
3. Click the **Search Results** link in the left pane.
4. Edit the fields based on descriptions in [the Admin Settings - Search Results Page fields table](#).
5. Click **Update**.

#### Admin Settings - Search Results Page Field Descriptions

Section	Field Names and Descriptions (* =required)	Default Value
<b>Search Parameters</b>	<ul style="list-style-type: none"> <li>• <b>*Card columns in contact card view—</b> (Only for contacts search views where the Results Display Type is set to Card) For contacts search results, enter the number of columns to display. The default is 3.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>*Card columns in contact card view—</b> 3</li> <li>• <b>Default option in Global Search—</b> Contacts</li> </ul>

	<p>Contact the solution developer for more information about contacts search view configuration.</p> <ul style="list-style-type: none"> <li>• <b>Default option in Global Search</b>—To specify a default Global Search type, select an option from this drop-down list. The options include system objects and custom objects.</li> <li>• <b>*Maximum Search results</b>—Enter the number of records to retrieve when users perform a search (Global or from a search view defined for an object type) in the <b>Maximum Search results</b> field.</li> </ul> <p>For more information, see <a href="#">Maximum Search Results</a>.</p> <ul style="list-style-type: none"> <li>• <b>*Maximum Search time (seconds)</b>—Enter the search timeout threshold. The default and minimum value is 10. If you are using the built-in reporting feature, you may wish to use a larger value.</li> </ul> <p>For more information, see <a href="#">Maximum Search Results</a>.</p> <ul style="list-style-type: none"> <li>• <b>*Minimum characters in Global Search</b>—Type the minimum number of characters required for Global searching across <b>All</b> (records) or <b>All Projects</b> in the <b>Minimum characters in Global Search</b> field. This value does not apply to Global searches for an individual system object or custom object.</li> </ul> <p><i>Important: The minimum value is 1; however, values less than 3 may lead to large search result sets and slow response times.</i></p> <ul style="list-style-type: none"> <li>• <b>*Records per result page</b>—Enter the number of records to display per page of search results. The default is 10.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>*Maximum Search results</b>—1000</li> <li>• <b>*Maximum Search time (seconds)</b>—10</li> <li>• <b>*Minimum characters in Global Search</b>—1</li> <li>• <b>*Records per result page</b>—10</li> </ul>
<p><b>Other Settings</b></p>	<ul style="list-style-type: none"> <li>• <b>Enable search across all projects for Project field</b>—Do one of the following:  To make searches across all custom objects (projects) available to users, select the <b>Enable search across all projects for Project field</b> check-box.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Enable search across all projects for Project field</b>—Disabled (not selected)</li> <li>• <b>Show approximate total</b></li> </ul>

	<p>To restrict searches to a custom object type, clear the <b>Enable search across all projects for Project field</b> check-box.</p> <ul style="list-style-type: none"> <li>• <b>Show approximate total number of Global Search results (including records users have no rights to)</b>—Do one of the following:  To display the approximate numeric count of Global Search results, select the <b>Show approximate total number of Global Search results (including records users have no rights to)</b> check-box.  To display the numeric count of Global Search results that the current user has rights to view, clear the <b>Show approximate total number of Global Search results (including records users have no rights to)</b> check-box.</li> <li>• <b>Automatically populate Contact field if search returns one result</b>—Do one of the following:  To automatically populate a Contact field when the related Contact Search Module returns one record result, select the <b>Automatically populate Contact field if search returns one result</b> check-box.  To display a link to a record if a search returns one record result, clear the <b>Automatically populate Contact field if search returns one result</b> check-box.</li> <li>• <b>Automatically populate Project field if search returns one result</b>—Do one of the following:  To automatically populate a Project field when the related Project Search Module returns one record result, select the <b>Automatically populate Project field if search returns one result</b> check-box.  To display a link to a record if a search returns one record result, clear the <b>Automatically populate Project field if search returns one result</b> check-box.</li> </ul>	<p><b>number of Global Search results (including records users have no rights to)</b>—Disabled (not selected)</p> <ul style="list-style-type: none"> <li>• <b>Automatically populate Contact field if search returns one result</b>—Enabled (selected)</li> <li>• <b>Automatically populate Project field if search returns one result</b>—Enabled (selected)</li> </ul>
<p><b>Custom Search</b></p>	<ul style="list-style-type: none"> <li>• <b>Enable Custom Search</b>—Select this check-box to put a New Search link on each record type's list page. Clicking New Search brings</li> </ul>	<p>Selected</p>

	up a set of search criteria and result fields that can be customized by the end user.	
<b>Global Search</b>	<ul style="list-style-type: none"> <li>• <b>Disable Global Search</b>—Select this checkbox to remove the Global Search drop-down list and text input box from all pages.</li> </ul>	Disabled ( <b>Disable Global Search</b> checkbox cleared)

### 2.3.1.7 Reporting Settings

Use the Reporting page to define the maximum number of search results when exporting and printing reports and the maximum number of data points allowed on a chart.

#### To define reporting settings

- For a Tabular View, in the **Maximum search results in Export and Printable View** box, enter the maximum number of search results that you can export to Excel and print in Printable View.
- For a Chart View, in the **Maximum number of data points allowed in chart display**, enter the maximum number of data points allowed when executing a chart.

#### 2.3.1.7.1 Defining Reporting Settings

Use the **Reporting** page to view information about your connection to Data Warehouse and to set search and data point values for the tabular view and chart view.

You can also set the parameters that will be used when generating the .pdf version of the report.

#### To view Data Warehouse connection information

- If you are successfully connected to the Data Warehouse, **Data Warehouse was last refreshed on <date> <time> appears.** <date> and <time> are displayed in your current preferred format.
- If you are not connected to the Data Warehouse, **Cannot connect to the data warehouse** appears.

#### To define report search settings

- For a Tabular View, in the **Maximum search results in Export and Printable View** box, enter the maximum number of search results that you can export to Excel and print in Printable View.
- For a Chart View, In the **Maximum number of data points allowed in chart display**, enter the maximum number of data points allowed when executing a chart.

#### To set .pdf parameters for reports

The **Default Page Settings** section lets you define the parameters that you want to use when generating a PDF version of the report:

- In the **Orientation** section, click the appropriate position for the report. Options are Portrait (default) or Landscape.
- In the **Margins** section, select the appropriate margin settings. You can use inches or centimeters by selecting an option in the Unit field. The default setting is 0.5.

Each **Header** and **Footer** area of the .pdf is divided into three sections. In each section, you can add one of the following options:

- Report Title
- Page #
- Page # of #
- Run Date
- Run Date/Time
- Logo

You can make a selection in one or more sections or leave all sections empty. Long lines of text will span across empty sections. For example, if you type a long string of text in section one of a header, and nothing in sections two and three, the text in section one will span across all three sections.

**Note:** *These settings apply to graphic and tabular reports that are exported as .pdf files and reports that are set up to run as .pdf's on an automatic schedule.*

### 2.3.1.8 Security Settings

You can perform the following tasks on the Admin Settings **Security** page:

- [Defining the User Password Policy](#)
- [Configuring User Authentication](#)

#### 2.3.1.8.1 Defining the User Password Policy

Use the **Security** page to define your company's password policy as follows:

- **Password Parameters**—Set the guidelines that users must follow when creating log in passwords and how frequently users must change their passwords.
- **Account Lockout**—Set whether or not users are locked out of the system after multiple unsuccessful login attempts.
- **Display Password Policy**—Choose to display your company's complete password policy on the change password page when a user is prompted to change a password.
- **Account Expiration**—Set whether or not user accounts expire after a specified period of inactivity.

- **Password Reset**—Set whether or not users can reset their own password from the login screen. This reset will be accomplished by an automated single-use reset email. For more information on how a user would reset in this method, see [Logging In](#).

**Note:** *If you are an existing user, your current password is not affected by password policy changes. If you attempt to change your password, you must then adhere to the new policies when creating a password. When you change an existing user's password on the **General** tab of the user account, all of the password policy settings apply to the new password, except for the **Minimum Age** setting. This allows you to change a user's password immediately when necessary.*

#### To define the user password policy

1. Click the **Admin** tab.
2. Click **Admin Settings** under the tab bar.
3. In the left pane, click the **Security** link.

The **Security** page opens.

See [the Security Page - Password Account Settings fields table](#) for specific field information.

#### Setting Password Parameters

(Optional) In the **Password Parameters** section, edit the appropriate fields. Leaving the default "0" in a field means that you do not want to set criteria for that option. See [the Security Page - Password Account Settings fields table](#) for specific field information. **Note:** *Passwords with more than 50 characters can result in a user being locked out of the system.*

**Password Parameters**

Minimum Age (days):

Maximum Age (days):

Minimum password length:

Maximum password length:

Minimum Alpha Characters:

Must contain both upper and lower case letters

Minimum Numeric Characters:

Minimum Special Characters:

Number of passwords stored:

Cannot contain more than two consecutive identical characters

Cannot contain user name, first name, or last name

Always show password policy on change password screen

Admin Settings Security Page - Password Parameters Section

## Displaying the Password Policy

(optional) Click the **Always show password policy on change password screen** check-box to display the password policy details that you selected on the Password Parameters screen whenever a user is prompted to change a password.

This also allows the password policy to appear on the **Reset Password** page in user preferences. See [Changing Password](#) for more information.

The graphic below displays the text that users may see on the change password screen and the **Reset Password** page if this check-box is selected:

## Setting Account Lockout

(Optional) In the **Account Lockout** section, check **Enable User Lockout**, and then enter the appropriate information. See [the Security Page - Password Account Settings fields table](#) for specific field information. If you clear this check-box, users are never locked out of their accounts for too many login attempts or time interval/duration violations.

**Account Lockout**

**Enable User Lockout**

Number of incorrect login attempts allowed:

Time interval for incorrect login attempts (minutes):

Lockout duration (minutes):

Admin Settings Security Page - Account Lockout Section

## Setting Account Expiration

(optional) In the **Account Expiration** section, enter the number of days of inactivity that cause the account to expire. The default value for this field is zero. If the value is set to zero, the inactivity expiration will never occur.

**Account Expiration**

Expire accounts after a period of inactivity (days):

Admin Settings Security Page - Account Expiration Section

To set account expiration, click **Update**.

TeamConnect enforces the password policy the next time users try to log in. Users are prompted to change their password if it does not meet the specified criteria.

## Enabling AutoComplete for User Name on the Log In Page

(optional) If you leave the **AutoComplete Settings** check-box selected, when you begin to enter a **User Name** on the TeamConnect log in page, a pop-up suggests names that you can choose from based on names that you previously entered on the log in page. If you unselect this check-box, the auto complete feature is disabled.

### Security Page - Password Account Settings Field Descriptions

Section	Field Name and Description
<b>Password Parameters</b>	<ul style="list-style-type: none"> <li> <b>Minimum Age (days)</b>—Enter the minimum number of days that a user must keep an existing password before it can be changed. This setting is based on the date that the password is set or changed.           <p>This setting only applies to an end user's ability to change their password on the Password Settings screen or from the login screen. Administrators can change user passwords from user accounts' General tab.</p> </li> </ul>



	<ul style="list-style-type: none"> <li>• <b>Maximum Age (days)</b>—Enter the maximum number of days before a user has to change an existing password. This setting is based on the date that the password is set or changed.</li> <li>• <b>Minimum password length</b>—Enter the minimum number of characters, (alpha and numeric) required for a password.</li> <li>• <b>Maximum password length</b>—Enter the maximum number of characters (alpha and numeric) required for a password.</li> <li>• <b>Minimum Alpha Characters</b>—Enter the minimum number of letters (A - Z and/or a - z) that must be included in the password.</li> <li>• <b>Must contain both upper and lower case letters</b>—If selected, the password must contain lower and upper case letters, such as WeDnesDay.</li> <li>• <b>Minimum Numeric Characters</b>—Enter the minimum number of numbers that must be included in the password.</li> <li>• <b>Minimum Special Characters</b>—Enter the minimum number of special characters that must be included in the password. For example *, &amp;, #, @ are all special characters.</li> <li>• <b>Number of passwords stored</b>—Enter the number of previously used passwords that TeamConnect should not allow users to reuse.</li> </ul> <p><i>Note: For the options above, leaving the default "0" in a field means that you do not want to set criteria for that option.</i></p> <ul style="list-style-type: none"> <li>• <b>Cannot contain more than two consecutive identical characters</b>—If selected, the password may not contain more than two consecutive letters or numbers.</li> <li>• <b>Cannot contain user name, first name or last name</b>—If selected, the password cannot contain the user's first or last name.</li> <li>• <b>Always show password policy on change password screen</b>—If selected, in the user interface, the complete password policy displays on the change password page when a user is prompted to change a password. This also allows the password policy to appear on the Reset Password page in user preferences.</li> </ul>
<b>Account Lockout</b>	<ul style="list-style-type: none"> <li>• <b>Enable User Lockout</b>—Select this check-box to lock a user's account after a specified number of consecutive, unsuccessful login attempts. Once this number is reached, the user can no longer log in and must contact an administrator.</li> </ul> <p>By default, this feature is disabled, which means that users can make an unlimited number of unsuccessful login attempts without getting locked out.</p>

	<p><b>Note:</b> The following fields are only available if the <b>Enable User Lockout</b> check-box is selected.</p> <ul style="list-style-type: none"> <li>• <b>Number of login attempts allowed</b>—Enter the maximum number of failed login attempts before a user's account is locked. 5 is the default.</li> <li>• <b>Time interval (minutes)</b>—Enter the number of minutes TeamConnect observes and counts failed login attempts. 30 minutes is the default.</li> <li>• <b>Lockout duration (minutes)</b>—The length of time that a locked out user must wait before trying to log in again.</li> </ul>
<b>Account Expiration</b>	<b>Expire accounts after a period of inactivity (days)</b> —Enter the number of days after which the account expires due to a lack of activity. The default value is zero, which means that accounts never expire.
<b>AutoComplete Settings</b>	<b>Allow AutoComplete for User Name</b> —If you select this option, when you begin to enter the first few characters of a <b>User Name</b> on the TeamConnect log in page, a pop-up suggests names that you can choose from based on names that you previously used on the log in page.

#### 2.3.1.8.2 Configuring User Authentication

Use the **Authentication Plug-in** section of the Admin Settings **Security** page to choose the default authentication method used by TeamConnect. Authentication is the process of comparing the username and password entered in the TeamConnect login screen to those stored in the database. If the information matches, the user is allowed access to TeamConnect.

The Standard Authentication plug-in is the default authentication method and logs the user into TeamConnect only. This is a secure method, but your organization might choose to integrate the TeamConnect authentication process with a network-wide method. The selected default method applies to all users except those that you specifically override.

Plug-ins are categorized based on their effect on the login process. The two main types of authentication plug-ins are:

- **Standard Authentication**—Allows users to bypass the login screen. If you load an authentication plug-in that has standard sign-on support, users do not see the login screen or enter a username and password because they are authenticated at the network level. If standard authentication is enabled, no other standard sign-on mechanism can be used.
- **Non-standard Authentication**—Allows the identity verification to be handled by an external mechanism used in your environment. This type of plug-in allows authentication to be performed by your local network method, but still requires users to log in to TeamConnect by entering a username and password. You can install various non-standard sign-in plug-ins, and then select which one is used on a per user basis.

**Note:** In order to have a choice of authentication methods, the appropriate configuration files must be edited and plug-ins must be installed as described in [Authentication](#).

Regardless of the plug-in that you select, only Standard authentication is used for the TeamConnect Admin and System user accounts and you cannot change it.

## Single Sign-on

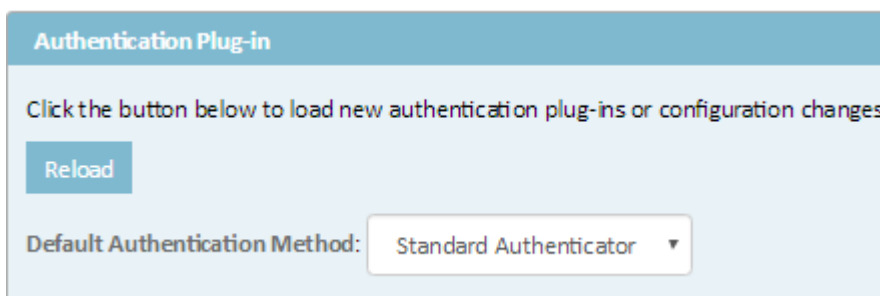
If you plan to use a single sign-on method such as NTLM or LDAP, it must be the default authentication method. It is not possible to have Standard Authenticator as the default while overriding a few individual users with NTLM or LDAP.

When single sign-on is the default authentication method, you cannot override a few individual users with standard authentication. Instead, for those users, you must ensure that they use the regular login URL, then append **standardLogin** to the end of the URL, to force the use of standard authentication.

For information about how to override authentication for individual users, see [Creating or Editing Users](#).

### To configure user authentication

1. Select the **Admin** tab.
2. Click **Admin Settings** under the tab bar.
3. In the left pane, click the **Security** link.  
The **Security** page opens.
4. In the Authentication Plug-in section, click the **Reload** button to refresh any recently changed or added authentication methods.



Admin Settings Security Page - Authentication Plug-in Section

5. In the **Default Authentication Method** drop-down list, select the method that you want to use as the default.  
The method that is currently the default has **(System default)** at the end of its name.
6. If additional fields appear as a result of the method you select, complete the appropriate fields, such as the URL where all user login information is stored for LDAP.
7. Click **Upload**.  
All users who are configured in their user accounts to be authenticated by the selected mechanism are now authenticated using that method.

### 2.3.1.9 Email Settings

You can configure the following from Email Settings:

- SMTP server information for outgoing email notifications
- IMAP server information for incoming email notifications
- TeamConnect IMAP server name and port
- TeamConnect login URL (for email notifications, HTTPS)

A conceptual overview of the settings follows. If you prefer to skip to the procedure for configuring the settings, see [Setting Up Email](#).

### Outgoing Mail Server Settings

Outgoing Mail Server (SMTP):

Reply-to Address:

Use SSL

Outgoing Server Requires Authentication

Username:

Password:

### Incoming Mail Server Settings

Account Type: IMAP

Incoming Mail Server:

Username:

Password:

Use SSL

### TeamConnect IMAP Server Settings

Enable SSL for IMAP

SSL Port:

Keystore Location:

Keystore Password:

Email Settings Page

#### 2.3.1.9.1 Email Notification

There are four sources of email notifications from TeamConnect:

- Default system notifications
- SMTP appender (sends email notifications for Error events from the TeamConnect system log)
- Scheduled action and custom action rules email notifications

- Approval routes email notifications

The following is an overview of steps to set up email notification support:

**Note:** Depending on which type of email notification you are setting up, some of the prerequisites may differ.

1. Set up the email recipients list (for Approval Routes and SMTP Appender).
2. Set up the SMTP Server information from the Admin Settings, Email page, Email Properties section. This is required for system email notifications.
3. Restart the TeamConnect application server.
4. Remind users to enable email notifications from their **Preferences** area.

### Prerequisites

- If your email system requires authentication each time an email is sent:
  - TeamConnect's application server must have JDK version 1.5 or above.
  - The SMTP server must have an email account set up for TeamConnect.
  - The SMTP server must allow the username with which TeamConnect authenticates to be different from the username used as the From address of the message. (The From address of the message is set to the contact email address for the user who made the assignment or appointment. The SMTP appender uses the value of the Reply-To Address field in the Email Properties). Check with your email administrator to correct this limitation if it exists.
- If you are using the SMTP appender or if authentication is required for sending email, you **MUST** specify the username and password for TeamConnect's email account in the Email Properties section of the Email tab in Admin Settings.

### Points to Remember

- Some [email notifications are provided by default](#) with TeamConnect and other [email notifications are created for Approval routes](#).
- The SMTP appender also uses the SMTP hostname and Reply-To address from the **Email Properties** section.
- When a user is the only recipient of an email, it is sent in the user's selected language. When there are multiple recipients, the email is sent in the selected system locale. See [Setting Regional Settings](#) for more information.

#### 2.3.1.9.2 TeamConnect IMAP Server

**Note:** IMAP settings pages are not visible to users who are running TeamConnect® Express.

The TeamConnect IMAP Server allows users to synchronize folders and email (from an email client) with Project records in TeamConnect (such as, Disputes, Matters, or Transactions).

Perform the following to set up and start the TeamConnect IMAP Server:

1. (optional) If you will require encrypted connections (SSL) to the TeamConnect IMAP Server, set up SSL for the IMAP Server.
2. Enable IMAP search views for custom objects (from TeamConnect; for example, Disputes). For more information, see [Enabling IMAP Search for Custom Object Search Views](#).
3. Configure the IMAP Server (from TeamConnect).
4. Start the IMAP Server (from TeamConnect).
5. Verify user/group rights for email messages and folders (from TeamConnect).
6. Configure users' email clients.

#### 2.3.1.9.2.1 Setting Up SSL for the IMAP Server

The TeamConnect IMAP Server supports encrypted connections (SSL). If you will require encrypted IMAP connections (SSL) from IMAP Clients, the following are required:

**Important:** *The JDK installation used for running the Keytool should be the same JDK installation used by the TeamConnect application server. For additional information about keystores, see your JDK vendor documentation.*

1. Get the Keytool security tool (provided in a JDK).

**Note:** *For remaining SSL setup requirements, high level descriptions are provided but you must refer to your JDK vendor's documentation for details.*

2. Using Keytool, do the following on the TeamConnect server:

- Create a keystore (including a public/private key pair). Use the RSA algorithm.

**Note:** *You will need the keystore file location, keystore password, and key password to set up the TeamConnect IMAP Server.*

- Generate a Certificate Signing Request (CSR). Send a request for a signed certificate to a Certification Authority (CA), using the information in the CSR.
- After you receive the resulting certificates from the CA, use Keytool to import the certificates to the keystore.

**Note:** *If you had generated a self-signed certificate above, you need to replace it by importing a signed certificate from the CA.*

**Important:** *The keystore used for IMAP over SSL must be generated by a Keytool from a JDK. Also it is recommended to generate a new CSR from the Keytool and order a new SSL certificate for the IMAP server.*

## 2.3.1.9.2.2 Configuring the IMAP Server

You can change the IMAP Port number and set up the IMAP Server to start automatically when TeamConnect is restarted.

### Tips and Warnings

- If you have already started the IMAP Server, and you update the IMAP Port, you need to stop and start the IMAP Server before the new port number will take effect.
- If you have already started the IMAP Server and you update the auto-start behavior of the IMAP Server, the changes take effect the next time you restart TeamConnect.
- If you are unable to start the IMAP Server, there might be a port usage conflict. Another IMAP Server may be running on the application server. There may be permissions restrictions where the default port is reserved for system processes or use by privileged users. You may need to change the port number (for example, 49152 or greater).
- If you have already started the IMAP Server, and you update settings related to SSL for IMAP Server, you need to stop and start the IMAP Server before changes will take effect.
- The TeamConnect IMAP Server supports encrypted connections (SSL) and unencrypted connections simultaneously. If SSL is enabled from IMAP Server system settings, IMAP Clients can still use unencrypted connections using the IMAP Port number

### To configure the IMAP Server

1. Select the **Admin** tab.
2. Click **Admin Settings** under the tab bar.
3. Click the **Email** link in the left pane.
4. Edit the fields based on descriptions in [the TeamConnect IMAP Server Email Settings table](#).
5. Click **Update**.
6. Restart TeamConnect on the application server.

#### TeamConnect IMAP Server Email Settings

Field	Description
<b>Enable SSL</b>	Indicates whether an encrypted IMAP connection is supported.
<b>SSL Port</b>	Enter the port number where you want IMAP server to listen for SSL connections. If you require IMAP Clients to connect to the IMAP Server using SSL, this value must match the IMAP Port specified on users' email client IMAP account. Also see the description for IMAP Port in this table.
<b>KeyStore Location</b>	The path to the keystore file (including the keystore file name). This location should be in the same machine that hosts TeamConnect. For example, /certs/



	ImapKeystore/imapkeystore.jks
<b>KeyStore Password</b>	The keystore password.
<b>Key Password</b>	Password for the generated private key, which you used to obtain the certificate.
<b>IMAP Port</b>	<p>Enter the IMAP port number (for non-SSL connections). This port must match the IMAP Port specified on users' email client IMAP account. The default value is 143.</p> <p><i><b>Note:</b> Even if you select the Enable SSL check-box, you can still set the IMAP Port and IMAP clients can connect to the IMAP Server using non-SSL connections. Also see the description for SSL Port in this table.</i></p>
<b>Status</b>	(Read-only) Indicates whether the TeamConnect IMAP Server is <b>Started</b> or <b>Stopped</b> .
<b>Run on Startup</b>	Indicates whether the TeamConnect IMAP Server will be automatically started when TeamConnect is started (or restarted). If this check-box is cleared, you would need to manually start the TeamConnect IMAP Server after TeamConnect is started.

### 2.3.1.9.2.3 Starting and Stopping the IMAP Server

By default, the IMAP Server is disabled. You must manually start the IMAP Server the first time.

#### Tips and Warnings

- If you restart the IMAP Server while users have IMAP Clients open, the users will need to close and reopen their IMAP Clients.
- You can start the IMAP Server from the Admin Settings without restarting the TeamConnect server.
- The **Start** button is available only if the IMAP Server status is **Stopped**.
- The **Stop** button is available only if the IMAP Server status is **Started**.

#### To start and stop the IMAP Server

1. Select the **Admin** tab.
2. Click **Admin Settings** under the tab bar.
3. Click the **Email** link in the left pane.

4. Do one of the following:
  - To start the IMAP Server, click **Start**.
  - To stop the IMAP Server, click **Stop**.
5. Click **Update**.

#### 2.3.1.9.2.4 Email Rights

Some prerequisite rights for working with email and email folders need to be set up from the Admin area of TeamConnect in user or group rights. There are additional email record and email folder-level rights that can be managed from the user area of TeamConnect.

For details, see [Rights for End-users](#).

#### 2.3.1.9.2.5 Email Clients

See the *Release Notes* for specific client versions supported.

#### 2.3.1.9.3 Setting Up Email

The following steps describe how to configure the Email page from Admin Settings. You can use the **Email** page to configure the connection to an SMTP Server (for system email notifications), configure the connection to an IMAP Server, and configure the TeamConnect URL for email notifications or HTTPS connections. Field descriptions are provided in a table below.

##### To set up Email

1. Select the Admin tab.
2. Click Admin Settings under the tab bar.
3. Click the Email link in the left pane.
4. Edit the fields based on descriptions in [the TeamConnect IMAP Server Email Settings table](#).
5. Click Update.
6. Restart TeamConnect on the application server.

Afterward, users must do the following:

- Verify that the default email address in their contact record is the email address to which notifications should be sent.
- From the Preferences area, select the Email Notifications that they want to receive.

**Important:** Each user who assigns other users to tasks and projects, or adds them as appointment attendees, should have a default email address in their contact

record. If the Email Properties are not defined properly, email notifications will not be sent from TeamConnect.

#### Admin Settings, Email Page Field Descriptions

Section	Field Names and Descriptions (* = required)	Default Value
<b>Outgoing Mail Server Settings</b>	<ul style="list-style-type: none"> <li data-bbox="595 528 1257 618">• <b>Outgoing mail server (SMTP)</b>—The host name or IP address of your Simple Mail Transfer Protocol (SMTP) email server.  <i>Note: TeamConnect routes SMTP connections through port 25 (the default SMTP port).</i>  For more information, see <a href="#">Email Notification</a>.</li> <li data-bbox="595 819 1257 1491">• <b>Reply-To Address</b>—A default email address that will be used as system email notification Reply-to address and email From address. A reply-to address is required in order for TeamConnect to automatically send email.  If a user performs an action that triggers a system email notification, that user's contact record primary email address will override the <b>Reply-To Address</b> value as the resulting email message's From and Reply-to email address. Otherwise, if the contact record has no email address populated, the <b>Reply-To Address</b> value will be used as the email message's From and Reply-to email address.  If the <b>Reply-To Address</b> is empty and the triggering user/contact's email address is empty, then a system email notification will not be sent.  <i>Note: The SMTP appender also uses the <b>Reply-To Address</b> as the <b>From</b> address for Error log email messages.</i></li> <li data-bbox="595 1536 1257 1738">• <b>Use SSL</b>—The check-box you select if the SMTP server uses SSL (secure socket layer) to send email.  <i>Note: TeamConnect uses port 465 as the default SSL port.</i></li> <li data-bbox="595 1783 1257 1839">• <b>Outgoing server requires authentication</b>—Select this check-box if the SMTP server requires</li> </ul>	empty

	<p>authentication for sending email. When selected, you must enter a username and password.</p> <ul style="list-style-type: none"> <li>• <b>Username</b>—TeamConnect's user name for authenticating with the SMTP server.</li> <li>• <b>Password</b>—TeamConnect's password for authenticating with the SMTP server.</li> </ul>	
<b>Incoming Mail Server Settings</b>	<ul style="list-style-type: none"> <li>• <b>Account Type</b>—Indicates that the email account that sends users incoming mail from TeamConnect has to have IMAP enabled.</li> <li>• <b>Incoming mail server</b>—The name of the server for the email account, for example, imap.gmail.com.</li> <li>• <b>Username</b>—The email address that TeamConnect uses to send emails.</li> </ul> <p>TeamConnect only pays attention to unread emails. As a result, you should create an email account specifically for the use of incoming mail.</p> <ul style="list-style-type: none"> <li>• <b>Password</b>—The password of the email account.</li> <li>• <b>Use SSL</b>—The check-box you select if the IMAP server uses SSL (secure socket layer) to send email.</li> </ul> <p><i>Note: TeamConnect uses port 993 as the default SSL port and port 143 as the default non-SSL port.</i></p>	empty
<b>TeamConnect IMAP Server Settings</b>	<p>These settings are described in detail in <a href="#">the TeamConnect IMAP Server Email Settings table</a>.</p>	empty

### 2.3.1.10 Notification Settings

Rules can be designed to send email notifications after certain actions have been triggered. To customize the templates used for these emails, you can update the notification system settings. Notifications can be defined for all objects, all custom objects, or a specific system or custom object.

See [Specifying Actions for Custom Action Rules](#) and [Defining Settings for Scheduled Action Rules](#) for more information about creating and editing rules to send notifications, as well as defining recipients of notifications.

Users must be given the appropriate rights to view, edit, create, and delete notification templates. For details, see the [Rights for End-users](#) appendix.

For notification templates that will be defined only for a custom object, the custom object must be created before creating the templates.

### To view notification templates

1. Click the **Admin** tab.
2. Click **Admin Settings** under the tab bar.
3. Click **Notifications** in the left pane.

The page of notification templates opens. Click the following table column headers to sort the list of notification templates: **Name**, **Active**, **Description**, **Defined for**.

4. To open a notification template, click its link in the **Name** column.

### To create or edit notification templates

1. Click the **Admin** tab.
2. Click **Admin Settings** under the tab bar.
3. Click **Notifications** in the left pane.
4. Select one of the following options:
  - To create a new notification template, click the **New** button.
  - To edit a notification template, click the **Edit** icon next to the record that you want to edit.

The notification template opens to its **General** page.

5. On the **General** page, enter or edit the appropriate information. See [the Admin Settings - General Page fields table](#) for more information.

Fields with an asterisk are required and must be completed before you can save this record.

6. Click the **Email Definition** link in the left pane.

7. On the **Email Definition** page, enter or edit the appropriate information. See [the Email Definition Page fields table](#) for more information.

Fields with an asterisk are required and must be completed before you can save this record.

8. Select a save option.

#### General Page Field Descriptions

Field (* = required)	Description
<b>*Name</b>	The name of the notification template.

<b>*Key</b>	<p>Uniquely identifies a template. Notification templates cannot have the same unique key.</p> <p>This key can be used to retrieve the template in a custom action rule through the notification API. It can also be used to identify the template in a custom design for use at the global level and at the object definition level.</p> <p>After you save the notification template, the unique key appears as read-only text and you cannot change it.</p>
<b>Is active?</b>	<p>The active status of the notification template. An inactive template will not show up as a selection option anywhere, such as in Admin Settings, object definitions, rules, etc. If a template that is associated with a rule or action is deactivated, the template notification will continue to use the deactivated template.</p>
<b>Description</b>	<p>A description of the notification template. Limited to 250 characters.</p>
<b>*Defined for</b>	<p>The object type the notification template is defined for. Available selections in the drop-down menu are <b>All Objects</b>, <b>All Projects</b>, and individual system or custom object types.</p>

#### Email Definition Page Field Descriptions

<b>Field</b> (* = required)	<b>Description</b>
<b>*Subject line</b>	<p>The subject line of the notification template. Limited to 250 characters. Supports Object Navigator variables.</p>
<b>*Body</b>	<p>The body text of the notification template. Limited to 250,000 characters. Supports Object Navigator variables.</p> <p>You can use the editor buttons in this field to format the email body text as you desire.</p> <p>Click the <b>Select Field</b> link under the <b>Body</b> field to show the <b>Object Navigator</b> button. The <b>Object Navigator</b> button opens a pop-up menu that displays the objects whose fields can be inserted into either the <b>Subject line</b> or <b>Body</b> of notification templates. This is determined by which field the text cursor was last active in when the <b>Insert</b> button is clicked. The objects available in Object Navigator are determined by which objects the template is defined for. In the Admin Settings pages, the object fields will be depicted in the text as variables, such as <code>#{currentUser}</code>.</p>

**Important:** The Object Link > Share URL value in Object Navigator will not display properly in notification emails if the **Login URL** setting in the **General** page of **Admin Settings** is not defined properly.

#### To make a copy of a notification template

1. Click the **Admin** tab.
2. Click **Admin Settings** under the tab bar.
3. Click **Notifications** in the left pane.
4. Click the link in the **Name** column of the desired notification template.
5. Click the **Copy** button.

The notification template editor opens with the contents of the copied notification template. The exceptions are the Name field, which is set to "Copy of [original template name]", and the Unique Key field, which is set to blank.

6. Edit the appropriate information on the **General** and **Email Definition** pages.
7. Select a save option.

#### To print a notification template

1. Click the **Admin** tab.
2. Click **Admin Settings** under the tab bar.
3. Click **Notifications** in the left pane.
4. Click the link in the **Name** column of the notification template you want to print.
5. Click the **Printable View** button.

#### 2.3.1.10.1 Customizing System Notifications

In the **System Notifications** section, you can specify which notification templates are sent in place of the default system notifications. This section lists general actions performed by objects that send system notifications. Beside each action is a drop-down list of available notification templates for that action.

Templates that are customized at this level will be used by all objects that perform the relevant action unless [another notification template is selected in an object definition](#), which supersedes these settings.

#### To customize a system notification for an action

1. Click the **Admin** tab.
2. Click **Admin Settings** under the tab bar.

3. Click **Notifications** in the left pane.
4. In the **System Notifications** section, select [the notification template](#) from the drop-down menu next to the desired action. To set a system notification back to its default, select **Default** from the drop-down menu.
5. Click the **Update** button.

## System Notifications Actions

The following list contains the types of email notifications. Updated the field in the **System Notifications** section to send a custom notification template instead of the default system notification.

- **Set as an assignee**—The notification that users receive when assigned to a record.
- **Workflow approval request**—The notification that users receive when assigned as the approver of a request.
- **Workflow approval notice**—The notification that users who complete an action that created an approval request receive when another user approves the request.
- **Workflow rejection notice**—The notification that users who complete an action that created an approval request receive when another user rejects the request.
- **Workflow review request**—The notification that users receive when assigned as a reviewer of a request.
- **Workflow expiration notice**—The notification that approvers receive to remind them that an approval request is about to expire.
- **Workflow review completion notice**—The notification that users receive when a review they requested is complete.

### 2.3.1.11 Invoices Settings

Invoices in TeamConnect represent bills that outside counsel and other vendors send to your organization. To customize how users work with invoices, you can update the invoice system settings.

#### To update Invoices settings

1. Click the **Admin** tab.
2. Click **Admin Settings** under the tab bar.
3. Click **Invoices** in the left pane.
4. Edit the fields based on descriptions in the [the Admin Settings, Invoice Page fields table](#).
5. Click **Update**.

#### Admin Settings, Invoice Page Field Descriptions



Section	Field Names and Descriptions	Default Value
<b>Invoice Settings</b>	<ul style="list-style-type: none"> <li>• <b>Reason for rejection is required</b>—Select the check-box if you want to require users to select a reason for rejection on the <b>My Approvals</b> page before they reject invoice approval requests.</li> <li>• <b>Comment to requestor is required</b>—Select the check-box if you want to require users to enter comments in the <b>Comment to Requestor</b> field on the <b>My Approvals</b> page before they reject invoice approval requests.</li> <li>• <b>Require In House Comments upon adjustment</b>—Select the check-box if you want to require the user to enter comments in the <b>In House Comments</b> field when adjusting an invoice line item.</li> <li>• <b>Allow users not part of workflow to edit line items</b>—Place a checkmark in the checkbox if you want to allow users who are not part of the invoice approval group to edit line items when an invoice is in the approval workflow. In addition to selecting this checkbox, users must have system rights to edit line items.</li> <li>• <b>Enable International Line Item Tax Details</b>—Place a checkmark in the checkbox to enable invoices to specify non-US taxes for individual line items.</li> </ul>	Check-boxes are empty
<b>Invoice Line Item Filter Criteria</b>	<p><b>Maximum number of items in list</b>—Enter the maximum number of items that can appear in multi-select lists when selecting values for a search of the <b>Line Items</b> fields on the <b>General</b> page of an invoice.</p> <p>If the number of items in a multi-select list is higher than the number you enter, a search box appears instead of a list. The search box allows the user to search across all databases.</p>	100
<b>Invoice Line Items Display</b>	<p>The following fields apply to the <b>Line Items</b> section on the <b>General</b> page of an invoice.</p> <ul style="list-style-type: none"> <li>• <b>Line Item Rate</b>—Specify up to how many numbers users can enter after the decimal in the <b>Rate</b> field.</li> <li>• <b>Line Item Units/Hours</b>—Specify up to how many numbers users can enter after the decimal in the <b>Unit</b> field.</li> </ul>	2 for both options

These settings apply immediately to all new invoices. Existing data does not change. For more information about setting decimal places, see [Setting Decimal Places for Invoice Line Items](#).

#### 2.3.1.11.1 Setting Decimal Places for Invoice Line Items

In the **Invoice Line Items Display** section, you can specify how many numbers users can enter after the decimal for the **Rate** or **Unit** line item fields on the **General** page of an invoice. Setting the number of decimal places for invoice line items is useful for rates or units that use fractions, such as photocopying at \$0.055 per page.

The system saves values in one of the following ways:

- If users enter values with more decimal places than the system setting, the system rounds the value. See examples 1 and 2 in the [Examples of Setting Decimal Places](#) table.
- Values an equal distance from the decimal place round down if the value before the 5 is even and up if the value before the 5 is odd. See examples 3 and 4 in the [Examples of Setting Decimal Places](#) table.
- If users enter values with less decimal places than the system setting, the system adds zeros to the value. See example 5 in the [Examples of Setting Decimal Places](#) table.

The following table provides examples of how the system formats values in the **Rate** and **Unit** fields.

**Examples of Setting Decimal Places**

Example Number	Number of Digits after Decimal	Value User Enters	Format of Value
1	2	0.0345	0.03
2	2	0.0379	0.04
3	3	1.1245	1.124 (because 4 is even)
4	3	1.1235	1.124 (because 3 is odd)
5	4	0.03	0.0300

#### 2.3.1.12 Approval Settings

Use the **Approvals** page to control the settings for approval requests and the **My Approvals** pages.

##### To update Approval settings

1. Click the **Admin** tab.
2. Click **Admin Settings** under the tab bar.
3. Click **Approvals** in the left pane.
4. Edit the fields based on descriptions in [the Admin Settings, Invoice Page fields table](#).
5. Click **Update**.

 Admin Settings, Approval Page Field Descriptions

Section	Field Names and Descriptions	Default Value
<b>Approval Settings</b>	<ul style="list-style-type: none"> <li>• <b>Allow batch approval</b>—Select this field to allow approvers to approve multiple requests at the same time by clicking the <b>Approve Selected</b> button below the table on the <b>Requests to Approve</b> page. Remove the checkmark to remove the <b>Approve Selected</b> button below the table so that approvers can only approve requests individually.</li> <li>• <b>Allow batch rejection</b>—Select this field to allow approvers to reject multiple requests at the same time by clicking the <b>Reject Selected</b> button below the table on the <b>Requests to Approve</b> page. Remove the checkmark to remove the <b>Reject Selected</b> button below the table so that approvers can only reject requests individually.</li> <li>• <b>Show days remaining before approval expires instead of date of expiration</b>—Select this field to include the <b>Days Pending Action</b> column in the <b>Requests to Approve</b> table. This column lists the number of days before the request expires. Without a checkmark, the table includes a <b>Due Date</b> column with the date of expiration.</li> <li>• <b>Require Internal Comments upon rejection</b>—Select this field to require the approver to enter comments in the <b>Internal Comments</b> field of the <b>Confirm Rejection</b> popup to reject the record(s).</li> <li>• <b>Allow this group to batch reassign approval requests</b>—Select a group from the drop-down field to allow the members of that group to use the <b>Reassignment page of My Workflow Processes</b>.</li> </ul>	The first check-box has a checkmark, and the other three have no checkmarks.
<b>Expiration Reminders</b>	<b>Enable reminder x day(s) prior to expiration</b> —Select this field to enable the ability for approvers to receive email reminders before requests expire. After you place a checkmark in the check-box, enter a number of days in the text field.	The first check-box has a checkmark and the text field has a 1.

You can set up to three reminders. To receive the email reminders, the user must enable those reminders under **Email Preferences**.

### 2.3.1.13 Maintenance Settings

The **Maintenance** link in Admin Settings displays a page that you can use to set up advisory messages warning of upcoming periods of system maintenance, during which TeamConnect will not be available to end users.



There might be no values in the fields on this page. You need to enter values only when a maintenance shutdown is approaching. There are two kinds of advisory messages:

**Maintenance Notification** messages appear when a user starts a session. These are intended for longer-term advisories when a shutdown is not imminent.

**Shutdown Alert** messages appear, brightly highlighted, at the top of every page of a user's session. These are intended to warn the user that shutdown is imminent and they should save their work and log out immediately.

**Maintenance Notification**

Notify all users about scheduled maintenance when they log in. Specify when the maintenance will be and when to start displaying a notification message.

Date & Time:    

Message: 

TeamConnect will be temporarily unavailable. Please plan accordingly. We apologize for any inconvenience this may cause.

Display before shutdown (days):

**Shutdown Alert**

Specify when an alert message will be shown to all users before the system is about to be shut down.

Display before shutdown (minutes):

#### Maintenance Settings

#### Maintenance Settings

Field	Description
<b>Date</b>	Enter the date at which the maintenance shutdown will occur. You can type the date, or you can click on the calendar icon to select a date.
<b>Time</b>	Enter the time at which the shutdown will occur. Select a time from the drop-down list.

<b>Message</b>	Enter the text of the advisory message to be displayed to users.
<b>Display before shutdown (days)</b>	Enter the number of days before the actual shutdown. During these days each user who logs in will see the advisory message in a separate window when their session begins. Default is 7 days.
<b>Display before shutdown (minutes)</b>	Enter the number of minutes before shutdown. During this period the advisory message will appear, highlighted, at the top of every page in a user's session. Default is 30 minutes.

Once the date and time have passed, TeamConnect will stop issuing advisory messages. However, the values in these fields will persist until they are manually erased or changed in the future.

### 2.3.1.14 Connections Settings

You can configure proxy authentication settings from the Connections page. Depending on your network configurations, proxy authentication may be required for your application to connect to Mitratesh's server when viewing the Available Updates (see [Viewing Information About Available Updates](#)).

**Proxy Settings**

Enable Proxy Authentication:

Proxy Host Name/IP Address:

Proxy Port:

Proxy Username:

Proxy Password:

Authentication Realm:

Enable NTLM Authentication:

NT Domain Name:

Admin Settings - Connections Page

Proxy authentication information on this page is used when connecting to Mitratesh's Available Updates server, and when the Collaborati Spend Management feature connects to Collaborati.

**Collaboration Portal**

The Collaboration Portal is designed to allow secure access to your TeamConnect instance from outside of your firewall. For more information on this feature, contact Mitratesch Support.

#### 2.3.1.14.1 Setting Up Connections

The following steps describe how to configure the Connections page from Admin Settings. Field descriptions are provided in a table below.

#### To set up Connections

1. Select the **Admin** tab.
2. Click **Admin Settings** under the tab bar.
3. Click the **Connections** link in the left pane.
4. Edit the fields based on descriptions in [the Admin Settings - Connections Page fields table](#).
5. Click **Update**.

#### Admin Settings - Connections Page Field Descriptions

Field Names and Descriptions	Default Value
<b>Enable Proxy Authentication</b> —To require TeamConnect to perform certain connections through a proxy server, select the check-box.	Disabled (check-box cleared)
<b>Proxy Host Name / IP Address</b> —Enter the node name, or IP address, by which your proxy server is known on your network.	None
<b>Proxy Port Number</b> —Enter the port number on which your proxy server listens for requests.	None
<b>Proxy Username</b> —Enter the user name which authorizes your requests to the proxy server.	None

<b>Proxy Password</b> —Enter the password which authorizes your requests to the proxy server. Enter plain text. (The password is stored in a secure format in the database.)	None
<b>Proxy Realm</b> —Enter the name of the realm that is associated with the protected area, for example, yourcompanyname.com. The realm name should be ASCII characters, with no spaces or punctuation except a hyphen (allowed only in the middle of the realm name) or a period.	None
<b>Enable NTLM Authentication</b> —If you check this setting, all requests to the proxy server will use the username and password that you use to log in to your operating system domain. Any values in "Proxy Username" and "Proxy Password" will be ignored when NTLM is enabled.	Disabled (checkbox cleared)
<b>NT Domain</b> —If NTLM Authentication is selected, enter the name of the NT domain against which your client should be authenticated.	None

### 2.3.1.15 Admin Settings History

For any changes made to Admin Settings, you can record notes on the **History** page.

Date	Description	Category
9/23/16	When the action was completed the following information was changed in Dispute 'DISP-000000-Charles Dispute 1' An existing Category 'Dispute' was updated Matter Description was changed	audit
10/4/16	When the action was completed the following information was changed in Dispute 'DISP-000023-m_Dispute' Main Assignee was "" and was changed to 'kumar2, nars.h2 (Attorney)' A new Project Assignee 'QA, limited_N (Administrative Assistant)' was added	audit

Admin Settings - History Page

#### 2.3.1.15.1 Recording a History for Admin Settings

The following steps describe how to use the **History** page from Admin Settings. Field descriptions are provided in a table below.

#### To record a history for Admin Settings

1. Select the **Admin** tab.
2. Click the **Admin Settings** link under the tab bar.
3. Click **History** from the left pane.
4. Click **New History**.

5. Enter a change **Description**.
6. Click **Update**.

#### Admin Settings - History Page Field Descriptions

Field (* -required)	Description
<b>*Date</b>	This field is automatically populated with the current date.
<b>Time</b>	This field is automatically populated with the time that the new history was opened.
<b>*Description</b>	Enter a change description.

#### 2.3.1.16 About TeamConnect

You can find the TeamConnect version, latest run database scripts, CSM version, and SOP version (if applicable) from the **About** page in Admin Settings.

In addition, you can view the details of the license file. See [Viewing Information About Your Product License](#) for explanations of licensing concepts.

Instance Information			
Name: <input type="text"/>			
Installed Product	Installation Date	Installation Type	Description
TeamConnect Enterprise Version 5.0 UI/UX Build 1189			
TeamConnect Legal Matter Management 5.0 Build 21			
TeamConnect Financial Management 5.0 UI/UX Build 43			
TeamConnect Collaborati Spend Management (CSM) 5.0 UI/UX Build 26			
TeamConnect AP Export 1.2			
TeamConnect SOP Manager UI/UX Build 20			
TeamConnect Com pulaw Integration 2.1.0			
SUPPS87			
TeamConnect Data Warehouse 4.1			
TeamConnect Office and Outlook Add-Ins 2.0.2.6			

#### Admin Settings - About Page for Version and License Key Information

##### 2.3.1.16.1 Viewing TeamConnect Versions and License Information

The following steps describe how to use the **About** page from Admin Settings. Field descriptions are provided in a table below.

#### To view TeamConnect versions and license information

1. Select the **Admin** tab.
2. Click the **Admin Settings** link under the tab bar.
3. Click **About** from the left pane. All fields on the resulting page are read-only.



4. Click the + sign next to TeamConnect to view a history of database scripts that have been run for the instance.
5. Click the + sign next to CSM to view a history of CSM versions that have been installed.
6. (only if SOP is installed) Click the + sign next to SOP to view a history of SOP versions that have been installed.

#### Admin Settings, About Page Field Descriptions

Field	Description
<b>Installed Product</b>	Displays the product name and version. For database scripts, a TeamConnect version displays as a parent to related scripts.
<b>Installation Date</b>	Displays the date the product was installed.
<b>Installation Type</b>	For database scripts, displays SQL Script.
<b>Description</b>	For database scripts, displays the script name.

#### 2.3.1.16.2 Viewing Information About Available Updates

On the **About** page, the block titled **Available Updates** displays product updates that you can install, download, or request, based on the product versions that you presently have licensed. In addition, this block may display information about products that you do not presently have licensed, but which relate to your licensed products.

**Note:** To take advantage of the Available Updates feature, you must allow your application server to make HTTP requests to Mitrtech's server. Connection parameters are built into TeamConnect, but if you cannot connect, you may need to configure your firewall to allow connection. Contact Mitrtech Support to get specific connection information.

The fields and links in this block are described below.

#### Admin Settings, About page, Available Updates block

Field	Description
<b>Product</b>	Displays the product name of the update. Updates are grouped by product.
<b>Version</b>	Displays the specific version of the update.
<b>File Size</b>	For updates that can be downloaded immediately, the file size of the download is shown. For updates that must be requested from Mitrtech, <b>N/A</b> is displayed.

<b>Instructio ns</b>	Displays a <b>Read Me First</b> link which, when clicked, opens a new window showing release notes and other detailed information about the update. If an update does not have detailed information available from this page, the text <b>Available Upon Request</b> is displayed, indicating that you should contact your Mitratech support representative for more information.
<b>Install or Downloa d</b>	<p>Some updates can be applied to your application immediately, while it is being used. Such updates show an <b>Install Now</b> link in this field. When you click the link, and answer a confirmation window, the update is downloaded and is immediately applied to your application. The update will affect logged-in users immediately. You should perform the Install action at times when other users are not using the application.</p> <p>(Collaborati Spend Management can be updated through Install Now, but it will not take effect until you stop and restart TeamConnect.)</p> <p>Some updates are more complex and cannot be installed while users are logged in. These updates can be downloaded now and applied at a time when the application has been stopped. Such updates show a <b>Download Now</b> link in this field. When you click the link, the update is saved to a file location that you specify.</p> <p>Updates for products that you have not licensed will have a blank entry in this field.</p> <p>Updates which do not have downloads will have an <b>Available Upon Request</b> entry in this field.</p>

#### 2.3.1.16.3 Choosing Your Support Options

TeamConnect can send minimal information about your application to Mitratech. In addition, it can send more detailed information about your application's environment, which can aid Mitratech Support in resolving problems more quickly. Check-boxes allow you to grant permission to send:

- Basic product information
- More detailed environmental information

#### 2.3.1.16.4 Viewing Information About Your Product License

On the **About** page, the block titled **License** displays the terms and status of your TeamConnect product license. Your license determines how many users can use TeamConnect.

##### 2.3.1.16.4.1 Licensing Schemes

The types of licensing schemes include the following:

- **Total Users**—In this scheme, both normal users and limited-privilege users are counted together when determining the license limit.
- **Normal Users and Limited-Privilege Users**—In this scheme, there are separate limits stored for [normal users and limited-privilege users](#).

A value of **Site** in any limit field causes that portion of the license to be treated as unlimited.

For specific information about user types, see [User General Page Field Descriptions](#) or [Creating or Editing Users](#).

#### 2.3.1.16.4.2 License Style

The **License Style** field indicates how these limits are enforced. If its value is **NAMED**, then you are restricted in how many active users you can enter into the database, even if they do not all use the application simultaneously. You may create more users than the license limit specifies, but the additional users cannot be active.

If the License Style value is **CONCURRENT**, then you may enter more active users than the license limit specifies, but your license restricts how many of them can log in to the application concurrently.

**Note:** *The count of concurrent sessions is affected by the application server's handling of expired sessions. For example, in BEA WebLogic, the default setting for checking for expired sessions is every 60 seconds. So if you had the maximum number of concurrent users logged in, and one user's session timed out, another user wouldn't be able to log in until WebLogic cleaned up the expired session, which might be up to 60 seconds from the logout time. See WebLogic documentation for information about changing this setting (invalidation-interval-secs).*

If you use the licensing scheme "Normal Users and Limited-Privilege Users", then there will be two limit fields visible and you must specify a License Style value for each limit. The two styles do not have to be the same. For example, you may wish to enforce a NAMED style for the normal user limit and a CONCURRENT style for the limited-privilege user limit.

#### 2.3.1.16.4.3 Uploading a License File

**Note:** *The following procedure assumes that you obtained the license file from Mitrtech Support and saved it to an accessible location.*

##### To upload and activate the license file

1. Log in to TeamConnect.
2. Click the **Admin** tab, and then click the **Admin Settings** sub-tab link.
3. In the left pane, click the **About** link.
4. In the **License** section, click the **Upload License** button.
5. In the **Upload a new license file** window, browse to the location of the license file and select it.
6. Click **Upload**.

The license file is now activated and users can log in to TeamConnect.

If you see error messages indicating that you exceeded or are about to exceed the maximum number of active users, contact Mitratech Support for an upgraded license. See [License Style](#) for specific information about types of licenses.

## 2.3.2 TeamConnect Logs

### Loggers and Appenders

Use the following terminology for this section. In the logging process, loggers perform the logging operation and capture event message data. Different types of appenders can be defined to physically store the event message data. File appenders or log files are the most common appenders. In addition, appenders are organized in two different ways, by data content and by data format or distribution.

In the area of log data content, there are two types of logs in TeamConnect:

- **System appenders**—Record what TeamConnect is doing in the background by capturing messages generated by the system. This can be useful for debugging or when customizing the application.
- **Audit appenders**—Track user activity, which may be necessary to meet certain policies or compliance regulations regarding user activity.

### Division of Logging Functionality

TeamConnect logging features are divided between the Admin and Designer areas.

In the Admin area, you can:

- View file appenders (log files)
- Clear or delete file appenders (log files)
- Change logging severity levels (for loggers). For example, change from Error message tracking level to Debug message tracking level if you need additional details for troubleshooting a TeamConnect component.

In the Designer area, you can define new log appenders (log files or email notifications), edit existing appenders, or toggle audit loggers on/off.

**Note:** TeamConnect logging uses the log4j® API.

#### 2.3.2.1 Default Logs

By default, TeamConnect has several active file appenders (log files in text format). These files are stored in the location specified for log folders at the time when TeamConnect was installed. There are three default logging appenders:

- **Default** (for system logging)—Logs all of the messages that are generated by all of the system loggers and writes them to the **system.log** file.
- **XML Appender**—Logs messages regarding XML requests, according to the level selected for the XML logger, and writes them to the **xml.log** file.

- **Default** (for audit logging)—Logs messages from all active audit loggers and writes them to the **audit.log** file.

*Note: If Collaborati Spend Management is installed on your system, the **CSM Appender** is installed by default.*

For information about how to view the default logs, see [Viewing Logs](#).

### 2.3.2.2 Working with Logs

From the Admin Logging area, you can perform the following:

- [Viewing Logs](#)
- [Clearing Logs](#)
- [Changing System Logging Levels](#)
- [Turning Audit Loggers On or Off](#)

See [Appendixes](#) for additional information about creating and editing appenders.

#### 2.3.2.2.1 Viewing Logs

The system and audit logs generated by file appenders are stored on the TeamConnect server. You can view logs by:

- Clicking a **View Log** link from TeamConnect.
- Opening the file on the TeamConnect server.

#### To view a log file

1. Click the **Admin** tab.
2. Click the **Logging** link under the tab bar.
3. To view a system log file
  - Click the **System** link from the left pane.  
The System Logging page displays.
  - From the **System Appenders** section, locate the name of the log you want to view and click the corresponding **View Log** link.
4. To view an audit log file
  - Click the **Audit** link from the left pane.  
The **Audit Logging** page displays.
  - From the **Audit Appenders** section, locate the name of the log you want to view and click the corresponding **View Log** link.

5. The current log file for the log you selected is displayed in a new browser window. The most recent log events are listed at the bottom of the log file.

#### 2.3.2.2.1.1 Things to Remember

- If a log file is very large, clicking the **View Log** link might cause a lockup in your browser session as it tries to transfer the large amount of data from the server. If this is a concern, right-click the **View Log** link, then choose context menu item **Save Link As**, then save the log to a file on a local disk drive. Open that local file with your browser.
- To view the latest log messages, press F5 or click your browser's **Refresh** button.
- For both System Appenders and Audit Appenders, if TeamConnect is running on a cluster of application servers, a drop-down list to the left of the **View Log** link allows you to choose the specific node whose log you wish to view. Node names consist of a machine name, followed by a numeric process ID.
- The log location is specified using the app.logFolder parameter in the Web Application Deployment Descriptor when TeamConnect is installed.

### Default TeamConnect Log Folders

If you did not update the Log Folder field (app.logFolder parameter) when updating the teamconnect.properties file using the TeamConnect Installer, you can find TeamConnect logs in the following locations:

- For Apache Tomcat (Linux)—/ <Apache\_HOME>/bin/logs  
For example:  
/opt/tomcat/apache-tomcat-6.0.16/bin/logs
- For Apache Tomcat (Windows)—C:\ <Apache\_HOME>\logs  
For example:  
C:\Program Files\Apache Software Foundation\Tomcat 6.0\logs
- For WebLogic (Linux)—<WebLogic\_Home>/user\_projects/domains/<domain\_name>/logs  
For example:  
/opt/boa/user\_projects/domains/Mitrstech/logs
- For WebSphere—<WebSphere\_Home>/AppServer/profiles/<profileName>/logs  
For example, on Windows:  
C:\WAS61\WebSphere\AppServer\profiles\AppSrv02\logs\

#### 2.3.2.2.2 Clearing Logs

For troubleshooting or maintenance purposes, you can manually clear the contents of a log. Clearing a log is different from rolling over a log in that the existing contents of the log will be deleted and not archived to a back-up file.

For example, one scenario in which clearing a log is useful is when you are trying to isolate the log messages related to a specific functional issue. You might clear a log, change the related logger's logging severity level to DEBUG, and then try to reproduce the issue. Afterward, the log would most likely isolate the event messages that can help troubleshoot the issue.

##### To clear a log file

1. Click the **Admin** tab.
2. Click the **Logging** link under the tab bar.
3. To clear a system log file
  - Click the **System** link from the left pane.  
The **System Logging** page opens.
  - From the **System Appenders** section, locate the name of the log you want to clear and click the corresponding **Clear Log** link.
4. To clear an audit log file
  - Click the **Audit** link from the left pane.  
The **Audit Logging** page opens.
  - From the **Audit Appenders** section, locate the name of the log you want to clear and click the corresponding **Clear Log** link.
5. To view any new log messages, click the corresponding **View Log** hyperlink.

#### 2.3.2.2.3 Changing System Logging Levels

Each message that can be written to a log has an associated log level that indicates the severity of the issue that caused the message to be written to the log. You can specify what severity of messages you want logged for each logger (or TeamConnect component).

**Note:** *This is a system-wide setting. When you set a logger to a specific log level, you are specifying the most verbose severity level of messages that you want written to appenders defined for that logger. Although there is an [analogous setting in the appender definition](#) for log threshold, the logger's logging level overrides the log threshold.*

##### To set the system logging level

1. Click the **Admin** tab.
2. Click the **Logging** link under the tab bar.

3. To change a system logger's logging level
  - Click the **System** link from the left pane.
  - From the **System Loggers** or **Detailed System Loggers** section, scroll to a logger. From the corresponding Level drop-down list, select a logging level:
    - **Debug**—Minor, frequent events in TeamConnect.
    - **Info**—Significant, frequent events in TeamConnect.
    - **Warn**—Minor issues external to TeamConnect.
    - **Error**—Serious issues that are usually associated with a Java exception.
    - **Fatal**—Rarely used in TeamConnect.

By default, system logger levels are set to **Error**, but you can adjust the severity based on the type of information you need to be captured in your logs.

See [System Loggers](#) and [Detailed System Loggers](#) for a description of the system loggers that come with standard TeamConnect.

4. Click **Update**.

Your changes take effect immediately. All appenders defined for these loggers are now logging messages according to the log level that you have specified, and according to each appender's threshold.

#### System Loggers

Logger	Description
<b>Request</b>	<p>Logs HTTP request information including headers, cookies, and parameters.</p> <p>This logger can automatically be set to the Debug level each time TeamConnect application is restarted. This depends on the value of the syslog.basic.request parameter in the Web Application Deployment Descriptor.</p>
<b>Response</b>	<p>Logs HTTP response header information.</p>
<b>Rule</b>	<p>Logs messages generated by rules that have been defined for your implementation of TeamConnect.</p>
<b>SQL</b>	<p>Logs TopLink-generated SQL statements.</p> <p>This logger can automatically be set to the Debug level each time TeamConnect application is restarted. This depends on the value of the syslog.basic.sql parameter in the Web Application Deployment Descriptor.</p>



<b>XML</b>	Logs messages regarding XML requests.
------------	---------------------------------------

**Detailed System Loggers**

<b>Logger</b>	<b>Description</b>
<b>Authentication</b>	Logs all messages regarding user authentication.
<b>Batch Display Component</b>	Logs all messages regarding the batch display component.
<b>Custom Blocks</b>	Logs all messages regarding custom blocks.
<b>Deletion</b>	Logs all messages regarding record deletion.
<b>IMAP</b>	Logs events related to the TeamConnect IMAP Server and Email Viewer.
<b>Portal</b>	Logs all messages regarding portal panes, including custom content.
<b>Reflection</b>	Logs all messages generated by TeamConnect's reflection factory.
<b>Search</b>	Logs all messages regarding search views and searching for records.
<b>Stores</b>	Logs all messages regarding TeamConnect stores.
<b>Sync Server</b>	Logs all messages regarding cache synchronization.
<b>Tools</b>	Logs all messages regarding both system and custom tools.
<b>UI Components</b>	Logs all messages regarding rendering of TeamConnect screens.
<b>WebDAV</b>	Logs all messages related to WebDAV document management.

## 2.3.2.2.4 Turning Audit Loggers On or Off

Audit loggers send their messages to logs defined by audit appenders. Audit loggers are turned on or off system-wide. You do not have to set levels for audit loggers.

You can turn audit loggers on or off as needed. When turned on, the logger's messages immediately start getting logged for all defined audit appenders.

**To turn audit loggers on or off**

1. Click the **Admin** tab.
2. Click the **Logging** link under the tab bar.
3. Click the **Audit** link from the left pane.  
The **Audit Logging** page opens.
4. From the **Audit Loggers** section, locate the name of the log you want to turn on and select **On** from the drop-down list.
5. Locate the name of the log you want to turn off and select **Off** from the drop-down list.
6. For descriptions of available audit loggers, see the following table.

#### Audit Loggers

Logger	Description
<b>API Read</b>	Turn on this logger to log read calls to the API.
<b>API Update</b>	Turn on this logger to log update calls to the API.
<b>Page Open</b>	Turn on this logger to log the opening of a page by a user.
<b>Login/Sign Off</b>	Turn on this logger to log user sign on/off events.
<b>Export/Download</b>	Turn on this logger to log PDF exports or downloads from users.

### Example Results for Users

For example, if **Internal Time** is selected in the **Budget for** section and **Annual+Month** is selected in the **Extend budget level** by section, the following corresponding accounts are created for each dispute or transaction record:

- Internal Fees
- Internal Fees : Annual (one for each fiscal year)
- Internal Fees : Monthly (twelve for each fiscal year)

### 2.3.3 TeamConnect Administrator Tools

Several administration functions are performed by starting the related tool from the Tools page (accessed from **All Services, Tools**.)

### 2.3.3.1 Scheduled Actions Monitor

Use this tool to observe and control actions that are scheduled to execute at specific times in the future.

**Note:** The right to this tool must be assigned before the tool is visible to a user. The right is found under System Rights, User Rights, Scheduled Action.

All actions pending execution that have been scheduled by Scheduled Action rules, or by Custom Action rules that utilize the Scheduled Actions API, automatically get listed in the Scheduled Actions Monitor. You can use this page to view and delete pending actions when necessary.

For example, the system's Scheduled Action rules may be set to automatically create tasks for a user, but those tasks might no longer need to be created because of other circumstances. As the Administrator, you can delete the scheduled actions that are going to eventually create these tasks on the scheduled dates.

You can also create entirely new scheduled actions on an as-needed basis.

#### 2.3.3.1.1 Accessing the Scheduled Actions Monitor

To access the Scheduled Actions Monitor, select **Scheduled Actions** from the **Tools** link. The Scheduled Actions Monitor opens, as shown in the following image. You must have the appropriate rights to access and view the Scheduled Actions Monitor.

Scheduled Actions

Scheduled Actions

Name:  Repeat:

Action:  [View Scheduled Actions folder](#)

Start Date:

Start Time:

Scheduled Actions 1 - 4 of 4

<input type="checkbox"/>	Next Run Time	Name	Action	Record and Rule	Repeat	End	Previous Run Time	Status
<input type="checkbox"/>	10/6/16 9:10 AM	ok	TestDeleteScheduledAction.class		Hours and Minutes (1 Hours and 5 Minutes)	No End Date	10/6/16 8:05 AM (1s)	<input type="button" value="Run Now"/>
<input type="checkbox"/>	10/6/16 9:53 AM	defect regression test	TestDeleteScheduledAction.class		Hours and Minutes (1 Hours)	No End Date	10/6/16 8:53 AM (1s)	<input type="button" value="Run Now"/>
<input type="checkbox"/>	10/6/16 9:53 AM		CompuLawScheduledAction.class		Hours and Minutes (1 Hours)	No End Date	10/6/16 8:53 AM	<input type="button" value="Run Now"/>
<input type="checkbox"/>	10/6/16 1:56 PM		BudgetEmailReminderScheduledAction.class		Hours and Minutes (24 Hours)	No End Date	10/5/16 1:56 PM	<input type="button" value="Run Now"/>

Scheduled Actions per page:

Startup Actions

Click the button to run all Startup Actions now.

**Scheduled Actions Monitor Screen**

You can refresh the information on this screen at any time by using your browser's **Refresh** button.

## 2.3.3.1.2 Tasks Available from the Scheduled Actions Monitor

From this page you can do any of the following tasks:

- [Create a New Scheduled Action](#)
- [Actions Against the Table](#)
- [Activate Scheduled Actions](#)
- [Delete Scheduled Actions](#)
- [Run a Scheduled Action](#)

## 2.3.3.1.2.1 Create a New Scheduled Action

Create a new scheduled action by entering values in the fields at the top of the page, then clicking **Add**. The new scheduled action is in active status and will run at the time you specified.

The following table describes these fields.

**Scheduled Actions Monitor - Creating a New Scheduled Action**

Field or column	Description
<b>Name</b>	Enter a name that describes the action.
<b>Repeat</b>	<p>If this action will execute only once, keep the default value of <b>None</b> in this drop-down list. If this action will be executed repeatedly, choose the time interval from the list of choices.</p> <p>Specify the period of time between rule action repetitions after the initial execution after the rule has been triggered. Additional options will appear depending on the selection in the <b>Repeat</b> drop-down list.</p> <ul style="list-style-type: none"> <li>• <b>Hours and Minutes</b>—The scheduled action will repeat after the specified number of <b>Hours</b> and <b>Minutes</b> have passed.</li> <li>• <b>Daily</b>—The scheduled action will repeat every day. Select the <b>Weekdays only</b> check-box to set the rule action to be repeated only from Monday through Friday.</li> <li>• <b>Weekly</b>—The scheduled action will repeat every week. Use the <b>Day</b> drop-down list to select on which day of the week the action will be repeated.</li> <li>• <b>Monthly</b>—The scheduled action will repeat every month. Use the <b>Day of the Month</b> drop-down list to select on which day of the month the action will be repeated, from <b>1</b> to <b>31</b>, or <b>Last</b> for the last day of the month.</li> <li>• <b>Yearly</b>—The scheduled action will repeat every year. Use the <b>Day and Month</b> drop-down lists to select on which month and day of the year the action will be repeated.</li> </ul>

<b>End</b>	<p>Specify when the scheduled action repetitions will stop.</p> <ul style="list-style-type: none"> <li>• <b>No End Date</b>—The scheduled action will repeat indefinitely until the rule is deleted or inactivated. This is the default option.</li> <li>• <b>After</b>—The scheduled action will execute for the specified number of repetitions.</li> <li>• <b>Until</b>—The scheduled action will repeat until the specified date.</li> </ul>
<b>Action</b>	<p>This drop-down list contains all the executable actions that have been uploaded to the <b>Top Level / System / Scheduled Actions</b> folder in Documents. Choose which action (usually a Java class file) should be associated with the new scheduled action that you are creating. The link <b>View Scheduled Actions Folder</b> opens this folder in Documents so that you can work with its contents, if necessary.</p>
<b>Start Date and Start Time</b>	<p>These two fields specify the first time that the new scheduled action should be run. For actions that are repeated, this time becomes the basis for all the future repetition intervals. If you enter a date and time that are in the past, and this is not a repeating action, an error message is shown.</p>

**Important:** Actions that are scheduled to repeat monthly on the 29th, 30th, or 31st day of the month will not occur in months that do not have those dates. To avoid this issue, use the **Last** option for the last day of the month.

#### 2.3.3.1.2.2 Actions Against the Table

The table in the lower portion of the Monitor shows which scheduled actions have run and which are pending. You can take several actions by using this table and its adjacent buttons. Here are the elements of the table:

##### Scheduled Actions Monitor Screen

Field or column	Description
<b>Next Run Time</b>	<p>All actions have a next run time, except startup actions and actions that have been deactivated.</p> <p>If something prevents an action from running at the scheduled time - for example, if TeamConnect itself has an outage - this time will remain unchanged.</p>
<b>Name</b>	This is an optional, user-friendly name that describes the action.
<b>Action</b>	This is the actual file that will be run when the action runs.

<b>Record and Rule</b>	Many scheduled actions are the result of a scheduled-action rule running against a specific record. In this case, the name of the record appears here as a hyperlink. Clicking the name will open the record. The name of the scheduled-action rule also appears here.
<b>Repeat</b>	Indicates the time interval in which the action is scheduled to repeat. If it is a one-time action, this field contains <b>none</b> .
<b>End</b>	For repeating actions, describes the condition under which the repetitions will end. See "Repeat" in <a href="#">the Scheduled Actions Monitor table</a> for details.
<b>Previous Run Time</b>	Shows the last date/time this action was run, and duration of the run. If an action failed during its last run, a failure message is shown here. For one-time actions, this field will always be empty.
<b>Status</b>	<b>Inactive</b> for actions that have been deactivated. <b>Running</b> for actions that are currently running. Otherwise, the <b>Run Now</b> button appears.  Clicking the <b>Run Now</b> button causes the action to run immediately. Such a run does not count towards the number of repetitions before the end, if a number had been configured for this action.

#### 2.3.3.1.2.3 Deactivate Scheduled Actions

If you deactivate a scheduled action, it remains pending but does not execute.

##### To deactivate a scheduled action

1. Select the check-box next to the scheduled action that you want to deactivate. You may select more than one check-box at once.
2. Click **Deactivate**.

The selected scheduled action(s) is deactivated and the Monitor refreshes to reflect this. The value in field **Status** now says "Inactive".

#### 2.3.3.1.2.4 Activate Scheduled Actions

If you activate a scheduled action that had been inactive, it will run at its next scheduled date and time. If that date and time are already past, it will run immediately.

##### To activate a scheduled action

1. Select the check-box next to the scheduled action that you want to activate. You may select more than one check-box at once.

2. Click **Activate**.

The selected scheduled action(s) is activated and the Monitor refreshes to reflect this.

#### 2.3.3.1.2.5 Delete Scheduled Actions

If you have the appropriate rights, you can delete scheduled actions to prevent them from being executed.

##### To delete a scheduled action

1. Select the check-box next to the scheduled action that you want to delete. You may select more than one check-box at once.
2. Click **Delete** and confirm that you want to delete the selected scheduled action.

The selected scheduled action(s) is deleted and the Monitor refreshes to reflect this.

#### 2.3.3.1.2.6 Run a Scheduled Action

The **Run Now** button, if visible in the Status field for a scheduled action, can be clicked to cause the action to run immediately, notwithstanding its next scheduled run date. This button is not visible if the action has been deactivated.

#### 2.3.3.1.3 Things to Remember

If a scheduled action is configured to end after a certain number of repetitions, that number is decremented each time the action is run. Even if the action fails due to an internal error or some external reason (e.g., database server is down), the number is decremented. However, running the action via the **Run Now** button does not decrement the number of repetitions remaining.

If an action is scheduled to run at a certain time interval, but then is deactivated, and is not re-activated until after that next scheduled run time, the action will run immediately. It will then revert to its regularly scheduled time interval.

## 2.4 Authentication

Authentication is the process of securely identifying a user who is requesting a connection to TeamConnect. TeamConnect contains a facility for standard authentication, in which the user name and password are stored within TeamConnect itself. In addition, TeamConnect provides two plug-ins to accommodate two other popular means of custom authentication: NTLM and LDAP.

### 2.4.1 Login URLs

TeamConnect has two types of login URLs.

Login URL	Description
-----------	-------------

<b>System Default</b>	<p>Give users the System Default login URL if you have selected <b>System Default</b> authentication in their <b>User Account</b> screen settings. This is the default login URL.</p> <p>If you have not installed any custom authentication mechanisms into TeamConnect, give this login URL to your users.</p> <p>The URL takes the form:</p> <pre>http://[host]/[context path]/login</pre> <p>For example:</p> <pre>http://tcserver/tc240/login</pre>
<b>Standard TeamConnect</b>	<p>Give users the Standard TeamConnect login URL if you have selected <b>Standard TeamConnect</b> authentication in their <b>User Account</b> screen settings. If you have installed custom authentication mechanisms into TeamConnect, give this login URL only to users who are not authenticated by those custom mechanisms.</p> <p>The URL takes the form:</p> <pre>http://[host][:port]/[context path]/standardLogin</pre> <p>For example:</p> <pre>http://10.0.1.72:5001/tc240/standardLogin</pre>

The **TeamConnectAdmin** account always uses the standard login URL.

The user who administers the Data Warehouse feature must log in with standard authentication.

For details on authentication and its effect on logging in to TeamConnect, see [Authenticating Yourself](#).

## 2.4.2 NTLM Authentication Plug-in

If you use TeamConnect with a Software as a Service (SaaS) license, you do not need to perform any of the tasks described in this section, except for [Enabling NTLM Authentication for TeamConnect Users](#).

This document provides the instructions to install and configure TeamConnect to use the NTLM Authentication plug-in to enable single sign-on and user authentication. For other general information about the plug-in, including the software prerequisites for running it, read the release notes.

The NTLM authentication plug-in allows TeamConnect to authenticate a user through a challenge/response system in conjunction with Microsoft® Internet Information Services (IIS). When authenticated through NTLM, the user no longer has to submit separate authentication credentials, such as username and password, to access TeamConnect.

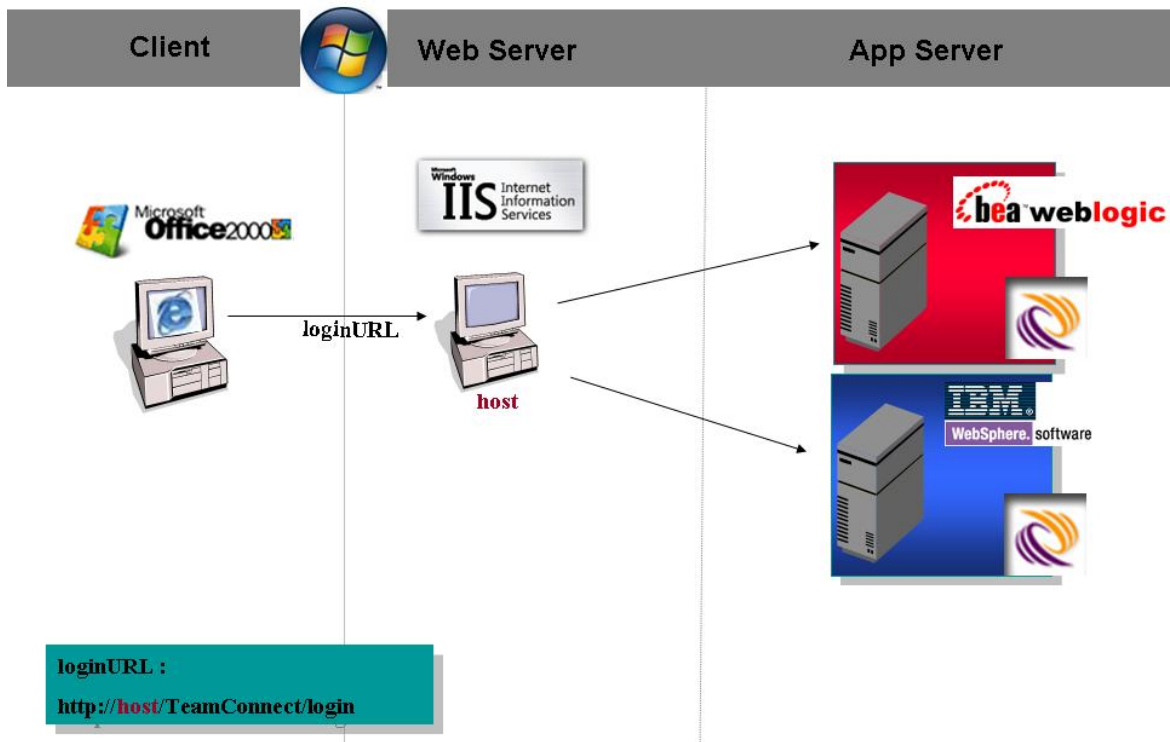
When the NTLM authentication plug-in is enabled, users who are successfully authenticated bypass the TeamConnect login page and go directly to their TeamConnect home page. Aside from the lack of a login page, there is no change to the user interface.

If the user account fails authentication through NTLM a denial page is displayed. When troubleshooting denials, verify that the login user ID corresponds to a valid user account name within



TeamConnect and that the appropriate domain is specified in the System Settings page in the Admin section of TeamConnect.

Shown below is a diagram of the general concept of NTLM authentication as it affects the application server and web server in TeamConnect:



Conceptual View of NTLM Authentication

#### 2.4.2.1 NTLM Prerequisites

The following requirements must be met in order to use the NTLM authentication plug-in for TeamConnect:

- Your TeamConnect environment must be a pure Microsoft solutions environment using Internet Information Services (IIS). The environment must include an application server that is compatible with IIS. For specific versions of IIS and application servers that are supported, refer to Application and Web Servers.
- Within TeamConnect, you must specify a list of one or more domains in which to authenticate all users. NTLM automatically authenticates users based on their Windows login name and domain. You can, however, change the behavior of the default plug-in.
- Each user that is authenticated through NTLM must have a corresponding user name defined in TeamConnect.

### 2.4.2.2 Configuring Internet Information Services

The process below describes the sequence of configuration requirement tasks:

- Configure Microsoft IIS plug-in to disable anonymous access.
- Customize Message Screens.
- Upload the NTLM authentication configuration class files to the Documents area of TeamConnect.
- Enable NTLM authentication through the **Admin Settings/Security** page of TeamConnect.
- Configure TeamConnect user accounts for using NTLM authentication.

It also provides two URLs that can be used for logging in and explains the difference between them.

The following sections explain each task in the order in which you should perform them.

#### 2.4.2.2.1 Disabling Anonymous Access

In order to use NTLM authentication you must disable anonymous access in IIS. When anonymous access has been disabled, IIS looks for other means of authentication and finds NTLM authentication.

**Note:** *The following steps are for Windows XP Professional. If you are using another version of Windows, these steps might differ.*

#### To disable anonymous access to IIS

1. On the computer running your web server, click **Start > Programs > Administrative Tools > Internet Services Manager**.
2. In the left pane, open **Web Sites**, click **Default Web Site**, and then be certain that the web site is stopped. Click the **Stop** button in the toolbar, if necessary.
3. Right-click **Default Web Site**, and then click **Properties**.
4. Click the **Directory Security** tab.
5. In the **Anonymous Access and Authentication Control** section, click **Edit**.
6. Clear the **Anonymous Access** check box. If a checkbox named **Integrated Windows authentication** is present, be sure that it is presently checked. Then click **OK**.
7. Click **OK**.
8. With **Default Web Site** highlighted, click the **Start** button in the toolbar to start the web site.
9. Close the Internet Information Services console.

**Important:** *You must also configure the proxy plug-in for IIS. For general information about doing this, see [Web Proxy Settings](#).*

Stop and restart IIS when these steps are completed.

### 2.4.2.3 Editing the NTLM Configuration File

The configuration file, **authenticationDescriptor.properties**, needs to be edited to match your particular network environment. Shown below is the default text of this file.

```
## This is a sample properties file for NTLM authentication. You can
tailor it to your specific
## situation to work with the NTLM authentication mechanism included
within TeamConnect. All
## un-commented properties in this file are required, though their
values can be changed.
# Identification properties for this authentication component
tc.displayName=NTLM based Single Sign On
tc.uniqueId=SNTLM
tc.isSSO=true
# Custom page mappings (omit any that do not apply)
#page.badCredentials=tryagain.html
#page.lockedAccount=giveup.html
#page.maximumLogins=sorry.html
#page.logout=goodbye.html
#page.sessionTimeout=missedyou.html
# NTLM configuration properties
#
# The NT domain against which clients should be authenticated.
ntlm.tcDomain=TCDOMAIN
ntlm.remoteUserHeader=
```

The value of **ntlm.tcDomain** can be a single domain name or it can be a list of domain names, delimited by semicolons. The domain portion of a user's login name (such as "terry@mycompany.com") is matched against this property value and, if a matching domain is not found, authentication fails.

If you expect to override the default error message pages, modify the "page mappings" section to supply your own HTML file names.

Property **ntlm.remoteUserHeader** must exist in the configuration file, but it does not always need to have a value associated with it. You may find that for WebLogic application servers, a value of `Proxy-Remote-User` works best and, for WebSphere application servers, a value of `$WSRU` works best.

Note that even if you do not use properties **page.logout** and **page.sessionTimeout**, TeamConnect will construct a logout page on the fly to use in these situations, to avoid redisplaying the login page, which would cause an automatic re-login with NTLM.

### 2.4.2.4 Installing the NTLM Configuration File

NTLM authentication plug-in installation consists of a configuration file that must be uploaded to a specific directory within the **Documents** area of TeamConnect: **authenticationDescriptor.properties**

You must edit the **authenticationDescriptor.properties** file, as described in [Editing the NTLM Configuration File](#).

**To install the NTLM Authentication configuration file**

1. In the TeamConnect tab bar, select **Documents**.
2. Navigate to **Top Level**, then **System**.
3. Create a folder named **Authentication**.
4. In the **Authentication** folder that you just created, create a sub-folder named **NTLM**.
5. In the **NTLM** folder that you just created, create the following sub-folders:

**Caution:** The folder names must be entered exactly as they appear below.

- **classes**
  - **pages**
6. In the **classes** folder that you just created, upload this file: authenticationDescriptor.properties
  7. The **pages** folder does not require any file uploads, unless you choose to override the default error message pages, using the page.xxx properties in the configuration file. If so, you would upload your matching custom error message pages to this folder.

The NTLM Authentication plug-in is now installed.

**2.4.2.5 Enabling NTLM Authentication for TeamConnect Users**

Before assigning authentication methods to individual users, be sure that NTLM has been chosen as the default authentication mechanism. To do so, refer to [Enabling a Default Authentication Method](#).

To enable other users to log in to TeamConnect using NTLM authentication, each user account must be properly configured except for **TeamConnectAdmin**. However, if you followed the directions in [Enabling a Default Authentication Method](#), NTLM is now the system default authentication mechanism, and users who are already set to use the default need no editing. You need to perform overrides only:

- For users who may have previously been overridden with a different authentication method and are now using NTLM. In this case you can set their method to **(System Default)**.
- For users who are not using NTLM.

For details on performing an override, see [Overriding Authentication for Individual Users](#).

**Important:** For user "TeamConnectAdmin", do not attempt to use NTLM authentication. Force this user to go through standard authentication by appending **/standardLogin** to the end of the URL that launches TeamConnect.

**Important:** When using Web Services and NTLM to connect to TeamConnect, be sure that the URL in your NTLM configuration points to the TeamConnect application server, not the Internet Information Server.

For information about which login URL to give your users based on the authentication mechanism, see [Login URLs](#).

### 2.4.3 LDAP Authentication Plug-in

By default, TeamConnect authenticates each user by comparing a username and password entered in the login screen with information stored in the TeamConnect database. This is a safe and secure method of authentication, but your organization might choose to integrate the TeamConnect authentication process with a network-wide authentication mechanism, such as an LDAP (Lightweight Directory Access Protocol) server.

You can integrate any version of TeamConnect 3.0 and higher with LDAP by uploading a configuration file to TeamConnect's **Documents** area. Once this is done correctly, the LDAP authentication plug-in is available on the **Authentication Plug-In** block of the **Admin Settings/ Security** page.

TeamConnect also provides the ability for you to enable LDAP authentication per user through a setting on the user's account.

**Note:** *TeamConnect's integration with LDAP for authentication does not provide single-sign-on (SSO) functionality to TeamConnect. Each user must sign on to TeamConnect, and then sign on to other applications.*

For additional assistance, such as customization for your LDAP directory structure or authentication using password comparison, contact Mitratech Client Services. In addition, if your server is using password encryption or an SSL connection, certain changes might be required in the plug-in. These modifications can be made by Mitratech.

This guide does not cover managing access rights in TeamConnect. You must define group rights to TeamConnect objects and tools directly in TeamConnect. TeamConnect is not compatible with LDAP Access Control Lists (ACL).

As with other applications to which users gain access through your LDAP server, users can log in to TeamConnect only when the LDAP server is available.

#### 2.4.3.1 LDAP Authentication Prerequisites

The process of configuring TeamConnect for LDAP integration includes the following basic steps:

1. [Creating Custom LDAP Authentication Error Pages](#)—(optional) Create custom HTML pages for use with LDAP authentication.
2. [Defining LDAP Parameters](#)—Enter information in TeamConnect's LDAP properties file.
3. [Installing LDAP Plug-in Files](#)—Upload a TeamConnect LDAP properties file to the TeamConnect Documents area. You can also upload customized authentication error messages.
4. [Enabling a Default Authentication Method](#)—Enable the LDAP plug-in from TeamConnect Admin Settings.
5. [Overriding Authentication for Individual Users](#)—Allows you to enable LDAP authentication for individual user accounts. This step is only required if you do not select LDAP as the default authentication mechanism in the previous step.

### 2.4.3.2 Creating Custom LDAP Authentication Error Pages

You have the option to create customized screens for logging into TeamConnect using LDAP authentication and related error messages. To enable custom error pages, you must do the following:

1. Create custom error pages.
2. [Update the LDAP authenticationDescriptor.properties](#) file properties for the new pages.
3. Upload the custom pages to the TeamConnect Documents area folder:  
     \Top Level\System\Authentication\LDAP\pages  
     See [Installing LDAP Plug-in Files](#) for more information.

**Note:** *If you choose not to create customized error and login pages then you should comment out related lines from the LDAP authenticationConfiguration.properties file.*

## Requirements

The custom pages should be in HTML format. The custom page file extensions should be .html

If any custom pages require images, those images should be uploaded to the TeamConnect Documents area

## Custom Error Pages List

A list of pages that can be customized follows. Each page has a corresponding property in the LDAP **authenticationDescriptor.properties** file whose value should match the file name you create per page. The corresponding properties are described in [Defining LDAP Parameters](#).

- Authentication Failed page
- Normal Logout page
- Session Timeout page
- Restore Session page
- Bad Credentials page
- Account Locked page

### 2.4.3.3 Defining LDAP Parameters

To enable TeamConnect to use your LDAP plug-in, you must provide certain parameters which represent your environment and save them in a properties file. You must modify the **authenticationDescriptor.properties** file for LDAP authentication and enter the appropriate information in these parameters according to the descriptions in the table below.

**Note:** *To comment out a property, type # at the beginning of the line to comment it out.*

LDAP Parameters and Definitions

Parameter	Description	Example
-----------	-------------	---------

<b>tc.displayName</b>	Required field. Do not change this value.	
<b>tc.uniqueid</b>	Required field. Do not change this value.	
<b>tc.isSSO</b>	Required field. Do not change this value.	
<b>ldap.host</b>	LDAP server host name.  This value is combined with ldap.port and ldap.root to form the ldap-server URL.	10.0.0.1  <b>Note:</b> When combined with ldap.port and ldap.root values, an ldap-server URL like ldap://10.0.0.1:389/dc=myserver,dc=org  would result
<b>ldap.port</b>	LDAP server port.  This value is combined with ldap.host and ldap.root to form the ldap-server URL.	389  <b>Note:</b> When combined with ldap.host and ldap.root values, an ldap-server URL like ldap://10.0.0.1:389/dc=myserver,dc=org  would result
<b>ldap.root</b>	Enter the top-level DN (distinguished name) under which the TeamConnect users can be found in the LDAP directory structure.  This value is combined with ldap.host and ldap.port to form the ldap-server URL.	dc=springframework,dc=org  <b>Note:</b> When combined with ldap.host and ldap.port values, an ldap-server URL like ldap://10.0.0.1:389/dc=myserver,dc=org  would result
<b>ldap.userName</b>	Enter the name of the attribute that stores the username on the LDAP server.  TeamConnect will log into the LDAP server with this username.	directory.username=uid

	<p><b>Note:</b> Do not confuse the attribute name with the actual value of the username that it stores.</p>	
<b>ldap.password</b>	<p>Enter the name of the attribute that stores the password on the LDAP server.</p> <p>TeamConnect will log into the LDAP server with this password.</p> <p><b>Note:</b> Do not confuse the attribute name with the actual value of the password that it stores.</p>	<pre>directory.password=userPassword</pre>
<b>ldap.userDnPatterns</b>	<p>The elements of the user's Distinguished Name that, when assembled, will form the complete DN to authenticate the user's identity to the LDAP Directory Service (if authentication is required).</p>	<pre>uid={0},ou=people</pre> <p><b>Note:</b> This example would obtain the DN for the user by substituting the user login name in the supplied pattern and attempting to bind as that user with the login password. This would work if all your users are stored under a single node in the directory.</p>
<b>ldap.groupDnPattern</b>	<p>The group's Distinguished Name. For authentication, users will need to belong to this group.</p> <p>You must create and configure this group on the LDAP server. Then add users to this group.</p>	<pre>teamconnect</pre>
<b>ldap.groupRoleAttribute</b>	<p>The attribute which contains the name of the authority defined by the group entry. This parameter is used to load authorities from groups in the LDAP directory.</p>	<p>The default value is cn</p>
<b>page.badCredentials</b>	<p>Page to display if the password does not match the password stored for the TeamConnect user account.</p>	<pre>/login.html?error=bc</pre>



	<b>Note:</b> If you do not create a custom file, then comment out this line.	
<b>page.lockedAccount</b>	Page to display if the TeamConnect user account is locked and User Locking is enabled in TeamConnect settings.  <b>Note:</b> If you do not create a custom file, then comment out this line.	/login.html?error=1
<b>page.maximumLogins</b>	Page to display if the LDAP account has already exceeded their maximum allowed sessions.  <b>Note:</b> If you do not create a custom file, then comment out this line.	/login.html?error=1
<b>page.logout</b>	This page is shown when a user logs out normally.  <b>Note:</b> If you do not create a custom file, then comment out this line.	/login.html
<b>page.sessionTimeout</b>	This page is shown when a user's session times out.  <b>Note:</b> If you do not create a custom file, then comment out this line.	login.html?error=2

### To define LDAP parameters in the properties file

1. Copy the LDAP **authenticationDescriptor.properties** file from the TeamConnect Installer installation path (subfolder utilities\config\ldap) to your local hard disk.

**Tip:** When editing a properties file, use a text-only editor to ensure no formatting codes get inserted by the application.

2. Edit the **authenticationDescriptor.properties** file according to your environment. For details about each parameter, see [the LDAP Parameters and Definitions table](#).
3. Save and close this file.
4. Make note of the location of this modified file so that you can use it in the next procedure.

Now that you have configured your LDAP plug-in, you must upload a file into TeamConnect, as described in [Installing LDAP Plug-in Files](#).

#### 2.4.3.4 Installing LDAP Plug-in Files

This section describes:

1. Creation of TeamConnect Documents area folders for LDAP files
2. Uploading LDAP configuration files for TeamConnect
3. Uploading custom LDAP screens

LDAP authentication plug-in installation consists of a configuration file, **authenticationDescriptor.properties**, that must be uploaded to a specific directory within the **Documents** area of TeamConnect:

You *must* edit the **authenticationDescriptor.properties** file, as described in [Defining LDAP Parameters](#).

##### To install the LDAP Authentication configuration file

1. In the TeamConnect tab bar, select **Documents**.
2. Navigate to **Top Level**, then **System**.
3. Create a folder named **Authentication**.
4. In the **Authentication** folder that you just created, create a sub-folder named **LDAP**.
5. In the **LDAP** folder that you just created, create the following sub-folders:

**Caution:** The folder names must be entered exactly as they appear below.

- **classes**
  - **pages**
6. In the **classes** folder that you just created, upload this file:  
**authenticationDescriptor.properties**
  7. The **pages** folder does not require any file uploads, unless you choose to override the default error message pages. using the page.xxx properties in the configuration file. If so, you would upload your matching custom error message pages to this folder.

The LDAP Authentication plug-in is now installed.

To complete the implementation of LDAP, make sure that you are using the correct default authentication method, as described in [Enabling a Default Authentication Method](#). If LDAP is not your default authentication method, override individual users who need LDAP, as described in [Overriding Authentication for Individual Users](#). If LDAP is your default authentication method, you need to perform overrides only:

- For users who may have previously been overridden with a different authentication method and are now using LDAP. In this case you can set their method to **(System Default)**.
- For users who are not using LDAP.

Also, user **TeamConnectAdmin** should always use the Standard Authentication method. You can force this behavior by appending **/standardLogin** to the end of the URL that launches TeamConnect for that user.

### 2.4.3.5 LDAP Authentication Sample Files

This section provides sample files described in this guide.

- [Sample Custom Log-in Page](#)
- [Sample Custom Error Page](#)
- [Sample Restore Session Page](#)
- [Sample LDAP Configuration File](#)

#### 2.4.3.5.1 Sample Custom Log-in Page

**Note:** You do not need to customize the TeamConnect login page for LDAP Authentication. If you customize the login page, however, you must reference the `j_spring_security_check` servlet, the user field input value should correspond to the `j_username` variable, the password field input value should correspond to the `j_password` variable as shown in the example below.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Log In - TeamConnect 3.0</title>
</head>
<body scroll="no">
  <div id="horizon" style="margin-top:150px">
    <div id="content">
      <div id="contentText">
        <center>
          <table class="logincontent" border="0" width="600" cellpadding="0"
            cellspacing="0">
            <tr>
              <td style="background-image: url(images/bg_login.jpg); background-
                repeat: no-repeat;">
                <!-- LOGIN TABLE [BEGIN] -->
                <table border="0" width="600" cellpadding="0" cellspacing="0">
                  <tr>
                    <td colspan="2" style="line-height: 1.5em;" align="center">

```

```
</tr>
<tr>
  <td height="150" align="center">
    <form method="post" id="loginForm" name="loginForm"
      action="j_spring_security_check">
      <table border="0" cellpadding="5" cellspacing="0"
        style="margin: 10px;">
        <tr valign="middle">
          <td width="90" align="right" nowrap><b>User Name:</b></td>
          <td align="left"><input type="text" name="j_username"
            id="j_username" size="18" maxLength=50 tabIndex="1"/></td>
        </tr>
        <tr valign="middle">
          <td align="right" nowrap><b>Password:</b></td>
          <td align="left"><input autocomplete="off"
            type="password" name="j_password" id="j_password"
            size="18" maxLength=50 tabIndex="2"/></td>
        </tr>
        <tr valign="middle">
          <td align="right" nowrap>&nbsp;</td>
          <td colspan="2" align="left" class="texthdr"
            nowrap><input type="submit" name="submit" value="Log
            In" class="mainSearchBtn"></td>
        </tr>
      </table>
    </form>
  </td>
</tr>
</table>
</td>
</tr>
</table>
</center>
</div>
</div>
</div>
</body>
<script>
$("j_username").focus();
</script>
</html>
```

#### 2.4.3.5.2 Sample Custom Error Page

**Note:** You do not have to customize error pages. If you create custom error pages, they must be in HTML format (with file extension .HTML). You can use existing TeamConnect error message JSP files but cannot upload custom JSP files.

```
<html>
  <head>
  </head>
  <body>
    Authentication error.
  </body>
</html>
```

#### 2.4.3.5.3 Sample Restore Session Page

**Note:** The following is a sample HTML page content you may choose to redirect to for the LDAP parameter, `ldap.restoreSessionUrl`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>You have another session open</title>
</head>
<body>
  <div id="horizon" style="margin-top:150px">
    <div id="content">
      <div id="contentText">
        <center>
          <table border="0" width="500" cellpadding="0" cellspacing="0"
            style="margin: 20px;">
            <tr valign="top">
              <td class="bgcolor16" align="left" style="padding: 20px;
                border-bottom: 1px solid #CCCCCC; line-height: 1.7em;">
                <table border="0" cellpadding="3" cellspacing="0">
                  <tr valign="top">
                    <td style="padding: 8px 0px 0px 15px;">
                      <span class="titlehdr5">You have another session
                        open</span>
                    </td>
                  </tr>
                </table>
              <td style="padding: 8px 0px 8px 15px;"><span
                style="padding:200px 0px 20px 0px;">You are either
                logged in at another location or did not log out of
```

```

        your previous session.<p>Do you want to recover the
        previous session and return to what you were working
        on, or start a new session (losing all previous work)?
    </span></td>
</tr>
<tr>
    <td align="center" class="texthdr">
        <form name="mainForm" action="restoreSession.htm"
        method="post">
            <input type="button" name="recover" value="Recover
            Session" class="mainSearchBtn"
            onclick="document.getElementById('restoreSession').valu
            e=true;document.mainForm.submit();" />&nbsp;
            <input type="button" name="new" value="New Session"
            class="mainSearchBtn"
            onclick="document.getElementById('restoreSession').valu
            e=false;document.mainForm.submit();" />&nbsp;
            <input type="hidden" id="restoreSession"
            name="restoreSession" value="true" />
        </form>
    </td>
</tr>
</table>
</td>
</tr>
</table>
</center>
</div>
</div>
</div>
</body>
</html>

```

#### 2.4.3.5.4 Sample LDAP Configuration File

```

## This is a sample properties file for LDAP authentication. You can
tailor it to your specific
## situation to work with the LDAP authentication mechanism included
within TeamConnect. All
## un-commented properties in this file are required, though their values
can be changed.
# Identification properties for this authentication component
tc.displayName=LDAP Team Connect Authenticator
tc.uniqueId=LDAP
tc.isSSO=false
# Custom page mappings (omit any that do not apply)
#page.badCredentials=tryagain.html
#page.lockedAccount=giveup.html

```

```
#page.maximumLogins=sorry.html
#page.logout=goodbye.html
#page.sessionTimeout=missedyou.html
# LDAP configuration properties
#
# The URL of the LDAP server to be used
ldap.host=10.1.2.48
ldap.port=389
ldap.root=dc=domain,dc=com
# DN of the user/administrator allowed to connect to the LDAP server
ldap.userDn=cn=Manager,dc=domain,dc=com
# Password of the user/administrator
ldap.password=secret
#
ldap.userDnPatterns=uid={0},ou=Department
#
ldap.groupDnPattern=ou=Department
#
ldap.groupRoleAttribute=cn
```

#### 2.4.4 Custom Authentication Plug-ins

You can write your own authentication plug-in and integrate it with TeamConnect security. The details of the plug-in code are beyond the scope of this document, but here are the general principles:

Write your own Java class, implementing one of these two interfaces:

- CustomSSOAuthenticator
- CustomUserPwdAuthenticator

Write an **authenticationDescriptor.properties** file. The file needs to have at least the following lines (sample data shown in italics):

- tc.displayName=*My Authenticator*
- tc.uniqueId=*MYAUTH*
- tc.isSSO=*true*
- tc.authenticatorClassName=*com.mitratesoft.teamconnect.authentication.provider.TestMySSOAuthenticator*

It's a good idea to override some default error-message HTML files, too:

- page.badCredentials=*loginFailed.html*
- page.logout=*logout.html*
- page.sessionExpired=*timeout.html*

Put your new class in Document folder **System/Authentication/MYAUTH/classes** (the folder path must incorporate the value of tc.uniqueId). Put the properties file in there too.

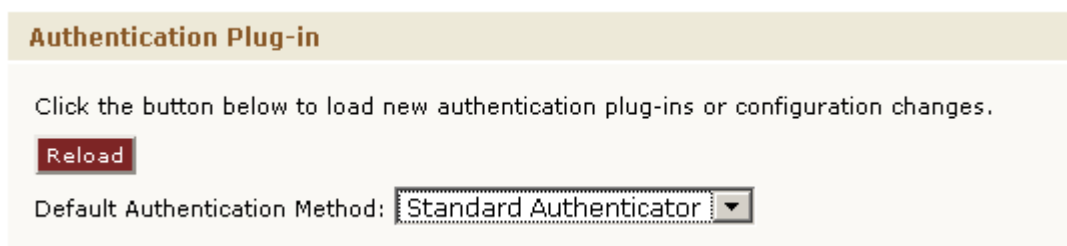
As a sibling of folder **classes**, create folder **pages**. Put in HTML files to be used for overriding the default error-condition pages. This is especially important if you're writing an SSO authenticator.

#### 2.4.4.1 Customizing Message Screens

If you enable an authentication plug-in mechanism in TeamConnect, you can customize the screens that you encounter while accessing and logging out of TeamConnect. You must write, test, and deploy your own Java class to display the customized screens. You should do this step only if you plan to provide extra information for users during their sessions.

Your custom Java class must implement interface **javax.servlet.Filter**. It should be deployed in the same Documents folder as the configuration files.

### 2.4.5 Enabling a Default Authentication Method



#### Enabling a Default Authentication Method

Choose a default method as described in [Configuring User Authentication](#). The names used in the default authentication drop-down correspond to the value of **tc.displayName** in the **authenticationDescriptor.properties** file that is associated with each authentication method.

### 2.4.6 Overriding Authentication for Individual Users

Now that a default plug-in has been configured, you must override the authentication method for each user account that requires a non-default method. (Users whose authentication matches the default method do not require any overrides.) For details on settings on the user account screen, see [Creating or Editing Users](#).

The example below presumes that your default authentication method is not LDAP, but some individual users do need to authenticate through LDAP. The steps described are similar no matter which method you are using to override.



**Account Information**

Authentication: LDAP Team Connect Authenticator

\*User Name: rog

\*Contact: Allenby, Roger

Description:

User Type: Normal

Account is active

Set expiration date:

User must change password at next log in

Enable assistive technology

Example: User Record - General Page

#### To configure an individual user account to use LDAP authentication

1. Click the **Admin** tab.
2. Click the **Users** link under the tab bar.
3. Open the desired user account.
4. From the **Authentication** drop-down list on the **General** page, select **LDAP TeamConnect Authenticator**.

The screen refreshes and all fields regarding TeamConnect passwords are removed since a user who is authenticated through LDAP does not need a TeamConnect password.

5. Click **Save**.

LDAP authentication configuration is completed.

The next time the user logs in to TeamConnect, even though the TeamConnect login screen is presented, he or she is actually authenticated through LDAP.

## 2.5 Troubleshooting and Frequently Asked Questions

This section provides additional frequently asked questions and troubleshooting information about users, groups, and the IMAP Server.

## 2.5.1 Frequently Asked Questions

This section includes additional information about application features, such as tips or helpful facts under the following topics.

- [Users and Groups](#)
- [Address Books](#)
- [Budgets](#)
- [Admin Settings](#)
- [Licensed Seats](#)

### 2.5.1.1 Users and Groups

**Q: I created an object view and assigned it to a user group, but the members of this group don't see the new object view.**

Make sure that you set the appropriate **Default Group** for the user. A user without a default group assigned uses the object views defined on the **Object Views** tab of the Designer application area **System Settings**.

If a default group is selected, the user uses the object views selected on the **Object Views** page of the group account. You can select the default group for a user on the **Group Memberships** page of the user account.

**Q: What happens if I deactivate or delete user accounts?**

When you deactivate a user account, the user's name is removed from the lists of Task and Project assignees, Appointment attendees, and Approval routes. However, it still appears on the **History** tab of records where they have entered History records, and other areas where that user is referenced as the **Created By**, **Modified By**, and so on.

If you delete a user account, all references to the user are removed from all records. In the records, the **Created By** and **Modified By** fields point to the TeamConnectAdmin user account instead.

**Q: If I enable the User Lockout feature, will a user that gets locked out be deactivated in TeamConnect?**

While a user is locked out of TeamConnect, only his or her ability to log in is disabled. The user account in TeamConnect remains active. This means that the user appears on all active lists, such as Appointment attendees, and can still be assigned to tasks or projects.

To unlock the user, clear the **Account is Locked** check box on the user account's **General** page. After doing this, the user is able to log into TeamConnect, and everything in his or her account is unaffected.

**Q: Why don't I see the authentication plug-in that I uploaded to TeamConnect in the list of available plug-ins on the system settings screen?**

After you upload a plug-in, in order for it to be visible from the **Plug-in** drop-down list for first time use, you must refresh the list. To do this, click **Load Plug-in**. Afterwards, the **Plug-in** drop-down list will contain the name of the plug-in that you uploaded.

**Q: Why don't I see custom objects' embedded objects on the Security tab's Other Restrictions section of custom object definitions?**

The additional record-level security restrictions of a custom object on its related objects does not include embedded objects.

**Q: What is the difference between cascading record-level security in the Related Object Security block and the Other Restrictions block on the Security tab of custom object definitions?**

In the **Related Object Security** block, you can set custom objects' record-level security to cascade to their related objects automatically. When read-only rights are cascaded from parent records to their related object records, users can still create related objects. For situations when parent projects have read-only access, due to security or workflow reasons, you can use the **Other Restrictions** block to deny users the ability to update, delete, or create related object records, even if they are the creators.

**Q: Why can't I create a record? (User with Create record rights or Create all record rights)**

You must assign both Create record rights and View record rights for a group member to be able to create records. For example:

- From global record rights for end-users, you must assign both **View all records** and **Create all records** to a group. Afterward, group members will be able to create all record types described in [Rights for End-users](#).
- For a specific record type (for example, contacts), you must assign both **View Contact records** and **Create Contact records** rights.

**Q: Why don't I see a new user in the New Users collection?**

Users in the **New Users** collection are sorted alphabetically. If a collection contains many records you may not see the expected record near the top of the list. Users in the **Recently Viewed** collection are sorted with the topic record as the most recently viewed record.

**Q: Why don't I see rights for the Involved object for a new custom object (project)?**

After you create a custom object (project), you need to create a corresponding **Involved**. Afterward, from the **User Rights** list (of the group's **Record Rights** page), you can select the

custom object and assign rights for the related **Involved** records.

**Q: Why was someone able to edit a user or group record while I was editing the record?  
Record locking was enabled.**

Record locking only applies to system objects and custom objects for end users (see [Rights for End-users](#)). Record locking does not apply to user or group records.

**Q: (TeamConnect®Enterprise Legal only) Why are some group members (users) unable to view the Legal tab?**

If group members do not have rights to view any record types that belong under the **Legal** tab, then the Legal tab does not display. For example, Advice and Counsel, Cost Centers, Disputes, Time Entry Settings, and Transactions are record types that can display under the **Legal** tab so if a user belongs to groups that do not have rights to any of those record types, then the **Legal** tab does not display.

**Q: I got a "Session Timeout" warning, tried to log back in, and got a "Recover Session" page?**

The actual expiration of a session might be significantly later than the session timeout warning. That's because of the application server's handling of expired sessions. For example, in BEA WebLogic, the default setting for checking for expired sessions is every 60 seconds. So if a session timed out, it would actually still be valid until WebLogic cleaned up the expired session, which might be up to 60 seconds from the timeout message. If you tried to log in again during this period, the still-valid session would cause a Recover Session page to appear. In this WebLogic example, if you wanted to alter the application server's behavior, you could consult WebLogic documentation for information about changing the applicable setting (invalidation-interval-secs).

**Q: I have a license for a number of concurrent users, but I can't use the entire number?**

This situation is also caused by application servers, as described in the question above. If you had the maximum number of concurrent users logged in simultaneously, and one user's session timed out, another user wouldn't be able to log in until the application server cleaned up the expired session.

### 2.5.1.2 Address Books

**Q: Where do I assign rights for Address Books?**

Address Book access rights are bundled with contact records access rights. If you assign the right, **View Contact records**, to a group then group members will also be able to view address books, create address books, edit address books (name and public vs. private status), and delete address books. If you assign the right, **Edit Contact records**, to a group then group members will also be able to add and remove contacts from address books.

### 2.5.1.3 Budgets

**Q: I can't see a Budget page from disputes or transactions. Why?**

Based on settings in the Legal Preferences tool, users may or may not be able to adjust budgets from dispute and transaction records. From the Legal Preferences tool, if the **Extend budget level by field** is set to **Do not budget by fiscal period**, budgets will not display within dispute and transaction records.

### 2.5.1.4 Admin Settings

**Q: From Admin Settings, I set the Read-only mode date format to mm/dd/yy but records display dates with the year in the format yyyy (4-digits). Why?**

When records are in edit-mode, the year must display with 4 digits.

### 2.5.1.5 Licensed Seats

**Q: How can I tell how many licensed user seats are remaining (unused)?**

You cannot view this number directly, but you can check your license limits, then view your current active users to determine how many have been used. To check license limits, click the **Admin** tab. Click the **Admin Settings** link under the tab bar. Click the **About** link from the left pane. From the resulting page, the **License** section displays the details of the license key, including limits. To count active users, click the **Users** link beneath the **Admin** tab, and view the count of users in the **Active** collection.

## 2.5.2 Troubleshooting

### IMAP

**Issue: Troubleshooting the TeamConnect IMAP Server**

Resolution: There are two ways to view IMAP Server events from a log.

You can view the system default log. See [Viewing Logs](#).

You can also create an IMAP Appender.

### License File

**Issue: A "License file is missing" error message displays**

Resolution: If you have not uploaded a license file for your application, an error message will display throughout the application ("License file is missing. Please check that it exists under Top Level

\System"). The license file should be uploaded to the **Documents** area \Top Level\System folder in TeamConnect.

**Issue: A "User account cannot be created because the number of active user accounts has exceeded the limit specified in the license key" error message displays**

Resolution: If you have reached the maximum number of users allowed in your current license file, this error displays when you attempt to create a new user. To add seats to the license file, contact Mitratech support. For more information, see [How can I tell how many licensed user seats are remaining \(unused\)?](#).

**Issue: A "User account cannot be updated because the number of active user accounts has exceeded the limit specified in the license key." error message displays**

Resolution: If you have reached the maximum number of users allowed in your current license file, this error displays when you attempt to activate an inactive user. For more information, see [How can I tell how many licensed user seats are remaining \(unused\)?](#).

**Issue: A "License file is invalid. Please check that it is correctly formatted." error message displays**

Resolution: If the license file you uploaded cannot be validated, this error message will display throughout the application. Contact Mitratech support for assistance.

## WebDAV

**Issue: When a user uploads files in batch loads using the WebDAV feature, the TeamConnect Documents displays all files as having the type, Data Mapping Files, instead of the actual file type (for example .DOC). The files that were uploaded in batch do not appear in full-text search results.**

Resolution: You may need to update a TeamConnect configuration file with information about document extensions and MIME types that you expect to upload in batch.

## Email Notifications

**Issue: TeamConnect system email notifications include links to TeamConnect that do not work.**

Resolution: Verify that you have set up the **Login URL** field in Admin Settings. For more information, see the topic [Setting Up Email](#) and description of the **Login URL** field.

## Clustered Environment

**Issue: In a clustered environment, data does not appear identically in each workstation session and/or the database server issues optimistic locking error messages.**

Resolution: Cache synchronization between nodes in the cluster may not be working properly.

## Searching

**Issue:** When performing a global search using "All as the qualifier, an error message appears.

Resolution: You may need to enable full-text search as follows:

1. On the SQLServer host, open **Services** and look for **SQL Full-text Filter Daemon Launcher (MSSQLSERVER)**.
2. Ensure that this service is running and is started by the same user who started the SQLServer (MSSQLSERVER) service.
3. Once the service is running, log-in to Microsoft SQL Server Management Studio as 'sa' user.
4. Open the master database and execute the following command:

```
sp_fulltext_service 'restart_all_fdhosts'
```

## Localization Spreadsheet

**Issue:** I cut and pasted translated text into the Excel localization spreadsheet and now I cannot delete it or make any changes.

Resolution: If you plan to copy and paste translated data from third-party translation software, such as Google Translate, you may have difficulty deleting or making changes to the text that you paste into Excel.

To avoid this, after you copy your text and are in the Excel spreadsheet, go to the appropriate cell, right-click, select **Paste Special**, and then select **Text**. This removes any hidden characters and allows you to make changes to your pasted text if necessary.

## 2.6 Rights for End-users

This appendix provides reference tables with information on how to assign rights to groups of end users. These rights involve access to a given TeamConnect record type, along with the appropriate pages and features associated with this type. You may access these features using the **View** menu in the **Record Rights** list page.

You can only assign rights at the group level and not specifically to users. Any user who is a member of a given group inherits all the rights assigned to that group.

### How To Use This Reference:

Each table contains the following information:

- Name of the right you may assign to a group
- Description of the operations this right allows a group to do
- Prerequisite rights a group needs in order to have this right (if applicable)
- TeamConnect pages that these rights enable a group to view

For information about rights that apply to administrator-type users, see [Rights for Administrators](#).

For information about rights that apply to solution developer-type users, see [Rights for Solution Developers](#).

## Rights to System Records

You may assign rights to the following system record types, listed in the **View** menu, to do primary operations in TeamConnect:

- Account
- Appointment
- Contact
- Document
- Expense
- History
- Invoice
- Task

## Rights to Custom Records

TeamConnect allows system designers to create custom records. For TeamConnect, custom records are referred to in this guide as Projects. For Matter Management, custom records are referred to in this guide as Matters. After a project or matter has been created, its name will appear in a group record's **Record Rights** page, **View** drop-down list.

## TeamConnect Legal Records

A few custom record (matter) types are included with TeamConnect Legal.

- Advice and Counsel
- Cost Center
- Dispute
- Transaction

**Note:** Record rights associated with Advice and Counsel, Cost Center, Dispute, Transaction records can be found in the [Project/Matter Rights](#) and [Project/Matter Involved Rights](#).

## Rights to TeamConnect Areas or Features

Utility operations allow group members to do specific operations with TeamConnect record types. You may assign rights to do these operations, using the following options listed in the **View** menu:

- Preferences
- Record Locking



- Scheduled Actions
- Search Settings

## 2.6.1 System Record Rights

This section lists and explains the rights you can assign to system record types in TeamConnect. Some of these rights also apply to TeamConnect Projects (custom record types).

**Note:** For more information on Projects, see [Custom Record Rights](#).

### 2.6.1.1 Account Rights

You must assign the appropriate rights for viewing and working with financial Account records, as explained in the following tables:

- The [Basic Rights to Account table](#) explains basic rights you can assign, allowing access to financial accounts, for example, **View Account records** and **Edit Account records**.
- The [Additional Rights to Account table](#) explains rights you can assign, allowing access to additional pages that are part of a financial account, for example, **View Account security** and **Create Account security**.

#### 2.6.1.1.1 Basic Account Rights

One or more basic account rights is required to view or work with financial accounts.

**Basic Rights to Account**

Right	Description	Prerequisite Rights	Pages
<b>View Account records</b>	Allows groups to search for, print, and see existing Accounts.	None	Accounts list <b>Advanced Account Search</b> Account record > <ul style="list-style-type: none"> <li>• <b>General</b></li> <li>• <b>Posting Criteria</b></li> <li>• <b>Workflow</b></li> <li>• <b>Child Accounts</b></li> </ul>
<b>Edit Account records</b>	Allows groups to change existing Accounts.	<b>View Account records</b>	Same as for <b>View Account records</b>
<b>Create Account</b>	Allows groups to create and print new Accounts. These groups can	<b>View Account records</b>	Same as for <b>View Account records</b>

<b>records</b>	<p>also create a copy of the current record.</p> <p><b>Note:</b> <i>To select a parent Account or create child Accounts, groups must be able to view the appropriate Accounts.</i></p> <p>To set any Account's posting criteria, groups must be able to view the appropriate:</p> <ul style="list-style-type: none"> <li>• Contacts</li> <li>• Projects and their categories</li> <li>• Expenses, Tasks, and their categories</li> </ul>		
<b>Delete Account records</b>	Allows groups to delete existing Accounts.	<b>View Account records</b>	Same as for <b>View Account records</b>
<b>Mass edit Account records</b>	Allows groups to mass edit existing Accounts.	<b>View Account records</b> <b>Edit Account records</b>	<b>Mass Edit page</b>

#### 2.6.1.1.2 Additional Account Rights

After you give groups one or more of the basic rights, additional rights and operations may become available to them, which were not accessible before. They also need to view pages where the operations are available. These Account rights enable a group to view the following additional TeamConnect pages:

- **Account record > Security**
- **Account record > Transactions**

The following table explains these additional group rights to Account:

**Additional Rights to Account**

Right	Description	Prerequisite Rights			Pages
		View Account Records	Edit Account Records	Additional	

<b>View Account all categories</b>	Allows groups to see existing Account categories and set one category as the default.	X		None	<b>Account record &gt; Categories</b>
<b>Add Account all categories</b>	Allows groups to add categories to Accounts.			None	
<b>Delete Account all categories</b>	Allows groups to delete categories from Accounts.	X	X	<b>View Account all categories</b>	
<b>View Account all custom fields</b>	Allows groups to see all the custom fields of Accounts.  <i>Note: To work with any custom field, groups must be able to view the field's category and its parents.</i>  To see custom fields in a Project, groups must be able to view the Project.	X		<b>View Account all categories</b>	As customized by design
<b>Edit Account all custom fields</b>	Allows groups to change custom fields in Accounts.	X	X	<b>View Account all custom fields</b>	
<b>View Account security</b>	Allows groups to see security access information for Accounts.	X		None	<b>Account record &gt; Security</b>
<b>Edit Account security</b>	Allows groups to change security settings for individual accounts.  <i>Note: Before you can enable security editing for a group and its users, the group must be able to view the</i>	X	X	<b>View Account security</b>	

	<i>appropriate Contacts.</i>				
<b>View Account all transactions</b>	Allows groups to see the transactions already posted to any Account.  <i>Note: To see transactions, groups must be able to view the appropriate Invoice Line Items, Expenses, and Tasks. Otherwise, these elements are displayed as "SECURED."</i>	<b>X</b>		None	<b>Account record &gt; Transactions</b>
<b>Create Account all transactions</b>	Allows groups to create and post transactions to an Account.	<b>X</b>		None	

### 2.6.1.2 Appointment Rights

You must assign the appropriate rights for viewing and working with appointments, as explained in the following tables:

- The [Basic Appointment Rights table](#) explains basic rights you can assign to work with calendar Appointments, for example, **View Appointment records** and **Edit Appointment records**.
- The [Additional Appointment Rights table](#) explains rights you can assign to work with additional pages that are part of a calendar appointment, for example, **View Appointment security** and **Edit Appointment security**.

#### 2.6.1.2.1 Basic Appointment Rights

One or more basic appointment rights is required to view or work with appointments.

The following table explains the basic rights to Appointment:

**Basic Rights to Appointment**

Right	Description	Prerequisite Rights	Pages
-------	-------------	---------------------	-------

<b>View Appointment records</b>	<p>Allows groups to search for, print, and see existing Appointments.</p> <p><i>Note: To view Appointments associated with Projects, groups must be able to see the appropriate Projects.</i></p> <p>Viewing rights also provide access to the Calendar view, which is one of the Appointment search views.</p> <p>To create Appointments with the Calendar view, groups need the <b>Create Appointment records</b> right to the appropriate Appointments.</p>	None	<p>Appointments list</p> <p><b>Advanced Appointment Search</b></p> <p>Appointment record &gt;</p> <ul style="list-style-type: none"> <li>• <b>General</b></li> <li>• <b>Attendees</b></li> <li>• <b>Workflow</b></li> </ul>
<b>Edit Appointment records</b>	<p>Allows groups to change existing Appointments.</p>	<b>View Appointment records</b>	<p>Same as for <b>View Appointment records</b></p>
<b>Create Appointment records</b>	<p>Allows groups to create and print Appointments. These groups can also create a copy of the current record.</p> <p><i>Note: To create appointments associated with Project records, groups must be able to view the appropriate Projects.</i></p>	<b>View Appointment records</b>	<p>Same as for <b>View Appointment records</b></p>
<b>Delete Appointment records</b>	<p>Allows groups to delete existing Appointments.</p>	<b>View Appointment records</b>	<p>Same as for <b>View Appointment records</b></p>
<b>Mass edit Appointment records</b>	<p>Allows groups to mass edit existing Appointments.</p>	<b>View Appointment records</b>  <b>Edit Appointment records</b>	<p><b>Mass Edit</b> page</p>

## 2.6.1.2.2 Additional Appointment Rights

There are additional appointment record pages that are available. Generally, access to these additional pages requires the assignment of the **View Appointment records** and/or **Edit Appointment records** rights.

- **Appointment** record > **Attendees**
- **Appointment** record > **Resources**
- **Appointments** list > **Scheduler**
- **Appointment** record > **Security**

The following table explains these rights to additional appointment pages:

**Additional Rights to Appointment**

Right	Description	Prerequisite Rights			Pages
		View Appointment records	Edit Appointment records	Additional	
<b>Add Appointment attendees</b>	Allows groups to add attendees to Appointments.  <i>Note: To view and add attendees, groups must be able to view the appropriate Contacts.</i>	X	X	None	<b>Appointment record &gt; Attendees</b>
<b>Edit Appointment attendees</b>	Allows groups to change an attendee's information.	X	X	None	
<b>Delete Appointment attendees</b>	Allows groups to remove attendees from Appointments.	X	X	None	
<b>View Appointment all categories</b>	Allows groups to see existing Appointment categories and set one of them as the default.	X		None	<b>Appointment record &gt; Categories</b>
<b>Create Appointment all categories</b>	Allows groups to add categories to Appointments.	X	X	<b>View Appointment all</b>	

				categories	
<b>Delete Appointments all categories</b>	Allows groups to remove categories from Appointments.	<b>X</b>	<b>X</b>	<b>View Appointment all categories</b>	
<b>View Appointments all custom fields</b>	Allows groups to see custom fields of Appointments.  <i>Note: To work with a custom field, groups must be able to view its associated category and the category's parents.</i>  To see custom fields in Project, groups must have be able to view the appropriate Projects.	<b>X</b>		None	As customized by design
<b>Edit Appointments all custom fields</b>	Allows groups to change custom fields of Appointments.	<b>X</b>	<b>X</b>	<b>View Appointment all custom fields</b>	
<b>View Appointments resources</b>	Allows groups to see existing resources for Appointments.	<b>X</b>		None	<b>Appointment record &gt; Resources</b>
<b>Edit Appointments resources</b>	Allows groups to change resource information.	<b>X</b>	<b>X</b>	<b>View Appointment resources</b>	
<b>Create Appointments resources</b>	Allows groups to add resources to Appointments.	<b>X</b>	<b>X</b>	<b>View Appointment resources</b>	

<b>Delete Appointment resources</b>	Allows groups to remove resources from Appointments.	<b>X</b>	<b>X</b>	<b>View Appointment resources</b>	
<b>Scheduler Tool</b>	Allows groups to create Appointments with the Scheduler Tool.	<b>X</b>	<b>X</b>	None	<b>Appointments list &gt; Scheduler</b>
<b>View Appointment security</b>	Allows groups to see the security access for Appointments.	<b>X</b>		None	<b>Appointment record &gt; Security</b>
<b>Edit Appointment security</b>	Allows groups to change the security for Appointments.  <i>Note: To set security for groups and users, the group must be able to view the appropriate Contacts.</i>	<b>X</b>	<b>X</b>	<b>View Appointment security</b>	

### 2.6.1.3 Contact and Address Book Rights

You must assign the appropriate rights for viewing and working with contacts, as explained in the following tables:

- The [Basic Rights for Contacts table](#) explains basic rights you can assign to work with contacts, for example, **View Contact records** and **Edit Contact records**.
- The [Additional Rights to Contacts table](#) explains rights you can assign to work with additional pages that are part of a contact, for example, **View Contact security** and **Edit Contact security**.

#### 2.6.1.3.1 Basic Contact and Address Book Rights

One or more basic contact rights is required to view or work with contacts and address books.

The following table explains the basic rights for working with contacts:

**Basic Rights for Contacts**

<b>Right</b>	<b>Description</b>	<b>Prerequisite Rights</b>	<b>Pages</b>
<b>View Contact</b>	Allows users to search for and view existing Contacts.	None	Contacts list page



<b>records</b>	Also allows users to view, edit (name and change public vs. private status), create, and delete address books.		<b>Advanced Contact Search</b> page Contact record > <ul style="list-style-type: none"> <li>• <b>General</b> page &gt; <ul style="list-style-type: none"> <li>○ <b>Address</b> section</li> <li>○ <b>Phone</b> section</li> <li>○ <b>Fax</b> section</li> <li>○ <b>Email</b> section</li> <li>○ <b>Internet Address</b> section</li> </ul> </li> <li>• <b>Workflow</b> page</li> </ul>
<b>Edit Contact records</b>	Allows users to change existing Contacts.  Also allows users to add or remove contacts from address books.	<b>View Contact records</b>	Same as for <b>View Contact records</b>
<b>Create Contact records</b>	Allows users to create new Contacts.  To specify a company, users must also have the Read right to Contact.  This right also determines whether users can add contacts in the Search Module window.	<b>View Contact records</b>	Same as for <b>View Contact records</b>
<b>Delete Contact records</b>	Allows users to delete existing Contacts.	<b>View Contact records</b>	Same as for <b>View Contact records</b>
<b>Mass edit Contact records</b>	Allows groups to mass edit existing Contacts.	<b>View Contact records</b>  <b>Edit Contact records</b>	<b>Mass Edit</b> page

## 2.6.1.3.2 Additional Contact Rights

There are additional contact record pages that are available. Generally, access to these additional pages requires the assignment of the View Contact records and Edit Contact records rights.

- **Contact** record > **Categories**
- **Contact** record > **Rates**
- **Contact** record > **Relations**
- **Contact** record > **Security**
- **Contact** record > **Skills**
- **Contact** record > **Territories**

The following table explains these rights to additional contact pages:

**Additional Rights to Contacts**

Right	Description	Prerequisite Rights			Pages
		View Contact records	Edit Contact records	Additional	
<b>View Contact all categories</b>	Allows users to view existing contact categories and set one of them as default.	x			<b>Contact</b> record > <b>Categories</b>
<b>Add Contact all categories</b>	Allows users to add categories to contacts.	x	x	<b>View Contact all categories</b>	
<b>Remove Contact all categories</b>	Allows users to delete categories from contacts.	x	x	<b>View Contact all categories</b>	
<b>View Contact all custom fields</b>	Allows users to view custom fields of contacts.  To work with custom fields, users must have at least the Read right to the associated category and its parents.  To see custom fields of type Custom Object, users must	x			As customized by design

	have at least the Read right to the custom object.				
<b>Edit Contact all custom fields</b>	Allows users to change custom fields of contacts.	<b>x</b>	<b>x</b>	<b>View Contact all custom fields</b>	
<b>View Contact involved</b>	Allows users to view a contact's involvement in various custom objects.  To view a contact's involvement, users must have at least the Read right to custom objects.  To view a contact's involved roles, users must have at least the Read right to the Involved object and its categories.	<b>X</b>		None	<b>Contact record &gt; Involved page</b>
<b>View Contact rates</b>	Allows users to view assigned contact rates.  <i>Note: To see tasks for which rates are set, users must have at least the Read right to Task.</i>	<b>x</b>		<b>View Task Records</b>	<b>Contact record &gt; Rates</b>
<b>Edit Contact rates</b>	Allows users to update existing rate information for contacts.	<b>x</b>	<b>x</b>	<b>View Contact rates</b>	
<b>Create Contact rates</b>	Allows users to add rates to contacts.  <i>Note: To add rates, the users must have at least the Read right to Task.</i>	<b>x</b>	<b>x</b>	<b>View Contact rates</b>  <b>View Task Records</b>	
<b>Delete Contact</b>	Allows users to delete rates from contacts.	<b>x</b>	<b>x</b>	<b>View Contact</b>	

rates				t rates	
<b>View Contact relations</b>	Allows users to view existing contact relations.	x			<b>Contact record &gt; Relations</b>
<b>Create Contact relations</b>	Allows users to add relations to contacts.	x	x	<b>View Contact relations</b>	
<b>Delete Contact relations</b>	Allows users to delete relations from contacts.	x	x	<b>View Contact relations</b>	
<b>View Contact security</b>	Allows users to view security access for contacts.	x			<b>Contact record &gt; Security page</b>
<b>Edit Contact security</b>	Allows users to change security for contacts.	x	x	<b>View Contact security</b>	
<b>View Contact skills</b>	Allows users to view existing contact skills.	x			<b>Contact record &gt; Skills page</b>
<b>Edit Contact skills</b>	Allows users to update existing skill information for contacts.	x	x	<b>View Contact skills</b>	
<b>Create Contact skills</b>	Allows users to add skills to contacts.	x	x	<b>View Contact skills</b>	
<b>Delete Contact skills</b>	Allows users to delete skills from contacts.	x	x	<b>View Contact skills</b>	
<b>View Contact territories</b>	Allows users to view existing contact territories.	x			<b>Contact record &gt; Territories page</b>

<b>territories</b>					
<b>Create Contact territories</b>	Allows users to add territories to contacts.	<b>x</b>	<b>x</b>	<b>View Contact territories</b>	
<b>Delete Contact territories</b>	Allows users to delete territories from contacts.	<b>x</b>	<b>x</b>	<b>View Contact territories</b>	

#### 2.6.1.4 Document Rights

Documents in TeamConnect are unique record types. A document uploaded to TeamConnect is a record and can be accessed from the Documents tab. In addition, documents can be associated with other record types from a different record's Documents page. For example, if you assign rights to Create Contact records and Create Document records, then group members can upload documents from a contact's [Documents page](#).

You must assign the appropriate rights to work with documents, as explained in the following tables:

- The [Basic Rights to Document table](#) explains basic rights you can assign to work with documents, for example, **View Document records** and **Edit Document records**.
- The [Additional Rights to Document table](#) explains rights you can assign to work with additional pages related to documents, for example, **View Document security** and **Edit Document security**.

##### 2.6.1.4.1 Basic Document Rights

One or more basic document rights is required to view or work with documents.

**Note:** Whereas other record types have multiple pages associated with each record, documents behave differently. From the Documents List page, when a document's Properties icon is clicked, a list of property pages associated with that document opens. Document property pages associated with basic document rights include General, and Workflow pages.

The following table explains the basic rights to Document:

**Basic Rights to Document**

<b>Right</b>	<b>Description</b>	<b>Prerequisite Rights</b>	<b>Pages</b>
<b>View Document records</b>	Allows groups to search for, view, print, and share existing	None	Documents list

	Documents, as well as search for and view folders.  <i>Note: For a group member to be able to access its own Document folder or other contact folders, the group member must be able to view the appropriate Contacts.</i>		<b>Advanced Document Search</b>  Document Properties > <ul style="list-style-type: none"><li>• <b>General</b></li><li>• <b>Workflow</b></li></ul>
<b>Edit Document records</b>	Allows groups to check out and check in Documents.	<b>View Document records</b>	Same as for <b>View Document records</b>
<b>Create Document records</b>	Allows groups to upload new Documents, as well as to create folders, links, and shortcuts to Documents.  <i>Note: To specify the author of a Document, groups must be able to view the appropriate Contact.</i>	<b>View Document records</b>	Same as for <b>View Document records</b>
<b>Delete Document records</b>	Allows groups to delete existing Documents or move them to another location in the Documents area.  <i>Note: To move Documents, a group must also have the Create Document records right.</i>	<b>View Document records</b>	Same as for <b>View Document records</b>
<b>Mass edit Document records</b>	Allows groups to mass edit existing Documents.	<b>View Document records</b>  <b>Edit Document records</b>	<b>Mass Edit page</b>

#### 2.6.1.4.2 Additional Document Rights

There are additional Document Property pages that are available. Generally, access to these additional pages requires the assignment of the **View Document records** and **Edit Document records** rights. These Document rights enable a group to view the following additional property pages, email folders, and email messages:

- **Document Properties > Categories**
- **Document Properties > Security**
- **Document Properties > Versions**
- **Email Folder**
- **Email record**

**Note:** E-mail features and rights are only available with TeamConnect Projects. For more information on Projects, see [Custom Record Rights](#). If the TeamConnect IMAP Server is set up and active, you must enable appropriate rights to groups to work with e-mail (treated as documents in TeamConnect).

The following table explains these additional group rights to documents:

**Additional Rights to Document**

Right	Description	Prerequisite Rights			Pages
		View Document records	Edit Document records	Additional	
<b>View Document all categories</b>	Allows groups to view existing Document categories and set one of them as the default.	X		None	<b>Document Properties &gt; Categories</b>
<b>Create Document all categories</b>	Allows groups to add categories to Documents.	X	X	<b>View Document all categories</b>	
<b>Delete Document all categories</b>	Allows groups to remove categories from Documents.	X	X	<b>View Document all categories</b>	
<b>View Document all custom fields</b>	Allows groups to view custom fields of Documents.  <i>Note: To work with any custom field, groups must be able to view the</i>	X		None	As customized by design

	<p><i>field's category and its parents.</i></p> <p>To see custom fields in a Project, groups must be able to view the Project.</p>				
<b>Edit Document all custom fields</b>	Allows groups to change custom fields of Documents.	<b>X</b>	<b>X</b>	<b>View Document all custom fields</b>	
<b>Generate New XML Based Document</b>	<p>Allows groups to create a new XML Document from an existing template.</p> <p><b>Note:</b> <i>To generate a Document, a group needs to be able to view the appropriate Documents list. However, to save a generated Document, the group also needs the ability to create Documents in that list.</i></p>	<b>X</b>		None	
<b>View Document security</b>	Allows groups to view security access for Documents.	<b>X</b>		None	<b>Document Properties &gt; Security</b>
<b>Edit Document security</b>	<p>Allows groups to change security for Documents.</p> <p><b>Note:</b> <i>Before you can enable security editing for a group and its users, the group must be able to view the appropriate Contacts.</i></p>	<b>X</b>	<b>X</b>	<b>View Document security</b>	



<b>View Document versions</b>	Allows groups to view the version history of Documents.  <i>Note: To generate a version history of Documents, a group must enable the Version Control feature, using the system settings.</i>	<b>X</b>		None	<b>Document Properties &gt; Versions</b>
<b>Revert Document versions</b>	Allows groups to remove the current most recent version of a Document and save the previous version as most recent.	<b>X</b>	<b>X</b>	<b>View Document versions</b>	
<b>View email</b>	Allows groups to search for and view email in TeamConnect.	<b>X</b>		<b>View Project records</b>	<b>Documents list &gt; Email Folder</b>
<b>Move email from non TeamConnect-IMAP folder to TeamConnect-IMAP folder</b>	Allows groups to move e-mail messages from an email client's non TeamConnect-IMAP folder to a TeamConnect-IMAP folder.		<b>X</b>	<b>Create Document records</b> <b>View Project records</b> <b>Edit Project records</b>	<b>Documents list &gt; Email Folder</b>
<b>Move email between TeamConnect-IMAP folders</b>	Allows groups to move email messages between an email client's TeamConnect-IMAP folders.	<b>X</b>	<b>X</b>	<b>Create Document records</b> <b>Delete Document</b>	<b>Documents list &gt; Email Folder</b>

				<b>record s</b> <b>View Project record s</b> <b>Edit Project record s</b>	
<b>Delete email</b>	Allows groups to delete IMAP email messages (for email clients and TeamConnect).	<b>X</b>		<b>Delete Document record s</b> <b>View Project record s</b> <b>Edit Project record s</b>	<b>Email record</b>
<b>Change email attributes (priority, read/unread)</b>	Allows groups to change email message attributes (for email clients).			<b>Create Document record s</b> <b>View Project record s</b> <b>Edit Project record s</b>	<b>Email record</b>
<b>Add TeamConnect-IMAP subfolder</b>	Allows groups to create TeamConnect-IMAP subfolders (for email clients).			<b>Create Document record s</b>	<b>Documents list</b>

				View Project records Edit Project records	
<b>Rename / Delete TeamConnect-IMAP subfolder</b>	Allows groups to delete TeamConnect-IMAP subfolders (for email clients and TeamConnect).	X	X	Delete Document records View Project records Edit Project records	Documents list
<b>Add TeamConnect-IMAP subfolder</b>	Allows groups to create TeamConnect-IMAP subfolders (for e-mail clients).			Create Document records View Project records Edit Project records	Documents list
<b>Rename / Delete TeamConnect-IMAP subfolder</b>	Allows groups to delete TeamConnect-IMAP subfolders (for e-mail clients and TeamConnect).	X	X	Delete Document records View Project records	Documents list

				<b>Edit Project records</b>	
--	--	--	--	-----------------------------	--

### 2.6.1.5 Expense Rights

You must assign the appropriate rights for viewing and working with expense records, as explained in the following tables:

- The [Basic Rights to Expense table](#) explains basic rights to work with budget Expenses, for example, **View Expense records** and **Edit Expense records**.
- The [Additional Rights to Expense table](#) explains rights to work with additional pages that are part of an expense, for example, **View Expense security** and **Edit Expense security**.

#### 2.6.1.5.1 Basic Expense Rights

One or more basic expense rights is required to view or work with expenses.

The following table explains the basic rights to expenses:

**Basic Rights to Expense**

<b>Right</b>	<b>Description</b>	<b>Prerequisite Rights</b>	<b>Pages</b>
<b>View Expense records</b>	Allows groups to search for and view existing Expenses.	None	Expenses list <b>Advanced Expense Search</b> Expense record > <ul style="list-style-type: none"> <li>• <b>General</b></li> <li>• <b>Transactions</b></li> <li>• <b>Workflow</b></li> </ul>
<b>Edit Expense records</b>	Allows groups to change existing Expenses.	<b>View Expense records</b>	Same as for <b>View Expense records</b>
<b>Create Expense records</b>	Allows groups to create and print expenses. These groups can also create a copy of the current record. Also allows usage of the Expense Entry Tool.  To be able to associate expenses with a specific Project and	<b>View Expense records</b>	Same as for <b>View Expense records</b>

	Contacts, groups must be able to view the Project and appropriate Contacts.		
<b>Delete Expense records</b>	Allows groups to delete existing Expenses.	<b>View Expense records</b>	Same as for <b>View Expense records</b>
<b>Mass edit Expense records</b>	Allows groups to mass edit existing Expenses.	<b>View Expense records</b> <b>Edit Expense records</b>	<b>Mass Edit page</b>

#### 2.6.1.5.2 Additional Expense Rights

There are additional expense record pages that are available. Generally, access to these additional pages requires the assignment of the View Expense records and Edit Expense records rights. These Expense rights enable a group to view the following additional expense pages:

- **Expense record > Categories**
- **Expense record > Security**

The following table explains these additional group rights to expenses:

##### Additional Rights to Expense

Right	Description	Prerequisite Rights			Pages
		View Expense records	Edit Expense records	Additional	
<b>Post</b>	Allows groups to post Expenses to accounts.  From any page of an open expense, a Post button becomes available unless the expense has posted.	<b>X</b>	<b>X</b>	None	
<b>Void</b>	Allows groups to void Expenses from accounts.  From any page of an open expense, a Void button becomes available unless the expense has not posted.	<b>X</b>	<b>X</b>	None	

<b>View Expense all categories</b>	Allows groups to view existing Expense categories and set one of them as the default.	<b>X</b>		None	<b>Expense record &gt; Categories</b>
<b>Create Expense all categories</b>	Allows groups to add categories to Expenses.	<b>X</b>	<b>X</b>	<b>View Expense all categories</b>	
<b>Delete Expense all categories</b>	Allows groups to remove categories from Expenses.	<b>X</b>	<b>X</b>	<b>View Expense all categories</b>	
<b>View Expense all custom fields</b>	Allows groups to view custom fields of Expenses.  <i>Note: To work with any custom field, groups must be able to view the field's category and its parents.</i>  To see custom fields in a Project, groups must be able to view the Project.	<b>X</b>		None	As customized by design
<b>Edit Expense all custom fields</b>	Allows groups to change custom fields of Expenses.	<b>X</b>	<b>X</b>	<b>View Expense all custom fields</b>	
<b>View Expense security</b>	Allows groups to view security access for Expenses.	<b>X</b>		None	<b>Expense record &gt; Security</b>
<b>Edit Expense security</b>	Allows groups to change security for expenses.  <i>Note: Before you can enable security editing for a group and its users, the group must be</i>	<b>X</b>	<b>X</b>	<b>View Expense security</b>	

	<i>able to view the appropriate Contacts.</i>				
<b>Expense Entry Tool</b>	Tool to add, post, and delete multiple expenses in batch.	<b>X</b>	<b>X</b>	<b>Create Expense records</b> <b>Post (Expenses)</b> <b>Delete Expense records</b>	<b>Expense Entry Tool</b>

### 2.6.1.6 History Rights

History records in TeamConnect are unique record types. A history in TeamConnect is a record and can be accessed from the All Services tab, History link. In addition, histories can be associated with other record types from a different record's History page. For example, if you assign rights to Create Contact records and Create History records, then group members can create a history from a contact's [History page](#).

You must assign appropriate rights to enable viewing or working with History records, as explained in the following tables:

- The [Basic Rights to History table](#) explains basic rights you can assign to work with History records, for example, **View History records** and **Edit History records**.
- The [Additional Rights to History table](#) explains rights you can assign to work with additional pages that are part of a history record, for example, **View History security** and **Edit History security**.

#### 2.6.1.6.1 Basic History Rights

One or more basic history rights is required to view or work with history records.

The following table explains the basic rights to history records:

**Basic Rights to History**

Right	Description	Prerequisite Rights	Pages
<b>View History records</b>	Allows groups to search for, print, and see existing History entries.	None	Histories list <b>Advanced History Search</b>

	<p><b>Note:</b> Because History entries can be created only from within records, groups must be able to view the appropriate records. Also, it is possible for users to search and view history records that are associated with another record type the user does not have access rights for. For example, if a user does not have rights to expense records, they can still search and view history records associated with expenses.</p>		<p>History record &gt;</p> <ul style="list-style-type: none"> <li>• <b>General</b></li> <li>• <b>Workflow</b></li> </ul>
<b>Edit History records</b>	Allows groups to change existing History entries.	<b>View History records</b>	Same as for <b>View History records</b>
<b>Create History records</b>	<p>Allows groups to create and print new History entries. These groups can also create a copy of the current record.</p> <p><b>Note:</b> Because History entries can be created only from within records, groups must be able to view the appropriate records.</p>	<b>View History records</b>	Same as for <b>View History records</b>
<b>Delete History records</b>	Allows groups to remove existing History entries.	<b>View History records</b>	Same as for <b>View History records</b>
<b>Mass edit History records</b>	Allows groups to mass edit existing History entries.	<p><b>View History records</b></p> <p><b>Edit History records</b></p>	<b>Mass Edit page</b>

#### 2.6.1.6.2 Additional History Rights

There are additional history record pages that are available. Generally, access to these additional pages requires the assignment of the View History records and Edit History records rights. These History rights enable a group to view the following additional expense pages:

- History record > Categories



- History record > Security

The following table explains these additional group rights to History:

**Additional Rights to History**

Right	Description	Prerequisite Rights			Pages
		View History records	Edit History records	Additional	
<b>View History all categories</b>	Allows groups to view the existing History categories and set one of them as the default.	X		None	<b>Histories</b> record > <b>Categories</b>
<b>Create History all categories</b>	Allows groups to add categories to History entries.	X	X	<b>View History all categories</b>	
<b>Delete History all categories</b>	Allows groups to delete categories from History entries.	X	X	<b>View History all categories</b>	
<b>View History all custom fields</b>	Allows groups to view custom fields of History entries.  <i>Note: To work with a custom field, groups must be able to view its category and the category's parents.</i>	X		None	As customized by design
<b>Edit History all custom fields</b>	Allows groups to change custom fields of History entries.	X	X	<b>View History all custom fields</b>	
<b>View History security</b>	Allows groups to view security access for History entries.	X		None	<b>History</b> record > <b>Security</b>

<b>Edit History security</b>	Allows groups to change security for History entries.  <i>Note: Before you can enable security editing for a group and its users, the group must be able to view the appropriate Contacts.</i>	X	X	<b>View History security</b>	
------------------------------	--	---	---	------------------------------	--

### 2.6.1.7 Invoice Rights

You must assign the appropriate rights to enable viewing and working with invoices, as explained in the following tables:

- The [Basic Rights to Invoice table](#) explains basic rights to work with invoices, for example, **View Invoice records** and **Edit Invoice records**.
- The [Additional Rights to Invoice](#) explains rights you can assign to work with additional pages that are part of a invoice, for example, **View Invoice security** and **Edit Invoice security**.

Invoice rights include rights to Invoice line items.

#### 2.6.1.7.1 Basic Invoice Rights

One or more basic invoice rights is required to view or work with invoices.

The following table explains the basic rights to Invoice:

**Basic Rights to Invoice**

Right	Description	Prerequisite Rights	Pages
<b>View Invoice records</b>	Allows groups to search for, print, and view existing Invoices.  <i>Note: To be able to view a specific vendor, groups must be able to view the appropriate Contacts.</i>	None	Invoices list <b>Advanced Invoice Search</b> Invoice record > <ul style="list-style-type: none"> <li>• <b>General</b></li> <li>• <b>Transactions</b></li> <li>• <b>Workflow</b></li> <li>• <b>Versions</b></li> </ul>
<b>Edit Invoice records</b>	Allows groups to change existing Invoices.	<b>View Invoice records</b>	Same as for <b>View Invoice records</b>

<b>Create Invoice records</b>	Allows groups to create and print new Invoices. These groups can also create a copy of the current record.  <i>Note: To be able to select a specific vendor, groups must be able to view the Contact.</i>	<b>View Invoice records</b>	Same as for <b>View Invoice records</b>
<b>Delete Invoice records</b>	Allows groups to delete existing Invoices.	<b>View Invoice records</b>	Same as for <b>View Invoice records</b>
<b>Mass edit Invoice records</b>	Allows groups to mass edit existing Invoices.	<b>View Invoice records</b> <b>Edit Invoice records</b>	<b>Mass Edit page</b>

#### 2.6.1.7.2 Additional Invoice Rights

After you give groups one or more of the basic rights, additional rights and operations may become available to them, which were not accessible before. They also need to view pages where the operations are available. These Invoice rights enable a group to view the following additional TeamConnect pages:

- **Invoices** record > **Categories** page
- **Invoice** record > **General** page > **Line Items** section
- **Invoice** record > **Security** page

The following table explains these additional rights to invoices:

**Additional Rights to Invoice**

Right	Description	Prerequisite Rights			Pages
		View Invoice records	Edit Invoice records	Additional	
<b>Post</b>	Allows groups to post Invoices to accounts.  From any page of an open invoice, a Post button becomes available unless the invoice has posted.	X	X	None	

<b>Void</b>	Allows groups to void Invoices from accounts.  From any page of an open invoice, a Void button becomes available unless the invoice has not posted.	<b>X</b>	<b>X</b>	None	
<b>View Invoice all categories</b>	Allows groups to view the existing Invoice categories and set one of them as the default.	<b>X</b>		None	<b>Invoices record &gt; Categories</b>
<b>Create Invoice all categories</b>	Allows groups to add categories to Invoices.	<b>X</b>	<b>X</b>	<b>View Invoice all categories</b>	
<b>Delete Invoice all categories</b>	Allows groups to remove categories from Invoices.	<b>X</b>	<b>X</b>	<b>View Invoice all categories</b>	
<b>View Invoice all custom fields</b>	Allows groups to view custom fields of Invoices.  <i>Note: To work with a custom field, groups must be able to view its category and the category's parents.</i>  To see the custom fields in a Project, groups must be able to view the Project.	<b>X</b>		None	As customized by design
<b>Edit Invoice all custom fields</b>	Allows groups to change custom fields of Invoices.	<b>X</b>	<b>X</b>	<b>View Invoice all custom fields</b>	
<b>View Line Item Records</b>	Allows groups to view Invoice line items.	<b>X</b>		None	<b>Invoice record &gt; General page &gt;</b>

<b>Edit Line Item Records</b>	Allows groups to change Invoice line items.  <i>Note: Because Invoice Line Items are associated with Projects, Task Categories, Expense Categories and Contacts, groups must be able to view these four items.</i>	X	X	<b>View Line Item Records</b>	<b>Line Items</b> section
<b>Create Line Item Records</b>	Allows groups to add line items to Invoices.  <i>Note: Because Invoice Line Items are associated with Projects, Task Categories, Expense Categories and Contacts, groups must be able to view these four items.</i>	X	X	<b>View Line Item Records</b>	
<b>Delete Line Item Records</b>	Allows groups to remove line items from Invoices.	X	X	<b>View Line Item Records</b>	
<b>View Line Item all categories</b>	Allows groups to view the existing Invoice line item categories and set one of them as the default.	X		None	<b>Line Item</b> record > <b>Categories</b>
<b>Create Line Item all categories</b>	Allows groups to add categories to Invoice line items.	X	X	<b>View Line Item records</b>  <b>View Line Item all categories</b>	

<b>Delete Line Item all categories</b>	Allows groups to remove categories from Invoice line items.	<b>X</b>	<b>X</b>	<b>View Line Item records</b> <b>View Line Item all categories</b>	
<b>View Line Item all custom fields</b>	Allows groups to view custom fields of Invoice line items.  <i>Note: To work with a custom field, groups must be able to view its category and the category's parents.</i>  To see the custom fields in a Project, groups must be able to view the Project.	<b>X</b>		None	As customized by design
<b>Edit Line Item all custom fields</b>	Allows groups to change custom fields of Invoice line items.	<b>X</b>	<b>X</b>	<b>View Line Item records</b> <b>View Line Item all categories</b> <b>View Line Item all custom fields</b>	
<b>View Invoice security</b>	Allows groups to view security access for Invoices.	<b>X</b>		None	<b>Invoice record &gt; Security</b>

<b>Edit Invoice security</b>	Allows groups to change security for Invoices.  <i>Note: Before you can enable security editing for individual invoices, the group must be able to view the appropriate Contacts.</i>	<b>X</b>	<b>X</b>	<b>View Invoice security</b>	
------------------------------	---	----------	----------	------------------------------	--

### 2.6.1.8 Report Rights

You must assign the appropriate rights to enable viewing and working with Reports, as explained in the following table:

#### Report Rights

Right	Description	Prerequisite Rights	Pages
<b>View Report records</b>	Allows groups to view Report lists, Report folders, view the results of Report execution, and view Report properties.	None	<ul style="list-style-type: none"> <li>Reports list</li> <li>Reports Chart View</li> <li>Reports Tabular View</li> </ul> <b>Report record &gt;</b> <ul style="list-style-type: none"> <li><b>General</b></li> <li><b>Data</b></li> <li><b>Chart</b></li> </ul>
<b>Edit Report records</b>	Allows groups to save changes to a Report and Report folders.	<b>View Reports records</b>	Same as for <b>View Report records</b>
<b>Create Report records</b>	Allows groups to create a new Report or Report folder.	<b>View Reports records</b> <b>Edit Report records</b>	Same as for <b>View Report records</b>
<b>Delete Report Records</b>	Allows groups to delete Reports and Report folders.	<b>View Reports records</b> <b>Edit Report records</b>	Same as for <b>View Report records</b>

		<b>Create Report records</b>	
<b>View Report security</b>	Allows groups to view security access for Reports	<b>View Reports records</b>	<b>Reports</b> record > <b>Security</b> page
<b>Edit Report security</b>	Allows groups to change security for Reports.	<b>View Reports records</b>	<b>Reports</b> record > <b>Security</b> page

### 2.6.1.9 Task Rights

You must assign groups the appropriate rights to enable them to work with Task records, as explained in the following tables:

- The [Basic Rights to Task table](#) explains basic rights you may assign a group to allow its members to work with Tasks, for example, **View Task records** and **Edit Task records**.
- The [Additional Rights to Task](#) explains rights you must assign a group to allow its members to work with additional features that are part of a Task, for example, **View Task security** and **Edit Task security**.

#### 2.6.1.9.1 Basic Task Rights

One or more basic task rights is required to view or work with tasks.

The following table explains the basic rights to Task:

**Basic Rights to Task**

Right	Description	Prerequisite Rights	Pages
<b>View Task records</b>	Allows groups to search for, print, and see existing Tasks.  <i>Note: To view the projects and Contacts associated with tasks, groups must be able to see Contacts and Projects.</i>	None	Tasks list  <b>Advanced Task Search</b>  Task record > <ul style="list-style-type: none"> <li>• <b>General</b></li> <li>• <b>Transactions</b></li> <li>• <b>Workflow</b></li> </ul>
<b>Edit Task records</b>	Allows groups to change existing Tasks.	<b>View Task records</b>	Same as for <b>View Task records</b>
<b>Create Task records</b>	Allows groups to create and print tasks. Also allows the usage of the Task Entry Tool.	<b>View Task records</b>	Same as for <b>View Task records</b>



	<p><b>Note:</b> To be able to use the Task Entry tool, groups must be able to view and edit rights to the appropriate Task categories.</p> <p>To be able to associate Tasks with specific Projects and Contacts, groups must be able to see Contacts and the appropriate Projects.</p>		
<b>Delete Task records</b>	Allows groups to delete existing Tasks.	<b>View Task records</b>	Same as for <b>View Task records</b>
<b>Mass edit Task records</b>	Allows groups to mass edit existing Tasks.	<b>View Task records</b> <b>Edit Task records</b>	<b>Mass Edit page</b>

#### 2.6.1.9.2 Additional Task Rights

After you give groups one or more of the basic rights, additional rights and operations may become available to them, which were not accessible before. They also need to view pages where the operations are available. These Task rights enable a group to view the following additional TeamConnect pages:

- **Task record > Assignees**
- **Task record > Security**

The following table explains these additional group rights to Task:

**Additional Rights to Task**

Right	Description	Prerequisite Rights			Pages
		View Task records	Edit Task records	Additional	
<b>Post</b>	<p>Allows groups to post Tasks to accounts.</p> <p>From any page of an open task, a Post button becomes available unless the task has posted.</p>	X	X	None	

<b>Void</b>	Allows groups to void Tasks from accounts.  From any page of an open task, a Void button becomes available unless the task has not posted.	<b>X</b>	<b>X</b>	None	
<b>View Task Assignees</b>	Allows groups to view the list of past and current Task assignees.  <i>Note: To view Task assignees, groups must be able to see Contacts.</i>	<b>X</b>		None	<b>Task record &gt;</b> <b>General page &gt;</b> <b>Assignment History</b> section
<b>Assign Task</b>	Allows groups to reassign Tasks to another group.  <i>Note: To view Task assignees, groups must be able to see Contacts.</i>	<b>X</b>	<b>X</b>	<b>View Task Assignees</b>	
<b>View Task all categories</b>	Allows groups to view the current default Task category.	<b>X</b>		None	<b>Task record &gt;</b> <b>Categories</b>
<b>Create Task all categories</b>	Allows groups to add categories to Tasks.	<b>X</b>	<b>X</b>	<b>View Task all categories</b>	
<b>Delete Task all categories</b>	Allows groups to delete categories from Tasks.	<b>X</b>	<b>X</b>	<b>View Task all categories</b>	
<b>View Task all custom fields</b>	Allows groups to see custom fields of Tasks.  <i>Note: To work with a custom field, groups must be</i>	<b>X</b>		None	As customized by design

	<p><i>able to view its category and the category's parents.</i></p> <p>To see the custom fields in a Project, groups must be able to view the Project.</p>				
<b>Edit Task all custom fields</b>	Allows groups to change custom fields of Tasks.	<b>X</b>	<b>X</b>	<b>View Task all custom fields</b>	
<b>View Task security</b>	Allows groups to view security access for Tasks.	<b>X</b>		None	<b>Task record &gt; Security page</b>
<b>Edit Task security</b>	<p>Allows groups to change security for Tasks.</p> <p><i>Note: Before you can enable security editing for a group and its users, the group must be able to view the appropriate Contacts.</i></p>	<b>X</b>	<b>X</b>	<b>View Task security</b>	
<b>Task Entry Tool</b>	Tool to add, post, and delete multiple tasks in batch.	<b>X</b>	<b>X</b>	<b>Create Task records</b> <b>Delete Task records</b> <b>Post (Tasks)</b>	Task Entry Tool

## 2.6.2 TeamConnect Features and Areas

This section explains rights related to TeamConnect features and areas, such as User Preferences.

### 2.6.2.1 Preferences Rights

You must assign the appropriate rights to view and edit User Preferences. Afterward, each group member can [edit their individual preferences](#) that apply to their personal TeamConnect experience.

The following table explains the rights to Preferences:

**Rights to Preferences**

Right	Description	Prerequisite Rights	Page
<b>View Preferences</b>	Allows groups to view their Preferences.	None	<b>Preferences</b>
<b>Edit Preferences</b>	Allows groups to change their Preferences.	<b>View Preferences</b>	

### 2.6.2.2 Release Edit Locks Right

This right only has an effect if the **Enable record locking in edit mode** Admin Setting is enabled. For more information about this system setting, see [Setting Up General Settings](#).

The following table explains the right for releasing record locks:

**Rights to Record Locking**

Right	Description	Prerequisite Rights	Page
<b>Release edit locks</b>	Allows group members to release the lock on a record when it is open for editing by another user. A Release Lock button becomes available after this right is assigned and the conditions above are met.  For more information, see <a href="#">Record Locking</a> and releasing the lock and <a href="#">Setting Up General Settings</a> .	None	N/A

### 2.6.2.3 Scheduled Actions Rights

You must assign the appropriate rights to view or delete Scheduled Actions. See [Scheduled Actions Monitor](#) for more information about Scheduled Actions.

The following table explains the rights to Scheduled Actions:

**Scheduled Actions Rights**

Right	Description	Prerequisite Rights	Page
-------	-------------	---------------------	------

<b>View Scheduled Actions records</b>	Allows groups to access the Scheduled Actions Monitor and view the pending actions listed.	None	<b>All page &gt; Tools page &gt; Scheduled Actions</b>
<b>Delete Scheduled Actions records</b>	Allows groups to delete scheduled actions in the Scheduled Actions Monitor so that they are not executed.	<b>View Scheduled Actions records</b>	

## 2.6.3 Custom Record Rights

Record types that solution developers can create and customize for your company are custom records. This section describes standard rights that would be assigned to a generic custom record, called a project (for TeamConnect) or matter (for TeamConnect Legal and TeamConnect® Express).

### 2.6.3.1 Project/Matter Rights

For TeamConnect, custom records are referred to in this guide as Projects. For TeamConnect Legal and TeamConnect® Express, custom records are referred to in this guide as Matters. In this guide, the terms project and matter can be considered as the same.

You must assign the appropriate rights to view or work with Projects/Matters, as explained in the following tables:

- The [Basic Rights to Project/Matter table](#) explains basic rights to work with a Project/Matter, for example, **View Project records** and **Edit Project records**.
- The [Additional Project/Matter Rights](#) explains rights you can assign to work with additional pages that are part of that Project, for example, **View Project security** and **Edit Project security**.

#### 2.6.3.1.1 Basic Project/Matter Rights

One or more basic project rights is required to view or work with projects.

The following table explains the basic rights to a Project/Matter:

**Basic Rights to Project/Matter**

Right	Description	Prerequisite Rights	Pages
<b>View Project records</b>	Allows groups to search for, print, and view existing Project records.	None	<b>Projects list</b> <b>Advanced Project Search</b> Project record > <ul style="list-style-type: none"> <li>• <b>General</b></li> </ul>

			• <b>Workflow</b>
<b>Edit Project records</b>	Allows groups to change existing Project records.	<b>View Project records</b>	Same as for <b>View Project records</b>
<b>Create Project records</b>	Allows groups to create new Project records.  <i>Note: If the Project is centered around Contacts, groups must be able to view the appropriate Contacts.</i>  If the Project requires a parent or is embedded in another Project, groups must be able to view the parent Project.	<b>View Project records</b>	Same as for <b>View Project records</b>
<b>Delete Project records</b>	Allows groups to delete existing Project records.	<b>View Project records</b>	Same as for <b>View Project records</b>
<b>Mass edit Project records</b>	Allows groups to mass edit existing Project records.	<b>View Project records</b>  <b>Edit Project records</b>	<b>Mass Edit page</b>

#### 2.6.3.1.2 Additional Project/Matter Rights

After you give groups one or more of the basic rights, additional rights and operations may become available to them, which were not accessible before. They also need to view pages where the operations are available. These Project/Matter rights enable a group to view the following additional TeamConnect pages:

- **Project/Matter record > Assignees**
- **Project/Matter record > Phases**
- **Project/Matter record > Involved**
- **Project/Matter record > Relations**
- **Project/Matter record > Security**

The following table explains these additional group rights to a Project/Matter:

##### Additional Project/Matter Rights

Right	Description	Prerequisite Rights	Pages
-------	-------------	---------------------	-------

		View Project records	Edit Project records	Additional	
<b>Create a copy</b>	Allows groups to create a copy of the current Project.  <i>Note: Giving this right causes the Create a copy button to appear on the Project record page.</i>	X	X	None	<b>Project record</b>
<b>View Project Assignees</b>	Allows groups to see existing Project assignees.  <i>Note: To view assignees, groups must be able to see Contacts.</i>	X		None	<b>Project record &gt; Assignees</b>
<b>Edit Project Assignees</b>	Allows groups to unassign or reassign Project assignees.  <i>Note: To change assignees, groups must be able to see Contacts.</i>	X	X	<b>View Project Assignees</b>	
<b>Create Project Assignees</b>	Allows groups to add assignees to Projects.  <i>Note: To add assignees, groups must be able to see Contacts.</i>	X	X	<b>View Project Assignees</b>	
<b>Delete Project Assignees</b>	Allows groups to remove assignees from Projects.	X	X	<b>View Project Assignees</b>	
<b>View Project all categories</b>	Allows groups to see the existing Project categories and set one of them as the default.	X		None	<b>Project record &gt; Categories</b>

<b>Create Project all categories</b>	Allows groups to add categories to Project records.	X	X	<b>View Project all categories</b>	
<b>Delete Project all categories</b>	Allows groups to delete categories from Project records.	X	X	<b>View Project all categories</b>	
<b>View Project all custom fields</b>	Allows groups to view custom fields in Project records.  <i>Note: To work with any custom field, groups must be able to view the field's category and its parents.</i>  To see custom fields in a Project, groups must be able to view the Project.	X		None	As customized by design
<b>Edit Project all custom fields</b>	Allows groups to change custom fields in Project records.	X	X	<b>View Project all custom fields</b>	
<b>Involved parties rights</b>	For details about rights to Involved parties, see <a href="#">Project/Matter Involved Rights</a> .  <i>Note: All of these rights appear in this segment of the menu.</i>				<b>Project record</b>
<b>View Project phases</b>	Allows groups to view Project phases.	X		None	<b>Project record &gt; Phases</b>
<b>Edit Project phases</b>	Allows groups to change Project phases.	X	X	None	



<b>View Project relations</b>	Allows groups to view existing relationships between Projects.	X		None	<b>Project record &gt; Relations</b>
<b>Create Project relations</b>	Allows groups to add relations between Projects.	X	X	<b>View Project relations</b>	
<b>Delete Project relations</b>	Allows groups to delete relations between Projects.	X	X	<b>View Project relations</b>	
<b>View Project security</b>	Allows groups to view security access for Project records.	X		None	<b>Project record &gt; Security</b>
<b>Edit Project security</b>	Allows groups to change security for Project records.  <i>Note: Before you can enable security editing for a group and its users, the group must be able to view the appropriate Contacts.</i>	X	X	<b>View Project security</b>	

### 2.6.3.2 Project/Matter Involved Rights

For TeamConnect, custom records are referred to in this guide as [Projects](#). For TeamConnect Legal and TeamConnect® Express, custom records are referred to in this guide as Matters. In this guide, the terms project and matter can be considered as the same.

[Involved Parties](#) (Involved) are contacts associated with the Project/Matter, but not assigned to the Project/Matter. You must assign the appropriate rights to enable specific Involved-related operations, as explained in the following tables:

- The [Basic Rights to Involved Contact table](#) explains basic rights you may assign a group to allow its members to work with Involved Parties, for example, **View Project involved** and **Edit Project involved**.
- The [Additional Involved Rights](#) explains rights you must assign a group to allow its members to work with additional features that are part of an Involved Contact, for example, **View Project involved security** and **Edit Project involved security**.

See the *User Guide* for more information on projects and involved records.

**Note:** After creating a custom object (project), you must create an associated Involved object. Afterward you can assign rights for that Involved object.

#### 2.6.3.2.1 Basic Involved Rights

One or more basic Involved rights is required to view or work with involved contacts.

The following table explains the basic rights to Involved:

**Basic Rights to Involved Contact**

Right	Description	Prerequisite Rights	Pages
<b>View Project involved</b>	Allows users to print and view existing Contacts who are Involved Parties in Projects.  <i><b>Note:</b> Before a group can view Involved Contact information, its members must be able to view the Involved Parties. The group must also be able to view and create the Involved Contact roles and categories before it can view Contact information.</i>	<b>View Project records</b>	<b>Project record &gt; Involved page</b>  Involved record > <ul style="list-style-type: none"> <li>• <b>General</b></li> <li>• <b>Relations</b></li> <li>• <b>Workflow</b></li> </ul>
<b>Edit Project involved</b>	Allows users to change existing Involved Contact records.	<b>View Project records</b>  <b>View Project involved</b>	Same as for <b>View Project involved</b>
<b>Create Project involved</b>	Allows users to create new Involved Parties.  <i><b>Note:</b> Before a group can create a new Involved Contact, its members must be able to view the Involved Parties. The group must also be able to view and create the Involved Contact roles and categories before it can view Contact information.</i>	<b>View Project records</b>  <b>View Project involved</b>	Same as for <b>View Project involved</b>

<b>Delete Project involved</b>	Allows users to delete existing Involved Parties.	<b>View Project records</b> <b>View Project involved</b>	Same as for <b>View Project involved</b>
--------------------------------	---	---	--

#### 2.6.3.2.2 Additional Involved Rights

After you give groups one or more of the basic rights, additional rights and operations may become available to them, which were not accessible before. They also need to view pages where the operations are available. These Involved rights enable a group to view the following additional TeamConnect pages:

- **Involved** record > **Categories**
- **Involved** record > **Relations**
- **Involved** record > **Security**

The following table explains these additional group rights to Involved:

**Additional Involved Rights**

Right	Description	Prerequisite Rights			Pages
		View Project involved	Edit Project involved	Additional	
<b>View Project involved all categories</b>	Allows groups to view the existing Involved categories and set one of them as the default.	X	X	<b>View Project records</b>	<b>Involved record &gt; Categories</b>
<b>Create Project involved all categories</b>	Allows groups to add categories to Involved Parties.	X	X	<b>View Project records</b> <b>View Project involved all categories</b>	
<b>Delete Project involved all categories</b>	Allows groups to delete categories from Involved Parties.	X	X	<b>View Project records</b> <b>View Project</b>	

				involved all categories	
<b>View Project involved all custom fields</b>	<p>Allows groups to view custom fields of Involved Parties.</p> <p><i>Note: To work with any custom field, groups must be able to view the field's category and its parents.</i></p> <p>To see custom fields in a Project, groups must be able to view the Project.</p>	X	X	<b>View Project records</b>	As customized by design
<b>Edit Project involved all custom fields</b>	Allows groups to change custom fields of Involved Parties.	X	X	<b>View Project records</b>  <b>View Project involved all custom fields</b>	
<b>View Project involved relations</b>	Allows groups to view existing relationships between Involved Parties.	X	X	<b>View Project records</b>	<b>Involved record &gt; Relations</b>
<b>Create Project involved relations</b>	Allows groups to create relationships between Involved Parties.	X	X	<b>View Project records</b>  <b>View Project involved relations</b>	
<b>Delete Project involved relations</b>	Allows groups to delete relationships between Involved Parties.	X	X	<b>View Project records</b>	

				<b>View Project involved relations</b>	
<b>View Project involved security</b>	Allows groups to view security access for Involved Parties.	<b>X</b>	<b>X</b>	<b>View Project records</b>	<b>Involved record &gt; Security</b>
<b>Edit Project involved security</b>	Allows groups to change security for Involved Parties.  <i>Note: Before you can enable security editing for a group and its users, the group must be able to view the appropriate Contacts.</i>	<b>X</b>	<b>X</b>	<b>View Project records</b>  <b>View Project involved security</b>	

### 2.6.3.3 Project/Matter Milestones Rights

For TeamConnect, custom records are referred to in this guide as [Projects](#). For TeamConnect Legal and TeamConnect® Express, custom records are referred to in this guide as Matters. In this guide, the terms project and matter can be considered as the same.

[Milestones](#) are measurable activities within a project that help bring the project to conclusion. You must assign the appropriate rights to view or work with milestones, as explained in the following tables:

- The [Basic Rights for Milestones table](#) explains basic rights you may assign to a group to allow its members to work with Milestones, for example, **View Project Milestones** and **Edit Project Milestones**.
- The [Additional Milestone Rights table](#) explains rights you can assign to work with additional pages that are part of that Project, for example, **View Project security** and **Edit Project security**.

See the User Guide for more information on projects and milestones.

**Note:** After creating a custom object (project), you must create an associated Involved object. Afterward you can assign rights for that Involved object.

## 2.6.3.3.1 Basic Project/Matter Milestone Rights

One or more basic milestone rights is required to view or work with milestones.

The following table explains the basic rights to a milestone:

**Basic Rights for Milestones**

Right	Description	Prerequisite Rights	Pages
<b>View Milestone records</b>	Allows groups to search for, print, and view existing Milestone records.	None	<b>Project</b> record > <b>Milestones</b> page <b>Milestones</b> record > <ul style="list-style-type: none"> <li>• <b>General</b></li> <li>• <b>Security</b></li> <li>• <b>Workflow</b></li> </ul>
<b>Edit Milestone records</b>	Allows groups to change existing Milestone records.	<b>View Project records</b> <b>View Project milestones</b>	Same as for <b>View Project milestones</b>
<b>Create Milestone records</b>	Allows groups to create new Milestone records.	<b>View Project records</b> <b>View Project milestones</b>	Same as for <b>View Project milestones</b>
<b>Delete Milestone records</b>	Allows groups to delete existing Milestone records	<b>View Project records</b> <b>View Project milestones</b>	Same as for <b>View Project milestones</b>

## 2.6.3.3.2 Additional Milestones Rights

After you give groups one or more of the basic rights, additional rights and operations may become available to them, which were not accessible before. They also need to view pages where the operations are available. These Milestones rights enable a group to view the following additional TeamConnect pages:

- **Milestones** record > **Categories**
- **Milestones** record > **Security**

The following table explains these additional group rights to a milestone:

**Additional Milestone Rights**

Right	Description	Prerequisite Rights	Pages
-------	-------------	---------------------	-------

		View Project milestones	Edit Project milestones	Additional	
<b>View Project milestones all categories</b>	Allows groups to view the existing Milestones categories and set one of them as the default.	X	X	View Project records	<b>Milestones record &gt; Categories</b>
<b>Create Project milestones all categories</b>	Allows groups to add categories to Milestones.	X	X	View Project records View Project milestones all categories	
<b>Delete Project milestones all categories</b>	Allows groups to delete categories from Milestones.	X	X	View Project records View Project milestones all categories	
<b>View Project milestones custom fields</b>	Allows groups to view custom fields of Milestones.  <i>Note: To work with any custom field, groups must be able to view the field's category and its parents.</i>  To see custom fields in a Project, groups must be able to view the Project.	X	X	View Project records	As customized by design
<b>Edit Project milestone</b>	Allows groups to change custom fields of milestones.	X	X	View Project records	

<b>all custom fields</b>				<b>View Project milestone all custom fields</b>	
<b>View Project milestone security</b>	Allows groups to view security access for Milestones.	<b>X</b>	<b>X</b>	<b>View Project records</b>	<b>Milestone record &gt; Security</b>
<b>Edit Project milestone security</b>	Allows groups to change security for Milestones.	<b>X</b>	<b>X</b>	<b>View Project records</b> <b>View Project milestone security</b>	

## 2.7 Rights for Administrators

This appendix provides reference tables for assigning rights to TeamConnect record types and areas that should be accessed by administrator-type users. From a group record's **Record Rights** page, the rights described in this appendix are a subset of options from the **View** drop-down list.

Record types for administrators include:

- **Groups**—Group accounts in TeamConnect.
- **Users**—User accounts in TeamConnect.

TeamConnect areas for administrators include:

- **Admin Settings**—From the **Admin** tab, these are TeamConnect system settings that apply to all users.

### 2.7.1 Rights for Group Records

You must assign the appropriate rights for viewing and working with group records. This section contains the following tables that explain these rights and their associated operations:

- The [Basic Rights for Group Records table](#) explains basic rights you can assign, allowing access to group accounts, for example, **View Group records** and **Edit Group records**.
- The [Additional Rights for Groups table](#) explains rights you can assign, allowing access to additional pages that are part of a group account, for example, **View Group Users** and **Create Group Users**.



### 2.7.1.1 Basic Group Rights

One or more basic group rights is required to view or work with group records.

#### Basic Rights for Group Records

Right	Description	Prerequisite Rights	Page
<b>View Group records</b>	Allows searching, viewing, and printing existing group accounts.	None	<ul style="list-style-type: none"> <li>• <b>Groups List</b> page</li> <li>• <b>Group Advanced Search</b> page</li> <li>• <b>Group</b> record               <ul style="list-style-type: none"> <li>○ <b>General</b> page</li> <li>○ <b>Object Views</b> page</li> <li>○ <b>Tool Rights</b> page</li> </ul> </li> </ul>
<b>Edit Group records</b>	Allows editing existing group accounts.	<b>View Group records</b>	See the <b>View Group records</b> right.
<b>Create Group records</b>	Allows creating group accounts.	None	<ul style="list-style-type: none"> <li>• <b>Groups List</b> page</li> <li>• <b>Group</b> record               <ul style="list-style-type: none"> <li>○ <b>General</b> page</li> <li>○ <b>Object Views</b> page</li> <li>○ <b>Tool Rights</b> page</li> </ul> </li> </ul>
<b>Delete Group records</b>	Allows deleting existing group accounts.	<b>View Group records</b>	See the <b>View Group records</b> right.

### 2.7.1.2 Additional Group Rights

There are additional group record pages that are available. Generally, access to these additional pages requires the assignment of the View Group Records and/or Edit Group Records rights.

- **Group** record > **Users** page
- **Group** record > **Record Rights** page

- **Group record > Category Rights** page

The following table explains rights to additional group pages.

#### Additional Rights for Groups

Right	Description	Prerequisite Rights			Page
		View Group Records	Edit Group Records	Additional	
<b>View Group Users</b>	Allows users to view existing group members.  <i>Note: To view existing group members, users must have the Read rights to user account.</i>	x			<b>Group record &gt; Users</b> page
<b>Create Group Users</b>	Allows users to add members to groups.  <i>Note: To view existing group members, users must have the Read rights to user account.</i>	x	x	<b>View Group Users</b>	
<b>Delete Group Users</b>	Allows users to delete members from groups.	x	x	<b>View Group Users</b>	
<b>View Group Rights</b>	Allows users to view existing group rights.	x			<b>Group record:</b> <ul style="list-style-type: none"> <li>• <b>Record Rights</b> page</li> <li>• <b>Category Rights</b> page</li> </ul>
<b>Edit Group Rights</b>	Allows users to set and change group rights.	x	x	<b>View Group Rights</b>	

## 2.7.2 Rights for User Records

You must assign the appropriate rights for viewing and working with user records. This section contains the following tables that explain these rights and their associated operations:

- The [Basic Rights for User records table](#) explains basic rights you can assign, allowing access to user accounts, for example, **View User records** and **Edit User records**.

- The [Additional Rights for Users table](#) explains rights you can assign, allowing access to additional pages that are part of a group account, for example, **View User Group Memberships** and **Create User Group Memberships**.

### 2.7.2.1 Basic User Rights

One or more basic user rights is required to view or work with user records.

#### Basic Rights for User records

Right	Description	Prerequisite Rights	Page
<b>View User records</b>	Allows searching, viewing, and printing existing user accounts.  <i>Note: To be able to view a user account, users must at least have Read rights to the Contact object.</i>	None	<ul style="list-style-type: none"> <li>• <b>Users List</b> page</li> <li>• <b>User Advanced Search</b> page</li> <li>• <b>User</b> record &gt; <ul style="list-style-type: none"> <li>○ <b>General</b> page</li> <li>○ <b>Tool Rights</b> page</li> </ul> </li> </ul>
<b>Edit User records</b>	Allows editing existing user accounts.	<b>View User records</b>	See the <b>View User records</b> right.
<b>Create User records</b>	Allows creating and printing user accounts.  <i>Note: To be able to create a user account, users must at least have Read rights to the Contact object.</i>	None	<ul style="list-style-type: none"> <li>• <b>Users List</b> page</li> <li>• <b>User</b> record &gt; <ul style="list-style-type: none"> <li>○ <b>General</b> page</li> <li>○ <b>Tool Rights</b> page</li> </ul> </li> </ul>
<b>Delete User records</b>	Allows deleting existing user accounts.	<b>View User records</b>	See the <b>View User records</b> right.

### 2.7.2.2 Additional User Rights

There are additional user record pages that are available. Generally, access to these additional pages requires the assignment of the basic **View User Records** and/or **Edit User Records** rights.

- **User** record > **Group Memberships** page
- **User** record > **Record Rights** page
- **User** record > **Category Rights** page

The following table explains rights to additional user pages.

## Additional Rights for Users

Right	Description	Prerequisite Rights			Page
		View Group Records	Edit Group Records	Additional	
<b>View Group Memberships</b>	Allows viewing and printing the groups the user belongs to.  <i>Note: To view groups, users must have the View Group records right.</i>	x			User record > Group Memberships page
<b>Create Group Memberships</b>	Allows adding the current user to groups.  <i>Note: To view groups, users must have View Group records right.</i>	x	x	View Group Memberships	
<b>Delete Group Memberships</b>	Allows removing the current user from groups.	x	x	View Group Memberships	
<b>View User Rights</b>	Allows viewing the user's record rights and category rights.	x			User record > Record Rights page
<b>Edit User Rights</b>	Allows editing the user's record rights and category rights.	x	x	View User Rights	Category Rights page

### 2.7.3 Rights for Admin Settings and Logs

From the **Admin** tab, there are two areas that can be accessed, **Admin Settings** and **Logging**. You can assign rights to both areas from the Admin Settings-related rights.

You must assign the appropriate rights for viewing and working with Admin Settings and logs. This section contains the following table that explains rights you can assign, for example, **View Admin Settings** and **Edit Admin Settings**.

#### Rights for Admin Settings and Logs

Right	Description	Prerequisite Rights
<b>View Admin Settings</b>	Allows viewing Admin Settings. Also allows viewing system logs and audit logs, and clearing existing system logs.	None
<b>Edit Admin Settings</b>	Allows editing Admin Settings. Also allows editing default system logging levels.	<b>View Admin Settings</b>

## 2.8 Rights for Solution Developers

This appendix provides reference tables for assigning rights to TeamConnect areas that solution developers need to access. From a group record's **Record Rights** page, the rights described in this appendix are a subset of options from the **View** drop-down list.

TeamConnect areas or features for solution developers include:

- **Administer Custom Tools**—Page that allows management of custom tools.
- **Global Navigation**—Page that allows customization of the global navigation toolbar and subnavigation links (displayed under a toolbar tab).
- **Home Pages**—Page that allows management and customization of home pages or the initial page in TeamConnect after logging in.
- **Lookup Tables**—Page that allows management and customization of system lookup tables in TeamConnect.
- **Custom Lookup Tables**—Page that allows management and customization of custom lookup tables in TeamConnect.
- **Multi-currency Lookup Table**—Page that allows management and customization of the multi-currency lookup table in TeamConnect.
- **Miscellaneous Settings**—Page that allows management of custom fields that should be considered part of a user's system preferences. Custom fields added to the Designer application, **Miscellaneous Settings** page correspond to fields that display on a user's **Preferences** area, **Other Settings** page.
- **Object Definitions**—Area that allows management of system and custom object definitions (record types).
- **Portal Panes**—Area that allows management and customization of applets that are available for user home pages. Portal Panes can be information-driven, for example, containing TeamConnect search views or RSS feeds. In addition, Portal Panes can be functionality-driven, for example, containing links to frequently used areas, such as workflow approval or record creation wizards.
- **Routes**—Area that allows management and customization of the part of a workflow approval rule where approvers, reviewers, process managers, and email notifications are defined.
- **Route Stops**—Page that allows management of route stops (required approvers)

- **System Settings**—From the Designer application, these are TeamConnect system settings that apply to all users. These rights also allow access to manage system appenders and audit appenders.

See [Access Rights to TeamConnect Designer](#) for information about minimal required rights to work with the TeamConnect Designer area.

## 2.8.1 Administer Custom Tools Rights

There are multiple steps to managing custom tools in TeamConnect.

1. Solution Developers must do the following:
  - a. Create custom tool source files and upload them to TeamConnect.
  - b. Afterward, from the **Administer Custom Tools** page, the developer must add the custom tool to the TeamConnect list of available custom tools.
2. Administrators or developers must enable rights to the custom tool from the Tool Rights page of appropriate group records.
3. Afterward, users will be able to view and work with custom tools.

Rights described in this section refer to step 1b, above.

After you assign any of the Administer Custom Tools rights (**View Administer Custom Tools records**, **Edit Administer Custom Tools records**, **Create Administer Custom Tools records**, or **Delete Administer Custom Tools records**), the following page will be available:

- (from the **Tools** menu) **Administer Custom Tools** page

### Administer Custom Tools Rights

Right	Description	Prerequisite Rights
<b>View Administer Custom Tools records</b>	Allows users to view the existing tools by the <b>Tools</b> drop-down on the <b>Admin</b> menu bar.	None
<b>Edit Administer Custom Tools records</b>	Allows users to change the information regarding existing tools.	<b>View Administer Custom Tools records</b>
<b>Create Administer Custom Tools records</b>	Allows users to add new (custom) tools.	<b>View Administer Custom Tools records</b>
<b>Delete Administer Custom Tools records</b>	Allows users to delete existing custom tools.	<b>View Administer Custom Tools records</b>

## 2.8.2 Global Navigation Toolbar Rights

The **Global Navigation** page allows solution developers to customize the Global Navigation toolbar (tabs, related links, and page content per tab) that displays when users log into TeamConnect. You must assign the appropriate rights for viewing and working with the **Global Navigation** page. This section contains a table that explains these rights and their associated operations

After you assign any of the Global Navigation rights (**View Global Navigation records**, **Edit Global Navigation records**, **Create Global Navigation records**, or **Delete Global Navigation records**), the following page will be available:

- (from the **Go To** menu) **Global Navigation** page

### Rights to Customize the Global Navigation Toolbar

Right	Description	Prerequisite Rights
<b>View Global Navigation records</b>	Allows users to view the Global Navigation area.	
<b>Edit Global Navigation records</b>	Allows users to edit the tab names, tab display order, and subnavigation links that display on the global navigation toolbar.	<b>View Global Navigation records</b>
<b>Create Global Navigation records</b>	Allows users to create tabs and links that display on the global navigation toolbar.	<b>View Global Navigation records</b>
<b>Delete Global Navigation records</b>	Allows users to delete tabs and links that display on the global navigation toolbar.	<b>View Global Navigation records</b>

## 2.8.3 Customize Home Pages Rights

The **Home Pages** page allows solution developers to customize the high level content that displays on the initial page after users log into TeamConnect. You must assign the appropriate rights for viewing and working with the **Home Pages** page. This section contains a table that explains these rights and their associated operations

After you assign any of the Home Pages rights (**View Home Pages**, **Edit Home Pages**, **Create Home Pages**, or **Delete Home Pages**), the following page will be available:

- (from the **Go To** menu) **Home Pages** page

### Rights to Customize Home Pages

Right	Description	Prerequisite Rights
<b>View Home Pages</b>	Allows users to search for and view existing Home Pages.	<b>View Portal Panes</b>

	<i>Note: Because Portal Panes are a major component of Home Pages, users must have at least the Read right to Portal Panes.</i>	
<b>Edit Home Pages</b>	Allows users to change existing Home Pages.	<b>View Home Pages</b>
<b>Create Home Pages</b>	Allows users to create new Home Pages. <i>Note: Because Portal Panes are a major component of Home Pages, users must have at least the Read right to Portal Panes.</i>	<b>View Portal Panes</b>
<b>Delete Home Pages</b>	Allows users to delete existing Home Pages.	<b>View Home Pages</b>

## 2.8.4 Lookup Tables Rights

Lookup Tables Rights refers to rights for solution developers to customize certain system lookup tables. The rights described below affect access to the Lookup Tables area > **System** tab. To a user, the result affects the list options that display from certain drop-down lists in the TeamConnect UI.

### Rights for System Lookup Tables

Right	Description	Prerequisite Rights	Page
<b>View System Tables</b>	Allows users to view existing system lookup tables and lookup table items.	<b>x</b>	<b>System tab</b>
<b>Edit System Tables</b>	Allows users to change existing lookup table items.	<b>View System Tables</b>	
<b>Create System Tables</b>	Allows users to add new system lookup table items.	<b>View System Tables</b>	
<b>Delete System Tables</b>	Allows users to delete existing system lookup table items.	<b>View System Tables</b>	



## 2.8.5 Custom Lookup Tables Rights

Custom Lookup Tables Rights refers to rights for solution developers to create and manage custom lookup tables. The rights described below affect access to the Lookup Tables area > **Custom** tab. To a user, the result affects the list options that display from certain drop-down lists in the TeamConnect UI.

**Rights for Custom Lookup Tables**

Right	Description	Prerequisite Rights	Page
<b>View Custom Lookup Table records</b>	Allows users to view existing custom lookup tables and lookup table items.		<b>Custom tab</b>
<b>Create Custom Lookup Table records</b>	Allows users to create new custom lookup tables.	<b>View User Tables</b>	
<b>Delete Custom Lookup Table records</b>	Allows users to delete existing custom lookup tables.	<b>View User Tables</b>	
<b>Create Custom Lookup Table Items</b>	Allows users to add new custom lookup tables items.	<b>View User Tables</b>	
<b>Delete Custom Lookup Table Items</b>	Allows users to delete existing custom lookup tables items.	<b>View User Tables</b>	
<b>Edit Custom Lookup Table Items</b>	Allows users to change existing custom lookup tables items.	<b>View User Tables</b>	

## 2.8.6 Multi-currency Lookup Table Rights

Multi-currency Lookup Table Rights refers to rights for solution developers to manage the multi-currency lookup table. The rights described below affect access to the Lookup Tables area > Multi-currency tab. To a user, the result affects the list options that display from certain drop-down lists in the TeamConnect UI.

## Rights for the Multi-currency Lookup Table

Right	Description	Prerequisite Rights	Page
<b>View Multi-currency Table</b>	Allows users to view the multi-currency lookup table and lookup table items.		<b>Multi-currency tab</b>
<b>Create Multi-currency Table</b>	Allows users to add new multi-currency lookup table items.	<b>View Multi-currency Table</b>	
<b>Edit Multi-currency Table</b>	Allows users to edit existing multi-currency lookup table items.	<b>View Multi-currency Table</b>	
<b>Delete Multi-currency Table</b>	Allows users to delete existing multi-currency lookup table items	<b>View Multi-currency Table</b>	

## 2.8.7 Miscellaneous Settings Rights

The **Miscellaneous Settings** page from the Designer application allows solution developers to manage custom fields that affect individual users' Preferences. After making changes to the **Miscellaneous Settings** page, those custom fields should be available from a user's **Preferences** area. You must assign the appropriate rights for viewing and working with the **Miscellaneous Settings** page. This section contains a table that explains these rights and their associated operations. In addition, rights to release edit locks and bypass search limits are located on the **Miscellaneous Settings** page.

After you assign any of the Miscellaneous Settings rights (**View Miscellaneous Settings**, **Edit Miscellaneous Settings**, **Create Miscellaneous Settings**, or **Delete Miscellaneous Settings**), the following page will be available:

- (from the **Tools** menu) **Misc. Settings**

## Rights for Miscellaneous Settings

Right	Description	Prerequisite Rights	Page
<b>Release edit locks</b>	Allows group members to release the lock on a record when it is open for editing by another user. A Release Lock button becomes available after this right is assigned and the conditions above are met.  See addition information about <a href="#">Record Locking</a> and releasing the lock and <a href="#">Setting Up General Settings</a> .	None  <i>Note: This right only has an effect if the <b>Enable</b></i>	

		<p><b>record locking in edit mode</b> Admin Setting is enabled. For more information about this system setting, see <a href="#">Setting Up General Settings</a>.</p>	
<b>Bypass search limits</b>	<p>Allows groups to bypass the Admin Settings for the Maximum Search results and <b>Maximum Search time (seconds)</b>.</p> <p><i>Note: If a search exceeds the limits, the group's users can click a <b>Return All Results</b> link to bypass the limit. TeamConnect then returns all records the group has access rights to, which may take more time.</i></p> <p>If a user does NOT have bypass rights and their search exceeds the maximum number of records or seconds, TeamConnect displays the records that were retrieved before the limit or time out was reached.</p>	None	
<b>View Miscellaneous Settings</b>	From the Designer application, allows access to the <b>Tools</b> drop-down list, <b>Misc. Settings</b> option. Also allows users to view existing custom fields that have been added to the <b>Misc. Settings</b> page.	x	<b>Miscellaneous Settings</b>
<b>Edit Miscellaneous Settings</b>	Allows users to change existing custom fields from the <b>Misc. Settings</b> page.	<b>View Miscellaneous Settings</b>	

<b>Create Miscellaneous Settings</b>	Allows users to create new custom fields to be used for <b>Misc. Settings</b> page.	<b>View Miscellaneous Settings</b>	
<b>Delete Miscellaneous Settings</b>	Allows users to delete existing custom fields from <b>Misc. Settings</b> page.	<b>View Miscellaneous Settings</b>	

## 2.8.8 Object Definitions Rights

The **Object Definitions** area from the Designer application allows solution developers to manage system and custom objects. After making changes to the **Object Definitions** area, those objects should have corresponding rights from a group's **Record Rights** page, **View** drop-down list. You must assign the appropriate rights for viewing and working with the **Object Definitions** area. This section contains a table that explains these rights and their associated operations.

After you assign any of the basic Object Definitions rights (**View Home Pages**, **Edit Home Pages**, **Create Home Pages**, or **Delete Home Pages**), the following pages will be available:

- (from the **Go To** menu) **Object Definitions** list
- Object Definition record for each of the following system objects:
  - **Account**
  - **Appointment**
  - **Contact**
  - **Document**
  - **Expense**
  - **Group**
  - **History**
  - **Invoice**
  - **Task**
  - **User**
- Object Definition record for any custom objects

Within each system object's Object Definition view, the following system blocks are represented by tabs:

- **Custom Fields**
- **Forms**
- **Object Views**
- **(Custom) Blocks**

- Rules
- Templates
- Wizards

There are corresponding rights to work with the system blocks (tabs), for example, **View Object Definition Custom Fields**. If you deny rights to a system block, a user will not be able to perform operations on the corresponding tab, but all system block tabs will still display. For details, see [Rights for System Record Types](#), [Rights for Custom Record Types](#), and [Requirements for Basic Record Rights](#), [Requirements for Additional Rights to Records](#).

#### Object Definition Rights

Right	Description	Prerequisite Rights			Page
		View Object Definitions	Edit Object Definitions	Additional	
<b>View Object Definitions</b>	Allows users to access and view existing object definitions.				<ul style="list-style-type: none"> <li>• <b>Object Definitions List</b> page</li> <li>• <b>Object Definition</b> for: <ul style="list-style-type: none"> <li>○ <b>Account</b></li> <li>○ <b>Appointment</b></li> <li>○ <b>Contact</b></li> <li>○ <b>Contact Group</b></li> <li>○ <b>Document</b></li> <li>○ <b>Expense</b></li> <li>○ <b>History</b></li> <li>○ <b>Invoice</b></li> <li>○ <b>Task</b></li> </ul> </li> </ul>
<b>Edit Object Definitions</b>	Allows users to change existing object definitions.	x			
<b>Create Object Definitions</b>	Allows users to create new custom object definitions and set the auto numbering, auto naming, add phases and add phase transitions.				
<b>Delete Object Definitions</b>	Allows users to delete existing custom object definitions.	x	x		
<b>View Object Definition</b>	Allows users to view existing user rights.	x			
					<b>Custom Fields</b> tab

<b>Custom Fields</b>					
<b>Edit Object Definition Custom Fields</b>	Allows users to change existing custom fields.	<b>x</b>	<b>x</b>	<b>View Object Definition on Custom Fields</b>	
<b>Create Object Definition Custom Rights</b>	Allows users to set and change user rights.	<b>x</b>	<b>x</b>	<b>View Object Definition on Custom Fields</b>	
<b>Delete Object Definition Custom Fields</b>	Allows users to delete existing custom fields.	<b>x</b>	<b>x</b>	<b>View Object Definition on Custom Fields</b>	
<b>View Object Definition Forms</b>	Allows users to view existing forms.				<b>Forms tab</b>
<b>Edit Object Definition Forms</b>	Allows users to change existing forms.	<b>x</b>	<b>x</b>	<b>View Object Definition on Forms</b>	
<b>View Object Definition Object Views</b>	Allows users to access and view existing object views.	<b>x</b>			<b>Object Views tab</b>
<b>Edit Object Definition Object Views</b>	Allows users to change existing object views.	<b>x</b>	<b>x</b>	<b>View Object Definition on Object Views</b>	

<b>Create Object Definition Object Views</b>	Allows users to create new customized object views.  <i>Note: To add custom blocks to the new object view, users must at least have Read rights to blocks.</i>	<b>x</b>	<b>x</b>	<b>View Object Definition on Object Views</b>	
<b>Delete Object Definition Object Views</b>	Allows users to delete existing object views.	<b>x</b>	<b>x</b>	<b>View Object Definition on Object Views</b>	
<b>View Object Definition (Custom) Blocks</b>	Allows users to view existing custom blocks.	<b>x</b>			<b>Blocks tab</b>
<b>Edit Object Definition (Custom) Blocks</b>	Allows users to change existing custom blocks.	<b>x</b>	<b>x</b>	<b>View Object Definition (Custom) Blocks</b>	
<b>Create Object Definition (Custom) Blocks</b>	Allows users to add new custom blocks.	<b>x</b>	<b>x</b>	<b>View Object Definition (Custom) Blocks</b>	
<b>Delete Object Definition (Custom) Blocks</b>	Allows users to delete existing custom blocks.	<b>x</b>	<b>x</b>	<b>View Object Definition (Custom) Blocks</b>	

<b>View Object Definition Rules</b>	Allows users to view existing rules	<b>x</b>			<b>Rule screen &gt; General tab Action tab</b>
<b>Edit Object Definition Rules</b>	Allow users to change rules	<b>x</b>	<b>x</b>	<b>View Object Definition Rules</b>	
<b>Create Object Definition Rules</b>	Allow users to create new rules.  To create valid rules with qualifiers, users must save the general rule data and then set the qualifiers. So, users must have Update rights to rules and the Read and Update rights to qualifiers.  To set actions for approval rules, users must have the Read right to the route.	<b>x</b>	<b>x</b>	<b>View Object Definition Rules</b>	
<b>Delete Object Definition Rules</b>	Allow users to delete rules	<b>x</b>	<b>x</b>	<b>View Object Definition Rules</b>	
<b>View Object Definition Rule Qualifiers</b>	Allow users to view rule qualifiers			<b>View Object Definition Rules</b>	<b>Rule screen &gt; Qualifier tab</b>
<b>Edit Object Definition Rule Qualifiers</b>	Allow users to change rule qualifiers			<b>View Object Definition Rules</b>  <b>Edit Object Definition Rules</b>	



				View Object Definition Rule Qualifiers	
<b>Create Object Definition Rule Qualifiers</b>	Allow users to add rule qualifiers.			View Object Definition Rules Edit Object Definition Rules View Object Definition Rule Qualifiers	
<b>Delete Object Definition Rule Qualifiers</b>	Allow users to delete rule qualifiers			View Object Definition Rules Edit Object Definition Rules View Object Definition Rule Qualifiers	
<b>View Object</b>	Allow users to view existing conditions	x			Conditions tab

<b>Definition Conditions</b>					
<b>Edit Object Definition Conditions</b>	Allow users to change conditions	x	x	<b>View Object Definition Conditions</b>	
<b>Create Object Definition Conditions</b>	Allows users to create new conditions	x	x	<b>View Object Definition Conditions</b>	
<b>Delete Object Definition Conditions</b>	Allow users to delete existing conditions	x	x	<b>View Object Definition Conditions</b>	
<b>View Object Definition Search Views</b>	Allow users to view search views.	x			<b>Search Views</b> tab
<b>Edit Object Definition Search Views</b>	Allow users to change existing search views  <i>Note: With this right, users can add, change, and delete search qualifiers and display columns.</i>	x	x	<b>View Object Definition Search Views</b>	
<b>Create Object Definition Search Views</b>	Allow users to create new search views including qualifiers and display columns.	x	x	<b>View Object Definition Search Views</b>	

<b>Delete Object Definition Search Views</b>	Allow users to delete search views	<b>x</b>	<b>x</b>	<b>View Object Definition on Search Views</b>	
<b>View Object Definition Templates</b>	Allow users to view Templates	<b>x</b>			<b>Templates tab</b>
<b>Create Object Definition Templates</b>	Allow users to create new Templates	<b>x</b>	<b>x</b>	<b>View Object Definition on Templates</b>	
<b>Edit Object Definition Templates</b>	Allow users to change Templates	<b>x</b>	<b>x</b>	<b>View Object Definition on Templates</b>	
<b>Delete Object Definition Templates</b>	Allow users to delete Templates	<b>x</b>	<b>x</b>	<b>View Object Definition on Templates</b>	
<b>View Object Definition Wizards</b>	Allow users to view existing wizards	<b>x</b>			
<b>Edit Object Definition Wizards</b>	Allow users to change wizards	<b>x</b>	<b>x</b>	<b>View Object Definition on Wizards</b>	<b>Wizards tab</b>
<b>Create Object</b>	Allows users to create new wizards.	<b>x</b>	<b>x</b>	<b>View Object Definition</b>	

<b>Definition Wizards</b>	<i>Note: To associate templates with wizards, users must at least have Read rights to Templates.</i>			<b>on Wizards</b>	
<b>Delete Object Definition Wizards</b>	Allow users to delete existing wizards	<b>x</b>	<b>x</b>	<b>View Object Definition on Wizards</b>	

## 2.8.9 Customize Portal Panes Rights

The Portal Panes area allows solution developers to customize the content within individual panes (portal panes) that display on the initial page (home page) that opens after users log into TeamConnect. You must assign the appropriate rights for viewing and working with the Portal Panes area. This section contains a table that explains these rights and their associated operations

After you assign any of the Portal Pane rights (View Portal Panes, Edit Portal Panes, Create Portal Panes, or Delete Portal Panes), the following page will be available:

- (from the Go To menu) Portal Panes page

### Rights to Customize Portal Panes

Right	Description	Prerequisite Rights
<b>View Portal Panes</b>	Allows users to search for and view existing Portal Panes.	
<b>Edit Portal Panes</b>	Allows users to change existing Portal Panes.	<b>View Portal Pane</b>
<b>Create Portal Panes</b>	Allows users to create new Portal Panes.	
<b>Delete Portal Panes</b>	Allows users to delete existing Portal Panes.	<b>View Portal Pane</b>

## 2.8.10 Routes Rights

The **Routes** area allows solution developers to manage routes (approvers, reviewers, process managers, and email notifications associated with workflow approvals). You must assign the appropriate rights for viewing and working with the **Routes** area. This section contains a table that explains these rights and their associated operations.

After you assign any of the basic Routes rights (**View Routes**, **Edit Routes**, **Create Routes**, or **Delete Routes**), the following pages will be available:

- (from the **Go To** menu) **Routes** List page
- Existing routes

For each route, the following tabs are available: **General**, **Stops**, and **Email Notifications**.

Before users/groups can view or work with the **Stops** tab, you must assign one of the corresponding rights (**View Route Stops**, **Edit Route Stops**, **Create Route Stops**, or **Delete Route Stops**); otherwise the tab will not display content. From the Routes area, the **Stop** tab defines required check points during an approval process.

#### Routes Rights

Right	Description	Prerequisite Rights			Page
		View Routes	Edit Routes	Additional	
<b>View Routes</b>	Allow users to view routes				<ul style="list-style-type: none"> <li>• <b>Route &gt; General</b> tab</li> <li>• <b>Route &gt; Email Notification</b> tab</li> </ul>
<b>Edit Routes</b>	Allow users to change routes	x			
<b>Create Routes</b>	Allow users to create new routes.  <i><b>Note:</b> To create routes with stops, users must first save the general route information and then create the desired stops. Thus, users must have the Update right to Routes and the Read and Update rights to Stops.</i>				<i><b>Note:</b> The Stop tab is available but doesn't display content or allow stop management until Route Stop rights are assigned.</i>
<b>Delete Routes</b>	Allow users to delete routes	x			
<b>View Route Stops</b>	Allow users to view existing stops.	x			<b>Route &gt; Stop</b> tab

<b>Edit Route Stops</b>	Allow users to add new stops to routes.	<b>x</b>	<b>x</b>	<b>View Route Stops</b>	
<b>Create Route Stops</b>	Allow users to change stops of routes.	<b>x</b>	<b>x</b>	<b>View Route Stops</b>	
<b>Delete Route Stops</b>	Allow users to delete stops from routes.	<b>x</b>	<b>x</b>	<b>View Route Stops</b>	

### 2.8.11 System Settings and Appenders Rights

From the Designer application, **System Settings** drop-down list, there are two areas that can be accessed, **System Settings** and **Appenders**. You can assign rights to both areas from the System Settings rights.

You must assign the appropriate rights for viewing and working with System Settings and appenders. This section contains the following table that explain these rights and their associated operations:

- The [Rights for System Settings and Appenders table](#) explains rights you can assign, for example, **View System Settings** and **Edit System Settings**.

After you assign any of the System Settings rights (**View System Settings** or **Edit System Settings**), the following pages will be available:

- From the Designer application, System Settings area:
  - **Rule Execution** page
  - **Custom Messages** page
  - **Clear Cache: 2.x UI Settings** page
  - **Default Object Views** (system-level) page
  - **Authentication Plug-In** page
  - **Backward Compatibility Settings** page
  - **History**
- From the Designer application, Appenders area:
  - **System Appenders** page
  - **Audit Appenders** page

#### Rights for System Settings and Appenders

Right	Description	Prerequisite Rights
-------	-------------	---------------------

<b>View System Settings</b>	Allows users to access and view the system settings.	None
<b>Edit System Settings</b>	Allows users to change the system settings.	<b>View System Settings</b>

## 2.9 Documents and Folders

The TeamConnect Documents area contains files that are related to your customized system design and documents that users attach to records. This appendix provides a reference to the default directory structure.

### 2.9.1 Directory Structure

The following directory structure of the [Documents area](#) includes not only files for TeamConnect records, but also all files that control the customized design of your application. <CustomObjects> and <UserNames> are placeholders for your system's custom objects and user names.

**Note:** Depending on the specific TeamConnect product that you are using, your username may be denied access to folders **System**, **Users**, **Attachments**, object definition folders, and subfolders **Rules** and **Screens** beneath each object definition folder.

```

Top Level
|
|-- /Attachments
|   |-- /Accounts
|   |-- /Appointments
|   |-- /Contacts
|   |-- /Expenses
|   |-- /Histories
|   |-- /Invoices
|   |-- /Line Items
|   |-- /Tasks
|   |-- /<CustomObjects> ...
|
|-- /System
|   |-- /Custom Rules
|   |-- /HTTPRoot
|   |-- /Icons
|   |-- /Legacy Rules
|   |-- /Libraries
|   |-- /Logging
|   |-- /Object Definitions
|   |   |-- /Accounts

```

```

|      |      |-- /Appointment
|      |      |-- /Contact
|      |      |-- /Document
|      |      |-- /Expense
|      |      |-- /History
|      |      |-- /Invoice
|      |      |-- /Involved
|      |      |-- /Line Item
|      |      |-- /Task
|      |      |-- /<CustomObjects> ...
|      |-- /Portal
|      |-- /Screens
|      |-- /StartUp
|      |-- /Tools
|
|-- /Users
|      |-- /<UserNames> ...

```

**Caution:** To avoid causing problems with your system, do not alter the system folders. The only folders that you should delete, move, or rename are subfolders that you created or the folders of custom object definitions.

This section describes the directory structure so that you can properly manage the files and folders related to records and to your system design. It includes the following topics:

- [Top Level Folder](#)
  - [Attachments Folder](#)
  - [HTTPRoot Folder](#)
  - [System Folder](#)
  - [Users Folder](#)
- [Automatically Created Folders](#)

For specific details about using the **Documents** area pages, see the *User Guide*.

### 2.9.1.1 Top Level Folder

At the top level of the TeamConnect document directory structure, there is a folder called **Top Level**. The **Top Level** folder contains the following folders:

#### Subfolders under the Documents Top Level Folder

Folder	Description
<b>Attachments</b>	Contains folders that store attached documents for each record in TeamConnect. Attachments to a specific record can also be accessed through the <b>Documents</b> page of the record.



	<p><b>Note:</b> These folders are automatically created when a user accesses the <b>Documents</b> tab of a record for the first time. The folder is created whether or not a user uploads a document to the corresponding record.</p> <p>For more details about the <b>Attachments</b> folder, see <a href="#">Attachments Folder</a>.</p>
<b>HTTPRoot</b>	<p>May contain scripts, custom style sheets, or images to be referenced by custom blocks via HTTP. You can create subfolders under this folder to better organize your items. For more details about the <b>HTTPRoot</b> folder, see <a href="#">HTTPRoot Folder</a>.</p>
<b>System</b>	<p>Contains several folders that store all of the design information for your application including blocks (sections), Document Generator templates, class files for rules, and so on. For more details about these folders, see <a href="#">System Folder</a>.</p>
<b>Users</b>	<p>Contains a folder for each TeamConnect user where personal files may be stored. When a user clicks the <b>Documents</b> tab, the user's individual Documents folder is automatically displayed.</p> <p>For more details, see <a href="#">Users Folder</a>.</p>

#### 2.9.1.1.1 Attachments Folder

The **Top Level/Attachments** folder contains automatically-created folders for all system and custom objects in TeamConnect (except for embedded objects, because their records cannot have attached documents).

Each object's folder contains subfolders for every record created for that specific object. The subfolders are named after the records whose attachments they contain, according to these patterns:

- Folders for system object records, such as Accounts, Contacts, and Expenses, are named using the name of the record and its primary key. For example: David\_Hendrikson(44).
- Folders for custom object records, such as claims, litigations, and so on, are named using the Unique ID of the record, according to the Unique ID settings in the object definition. For example, if claims are set to auto-numbering, then the folder is named using the number of the claim record.

You can use the **Attachments** folder to navigate to the specific attachments of a record; however, you may find it easier to open the record and click its **Documents** tab.

#### 2.9.1.1.2 HTTPRoot Folder

The **Top Level/HTTPRoot** folder stores scripts, style sheets, and image files that a custom block or block template can reference through HTTP. You can also create subfolders under this folder to organize your files better.

### 2.9.1.1.3 System Folder

By default, the Top Level/System folder contains the following folders:

- [Authentication](#) (requires manual creation)
- [Custom Rules](#)
- [Icons](#)
- [Legacy Rules](#)
- [Libraries](#)
- [Logging](#)
- [Object Definitions](#)
- [Portal](#)
- [Screens](#)
- [StartUp](#)
- [Tools](#)

## Authentication

If you use an authentication plug-in with TeamConnect, you must manually create the **Top Level/System/Authentication** folder.

Afterward, for each authentication plug-in you plan to use, you must manually create the following sub-folders:

**Top Level/System/Authentication/<Plug-in\_Name>**

**Top Level/System/Authentication/<Plug-in\_Name>/classes**

**Top Level/System/Authentication/<Plug-in\_Name>/pages**

The folder specified by *<Plug-in\_Name>* should be the unique four-letter code specified in the plug-in's class file. Upload the authentication plug-in file to the corresponding authentication plug-in subfolder (for example, **Top Level/System/Authentication/NTLM**).

Use an authentication plug-in's subfolders as follows:

- **classes**—Stores necessary Java class files
- **pages**—Stores necessary XML files

For more information about installing plug-ins and a full description of the **Authentication** folder's contents, see the specific document for your authentication mechanism, which is provided by Mitratech.

**Important:** Make sure to replace the folder name examples, *AuthMechanism1* or *AuthMechanism2*, with the unique four letter code specified by the plug-in class file used in your environment.

## Custom Rules

The **Top Level/System/Custom Rules** folder is a common folder used for storing Java files that are shared by multiple object definitions. Files in this folder can be referenced by:

- Custom action rules
- Scheduled action rules
- Wizard page actions
- Wizard page transition rules

For example, if three different custom objects require the same Java files for their custom action rules, then you can use this folder to upload the necessary Java files for all three custom objects.

Files in the **Custom Rules** folder do not become available on the **Qualifiers** or **Action** tab of rules. The **Custom Rules** folder can store only supporting files that may be referenced by custom qualifiers or actions for specific object definitions.

## Icons

The **Top Level/System/Icons** folder contains images used by various objects in TeamConnect. You can upload images to this folder to be used as icons that appear in the corner of object records and in the button for the record on the TeamConnect task bar.

The images in this folder that are in web-friendly formats (JPG, GIF, and so on) are listed in the **Icon Image File** drop-down list on the **General** page of all object definitions.

If no image files are stored in this folder, each object will use its default image.

## Legacy Rules

The **Top Level/System/Legacy Rules** folder is used for storing any Java class files for rules that were written for TeamConnect version 1.6.x. Previously, each object definition had its own **Legacy Rules** folder for storing these files. This centralized folder allows the files to be shared by multiple objects, instead of being uploaded to each individual object definition. This folder is not applicable if you do not have any TeamConnect version 1.6.x rules.

## Libraries

The **Top Level/System/Libraries** folder allows developers to deploy JAR files outside of the **TeamConnect.war** file or the **WEB-INF/lib** folder to implement a custom solution. The class loader that loads classes from this folder is the parent of the following types of custom Java code:

- Custom appenders
- Custom Java blocks
- Custom portlets
- Custom tools
- Rules
  - Automated qualifiers
  - Automated actions

- Scheduled actions
- Wizard initializers
- Wizard page transitions

The JAR files uploaded to this folder do not override the ones included in the TeamConnect WAR file or those that are present in the application server's classpath.

It is not necessary to restart TeamConnect if new JAR files are uploaded to the **Libraries** folder because they are automatically added to the class path. However, if JAR files are updated, you must restart TeamConnect in order for the changes to be applied.

## Logging

The **Top Level/System/Logging** folder contains any Java class files used for modifying the log level that triggers a log message to be sent via email.

By default, the log defined by an SMTP appender is emailed when an event with the level of **Error** is generated. However, you can [set the appender to email the log](#) at a different log level.

## Object Definitions

The **Top Level/System/Object Definitions** folder contains subfolders for each object definitions, including system object and custom object definitions. Each subfolder contains all of the files needed for the corresponding object's design, such as blocks, Document Generator templates, rule class files, and so on. The subfolders for each object are automatically created when the object is saved.

The following folders are found in each system object's folder—**Document Templates, Rules, Screens, and Template Folders.**

Two additional folders, named after the **Involved** object, are also found in the folder of any custom object that has these objects defined for it.

### Object Definition Subfolders

Folder	Description
<b>Document Templates</b>	Contains the document templates that have been created and uploaded for use with <a href="#">the Document Generator</a> .
<b>Rules</b>	<p>Contains three subfolders:</p> <ul style="list-style-type: none"> <li>• <b>Automated Actions</b>—Contains Java or JavaScript files for the corresponding object definition. These files define an action that will be executed or a method that will be called wherever automated actions are used.</li> </ul> <p>For example, the custom actions or scheduled actions for rules are stored in this folder. Also, the files for wizard page actions and wizard page transition rules are stored in this folder.</p>

	<ul style="list-style-type: none"> <li>• <b>Automated Qualifiers</b>—Contains files that define <a href="#">custom qualifiers</a> for the corresponding object definition. Rules that use these files for their qualifiers can be of any rule type.</li> <li>• <b>Legacy Rules</b>—Contains any Java class files for rules that were written for the corresponding object definition in TeamConnect version 1.6.x. This folder is not applicable if you do not have any TeamConnect version 1.6.x rules.</li> </ul>
<b>Screens</b>	Contains all uploaded XML files that can be used for blocks for the corresponding object. They are available on the object definition's <b>Custom Blocks</b> tab.
<b>Template Folders</b>	<p>Contains the directory structures, including all necessary documents, that have been created for use with object templates.</p> <p>When a record is created using a <a href="#">template</a> or <a href="#">wizard</a>, the directory structure for the template that is stored in <b>Template Folders</b> is automatically added to the <b>Attachments</b> folder of the record.</p>
<b>Involved</b>	(Custom objects only) Contains the folders <b>Document Templates, Rules, Screens,</b> and <b>Template Folders</b> for the Involved Object.

## Portal

The **Top Level/System/Portal** folder allows you to add custom content to [portal panes](#) on your users' home pages. Files placed in this folder can be used to obtain external data to be displayed in a portal pane.

## Screens

The **Top Level/System/Screens** folder stores the following files that apply a common look and feel to [system and custom blocks](#):

- **template.xml**—An XML template that defines the layout of block headers and footers
- **global.css**—A global cascading style sheet that contains definitions used by **template.xml** and custom blocks

## StartUp

The **Top Level/System/StartUp** folder allows developers [to deploy class files that should be executed](#) when TeamConnect starts. This can be useful for installation scripts, to initialize TeamConnect using custom code, and to check consistency in the application for the session.

**Important:** You should limit access to the **StartUp** folder to prevent the addition of malicious code.

To enable the execution of custom code in the **StartUp** folder, you must set the app.runStartUp application parameter to YES in the **web.xml** file.

If you want to stop the execution of all class files in the **StartUp** folder in the event of any runtime errors or exceptions, you can set the `app.runStartUp.failOnError` parameter to YES.

TeamConnect finishes its standard initialization before it executes the class files in the **StartUp** folder. Your custom code can affect TeamConnect's initialization after the standard initialization and before TeamConnect becomes available via a web browser.

This start up functionality only supports Java class files. TeamConnect ignores non-Java class files and any sub-folders to the **StartUp** folder. However, you can include JAR files and reference to them from your class files. You can also upload Java classes that may be referenced from the executing class.

If multiple class files are present in the **StartUp** folder, TeamConnect executes them according to the alphabetical order of the file names.

## Tools

The **Top Level/System/Tools** folder contains a subfolder for each custom tool that has been uploaded to TeamConnect. Each custom tool's folder contains two subfolders:

- **classes**—Stores necessary Java class files
- **resource**—Stores necessary XML files

You must manually create these folders when you [create a custom tool](#).

### 2.9.1.1.4 Users Folder

The Top Level/Users folder contains a <username> subfolder for each user where their personal files may be stored. When a user selects Documents from the Go to menu, their personal Top Level/Users/<UserName> folder opens by default. Each user can create subfolders in their personal folder, provided that they have been given the rights to do so.

## 2.9.2 Automatically Created Folders

The following table summarizes the folders that are automatically created when you create items in TeamConnect.

Certain folders are created automatically, the first time a user accesses the folder, rather than when the item itself is first created by a user. This helps keep the database clear of unnecessary folders.

### Automatically Created Document Folders

New item created in TeamConnect	When folder is created	Path of automatically created folder	How folder is named	Folder contents
User	When users first access their personal folder by selecting	Top Level/ Users/ UserName  For example:	Username (login name) specified in user account	Folder is empty upon creation.  After the folder is created, it may

	<b>Documents</b> from the <b>Go to</b> list	Top Level/ Users/Jack		contain documents uploaded by and subfolders created by the user.
Custom object (including Embedded objects)	Upon creation of the custom object	Top Level/ System/Object Definitions/ ObjectName  For example:  Top Level/ System/Object Definitions/Case	Name of custom object as defined in the object definition	<p>Folders for files needed for the object's design, including:</p> <ul style="list-style-type: none"> <li>• <b>Document Templates</b></li> <li>• <b>Rules</b></li> <li>• <b>Screens</b></li> <li>• <b>Template Folders</b></li> </ul> <p>For custom objects, the folders for Involved are stored here, if they are defined for the object.</p> <p><i><b>Note:</b> Embedded object folders do not contain any of the subfolders listed above except for Screens.</i></p>
Involved Object	Upon creation of the Involved Object	Top Level/ System/Object Definitions/ CustomObjectName/ InvolvedName	Name of Involved object as defined in the object definition	<p>Folders for files needed for the Involved object's design, including the following:</p> <ul style="list-style-type: none"> <li>• <b>Document Templates</b></li> <li>• <b>Rules</b></li> <li>• <b>Screens</b></li> <li>• <b>Template Folders</b></li> </ul>

Record created without Template files or folders	When a user first opens the record's <b>Attachments</b> folder by opening the record's <b>Documents</b> tab	Top Level/ Attachments/ ObjectName/ RecordName  For example:  Top Level/ Attachments/ Account/ Jones_Account _0402	<ul style="list-style-type: none"> <li>• <b>System object records</b>— named using the record's name and its primary key</li> <li>• <b>Custom object records</b>— named using the Unique ID of the record, according to the settings in the object definition</li> </ul>	<ul style="list-style-type: none"> <li>• Folder is empty upon creation.</li> <li>• After the folder is created, it may contain any documents uploaded, or folders created, that are attached to the record by being added through its <b>Documents</b> block.</li> </ul>
Record created with Template files or folders	Upon creation of the record.	Top Level/ Attachments/ ObjectName/ RecordName/ FolderName  (or FileName)  For example:  Top Level/ Attachments/ Account/ Settlements_Ac count_0402/ TaxForms		<ul style="list-style-type: none"> <li>• Upon creation, the folder includes files or folders that are automatically attached to the record because they are included in the Template.</li> <li>• After creation, the folder may contain any documents uploaded or folders created through the record's <b>Documents</b> block.</li> </ul>

## 2.10 Glossary

### A

#### **accounts**

Entities used for tracking financial accounts or budgets electronically. Accounts are typically defined for transactions related to a project. In addition, posting criteria are typically defined per account (which financial transactions can post to the account).

#### **Admin Rights**



These are access rights to features available from the **Admin** tab in the user interface. These include user records, group records, Admin Settings area, and logger rights.

### ***Administrator (TeamConnectAdmin user)***

A user (such as TeamConnectAdmin) whose job responsibilities include creating or maintaining user and group accounts, specifying the appropriate admin settings for the organization, and managing logs.

### ***appender***

A definition of a system or audit log. Appenders can be of various types, including SMTP, socket and file. File appenders are the most commonly used, as the logging statements are simply captured in a file. A log can be a general TeamConnect log, or it can be associated with a specific *logger*, so that it captures messages only from a certain TeamConnect area, such as portal panes or custom blocks.

### ***applet***

A small utility written in Java. In TeamConnect, applets can be used for the document screen interface and the hierarchical item selector. If applets are enabled on the system, users can enable them for their own use on the **Preferences** tab of the **Preferences** screen.

### ***Application Programming Interface (API)***

A set of object classes that expose public methods the TeamConnect system.

A language and message format used by a program to communicate with an operating system, database management system, communications protocol, or some other control program. For TeamConnect, the Web Services API provides services equivalent to end-user functionality in the application user interface (UI).

### ***application server***

In a TeamConnect environment, the application server hosts the TeamConnect .WAR or .EAR file. In TeamConnect properties files, the application server might also be called the host.

### ***Appointment***

A *System Object* that allows users to track their scheduled events, engagements, or meetings. Appointments are automatically available for custom objects.

### ***Approval***

In TeamConnect workflow process, an approval is an authorization action performed by a approver. After all required approvers perform approval actions on workflow requests, the original TeamConnect operation completes.

### ***Approvers***

In TeamConnect workflow process, an approver is a user with the authority to approve or reject certain actions attempted by requestors. There can be several approvers at each level of an *Approval* process.

During an approval process, approvers may be able to send the approval request to reviewers for review before they make their decision. Approvers can view their approval requests on the **My Approvals screen**.

### ***Assignees***

TeamConnect *Users* who are assigned to one or more Projects or Tasks. There can be one or several users assigned to a project. In projects each assignee can have a different role, such as attorney, agent, appraiser, paralegal, and so on. Typically, the main assignee is in charge of the whole project, whereas the rest of the assigned users are responsible for certain parts of the project. Tasks can have only one current assignee.

### ***Attendees***

TeamConnect Users and members of Contact Groups who are expected to take part in an event, such as a meeting or conference call, for which there is an Appointment scheduled.

### ***authorization***

Permission to access a resource controlled by a server. Authorization typically occurs after a server evaluates the access control list (ACL) associated with a resource. The types of events that require authorization to TeamConnect include user login, running TeamConnect rules, and running Web Service requests.

## **B**

### ***Block***

A group of system and/or custom fields included into a single file, XML (for custom blocks) or JSP (for system blocks). In the end-user interface, each block is typically displayed as a separate tab or a section on the record screen.

TeamConnect *Object Views* provided by Mitratech consist of system blocks, such as **General**, **Categories**, **Workflow**, **Assignees**, **Attendees**, and so on. You can also create your own Custom Blocks using *Custom Fields*.

## **C**

### ***Category***

Lookup items that allow you to organize custom fields into blocks or sections on the screen for the **Main** user interface. By adding or deleting categories in a record, users display or hide respectively the associated blocks of details in that record. Categories also help users to organize their records by certain types, for example, employee contact records vs. non-employee contact records, domestic vs. international accounts, and so on.

### ***Category Rights***

See [Category Security](#).

### ***Category Security***

The *Rights* to the object's categories and their custom fields. This is the second level of security, after the object level. All rights assigned at this level take effect only if the access to the object itself is granted.

Category-level rights add more granularity to the object-level rights. A user might have access to the object, but you can still control access to the organization-specific information stored in the custom fields and displayed in forms or custom blocks.

## **CLASSPATH**

An environment variable or argument set on the command-line that tells the Java Virtual Machine where to look for user-defined classes and packages in Java programs.

## **COD**

Custom Object Definition

## **Contact**

A *System Object* definition used to create a TeamConnect record for an individual, organization or group within a organization (such as department). Contacts can be organized into groups by creating *Address books*.

## **CSM**

Collaborati Spend Management

## **Custom Fields**

Custom fields can be created per system or custom object. These fields allow you to meet the individual needs of your organization.

## **Custom Object**

*Business Objects* created for your organization by a TeamConnect Solution Developer TeamConnect Certified Professional. [Custom Objects](#) can have names used within your industry for files, such as matters, claims, litigation, or policies.

The individual instances of Custom Objects are sometimes referred to as Projects or Matters within this documentation.

## **Custom View**

See *Object View*.

## **D**

## **DataSource**

An object that provides the means by which connection pooling may be neatly hidden and performed. A DataSource object is a factory for Connection objects. An object that implements the DataSource interface will typically be registered with a JNDI service provider. A JDBC driver that is accessed via the DataSource API does not automatically register itself with the DriverManager. This is the preferred way of communicating with a data source (for example, a database) and provides a good alternative to using a DriverManager.

### ***Debug Settings***

Refers to a scenario when an administrator changes a logger's logging level to Debug. This results in the highest possible detail level of output to the log file. Setting a logger's logging level to Debug is useful to help troubleshoot an issue but afterward you should change the logging level to Warning (or a less verbose setting) to maintain optimal system performance.

### ***Document***

A *System Object* definition used to manage files, folders, shortcuts, or hyperlinks in the TeamConnect Documents area. Documents are usually "attached to" other TeamConnect Records by uploading a file from the Documents page of a TeamConnect record.

Additional files necessary for customization of TeamConnect are also uploaded as Documents. These include XML files for *Custom Blocks*, *Custom Actions* for Java *Rules*, files included in *Templates*, and other files related to your customized system design.

## **E**

### ***end-user***

A person who uses TeamConnect in a non-administrator or non-Solution developer capacity. This user type would typically not have rights to the **Admin** tab or the Designer application area.

For example, an end-user would typically have rights to work with system objects such as Documents, Contacts, and may have rights to work with custom objects such as legal Disputes. An end-user would typically have rights to create, edit, view, search, and delete contacts in TeamConnect.

### ***Embedded Objects***

Simple Custom Objects created as children of other *Custom Objects (parent objects)*. An Embedded Object can only exist within the context of a parent Custom Object. Within *Wizards* and *Templates*, embedded objects are considered *Related Objects*.

### ***Expense***

A *System Object* definition that is intended to allow *Users* to track internal costs of doing business in their organization. Expenses are automatically available as a system *Block* in *Object Views* for *Custom Objects*.

## **F**

### ***Full-Text Search***

Allows you to search for items in the database according to their content. TeamConnect provides the feature of content searching in uploaded documents, and certain fields in *Records*, if Microsoft® SQL Server Full-Text Search or Oracle® Text (interMedia) is available on your database server. For

example, using content searching, you could search the content of uploaded documents to find a file in which a certain person's name is mentioned. This feature is available in search screens if the appropriate fields are included in the corresponding *Search View*.

## G

### **Group**

Typically refers to Group accounts (used in TeamConnect account administration to group multiple users with similar access rights or day-to-day business responsibilities). A group could also refer to a *Contact Group*.

### **Group Rights**

Access *rights* to features and application areas that can be accessed in the user interface. These primarily include the rights to all *System Object* and *Custom Object* records, and certain *Tools*, such as Scheduler, Batch Task and Expense Entry. Rights are assigned by group account.

## H

### **History**

A *System Object* definition used to create notes associated with a specific Record. History records can only be created in relation to an already existing record. You can search for a history record from a global History search page.

### **Home Page**

The first screen that users see after logging in to TeamConnect. Home Pages function as a starting point for all TeamConnect users and can be personalized to fit their individual needs. Users can access various types of information or do different actions from *Portal Panes* that are available on the Home Page.

### **host**

A computer that hosts (is the source of) information or services used by other devices.

### **hostname**

The logical name of a computer, such as machine.domain.domain, which is translated into an IP address.

### **HTTPS**

A secure version of HTTP, implemented using the Secure Sockets Layer (SSL).

**HyperText Transfer Protocol (HTTP)**

The method for exchanging information between HTTP servers and clients.

**I****internationalization**

See [Localization](#).

**Internet Message Access Protocol (IMAP)**

A standard mail server used on the Internet. It provides a message store that holds incoming email until users log in and download it.

**Invoice**

A *System Object* definition used to electronically track invoices or bills sent by *Vendors*. Invoices are typically billing records for goods or services provided by 3rd party vendors to TeamConnect customers.

**Involved**

A *System Object* definition used to keep track of various involved parties associated with a *Project*. An Involved object is always related to a *Custom Object*.

**J****JAR**

(Java Archive file) A file type packaged with the ZIP file format, for data compression. Libraries shipped with TeamConnect are typically distributed in .JAR format.

**JDBC**

Java database connectivity

**JRE**

Java runtime environment is a software bundle from Sun Microsystems that allows a computer system to run a Java application. The bundle includes the JVM and API. The API provides a set of standard class libraries. You can download the current JRE version by testing their JVM at the Java.com test page.

**Java Virtual Machine (JVM)**

A virtual machine is a simulated computer; it runs on a host computer but behaves as if it is a separate computer. The Java Virtual Machine works as a self-contained operating environment to run Java applets, with the advantage that Java applets can be run independently from the host operating system.

## L

### ***Lightweight Directory Access Protocol (LDAP)***

A protocol used to access a directory listing. LDAP support is implemented in browsers and email programs, which can query an LDAP-compliant directory. In a TeamConnect environment, if you already have a 3rd party LDAP Server set up, you can configure TeamConnect user authentication to occur through your LDAP Server.

### ***Localization***

The process of customizing system and custom data for multiple languages and regions. In TeamConnect, you can localize the user interface and features, such as the time and data display. To localize in TeamConnect, you export the English data to an Excel spreadsheet, enter the translated data into corresponding fields, and import the translated data back into TeamConnect. After you import the data, the translated elements appear in your preferred language.

The term *localization* can be confused with *internationalization*, which involves selecting a language at runtime.

### ***Line Item***

A *System Object* definition used to list the goods or services specified in an *Invoice*. The details of each line item, such as the type (fee or expense), price, quantity, and dates must be entered in the Line Items screen of the corresponding invoice. Line items must be created from an existing invoice.

### ***Logger***

A component for a specific area of TeamConnect for which a system or audit log can be defined. For example, TeamConnect has loggers for portal panes, custom blocks, the XML layer, and user login/sign off. When you define a log (or *appender*), you select which logger's messages it captures.

### ***Lookup Table***

Tables that organize, store and quickly access multiple items, such as contacts' roles, types of address, phone, fax and other contact information, and so on. In addition to adding, renaming, inactivating, and deleting items in system lookup tables, you can define new custom lookup tables and their contents. In the end-user interface, lookup tables are represented by either drop-down lists (most commonly) or radio buttons, where the user can look up the necessary information and make the appropriate selection.

Users have the option of displaying item hierarchies in a regular drop-down list or in a Java applet called a hierarchical item selector.

## N

### ***NTLM***

NT LAN Manager. Former name for Windows Integrated Security. In a TeamConnect environment, you can configure TeamConnect user authentication to occur through a Microsoft Internet Information Services Server.

## O

### ***Object Definition***

A specification of all properties of a *System Object* or *Custom Object*. A TeamConnect Solution Developer TeamConnect Certified Professional may edit a System Object Definition to create custom rules, create additional search views, customize additional object views, create wizards, or create custom blocks, among other operations. A TeamConnect Solution Developer TeamConnect Certified Professional would create a new object definition for a custom object.

Properties that are part of an Object Definition include its name, icon, categories and custom fields, look and feel of corresponding records, search views, wizard contents for creating its *Records*, and associated rules, among other properties.

### ***Object View***

A layout definition of tabs, sections/blocks, and other user interface elements per object definition. For an object definition, you can create multiple object views as an alternate method of customizing the available system and custom blocks for an object's records. For each group account, you can set each object definition's default Object View as an alternate method of setting record rights.

## P

### ***Portal Pane***

A component of a *Home Page*, which represents a specialized menu that allows users to access different information or do specific tasks directly from the Home Page.

### ***Preferences***

System preferences that a user can set for his individual TeamConnect session. Preferences include color scheme and time zone, among other settings. Preferences settings are available from the **Preferences** link in the upper right corner of the user interface.

### ***Process Manager***

In TeamConnect workflow process, a TeamConnect group whose members are responsible for monitoring approval processes that are sent to approvers. Process Managers receive notifications of approval requests when there is an error. They may also be notified when a request is rejected or



expired. They can restart an approval process to resolve errors, rejections, or expirations, or reassign the task of approving a request to other users when necessary. They can also reject approval requests.

### ***Project***

Refers to a Record or instance of a *Custom Object*.

## **R**

### ***record***

An instance of a *System Object* or *Custom Object* in TeamConnect, created and accessed in the user interface.

### ***Record Security***

Access protection set at the level of individual object records. For example, you can make a *Record* Private or Public or you can grant or deny the rights to read, update, and delete a record to other *Users* or *Groups*. This level of security can be set only on the **Security** tabs of individual records in the end-user interface. This is the lowest level of access rights. All rights assigned at this level take effect only if the appropriate rights are assigned at the object level.

### ***request***

In TeamConnect workflow process, an attempted operation by a user that is [initially pending approval](#) before the original operation can complete.

### ***requestor***

In a TeamConnect workflow process, [a user who performs an action](#) that triggers an approval process, such as posting an *invoice*. The *request* is sent to *approvers* according to the approval route that is defined for that workflow process.

### ***reviewer***

In TeamConnect workflow process, a user who reviews an approval request. An approver designates the reviewer for a request.

### ***rights***

Access permissions to records (categories, custom fields) and functional areas in TeamConnect. In TeamConnect, access rights are applied to enforce record or system security, preventing users from viewing or changing records/settings they should not be able to access. Rights are assigned to *Group accounts*.

**Route**

In TeamConnect workflow process, a defined hierarchy of groups whose members' *Approval* is required to authorize a requested operation. It is an approval path that can consist of one or more nodes, called *Stops*, that can contain one or more *Members* who can approve or reject the operation. Routes are used for the Workflow purposes, to control the actions that take place in your business processes.

**Rules**

TeamConnect allows you to create rules that control whether the operations attempted by users are in accordance with your business processes. A rule is triggered when the operation is attempted, and the *Qualifiers* are checked to determine what should occur. For example, rules can ensure that certain *Custom Fields* are filled out if a particular *Category* is selected.

*Approval rules* postpone the operation until all members in the approval process, or *Route*, have approved the operation.

Rules written using Java class files or JavaScript can also perform *Automated Actions* when they are triggered.

**S****Search Module**

A special utility that allows the user to quickly search for a specific record and link it to the record with which you are currently working without leaving its screen. The [Search Module](#) is represented by a field with two icons, **Find** and **Open**, next to it.

**Secure Sockets Layer (SSL)**

An industry standard protocol developed by Netscape for securing communications. It works by using public-key cryptography and certificates.

**Service Of Process (SOP)**

The procedure employed to give legal notice to a person (defendant etc.) of a court or administrative body's exercise of its jurisdiction over that person so as to enable that person to respond to the proceeding before the court, body or other tribunal. Usually, notice is furnished by delivering a set of court documents to the person to be served.

**SQL\*Plus**

A command line SQL and PL/SQL language interface and reporting tool that ships with the Oracle Database Client and Server. It can be used interactively or driven from scripts.

**Structured Query Language (SQL)**

A computer language used to create, modify, retrieve and manipulate data from relational database management systems and object-relational database management systems. It is an ANSI/ISO standard.

### **System Object**

Default *Business Objects* provided by Mitrtech to your organization. They represent business objects that are common to most business models regardless of the specific industry.

TeamConnect System Objects include: *Account, Contact, Appointment, Expense, Invoice, Line Item, Task, History, Document, Involved, User account, and Group account.*

## **T**

### **Task**

A *System Object* definition used to track *internal* assignments, usually related to *Projects*. For example, preparing reports, following up on cases, reviewing documentation, and so on.

### **Template**

This term can refer to the following:

- Document templates
- Templates used with Wizards
- Templates used with rules
- Templates used with wizards and rules

A document template is a template designed for use with the Document Generator. You can use this capability to generate documents from Records using certain fields, such as the name of the contact associated with the record.

Templates used with wizards or rules define default values that should be automatically filled in for certain fields, *Sub-objects*, and *Related Objects*. For more details on templates, see [Creating Custom Pages](#).

### **tool**

There are system tools that come by default with TeamConnect; however, it is also possible to [create custom tools](#).

### **transaction**

Events or happenings in your business that change its financial position, such as buying supplies, paying bills, withdrawing money, or buying equipment. For example *Expenses, Tasks, and Invoices* and regular money withdrawals from and transfers between *Accounts* incur transactions against accounts.

## U

### *User*

An account created for a person who logs into TeamConnect. Users must be members of *Groups*.

### *User Interface (UI)*

The graphical representation of TeamConnect on the computer screen, where the user can click buttons and hyperlinks, make selections from drop-down lists, enter text into fields, and so on.

## V

### *Version Control*

A method of tracking the edits made to a document that is uploaded to TeamConnect. Version control allows users to access previous versions of a document, and even revert the current version back to a previous version if needed. Version control must be activated by the TeamConnect *Administrator*.

## W

### *Workflow*

In TeamConnect, [workflow](#) is defined through Approval Rules at the Object Definition level.

For example, in a financial setting, a financial officer may trigger [an approval rule](#) by attempting to delete an account. A corresponding approval request would be sent to an accounting manager (approver) who would approve or reject the request.

## X

### *XML (extensible markup language)*

In TeamConnect environments, XML is commonly used as the file format for properties files. When creating custom blocks in the TeamConnect Designer application area, you upload .XML files to designated screens folders (under TeamConnect Documents) that define a custom block.





















### 3 TeamConnect Setup and Development

Welcome to the *TeamConnect® Setup and Development Help*.

I use TeamConnect as an...	I need help with...
IT Manager	<a href="#">Enterprise Setup and Design</a> <a href="#">Legal Setup</a> <a href="#">Collaborati Management Setup</a> <a href="#">API Development</a> <a href="#">Building Applications</a>

#### 3.1 Enterprise Customization Help

Welcome to the *TeamConnect® Enterprise Customization Help*.

 <a href="#">System Settings</a>	 <a href="#">Objects</a>
 <a href="#">Categories</a>	 <a href="#">Lookup Tables</a>
 <a href="#">Custom Pages</a>	 <a href="#">Search Views</a>
 <a href="#">Home Pages</a>	 <a href="#">Portal Panes</a>
 <a href="#">Global Navigation</a>	 <a href="#">Templates</a>
 <a href="#">Wizards</a>	 <a href="#">Rules</a>
 <a href="#">Conditional Expressions</a>	 <a href="#">Automated Qualifiers</a>
 <a href="#">Matter Management Design</a>	 <a href="#">Localization</a>
 <a href="#">Custom Designs</a>	 <a href="#">Document Generator</a>
 <a href="#">EasyDocs</a>	 <a href="#">Object Navigator</a>

### 3.1.1 Introduction to Customization

Developing your custom TeamConnect design involves research and planning. Business analysts gather the requirements to represent your organization's business model, and technical analysts develop the corresponding functional specifications.

Be sure to develop or obtain all of the necessary design documents before you and your team begin the customization process. Once your design is completed, you can set up and configure TeamConnect to translate the design into a fully functional application to meet your organization's needs.

Before you start implementing your design in TeamConnect, do the following operations:

- Install TeamConnect according to the requirements.
- Create your own user account and ensure that it belongs to a user group with **Setup** rights.
- Familiarize yourself with TeamConnect and its Designer user interface.
- Plan your implementation strategy according the order, process, and dependencies listed in [Customization Sequence](#).

#### 3.1.1.1 The Meaning of Design

As used in this document, the word "design" has several definitions:

- The abstract models and requirements that your organization develops before implementing TeamConnect.
- The collection of object model definitions, rules, templates, and other system components that results from your implementation of those abstract models and requirements in TeamConnect.
- That part of the TeamConnect database that defines the system components only, excluding data entered by end users. That is the specific definition used by [Migrating Custom Designs](#).

Most of this document uses the second definition in the list above.

#### 3.1.1.2 Preparing User and Group Accounts

Even though users and user groups are not administered in the Designer, it is important to have a well-developed structure of groups already in place before doing customization. Some of the tasks in customization, such as assigning object rights and designating workflow stops in routes, require that the appropriate groups already exist.

Even if you are testing with only a small number of users, the structure of your groups should reflect how TeamConnect will be used in full production. It is relatively simple to add many more users to existing groups later, in production, without requiring any further customization.

You will need your own user account if one has not already been made for you during the installation process. Your account must belong to a group that has **Setup** rights. You can also get access to Setup if your user account is designated as a "superuser". However, for realistic testing, it is preferable that your user account is designated as "normal", with appropriate group memberships and Setup rights.

Consult the person who installed TeamConnect to see if you already have such a user account. If not, have [an account](#) created for you.

As you customize TeamConnect by adding custom objects or tools, assign rights to them to the appropriate user groups, including your own user group. Otherwise, you will not be able to access and test them.

### 3.1.1.3 TeamConnect Designer User Interface

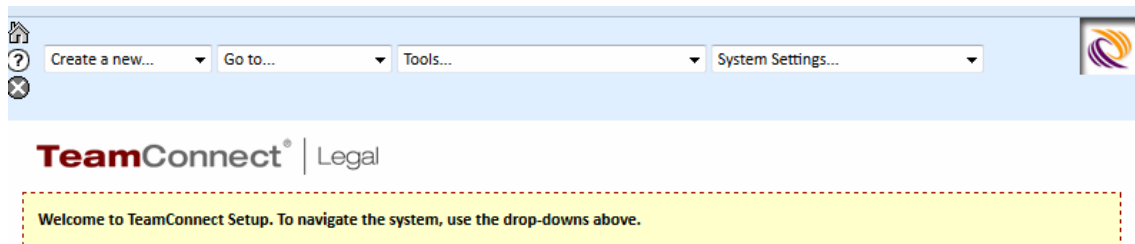
Most customization tasks do not require extensive technical skills because you perform them using the TeamConnect Designer user interface. For example, you can develop most business rules directly through the user interface.

Depending on your design, you may also need certain technical skills, such as a knowledge of Hypertext Markup Language (HTML) and Extensible Markup Language (XML), to create custom pages. Some customization of rules and pages requires expertise with Java and JavaScript.

#### To access the Designer user interface

1. Log in to TeamConnect with Setup rights.
2. Click the Setup link at the top of the page (upper right).

The main access page of the Designer user interface opens in a separate window, as shown in the following image.



Designer Main Page

#### 3.1.1.3.1 User Interface Overview

Most Designer user interface elements are self-explanatory. However, some procedures require knowledge of TeamConnect itself.

For example, you may define patterns to automatically name records, such as the date on which the record was created, the record's default category, and the name of the main involved party. To create such patterns, you must identify the location of these attributes within TeamConnect's object model (see the various Object Model appendices for details). Thus you would require Object Model knowledge for this task.

The Designer user interface has the following types of screens:

- Record Screens—Most Designer configuration screens include tabs and a toolbar. For example, object definition screens include tabs where you can create and define the corresponding components, such as the **Templates** and **Wizards** tabs of the Account object.

Unlike the end-user interface, the Designer user interface does not have the Read-only display mode. Most fields in the screens are editable at all times.

- **Batch Screens**—Many configuration screens include batch screens, which have a row of data entry fields for entering multiple items at the same time. For example, the **Lookup Tables** screen allows you to enter the names and tree positions of dozens of lookup table items at the same time.
- [Using Object Navigator](#)—A user interface tool for defining paths to attributes in the object model. See [Using Object Navigator](#) and [Object Model: Read This First](#) provides information about the objects, attributes and relationships in TeamConnect's object model that you can use to create your custom designs. This reference points you to additional reference tables that provide more information about object models.

While developing your design, you are testing it in the end-user interface as well. You may also need to use that interface to create and define [users](#) and [groups](#), access rights, [system settings](#), and so on.





### 3.1.1.3.2 Designer Menu Bar

The Designer menu bar contains a set of drop-down lists, which provide a quick access to the appropriate records and functions.



Designer Menu Bar

#### Designer Menu Bar Fields

Field or control	Icon or option	Description	
Logo		TeamConnect logo. When the image in the icon is "spinning," it indicates that the system is processing a request, such as opening a record, searching for a records, or another operation.	
Home		Opens your <b>Home</b> page. For details on home pages, see <a href="#">Creating Home Pages and Portal Panes</a> .	
Help		Provides access to the Designer help files.	
Sign Off		Allows you to sign off from TeamConnect..	
Create a new	<b>Object Definition</b>	Displays the <b>General</b> tab of the corresponding screen.	For details on how to define objects, see <a href="#">Creating and Defining Objects</a> .
	<b>Route</b>		For more information on creating routes, see the <a href="#">Creating Routes</a> .



	<b>Home Page</b>		See <a href="#">Home Pages</a> .
	<b>Portal Pane</b>		See <a href="#">Portal Panes</a> .
<b>Go to</b>	<b>Object Definitions</b>	Displays a list of available system and custom objects. Click the appropriate hyperlink and the corresponding object definition screen appears with its <b>General</b> screen opened by default. For details on the information displayed in this screen, see <a href="#">General System Object Information</a> and <a href="#">General Custom Object Information</a> .	
	<b>Routes</b>	Click the appropriate hyperlink and the <b>General</b> tab of the corresponding screen appears by default.	For more information on routes, see <a href="#">Creating Routes</a> .
	<b>Home Pages</b>		See <a href="#">Home Pages</a> .
	<b>Portal Panes</b>		See <a href="#">Portal Panes</a> .
	<b>Lookup Tables</b>	Displays the <b>System</b> tab of the tables screen, where you can add, modify, inactivate, and delete the necessary items in the appropriate system tables. For more details, see the <a href="#">Types of Lookup Tables</a> .  For details on creating custom lookup tables and using them in custom fields of type List, see <a href="#">Using Lookup Tables</a> .	
	<b>Global Navigation</b>	Click the hyperlink to show the tabbed work screen.	See <a href="#">Using Global Navigation</a> .
<b>Tools</b>	<b>Administer Custom Tools</b>	Displays the <b>Administer Tools</b> screen where you can add or delete custom tools, and rename system tools. See <a href="#">Creating Custom Tools</a> .	
	<b>Misc. Settings</b>	Displays the <b>Miscellaneous Settings</b> screen where you can add custom fields for integration of TeamConnect with other products. See <a href="#">Miscellaneous Settings</a> .	
	<b>Contact Sweeper</b>	Run a tool that identifies possible duplicate contacts and allows you to choose which ones to consolidate and delete. See <a href="#">Using Contact Sweeper</a> .	
	<b>Design Snapshot</b>	Run a tool that captures complete information about the object definitions and other TeamConnect components in the current application.	

	<b>Design Import</b>	Run a tool that uses information from Design Snapshot to change the current application design to match the snapshot.	
	<b>XML Worksheet</b>	Launch a tool that allows you to upload XML requests to the application and view the responses.	
<b>System Settings</b>	<b>Rule Execution</b>	Displays the corresponding tab in the <b>System Settings</b> screen, where you can configure the appropriate setting for your application.	See <a href="#">Accessing System Settings</a> .
	<b>Custom Messages</b>	Lets you create custom messages and assign unique messages keys.	See <a href="#">Creating Custom Messages</a> for more information.
	<b>Clear Cache 2.x user interface Settings</b>	Allows you to clear the cache, which may be required when you create custom pages. See <a href="#">Creating Custom Pages</a> .	
	<b>Default Object Views</b>	See <a href="#">Object Views</a> .	
	<b>History</b>	Displays a Search page where you can set your search criteria for the <a href="#">history entries</a> associated with system settings.	
	<b>Backward Compatibility Settings</b>	See <a href="#">Backward Compatibility Settings</a> .	
	<b>System Appenders</b>	Displays the corresponding tab in the <b>Logging</b> screen, where you can <a href="#">define logs</a> and set the status of loggers.	
	<b>Audit Appenders</b>		

#### 3.1.1.3.2.1 Backward Compatibility Settings

Each new version of TeamConnect contains new and updated features. The **Backward Compatibility Settings** page provides several options that let you return to previous behavior until you are ready to shift to the new features.

**Important:** You can check and uncheck these boxes at any time, but it is not recommended for settings that affect how data is stored. Make the switch in a test environment to ensure that everything works properly.

#### To open the Backward Compatibility Settings page

1. Open the Designer.
2. In the **System Settings** drop-down list, select **Backward Compatibility Settings**.
3. On the **Backward Compatibility Settings** page, select one or more of the options in [the Backward Compatibility Settings Page table](#).

**Note:** Most of the boxes on this page are unchecked by default. Adding a check-mark to a box indicates that you want to use the old behavior. This does not apply to the **Optimize cascading security** option, which is checked by default. Please see the details below.

#### Backward Compatibility Settings Page

Control	Options
<b>Allow Duplicate Invoices (Same invoice number, invoice date, year, and vendor)</b>	<ul style="list-style-type: none"> <li>• If checked, an error message appears when posting duplicate invoices.</li> <li>• If unchecked, allows duplicate invoices for the purposes of data migration of duplicate invoices for the Dynamic Fields Conversion upgrade tool and Workflow upgrade tool.</li> </ul>
<b>Enable wild card characters in XML layer search requests</b>	<ul style="list-style-type: none"> <li>• If checked, it is not possible to use wild cards when writing search requests in the XML layer.</li> <li>• If unchecked, wild cards are valid.</li> </ul>
<b>Always use 'ISO- 859-1' for encoding characters entered in text/memo fields</b>	<ul style="list-style-type: none"> <li>• If checked, the default is the ISO/IEC 10646 character set.</li> <li>• If unchecked, the eight-bit coded character set is used in text and memo fields.</li> </ul>
<b>Optimize Cascading security</b>	<ul style="list-style-type: none"> <li>• If unchecked, cascading security proceeds in a single batch with one commit.</li> <li>• If checked, if cascading security is enabled for a custom object definition, the security cascade proceeds in batches of 100 and commits after each batch. Any failures in batches is logged and do not prevent the completion of later batches.</li> </ul>

	<b>Note:</b> This box is checked by default.
<b>Ignore security cascading from document folders</b>	<ul style="list-style-type: none"> <li>• If checked, security propagates based on the Replace Rights on subfolders and/or Replace Rights on existing files sections in the security block of the document folder.</li> <li>• If unchecked, modifying the security of a document folder will not propagate the security changes to its child files and folders.</li> </ul>
<b>Take user to document properties page when clicking on a shared persistent URL of a document--</b>	<ul style="list-style-type: none"> <li>• If checked, clicking a document or document folder URL link <a href="#">in an email message</a> opens the TeamConnect <b>Documents General</b> page for the referenced document.</li> <li>• If unchecked: <ul style="list-style-type: none"> <li>○ <b>For documents</b>—Clicking a document URL link in an email message opens the document in a browser.  If the document URL links to a file that may contain other files, such as a zip file, the browser page opens the TeamConnect Documents page that contains the referenced file.</li> <li>○ <b>For document folders</b>—Clicking a document folder URL link in an email message opens the TeamConnect <b>Documents</b> page that contains the referenced folder.</li> </ul> </li> </ul> <p><b>Notes:</b> Users must be logged in to TeamConnect and have the appropriate rights to access documents, document properties, or document folders.</p> <p>Browser settings determine the way that documents are displayed. For example, if the browser is set to open a new tab whenever a document is opened, clicking a persistent document URL link always opens the document in a separate tab.</p>

#### 3.1.1.4 Customization Sequence

Some customization tasks are sequential and others can be performed concurrently. For example, you cannot create custom fields until you have created the categories to which the fields belong.

Once you are done creating custom fields, you can create rules, define wizards, or define home pages in any order.

Creating custom objects and fully developing their pages, rules, search views, and so forth usually requires the most work. System objects come "out of the box" fully developed, although your organization's design may require some custom pages, wizards, and other extras for system objects.

**Important:** *Collaborati Spend Management (CSM) is delivered as a group of custom objects. Although it is possible to alter the properties of these objects, you should not do so, since the alterations could prevent future updates to CSM from installing properly.*

## Customizing System Objects

Customizing system objects (optional) is typically done in the following sequence:

1. Define general object information.
2. Add items to system lookup tables.
3. Define categories.
4. Create custom lookup tables and their items.
5. Create custom fields.
6. Create custom pages.
7. Configure Rules and workflow.
8. Create Document Generator templates.
9. Test the components of your custom design.

## Creating Custom Objects

You must create custom objects in the following sequence:

1. Define general object information.
2. Define unique identifiers for records.
3. Define record name patterns.
4. Define phases and their transitions.
5. Add assignee roles.
6. Add items to system lookup tables.
7. Define categories.
8. Create custom lookup tables and their items.
9. Create custom fields.
10. Once the previous tasks are completed, you can perform the following tasks in parallel or in any sequence:

- Create custom pages.
- Configure Rules and workflow.
- Create Document Generator templates.

11. Test the components of your custom design.

## Creating Custom Pages

You can define the following types of custom pages:

- **Custom object views**—Layout of fields and record pages
- **Wizards**—An tool to guide end-user through creating or updating records
- **Search views**—Pages for finding and displaying lists of records
- **Home pages**—Provide a range of information and links in one location
- **Portal panes**—Components of home pages that display sets of information and links

You can create custom pages in the following sequence:

1. Define blocks.
2. Define object views.
3. (optional) Define templates for wizards.
4. Define any user groups necessary for wizard rules.
5. Define wizards.
6. Define search views.
7. Define portal panes.
8. Define home pages.

## Creating Custom Object Views

Object views define the layout of fields and record pages. You can customize and assign object views for different user groups.

To successfully create object views, you should understand the following:

- Object categories, their purpose, full tree positions, how they are created, where they appear in the end-user interface, and so on. See [Using Categories](#).
- The differences between system and custom object views. See [Custom and System Views](#).

You can create custom object views in the following sequence:

1. Create a field specification document, including categories and display requirements for the object definition.
2. Create the custom object definition (except for system objects).
3. Create any necessary categories.

4. Create custom fields.
5. Create blocks.
6. Create object views.

## XML Templates

The design of Custom Java Blocks is outside the scope of this documentation. However, if you do work with Custom Java Blocks, be aware of the XML templates available in the `\utilities\samples\screens` subfolder of your program directory. By using these templates for your work, your CJBs will preserve the look and feel of "out of the box" TeamConnect.

### 3.1.1.4.1 Creating Custom Messages

For error messages that may appear as a result of custom java rules, custom java blocks, or custom tools, TeamConnect lets you create custom message keys and associated message text for localization purposes.

You can define for the following:

- Object-definition-specific custom messages used in custom rules and screens.
- Common custom messages that are used in custom tools, custom portal pane content, and custom screens.

Once defined, when you export data from TeamConnect for localization, the message keys and message text appear on the Excel spreadsheet on the **Custom Resources** tab. You can enter your translations on this tab and then [import the translated text back to TeamConnect](#). This allows users to see custom messages in their preferred language.

Once you create a message key, you cannot edit it, but you can edit the associated message. You can delete a custom key and its associated message.

**Note:** *If the text of a custom message contains parameter tokens (`{0}`, `{1}`, etc.) these tokens must be retained in the translated text for each locale.*

### To create custom messages

1. Open the Designer.
2. Select one of the following options:
  - **For object definition-specific custom messages**—Click the **Go to** drop-down list, and then select **Object Definitions**. Select the object definition for which you want to create custom messages. Click the **Custom Messages** tab.
  - **For common custom messages**—Click the **System Settings** drop-down list, and then select **Custom Messages**. If necessary, click the **Custom Messages** tab.
3. (optional) In the **Number of entries you would like to add** drop-down list, enter the number of keys with messages that you want to create.
4. In the **Message Key** field, enter the appropriate message key.

All common custom message keys are prefixed with *custom.common.*; all object definition message keys are prefixed with *custom.* only. You do not have to enter these prefixes or the period—they are automatically added to the front of the message key when you click **add more**.

A message key may contain up to 250 alphanumeric characters.

5. In the **Default Value** box, enter the appropriate message text. The message may contain up to 4000 characters. A Default Value message must have an associated message key.
6. Click **add more**.

### To edit custom messages

**Note:** You cannot edit the **Message Key**, but you can edit the associated **Default Value** message.

Click the check-box for the **Default Value** that you want to edit, and then click **edit**. Make the appropriate changes and then click **ok**.

### To delete custom messages

Click the check-box for the **Message Key** that you want to delete, and then click **delete**. Deleted keys are no longer available when you re-import localized data into TeamConnect.

#### 3.1.1.4.2 Designing for Data Warehouse and Microsoft SQL Server

Before you begin changing your TeamConnect design, you must be aware of some limitations imposed by Microsoft SQL Server, particularly with regard to the Data Warehouse feature of TeamConnect.

The system objects and custom objects contained in TeamConnect can be synchronized with database tables in the Data Warehouse, a feature that is used for reporting. However, tables in Data Warehouse can sometimes have a very large number of columns, particularly when the related TeamConnect object has a large number of categories. In Microsoft SQL Server, Data Warehouse tables support a maximum of 8060 bytes per row. You should design your categories and reportable fields so that the 8060 byte limit cannot be exceeded. (For the bytes allocated per column for each field type, see [Data Warehouse Requirements for Custom Fields](#).) These amounts are subtracted from MS SQL Server's maximum allocation of 8060 bytes per row.

For example, if there are 30 custom fields of the type Text within a single category, the Data Warehouse table has 30 columns that use 250 bytes each. Altogether, 30 custom fields of the type Text use 7500 bytes of the 8060-byte allocation, which only leaves 560 bytes for other field types in that category.

When run with Microsoft SQL Server, Data Warehouse has a 250-byte limit for custom fields of the type **Text**. If a user enters data in excess of this 250-byte limitation, the Data Warehouse scripts truncate the text when populating the table. In addition, some Non-ASCII characters use two bytes per character of the 250-byte data length limitation. To make sure that users do not exceed the 250-byte limit, which results in truncated data in reports, choose one or more of the following options:



- Use memo text fields instead of custom fields of the type **Text** if users might exceed the 250-byte limit.
- Develop rules in TeamConnect to enforce the 250-byte limit for each custom field of the type **Text**.

### 3.1.2 Working with System Settings

Most customization tasks involve creating or changing object definitions, search views, new custom pages, or other TeamConnect components, then modifying TeamConnect to use these new components. However, some tasks use existing TeamConnect pages and features primarily to configure or reconfigure the system's behavior settings.

You may access these configuration settings using:

- **Tools** drop-down list
- **System Settings** drop-down list
- Screens inside the related Object Definitions, which allow you to change the behavior of specific system objects

As necessary, you may also create and upload custom tools to help you with setting these configurations.

All of these configuration or reconfiguration features are available to you in TeamConnect.

#### 3.1.2.1 Accessing System Settings

Many settings and preferences that control the behavior of TeamConnect are accessed through the [Admin tab in the user interface](#). However, some settings and preferences are accessed through Designer.

In addition to the TeamConnect system settings and predefined tools, your organization may require more configuration options to be set. This documentation provides only the following information:

- How to configure custom tools that you may develop or purchase, such as Time Entry, Batch Invoicing, or Custom Fields for Invoice Line Items tool. See [Configuring Custom Tools](#).
- How to create custom fields, such as password fields, for integrating your TeamConnect application with other systems. See [Miscellaneous Settings](#).

##### 3.1.2.1.1 Rule Execution Properties

The **Rule Execution Properties** entry in the **System Settings** dropdown list allows you to temporarily disable all of the rules in TeamConnect.

It is useful to disable rules during some processes. For example, when running an upgrade tool, rules that activate automatically during the upgrade may cause the upgrade tool to fail.



**Rule Execution Properties**

Both options are selected by default.

The options in the **Rule Execution Properties** section are described as follows:

**Rule Execution Properties**

Field	Description
Enable Rule Execution	Clear this check-box if you want to temporarily disable all of the rules in TeamConnect.
Enable Legacy Rule Execution	Clear this check-box if you want to temporarily disable all of the legacy rules in TeamConnect.  The <b>Top Level/System/Legacy Rules</b> folder is used for storing Java class files for legacy rules that were written for TeamConnect. This centralized folder allows the files to be shared by multiple objects instead of being uploaded to each individual object definition.

Once you select the appropriate options, you must click **Save** in order for the settings to take effect.

**Important:** *If you temporarily disable rules and you want to re-enable them, you must go back to the **Rule Execution Properties**, select the appropriate check-box, and then click **Save**. If you want to enable legacy rule execution, you must select the **Enable Legacy Rule Execution** check-box and the **Enable Rule Execution** check-box. Legacy rule execution does not succeed unless both check-boxes are selected.*

For more information about rules, see [Rule Components](#).

#### 3.1.2.1.2 Miscellaneous Settings

If there is another system which your organization wants to use in tandem with TeamConnect, such as Microsoft Exchange or Lotus Notes, Mitratech can service a custom integration project for your organization.

In order for a TeamConnect to integrate with another system, the user must enter specific information into the available fields in the **Other Settings** page accessed from the user's **Preferences** page. However, these fields are unavailable to the user unless you create them.

You may create custom fields for system integration in the **Miscellaneous Settings** screen. Depending on the system your organization wants to integrate, Mitratech may provide you with specific guidelines and information for creating custom fields for your integration. However, the procedure itself is the same as creating custom fields for an object definition.

#### To access the Miscellaneous Settings screen

Select **Misc. Settings** from the **Tools** drop-down list.

The corresponding [Miscellaneous Settings screen](#) appears.

The Misc. Settings tool is only available in the **Tools** menu if you have been given the right to it on the **Rights** tab in either your user account or a group account.

**Miscellaneous Settings**

Name	Label	Type	Order
Number of entries you would like to add: <input type="text"/>			
1	Exchange_Pass	Password	0
			<input type="button" value="+ add more"/>
Name	Label	Type	Order
1	<input type="checkbox"/> Exchange_Date	Today's Date	1
2	<input type="checkbox"/> Exchange_EnableCheckBox	Enable Exchange	2
<input type="button" value="Check All"/> <input type="button" value="Uncheck All"/> <input type="button" value="edit"/> <input type="button" value="delete"/>			

Miscellaneous Settings Tool Screen

The following table describes the items in the **Miscellaneous Settings** screen.

Miscellaneous Settings Tool Screen

Field	Description
<b>Name</b>	Enter an alphanumeric name (maximum 250 characters) to uniquely identify the field.
<b>Label</b>	Enter a label (maximum 250 characters) for the field as you would like it to display in the <b>Other Settings</b> page of the user's <b>Preferences</b> page.
<b>Type</b>	<p>Select an option to specify the type of custom field that displays in the <b>Other Settings</b> page of the user's <b>Preferences</b> page. The options in this drop-down list are the same as the field types used for creating custom fields for objects, except for passwords.</p> <p>When creating a <b>Password</b> custom field for system integration, the following two fields appear on the user's <b>Other Settings</b> page of the <b>Preferences</b> page:</p> <ul style="list-style-type: none"> <li>• <b>New Password</b>—The user must enter the password required for the system that is integrating with TeamConnect.</li> <li>• <b>Confirm Password</b>—The user must confirm the password used for the system that is integrating with TeamConnect.</li> </ul>
<b>Order</b>	<p>Enter an integer to indicate the display <b>Order</b> of the custom field in the <b>Other Settings</b> page of the user's <b>Preferences</b> page. If you create more than one custom field for the user, the custom fields appear in the order specified, lowest number first.</p> <p><b>Note:</b> <i>Items with the same Order are sorted and displayed alphabetically.</i></p>
<b>add more</b>	Click to add the custom field to the <b>Other Settings</b> page of the user's <b>Preferences</b> page.

### To create a custom field for system integration

1. In the **Tools** drop-down list, select **Misc. Settings**.
2. Fill in the [appropriate fields](#).  
If necessary, you can learn more about custom fields, their name requirements and types under [Creating Custom Pages](#).
3. Click the **add more**.  
The custom field added is immediately available in the **Misc. Settings** tab of the user's **Preferences** screen.
4. Give users the Read and Update rights for **Preferences** in their user or group accounts, so that they can use the newly added fields.

*Tip: You can view and access the information entered by the user in their **Misc. Settings** tab by clicking the **Preferences** hyperlink in the user's account.*

### 3.1.2.2 Setting up Invoices and Line Items

Options and default behavior across all invoices records and for invoice record - Line Items sections can be configured from the Object Definitions area, using the following operations:

- [Defining Non-US Tax Categories](#)
- [Setting up Invoice Duplicate Checking](#)
- [Setting up Invoice Line Items](#)

#### 3.1.2.2.1 Defining Non-US Tax Categories

If you use CSM, TeamConnect lets you define Non-US Tax categories to support e-Billing invoices that contain Non-US taxes (such as VAT or GST).

#### To define non-US tax categories

1. In the Designer, click **Go to**, and then click **Object Definitions**.
2. Click the **Invoice** object definition.
3. Click the **Non-US Tax Categories** tab.
4. In the **Show items node** drop-down list, select **Fee** or **Expense**.
5. In the **Item Name** field, enter the a unique name for this item.
6. In the **Order** field, enter a number to indicate the order in which this item appears to the user. Items with the same order number are sorted alphabetically.
7. In the **Tree Position** field, enter a four-character alphanumeric code that is unique to any other code.

8. In the **Tax Code** field, enter the name for the tax code that applies to this item.  
**Item Name**, **Tree Position**, and **Tax Code** only need to be unique for items of the same type.
9. Click **+add more**.
10. To add more items, click +add more and repeat steps 4 through 8.
11. Click **Save** or **Save and Close**.

You will be able to create financial accounts in the TeamConnect user interface to [track Non-US Taxes separately](#). The Non-US Tax category **Item Name** values appear in the **Post Non-US Tax of Type** drop-down list on the Accounts [Posting Criteria page](#).

#### 3.1.2.2.2 Setting up Invoice Duplicate Checking

A system setting will determine whether invoices will be checked for duplicates. For more information about invoice duplicates and versions, see [Invoice Duplicates](#) and [Invoice Versions](#).

##### To set up invoice duplicate checking

1. From the **System Settings** drop-down list, select **Backward Compatibility Settings**.
2. Do one of the following actions:
3. To allow duplicate invoices, select the **Allow Duplicate Invoices (Same invoice number, invoice date year and vendor)** check-box.
4. To check for and prevent duplicate invoices, clear the **Allow Duplicate Invoices (Same invoice number, invoice date year and vendor)** check-box.
5. Click **Save**.

#### 3.1.2.2.3 Setting up Invoice Line Items

This section describes how to set up the ways users will work with Invoice Line Items.

**Note:** The terms "fee" and "task" are used interchangeably for Invoice Line Items. For example, in the user interface, Line Item types include Fee. In the developer's view for Line Item object definitions, Task Categories refer to categories for Fee Line Items.

##### 3.1.2.2.3.1 Setting up General Invoice Line Items Behavior

General Invoice Line Items behavior includes:

- Setting up which Line Item fields display in Concise view
- Setting up which existing Line Item field values can be copied to a new Line Item
- Setting up whether Line Item discount adjustments are applied by flat-rate or as a percentage

### To set up general Line Item fields behavior

1. In the **Go to** drop-down list, select **Object Definitions**.
2. Expand **Invoices** and click the **Line Items** link.
3. Select the **Line Items** tab.
4. To set up which fields display on an invoice **General** page, **Line Items** section area for line item entry (in edit-mode), do one of the following:
  - For each field to display in the invoice **Line Items** section (for line item entry), from the corresponding **Line Item Entry** column drop-down list, select **Show**.
  - For each field to hide in the invoice **Line Items** section (for line item entry), from the corresponding **Line Item Entry** column drop-down list, select **Hide**.
5. To set up which fields display on an invoice **General** page, **Line Items** section area for existing line items (in edit-mode), do one of the following:
  - For each field to display in the invoice **Line Items** section (table for existing line items), from the corresponding **Table Display** column drop-down list, select **Show**.
  - For each field to hide in the invoice **Line Items** section (table for existing line items), from the corresponding **Table Display** column drop-down list, select **Hide**.
6. For each **Line Item** field whose existing value to copy to a blank Line Item block using the **Copy values from previous line** link, check the corresponding **Copy Capability** check-box.
7. Click **Save**.

### Setting the Discount Application Type for Line Item Adjustments

When users adjust Invoice Line Items, an Original Discount amount can be applied in one of the following ways:

- As a flat amount
- As a percentage

By default, Discounts are applied to Line Item adjustments as flat amounts. The Original Discount value will be automatically copied to the Current Discount field (Batch Adjustment mode) or will be displayed in the Current Discount field of the **Invoice - Adjust Line Items** screen (Detail Adjustment mode) after invoice changes are saved.

When Discounts are applied as percentages, the percentage is calculated in the following way:

$$\text{Discount \%} = (\text{Original Discount}) * 100 / [(\text{Original Rate}) * (\text{Original Hrs/Units})]$$

Based on Line Item Object Definition settings, a Line Item can be adjusted using Current Rate, Current Hrs/Units, Current Disc (Batch Adjustment only), Current Amount.

The Current Discount will be applied as the percentage value only if Current Rate or Current Hrs/Units fields are adjusted.

The Current Discount is not applied as a percentage, if a Line Item is adjusted by any of the following methods:

- The Current Amount is manually adjusted.
- The Current Discount is manually adjusted.
- The Current Amount is indirectly adjusted by an Invoice Header adjustment.

For more information, see [Adjusting Line Items with Discounts](#).

### To toggle between setting adjustment discounts as a flat amount or percentage

1. In the **Go to** drop-down list, select **Object Definitions**.
2. Click the **Invoices** link.
3. From the **General** tab, do one of the following actions:
  - To apply the adjustment discount as a percentage, check **Discounts are applied as percentage of the charge amount**.
  - To apply the adjustment discount as a flat rate value, uncheck **Discounts are applied as percentage of the charge amount**.
4. Click **Save**.

#### 3.1.2.2.3.2 Setting up Invoice Line Items Adjustment Behavior

Invoice Line Items Adjustment Behavior includes:

- Setting up whether Line Items can be adjusted in Batch Adjustment mode or in Detail Adjustment mode
- Setting up which Line Item fields will be read-only in Batch Adjustment mode
- Entering adjustment reasons

### Invoice Line Items Adjustment Options

You may adjust invoice Line Items as follows:

- **Detail Adjustment:** (recommended for Invoices with few Line Items) For TeamConnect 2.4 SP3 and earlier, Line Items have been adjusted in this mode. When enabled adjusters must select each Line Item, adjust the Line Item, and save changes to individual Line Items separately.
- **Batch Adjustment:** (recommended for Invoices with many Line Items) When enabled adjusters can make all Line Items for an invoice adjustable with one button click, adjust the Line Items, and save changes to multiple Line Items with one button click. This option is helpful for tabbing through multiple Line Items.
- **In-Line & Bulk Adjustment:** (recommended for Invoices with many Line Items) Introduced in TeamConnect 3.4 SP1, this setting allows adjusters to make changes to multiple Line Items at a time, with improved screen navigation, the ability to add comments to Line Items directly from the In-Line & Bulk Adjustment (IBA) screen, and other additional features.

### To select between Individual Invoice Line Items Adjustment, Batch Adjustment, or In-Line & Bulk Adjustment

1. In the **Go to** drop-down list, select **Object Definitions**.
2. Expand the **Invoices** link and click **Line Item**.
3. Do one of the following actions:
  - To adjust invoice record line items individually, from the **Default Adjustment Mechanism** drop-down list, select **Detail Adjustment** (default).
  - To adjust invoice record line items in batch mode, from the **Default Adjustment Mechanism** drop-down list, select **Batch Adjustment**. This option allows you to adjust multiple Line Items simultaneously and save changes with one button click.
  - To adjust invoice record line items in in-line & bulk mode, from the **Default Adjustment Mechanism** drop-down list, select **In-Line/Bulk Adjustment**. This option allows you to adjust multiple Line Items simultaneously and add comments to individual Line Items from the same screen.
4. Click **Save**.

### Setting Batch Adjustment Line Item Fields Read/Write Permissions

If Line Items are set to Batch Adjustment mode, you can restrict which Invoice Line Item fields can be used to adjust invoices by setting certain fields to read-only status.

***Note:** The following description only applies if you have set the Line Items **Default Adjustment Mechanism** to **Batch Adjustment**.*

Fields that can be set to read-only include:

- Current Total
- Current Discount
- Current Qty
- Current Rate

### To set Batch Adjustment Line Item fields to read-only

1. In the **Go to** drop-down list, select **Object Definitions**.
2. Expand the **Invoices** link and click **Line Item**.
3. Select the **Line Items** tab.
4. For each **Line Item** field (**Current Total**, **Current Discount**, **Current Qty**, **Current Rate**) to set as read-only when performing a batch adjustment, check the corresponding **Read Only in Adjustment** check-box.



At least one of the adjustable Line Item fields above needs to be left as non-read only. Users need to be able to adjust at least one field.

5. Click Save.

### To access the Adjustment Reasons page

1. In the **Go to** drop-down list, select **Object Definitions**.
2. Expand the **Invoices** link and click **Line Item**.
3. Select the **Adjustment Reasons** tab.

Adjustment reasons are organized in a parent-child hierarchy. Initially there is only one node, "Root", and any adjustment reasons that you enter become children of the "Root" node. If you want to enter reasons that are children of some other node, choose that node name from the **Show items in node** drop-down list, as shown below.

### Adjustment Reasons

Show items in node: Root ▼

Number of entries you can add: 1

Item ID	Item Name	Tree Position	Is Active
1	<input type="text"/>	<input type="text"/>	YES

+ add more

Item ID	Item Name	Order	Tree Position	Is Active
1	<input type="checkbox"/> Other	0	OTHE	YES
2	<input type="checkbox"/> Prenegotiation	0	PRNG	YES

[Check All](#) - [Uncheck All](#)

Adjustment Reasons Hierarchy

### To work with Adjustment Reasons

1. Choose a node from the **Show items in node** drop-down list.
2. The adjustment reasons that are children of that node are shown.
3. If you wish to delete an existing reason, click its checkbox and click **delete**.
4. If you wish to inactivate an existing reason, click its checkbox and click **inactivate**. Inactive reasons won't appear to the end user who is doing an adjustment.
5. If you wish to reactivate an existing reason, click its checkbox and click **activate**.

6. To edit an existing reason, click its checkbox and click **edit**.  
The **Name** and **Order** fields become editable and you can change their values. When finished, click **OK**.
7. To enter a new adjustment reason, begin typing in the editable fields above the existing items.
  - Enter a **Name** for the adjustment reason.
  - Enter an **Order** value. **Order** determines the sequence in which the reasons appear to the end user.
  - Enter a **Tree Key**. This value must be unique, different from all other adjustment reasons.
  - **Is Active** is presumed to be Yes when a reason is first entered. You can subsequently **inactivate** it after it is saved.
  - When finished, choose one of the **Save** buttons at the top of the page. Clicking **Add More** will also save your work and present you with new editable fields for additional adjustment reasons.

### 3.1.2.3 Using Duplicate Contact Manager

Duplicate Contact Manager enables TeamConnect designers, developers, and other highly trained IT specialists to identify duplicate contact records. A Contact Sweeper Tool is provided allowing the tool users to select which duplicate contact records to keep and which contact records to remove from the system.

Contact Sweeper also allows for the redirection of existing pointers to duplicate contact records, to reference the contacts you choose to save. Duplicate Contact Manager may also be used to prevent end users from creating or updating contact records that are intentional duplicates.

Duplicate Contact Manager includes:

- A Contact Sweeper Tool that can identify potential duplicate contact records based on specified contact fields and sample contact data values.
- A duplicate contact checking option with a settings screen to select which contact fields should contain unique data across records. After duplicate contact checking is enabled, users will be warned when they create/update contacts with possible existing duplicate records.

Duplicate Contact Manager must be configured by a solution developer, but the Contact Sweeper can be invoked by any user who is granted the right to that tool.

It is recommended that you use Duplicate Contact Manager in the following order:

1. [Using Contact Sweeper](#)
2. [Preventing Duplicate Contacts](#)

### 3.1.2.3.1 Using Contact Sweeper

Contact Sweeper helps identify, replace, and remove contacts that are duplicates of other existing contacts. Since the contact attribute is found in a large number of database tables, the process of manually changing a contact throughout TeamConnect is complex. Contact Sweeper hides that complexity and makes the process faster and more reliable.

#### 3.1.2.3.1.1 Prerequisites

Contact Sweeper is a system tool and, like other system tools, it requires the user running it to have sufficient authority. You must use caution when granting rights to this tool because its use may greatly alter the information in the database.

#### To give a user group rights to the Contact Sweeper tool

1. Click **Groups** in the **Admin** tab of TeamConnect.
2. Click on the specific group to give rights to the Contact Sweeper tool.
3. Click the **Tool Rights** tab of the group's settings.
4. Click the **Edit** button.
5. Select the check-box for **Contact Sweeper**.
6. Save the group's tool rights settings.

**Note:** *If the Contact Sweeper window is prematurely closed before steps have finished running, all steps done in the session must be repeated.*

It is essential that you back up your database before running Contact Sweeper. Contact Sweeper makes extensive changes to the TeamConnect database. After you have run the Fix References step in Contact Sweeper, the only way to reverse its effects is by restoring the database from the backup.

**Important:** *Several prompts in Contact Sweeper may display in a pop-up window. If your web browser is configured to not allow pop-up windows, you must reconfigure it to allow pop-ups from the TeamConnect web server.*

Using Contact Sweeper entails the following steps:

- **Step 1 of 5 (Generate Report)**—Specifies the conditions to be used in searching for duplicates, then performs the search and prepares an Excel spreadsheet containing the generated report, which you then save.
- **Step 2 of 5 (Edit Report)**—Indicates which duplicate contacts must be replaced by other contacts. Do this by editing the values in the generated report spreadsheet (offline step).
- **Step 3 of 5 (Upload Report)**—Uploads your edits to TeamConnect and makes them available for the following steps.
- **Step 4 of 5 (Fix References)**—Performs an optional preview and performs the actual replacement of references.

- **Step 5 of 5 (Delete Duplicates)**—Deletes the contacts whose references were replaced in the previous step.

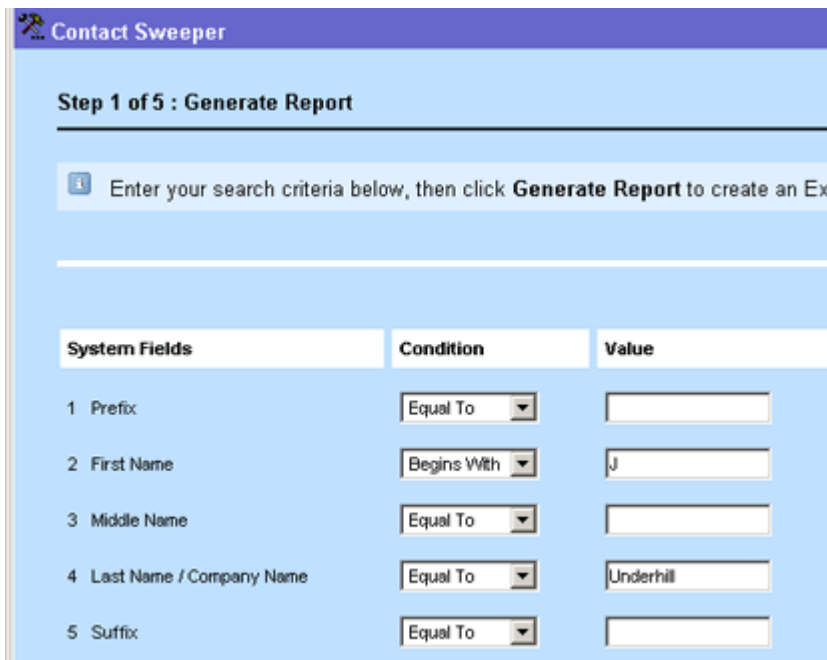
You may do all of these steps, except step 2, using the Contact Sweeper tool window.

### To start the Contact Sweeper tool

Select **Contact Sweeper** from the **Tools** drop-down list in Designer.

#### 3.1.2.3.1.2 Generate Report (Step 1 of 5)

This step creates a report of all contacts that meet the conditions you specify. You may run this step repeatedly, specifying a different set of conditions each time. Contacts meeting your conditions appear in an Excel spreadsheet.



The screenshot shows the 'Contact Sweeper' tool window at 'Step 1 of 5 : Generate Report'. It features an information icon and a prompt: 'Enter your search criteria below, then click **Generate Report** to create an Excel spreadsheet.' Below this is a table for defining search criteria.

System Fields	Condition	Value
1 Prefix	Equal To	
2 First Name	Begins With	J
3 Middle Name	Equal To	
4 Last Name / Company Name	Equal To	Underhill
5 Suffix	Equal To	

Generate Report Interface (1 of 2)

The screenshot shows a web interface for generating a report. It features two search criteria: '23 Internet Address' with a dropdown set to 'Equal To' and an empty text input; and '24 Default Category' with a dropdown set to 'Equal To' and a dropdown menu showing '(None Selected)'. Below these is a section 'Search for records of type:' with radio buttons for 'Person' (selected) and 'Company'. Another section 'A contact is a duplicate when:' has radio buttons for 'All of the above conditions are met (AND logic)', 'Any of the above conditions are met (OR logic)', and 'The following combination of the above conditions is met' (selected). A text input next to the third option contains '(2 and 4)'. Below this is a label 'Use condition numbers'. At the bottom left is a 'Generate Report' button.

Generate Report Interface (2 of 2)

Contact Sweeper displays some of the possible criteria. All the possible criteria are available for your use in specifying conditions. Ordinarily, only some, not all, of the criteria are actually used. For example, your conditions may test for equality with a criterion, a partial string match, and other conditional expressions.

Option buttons specify whether to form a match using all conditions, any of the specified conditions, or some custom combination of conditions. In the previous example, a custom combination is used, specifying that condition 2 (First Name match) and condition 4 (Last name / Company Name match) must both be true.

The initial conditions that appear when you run this step are those that were used the last time you ran this step. However, you have the ability to change any condition that you wish.

## Points To Remember

Keep the following points in mind when setting up Contact Sweeper reports:

- Criteria that are left blank will be ignored when generating the report.
- Try to keep your conditions restrictive, so a relatively small set of contacts is returned. You may wish to search on a last name plus postal code, or other such conditions that should yield a small number of results.

**Note:** *You may be even more restrictive by searching on unique identifiers such as Social Security number or driver's license number. If your conditions are defined too loosely, many contacts will be returned, and it can be difficult to find the true duplicates.*

- Some of the criteria may relate to attributes that have multiple values per contact. For example, a contact can have multiple addresses, multiple phone numbers, and multiple email addresses. When you use a criterion like these to specify a condition, be aware that only the default value for the contact is tested. Contact Sweeper does not search through all of the multiple values for the contact, just the default.

- Contact Sweeper understands the difference between a contact that represents a company and a contact that represents a person. One of the search conditions is whether you are looking for persons or companies.

For example, if your search involves the person name "Pierre Cardin", the company "Pierre Cardin" does not appear in the report on potential duplicates. Similarly, if you have asked to search for companies, but you have also specified conditions that apply only to persons (such as a First Name value), those conditions are ignored.

## Generating a Report

Generating a report operates as follows:

- After you click **Generate Report** and Contact Sweeper performs its search, you are prompted to save the resulting "duplicate reference report" as an Excel spreadsheet. The default name of the report is **DuplicateContactsReport.xls**.
- Choose an easily remembered name and location for this file. It is possible to store many different sets of results by running this step repeatedly, using different conditions each time, and saving each resulting spreadsheet with a different name.
- In some web browsers, the spreadsheet may open automatically. If so, you must save it, using a name and location that you remember. Other browsers may download the spreadsheet automatically without prompting you. If nothing appears to happen after clicking the **Generate Report** button, check your web browser's default download location for **DuplicateContactsReport.xls**.

### 3.1.2.3.1.3 Edit the Report (Step 2 of 5)

The spreadsheet produced by the Generate Report step contains one row per contact. Contact Sweeper has identified all the contacts in the spreadsheet as being potential duplicates of each other, because they all match the conditions you specified in step 1. Review these contacts and decide which ones are actual duplicates.

Microsoft Excel - DuplicateContactsReport.xls

File Edit View Insert Format Tools Data Window Help

Type a question for help

100% Arial 10

1

2 Contact Sweeper instructions

3

4 \*If you decide that two or more contacts are duplicates, you must choose one of them as the base contact.

5 All TeamConnect information that presently refers to the other duplicates will instead refer to the base contact after step 4 runs.

6

7 \*Find the 'Contact key' value (column B in the spreadsheet) for the base contact's row. Then enter that value in the

8 'Base key' column for each of the duplicate contact rows. (Don't enter it in 'Base key' of the base contact; that's unnecessary.)

9

10 \*If you edit a contact in TeamConnect, verify that you have not placed any value in the 'Base key' column of the

11 corresponding row in the spreadsheet.

12

13 \*If a specific contact is not a duplicate and does not require any editing in TeamConnect,

14 verify that you have not placed any value in the 'Base key' column of the corresponding row in the spreadsheet.

15

16	BASE KEY	CONTACT KEY	INVOLVEMENT COUNT	ASSOCIATED USER	PREFIX	FIRST NAME	MIDDLE NAME	LAST NAME
17		811	2			John		Underhill
18		2171	0			Jacqueline		Underhill
19		2169	2			Jacqueline		Underhill
20		2235	1			Jackie		Underhill
21								
22								

Ready NUM

Report Editor Interface

Review and edit the report as follows:

- If the spreadsheet from step 1 is not already open, open it now.

The spreadsheet contains blank cells in column A ("Base key"), the primary key of the contact in column B ("Contact key"), then a large number of contact information fields in the following columns. The columns "Involvement count" (indicating how many projects or other objects refer back to this contact) and "Associated user" (indicating that this contact is related to a TeamConnect user) are useful in deciding which contacts to keep and which to replace.

If you decide that two or more contacts are duplicates, you must choose one of them as the base contact. All TeamConnect information that presently refers to the other duplicates will instead refer to the base contact after step 4 runs.

- Find the "Contact key" value (column B in the spreadsheet) for the base contact's row. Then enter that value in the "Base key" column for each of the duplicate contact rows. (Don't enter it in "Base key" of the base contact; that's unnecessary.)

**Note:** Most of the cells in the spreadsheet are protected. Do not turn off this protection, or you may cause the spreadsheet to become unusable. Only the "Base key" column of the spreadsheet is editable. You cannot move, insert, or delete rows and columns.

In the example shown above, the row with contact key 2169 is the base contact that will remain after Contact Sweeper completes. Therefore, you would type 2169 in the Base Key cell for contacts 2171 and 2235. Contact 811 is a completely different person, not a duplicate, so we make no entry in that row.

**Note:** Contacts where the column "Associated user" contains a value will not be replaced or deleted, even if you request this.

- Instead of replacing some of the potential duplicate contacts, you may want to edit them so that they do not appear as duplicates anymore. If you edit a contact in TeamConnect, verify that you

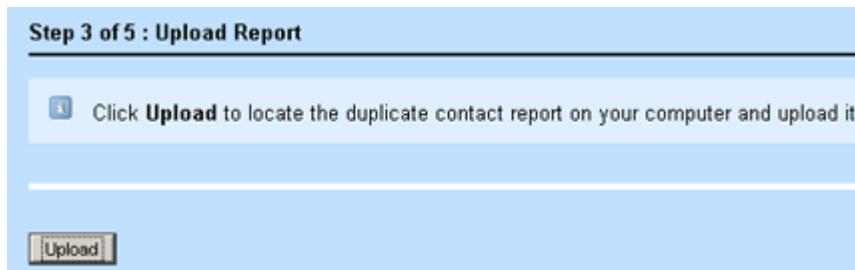
have not placed any value in the "Base key" column of the corresponding row in the spreadsheet.

- If a specific contact is not a duplicate and does not require any editing in TeamConnect, verify that you have not placed any value in the "Base key" column of the corresponding row in the spreadsheet.
- Save your changes to the spreadsheet.

#### 3.1.2.3.1.4 Upload Report (Step 3 of 5)

When you have finished all your data entry from the previous step, upload the spreadsheet to TeamConnect as follows:

1. Click the **Upload** button.
2. In the resulting **Upload New File** window, type the full path and file name in the **File Name** field or use the **Browse** button to locate the file on your computer.
3. Click upload file to move your spreadsheet into TeamConnect.



Upload Report Interface (1 of 2)



Upload Report Interface (2 of 2)

Now you are ready to run the Fix References step.

#### 3.1.2.3.1.5 Fix References (Step 4 of 5)

The Fix References step provides you with the option to preview the database changes that are about to occur, then updates the TeamConnect database with the contact changes you requested in the spreadsheet.

**Caution:** Perform this step when there are no other active TeamConnect users working with your TeamConnect database. This step puts the database into an inconsistent state until it

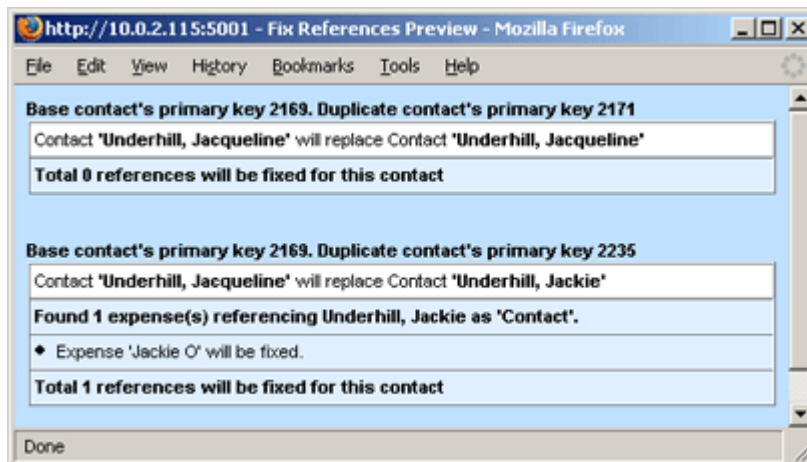


completes. It is strongly recommended that you back up your database before performing this step. The changes that are done in this step are not reversible, except by restoring a database backup.

**Important:** Custom rule processing could potentially interfere with the database updates done by this step. Deactivate all rules that relate to the Contact object before performing this step. When the step is complete, you may activate the rules again.

Preview the suggested changes and update the database as follows:

- Click **Preview** to display a pop-up window with text that describes the proposed database changes. No actual changes are done during Preview.

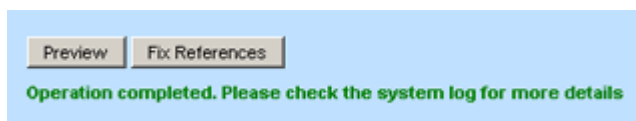


Fix References Preview Screen

In the previous example, there were references in existence for each duplicate contact. However, it is possible to have duplicate contacts that have never been used as a reference in any other record.

- Click **Fix References** to apply the changes in the spreadsheet to the database. Read the confirmation pop-up window, determine if you wish to continue, and click **OK** to continue or **Cancel** to stop.

This step may take some time, depending on the number of duplicate contacts and references that must be replaced. When finished, the Contact Sweeper screen refreshes and a completion message will appear below the **Fix References** button.



Contact Sweeper Completion Message

Contacts can become references in several other types of TeamConnect objects. Contact Sweeper fixes the following kinds of references:

- Every contact-related role in every contact-centric custom object record.
- Contacts that appear in the Relations tab of other Contact records.

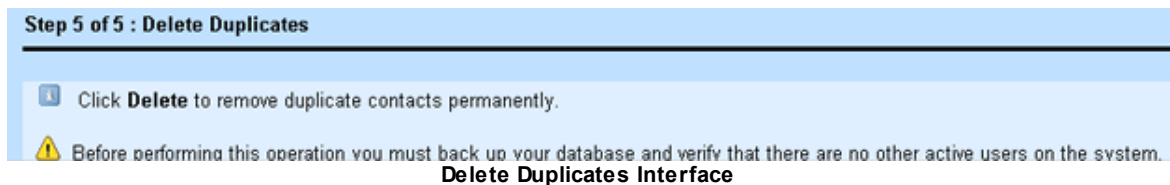
- Contacts that appear as "Vendor" or "Involved" in the **Posting Criteria** tab of an Account record.
- The "Contact" reference in an Expense record.
- The "Contact" reference in a Task record.
- The "Vendor" reference in an Invoice record.
- The "Vendor" and "Timekeeper" references in an Invoice Line Item record.
- The "Involved" references in a Project record.

By default, the changes made during the Fix References process are written to the system log at the Debug level. This level provides complete information about the changes that have been made.

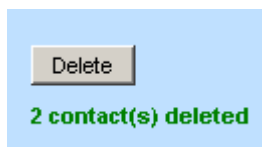
It is possible that you have previously set the system logging level to some more restrictive level, such as Warn, using the **System Preferences** screens. If so, logging occurs at that more restrictive level, and some of the detailed information is lost. When you determine the level of logging you want during Contact Sweeper operation, verify that the system logging level setting is compatible with your needs.

#### 3.1.2.3.1.6 Delete Duplicates (Step 5 of 5)

Click **Delete** to delete the contacts that have just been replaced by the base contact. Answer **Yes** to the confirmation pop-up message.



After the deletion operation, the Contact Sweeper screen refreshes, and a success message indicating the number of contacts deleted appears below the **Delete** button.



#### Confirmation Message

**Note:** The count reflects the actual number of contacts deleted. This number may be less than the number of contacts listed in step 4, if one or more of those contacts is associated with a user account. Such contacts are not deleted by Contact Sweeper. The system log notes such cases.

**Note:** If you click **Delete** without first doing **Fix references**, TeamConnect attempts to delete all the contacts for which you have entered a value in the "Base key" column of the spreadsheet. But since these contacts were not fixed yet, many of them may have references, such as being used in an "involved" role. In such cases, any contact with references is not deleted, and an error message is entered in the log file noting each case.

### 3.1.2.3.2 Preventing Duplicate Contacts

The Duplicate Contact Manager can perform day-to-day duplicate contact checking and prevent the creation or update of duplicate contact records. You must select which contact fields require unique field data across contact records and activate the duplicate contact checking setting.

Afterwards, when a user creates or updates a contact, TeamConnect compares the new contact's data against existing contacts to determine whether duplicates exist and warn the user.

If you activate the Duplicate Contact check for day-to-day usage, the following activities are affected:

- [Creating a new contact](#) from the **Create a new Contact** screen
- Updating an existing contact (person or company)
- Creating a new contact (person or company) from a Contact search module pop-up window
- Creating a new contact from a wizard using a contact search module (**Search for Contacts** pop-up window)
- Creating a new contact through the XML layer

#### 3.1.2.3.2.1 Activating Duplicate Contact Checking

Consider the following before using the Duplicate Contact Manager and Contact Sweeper:

- Contacts that are updated or added through the actions of wizards, rules, and custom views are not automatically checked for duplicates. If you wish to prevent potential duplicates that result from these actions, you must modify your wizards, rules, and views. For wizards and custom views, one option is to add the **Potential Duplicates** block to your wizard or view. Also, the `ContactService.readDuplicateContacts` API method is available for use in your custom code. This method allows you to check for duplicates before committing a contact record. See [Using Rules](#) for more information.
- Review the design of wizards, rules, and custom objects to verify that the criteria for identifying duplicate contacts is narrow enough not to prevent users from completing wizards successfully. You may need to modify existing rules and wizards to handle duplicate exceptions wherever a contact is created or updated.
- After you use the Contact Sweeper to remove existing duplicate contact records from TeamConnect, you should determine which contact fields should be required to create a contact and for day-to-day duplicate contact checking. For example, a person contact might require the **First Name**, **Last Name**, and **Phone Number** fields. This standard could be enforced by creating validation rules. Company contacts might require the **Company Name**, **Phone Number**, **Street Address**, and **Postal Code** fields.

The contact fields you set as Active from the Duplicate Contact Manager settings screen determine which contact fields require unique data when users create or update contact records (when duplicate contact checking is enabled).

**Note:** You may select one or more contact fields to use for search criteria.

**General Settings**

Activate Duplicate Matching

Allow users to override

Enter the number of duplicate results you would like to display

**Duplicate Matching Settings**

	Label	Field	Is Active	Condition
1	<input type="checkbox"/> Prefix	prefix	NO	Begins With
2	<input type="checkbox"/> First Name	firstNameUpper	<input checked="" type="checkbox"/>	Equal To <input type="button" value="ok"/> <input type="button" value="cancel"/>
3	<input type="checkbox"/> Middle Name	middleName	NO	Equal To
4	<input type="checkbox"/> Last Name / Company Name	nameUpper	<input checked="" type="checkbox"/>	Equal To <input type="button" value="ok"/> <input type="button" value="cancel"/>

**Duplicate Contact Manager Settings Screen**

The Duplicate Contact Manager settings screen determine which contact fields are used to find potential duplicate contact records.

**To select a contact field to use for duplicate contact checking**

1. From the Designer **Go To** drop-down list, select **Object Definitions**.
2. Select the **Contact** object definition.
3. Select the object definition's **Duplicate Contact Manager** tab.
4. Select the far left check-box for the contact field to use for duplicate contact checking.
5. Click **edit**.
6. Select the **Is Active** check-box.
7. Select the value-matching condition (for example, **Equal To**).
8. Click **ok**.

In the following steps, you can:

- Select whether to enable day-to-day duplicate contact checking (and whether to enable a user override to allow creation of duplicate contact records)
- Select which contact fields to activate for comparison of a new contact's field data against existing contacts' data
- Select logic for combining multiple contact fields as duplicate record identifiers

### To activate duplicate contact checking

1. From the Designer **Go To** drop-down list, select **Object Definitions**.
2. Select the **Contact** object definition.
3. Select the **Object Views** tab.
4. Choose an object view. On the General tab of the object view, ensure that the block **Duplicate Contacts** is used by the object view. If it is not, add that block to the object view.
5. Optionally, repeat the previous step for as many object views as are affected by duplicate contact checking.
6. Select the object definition's **Duplicate Contact Manager** tab.
7. To enable day-to-day duplicate contact checking, select the **Activate Duplicate Matching** check-box. If users try to create or update a contact that match existing duplicate contact records, a warning message will display.

To allow users to override duplicate contact checking and continue with the create or update operation, select the **Allow users to override** check-box.

8. To set the maximum number of potential duplicate contacts results that display during day-to-day duplicate contact checking, type the number in **Enter the number of duplicate results you would like to display**. The default value is 20.

If during a contact creation operation, the Duplicate Contact Manager finds more potential duplicate contacts than the maximum duplicate results value you set above, a warning message will display. For example, **25 possible duplicate(s) found. Displaying 20 of 25**.

9. From the **Duplicate Contact Manager** tab, check one or more of the following Labels and click **edit**. Select the **Is Active** check-box, and select a **Condition** from the drop-down menu to use the corresponding contact field for duplicate matching. Click **ok** to save your selections for the corresponding contact field.

A full list of available items is described in [the Duplicate Matching Settings table](#). Labels and Conditions marked with an asterisk are recommended for duplicate matching.

10. If you selected multiple contact fields above, select one of the following for **A contact is a duplicate when**:
  - **All of the above conditions are met (AND logic)**—Detects duplicate contacts if the values of all selected fields above match between the contact records being compared
  - **Any of the above conditions are met (OR logic)**—Detects duplicate contacts if the values of any selected fields above match between the contact records being compared
  - **The following combination of the above conditions is met**—Type a custom logical rule for contact field. For example, to compare contacts by (First Name, Last Name, and Social Security Number) or (Default Postal Code and Phone number), type **(2 and 4 and 12) or (17 and 20)**

The field numbers you enter when using **The following combination of the above conditions is met** must refer to contact fields that are set as **Active**.

11. Click **Save**.

## Duplicate Matching Settings

Label	Field	Condition
<b>Prefix</b>	prefix	Begins With *Equal To Contains Ends With Has No Value Has Value
<b>*First Name</b>	firstNameUpper	Begins With *Equal To Contains Ends With Has No Value Has Value
<b>Middle Name</b>	middleName	Begins With *Equal To Contains Ends With Has No Value Has Value
<b>*Last Name / Company Name</b>	nameUpper	Begins With *Equal To Contains Ends With Has No Value Has Value
<b>Suffix</b>	suffix	Begins With *Equal To Contains

		Ends With Has No Value Has Value
<b>Nickname/Alias</b>	aliasUpper	Begins With *Equal To Contains Ends With Has No Value Has Value
<b>Company</b>	company	Equal To
<b>Title</b>	title	Begins With *Equal To Contains Ends With Has No Value Has Value
<b>ID</b>	numberStringUpper	Begins With *Equal To Contains Ends With Has No Value Has Value
<b>*Driver's License</b>	driverLicense	Begins With *Equal To Contains Ends With Has No Value Has Value

<b>*SSN / Tax ID</b>	SsOrTaxNumberString	Begins With *Equal To Contains Ends With Has No Value Has Value
<b>*Birth Date</b>	birthDate	*Equal To More Than X Days After During X Days After During X Days Before More Than X Days Before Between X and Y Days
<b>Street</b>	defaultAddress_Street	Begins With *Equal To Contains Ends With Has No Value Has Value
<b>City</b>	defaultAddress_City	Begins With *Equal To Contains Ends With Has No Value Has Value
<b>State</b>	defaultAddress_State	Begins With *Equal To Contains Ends With



		Has No Value Has Value
<b>Postal Code</b>	defaultAddress_PostalCode	Begins With *Equal To Contains Ends With Has No Value Has Value
<b>County</b>	defaultAddress_County	Begins With *Equal To Contains Ends With Has No Value Has Value
<b>Country</b>	defaultAddress_CountryItem	Equal To
<b>Phone</b>	defaultPhone_PhoneString	Begins With *Equal To Contains Ends With Has No Value Has Value
<b>Fax</b>	defaultFax_FaxString	Begins With *Equal To Contains Ends With Has No Value Has Value
<b>Email</b>	defaultEmail_EmailString	Begins With

		*Equal To Contains Ends With Has No Value Has Value
<b>Internet Address</b>	defaultInetAddress_inetAddressString	Begins With *Equal To Contains Ends With Has No Value Has Value
<b>Default Category</b>	defaultCategory	Equal To

### Using AND/OR Logic

If AND logic or OR logic is selected, when a user is creating/updating contacts, the duplicate check only evaluates field criteria that apply to the type of contact being created/updated.

For example, suppose duplicate contact checking is enabled for First Name, Last Name/Company Name, and Social Security/Tax ID using AND logic. If a user creates a company contact, duplicate contact checking performs a comparison against existing company records using the Company Name and Tax ID fields (ignoring the First Name criteria).

If custom logic is enabled, for example, "(First Name, Last Name/Company Name, and Social Security Number/Tax ID) or (Default Postal Code and Phone Number)," when creating/updating a company contact, only "(**Last Name/Company Name** and **Social Security Number/Tax ID**) or (Default Postal Code and Phone Number)" is checked.

### Creating Contacts in the Search for Contacts Pop-up Window

When users create contacts from a **Search for Contacts** pop-up window, duplicate contact checking only compares certain fields to existing contact records (for example, when an Involved record is added to a project).

For example, when a user creates a contact from the **add person** button, the following fields' values are compared to existing contacts: **First Name**, **Last Name/Company Name**, **Title**, **ID**, **SSN/Tax ID**, default **Street Address**, default **City**, default **State**, default **Country**, default **Postal Code**, and default **Category** field.

For another example, when a user creates a company contact from the **add company** button, the following fields' values are compared to existing contacts: **Last Name/Company Name**, **SSN/Tax ID**, default **Street Address**, default **City**, default **State**, default **Country**, default **Postal Code**, and default **Category** field.

---

As an example of a person from the contact screen, you might set up the Duplicate Contact Manager settings screen as follows:

- **Activate Duplicate Matching** has been selected.
- **Allow users to override** has been selected.
- **Enter the number of duplicate results you would like to display** is set to 5.

2	<input type="checkbox"/> First Name	firstNameUpper	YES	Equal To
3	<input type="checkbox"/> Middle Name	middleName	NO	Equal To
4	<input type="checkbox"/> Last Name / Company Name	nameUpper	YES	Equal To
5	<input type="checkbox"/> Suffix	suffix	NO	Equal To
6	<input type="checkbox"/> Nickname/Alias	aliasUpper	NO	Equal To
7	<input type="checkbox"/> Company	company	NO	Equal To
8	<input type="checkbox"/> Title	title	NO	Equal To
9	<input type="checkbox"/> ID	numberStringUpper	NO	Equal To
10	<input type="checkbox"/> Driver's License	driverLicense	NO	Equal To
11	<input type="checkbox"/> SSN / Tax ID	SsOrTaxNumberString	NO	Equal To
12	<input type="checkbox"/> Birth Date	birthDate	NO	Equal To
13	<input type="checkbox"/> Street	defaultAddress_Street	NO	Equal To
14	<input type="checkbox"/> City	defaultAddress_City	NO	Equal To
15	<input type="checkbox"/> State	defaultAddress_State	NO	Equal To
16	<input type="checkbox"/> Postal Code	defaultAddress_PostalCode	YES	Equal To
17	<input type="checkbox"/> County	defaultAddress_County	NO	Equal To
18	<input type="checkbox"/> Country	defaultAddress_CountryItem	NO	Equal To
19	<input type="checkbox"/> Phone	defaultPhone_PhoneString	YES	Equal To
20	<input type="checkbox"/> Fax	defaultFax_FaxString	NO	Equal To
21	<input type="checkbox"/> Email	defaultEmail_EmailString	NO	Equal To
22	<input type="checkbox"/> Internet Address	defaultInetAddress_InetAddressString	NO	Equal To
23	<input type="checkbox"/> Default Category	defaultCategory	NO	Equal To

[Check All](#) - [Uncheck All](#) [edit](#)

**A contact is a duplicate when:**

- All of the above conditions are met (AND logic)
- Any of the above conditions are met (OR logic)
- The following combination of the  above conditions is met

**Duplicate Matching Settings**

In this example, a user may try to create a contact (type person) with the following field data:

- **First Name** - Len
- **Last Name** - Marr
- **Default Phone Number** - 213-380-5512

In this case, when the user clicks **Save**, a message appears showing possible duplicates that have been found:

The screenshot shows the TeamConnect Legal interface. At the top, there is a search bar with 'Contacts' entered. Below the search bar are navigation tabs: Home, Legal, Finance, Contacts (selected), Calendar, Documents, Admin, and All Services. The main content area is titled 'Contacts' and contains a 'New Person' form. A yellow warning box titled 'Possible Duplicates Found!' is displayed, containing a checkbox 'Allow contact to be saved' and a table with three columns, each containing the text 'Marr, Len'. Below the warning box is the 'Personal Information' form with fields for Prefix, Job Title (Plumber), \*First Name (Len), Middle Name, \*Last Name (Marr), and Suffix. The Company field has a 'New' button.

**Possible Duplicate Contacts Found Message Example 1**

**Note:** A message like "Displaying 5 of 7" (not shown in this example) would indicate that the Duplicate Contact Manager was configured to display a maximum of 5 duplicate results but 2 additional possible duplicates were found and not displayed.

The user may perform one of the following actions:

- Override the duplicate contact check by selecting the **Allow contact to be saved** check-box and clicking **Save**.
- Click an existing contact link to verify that the contact is actually unique.

As an example of the contact screen, you might set the Duplicate Contact Manager settings screen as follows:

- **Activate Duplicate Matching** is selected.
- **Allow users to override** is selected.
- **Enter the number of duplicate results you would like to display** is set to 5.

2	<input type="checkbox"/>	First Name	firstNameUpper	YES	Equal To
3	<input type="checkbox"/>	Middle Name	middleName	NO	Equal To
4	<input type="checkbox"/>	Last Name / Company Name	nameUpper	YES	Equal To
5	<input type="checkbox"/>	Suffix	suffix	NO	Equal To
6	<input type="checkbox"/>	Nickname/Alias	aliasUpper	NO	Equal To
7	<input type="checkbox"/>	Company	company	NO	Equal To
8	<input type="checkbox"/>	Title	title	NO	Equal To
9	<input type="checkbox"/>	ID	numberStringUpper	NO	Equal To
10	<input type="checkbox"/>	Driver's License	driverLicense	NO	Equal To
11	<input type="checkbox"/>	SSN / Tax ID	SsOrTaxNumberString	NO	Equal To
12	<input type="checkbox"/>	Birth Date	birthDate	NO	Equal To
13	<input type="checkbox"/>	Street	defaultAddress_Street	NO	Equal To
14	<input type="checkbox"/>	City	defaultAddress_City	NO	Equal To
15	<input type="checkbox"/>	State	defaultAddress_State	NO	Equal To
16	<input type="checkbox"/>	Postal Code	defaultAddress_PostalCode	YES	Equal To
17	<input type="checkbox"/>	County	defaultAddress_County	NO	Equal To
18	<input type="checkbox"/>	Country	defaultAddress_CountryItem	NO	Equal To
19	<input type="checkbox"/>	Phone	defaultPhone_PhoneString	YES	Equal To
20	<input type="checkbox"/>	Fax	defaultFax_FaxString	NO	Equal To
21	<input type="checkbox"/>	Email	defaultEmail_EmailString	NO	Equal To
22	<input type="checkbox"/>	Internet Address	defaultInetAddress_InetAddressString	NO	Equal To
23	<input type="checkbox"/>	Default Category	defaultCategory	NO	Equal To

[Check All](#) - [Uncheck All](#) [edit](#)

**A contact is a duplicate when:**

- All of the above conditions are met (AND logic)
- Any of the above conditions are met (OR logic)
- The following combination of the  above conditions is met

**Duplicate Matching Settings Interface**

In this example, the user may try to create a contact (type company) with the following field data:

- **Company Name** - Omnivision Private Investigators
- **Postal Code** - 90032
- **Default Phone Number** - 323-555-0770

In the same way as explained previously, when the user clicks **Save**, a message appears showing possible duplicates that have been found:

The screenshot shows the TeamConnect Legal interface. The search bar contains 'Contacts'. The navigation menu includes Home, Legal, Finance, Contacts, Calendar, Documents, Admin, and All Services. The main content area is titled 'Contacts' and shows a 'New Company' form. A yellow warning box titled 'Possible Duplicates Found!' contains the following information:

Possible Duplicates Found!		
<input type="checkbox"/> Allow contact to be saved		
Omnivision Private Investigators 4500 Wilshire Blvd #213 Los Angeles, CA 90032	Omnivision Private Investigators	Omnivision Private Investigators 4500 Wilshire Blvd #214 Los Angeles, CA 90032

Below the warning box, the form fields are as follows:

- Company Information:** \*Company Name: Omnivision Private Investigators
- Address Information:**
  - Type: Business
  - Street: 4500 Wilshire Blvd #214
  - City: Los Angeles
  - State/Province: CA
  - Zip/Postal Code: 90032
  - County: (empty)
  - Country: (Select)
- Phone Number:** Business, 323-555-0770
- Fax Number:** (Select)

**Possible Duplicate Contacts Found Message Example 2**

Notice that the duplicates differ slightly from each other. In some cases, the Duplicate Contact Manager found duplications in company name and phone number. In others, it found duplications in company name and postal code. Both kinds of cases were specified in the Manager, so duplicates of both kinds are reported.

In this situation, the user may take one of the following actions:

- Override the duplicate contact check by selecting the **Allow contact to be saved** check-box and clicking **Save**. This option is only displayed if you selected the **Allow users to override** check-box on the Duplicate Contact Manager Settings screen.
- Click an existing contact link to verify that the contact is actually unique.

#### 3.1.2.4 Using XML Worksheet Tool

The XML Worksheet Tool executes Extensible Markup Language (XML) requests that can perform functions such as inserting, updating, and deleting data in TeamConnect. This can be useful in cases where you need to create a large number of records for testing performance or search functionality, or to populate a fresh instance of TeamConnect with "stock" data.

##### To execute XML code using XML Worksheet Tool

1. In the Designer window, in the **Tools** drop-down list select **Xml Worksheet**.  
The corresponding **Xml Worksheet** screen appears.
2. Enter your XML code in the **Xml** text box. Alternately, click the **Choose File** button to locate a local XML document that contains the code you want to run. Click the **Upload** button after the local XML document is selected to populate the **Xml** text box with the contents of the document.  
Click the **Clear Request** link to reset the **Xml** text box to empty.
3. Click the **Post Data** button to execute the XML code in the **Xml** text box.  
The results of your XML code display in the **Results** text box.

#### XML Worksheet Code Format

The XML code used in XML Worksheet Tool requires several sections that must be organized in the following order:

- Request—XML document wrapper.
- Authentication—The TeamConnect username and password of an account that has the proper rights to perform the operation defined in the XML code.
- One or more records with the following components and their respective attributes:
  - **Entity** and **Operation**—Entities correspond to System Object definitions with special case for Custom Object Definitions. Operations are used to insert, delete, or update records.
  - **Elements**—These represent System or Custom Fields. System Field tags are generally their labels. Custom Field tags are their field names.

In TeamConnect, tags represent fields that have corresponding methods in the API. All System Objects and their system fields can be accessed with XML. Custom Objects and their fields can also be accessed with XML.

#### Sample XML Worksheet Code



The following XML code sample will create a contact of type Person named "John Doe" in TeamConnect. The sample contains all the sections described in [XML Worksheet Code Format](#).

```
<TeamConnectRequest>
  <Authentication>
    <Username>TCUser</Username>
    <Password>TCPassword</Password>
  </Authentication>
  <Contact op="insert">
    <Type>P</Type>
    <FirstName>John</FirstName>
    <Name>Doe</Name>
    <Alias>Test Contact</Alias>
  </Contact>
</TeamConnectRequest>
```

### 3.1.2.5 Creating Custom Tools

You may develop and create custom tools to meet the specific needs of your organization using Java and XML code. For information on creating custom tools, see [Using the TeamConnect API](#).

#### Points To Remember

When creating custom tools, keep in mind:

- If you grant access to a tool, the user is able to both view and use the tool.
- You may grant group rights so that specific user groups can use the tool.
- You may set up loggers to monitor tools.

**Note:** Custom tools are developed using Java, and the user interface for the tools is developed using XML.

#### Setting Up Custom Tools

After you have developed a custom tool, you must set it up to work within TeamConnect by doing the following operations:

- Upload the Java and XML files to TeamConnect.
- Configure the tool in the **Administer Custom Tools** screen.
- Grant rights to solution developer users who are to test the tool.
- Test the tool on a test instance of TeamConnect before uploading it to your production system.
- After testing the tool, you must [grant access to the tool](#) on the **Tools** option of group accounts. That way, end-users in the appropriate user groups can access the tool from the **Tools** link in **All Services > Tools** drop-down list in the end-user interface.

#### 3.1.2.5.1 Uploading Custom Tool Files

Before you can use or operate custom tools, you must first upload the code files you have written to TeamConnect.

##### To upload the custom tool files

1. In the end-user application, select **Documents** in the tab bar.  
The **Documents** screen appears.
2. Navigate to **Top Level > System > Tools**.
3. Create a folder for the tool (for example, **Batch\_Invoicing**).

**Important:** *The name of the custom tool folder cannot have spaces and it must be the same as the name of the main class file.*

4. In the new folder, create the following folders:
  - **Classes**
  - **Resource**
5. Upload the Java files to the **Classes** folder.
6. Upload the XML files to the **Resource** folder.

After you have uploaded the necessary files, you must configure the custom tools in TeamConnect.

#### 3.1.2.5.2 Configuring Custom Tools

To enable your custom tools, you must name them and point to their folder from the **Administer Tools** screen.

##### To configure a custom tool

1. In the Designer window, in the **Tools** drop-down list select **Administer Tools**.  
The corresponding **Administer Tools** screen appears.

Administer Tools

Number of entries you would like to add: 1

	Name	FolderName
1	Batch Invoicing	Batch_Invoicing
+ add more		
	Name	FolderName
1	<input type="checkbox"/> Export Design	Export
2	<input type="checkbox"/> Import Design	Import
3	<input type="checkbox"/> Reassign Work	Reassignment
4	<input type="checkbox"/> Xml Worksheet	Worksheet

[Check All](#) - [Uncheck All](#)

Administer Tools Screen

2. Enter a name for the tool in the **Name** field.
3. From the **Folder Name** drop-down list select the directory in which the class and XML files are located.
4. Choose whether this tool will appear in the end-user interface (**User Tool**) or in the Designer interface (**Setup Tool**).
5. Click **add more**.

The custom tool is now available in TeamConnect.

**Important:** You must give rights to users on the **Tools** tab of the user or group accounts in order for them to access the custom tool.

### 3.1.2.6 Appenders

Appenders are definitions of logs. They are classified according to the type of output they generate and whether they are defined for system or audit logging.

Although many features of appenders and loggers are configured in Designer, there are some features that are controlled by the **Admin** tab in the main TeamConnect user interface. These Admin features include:

- [Turning loggers on and off](#)
- [Selecting the severity levels](#) that generate log entries

When the TeamConnect application starts running, all logs defined by active appenders start to log generated messages. When you define a new appender or make changes to an existing appender, the changes take effect the next time you restart TeamConnect on the application server. When you delete an appender, the log continues to be generated until you restart TeamConnect.

You can define the following types of appenders which generate logs in corresponding output formats:

- [File Appenders](#)—Define logs that capture messages in a text file as they are generated by the system.
- [SMTP Appenders](#)—Define logs to be sent by email when a log event and when a severity of **Error** or higher is logged to a buffer.
- [Socket Appenders](#)—Define logs that transmit logger messages to a port on a remote host.

If you need a system log that only logs messages for a certain area of TeamConnect, such as custom blocks, you can associate the appender with the corresponding logger. For more details, see [Loggers](#).

#### 3.1.2.6.1 Default System and Audit Appenders

By default, TeamConnect has several active file appenders—that is, the logs that they generate are written as text files. These files are stored in the location specified for log folders at the time when TeamConnect was installed. There are three default logging appenders:

- **Default** (for system logging)—Logs all of the messages that are generated by all of the system loggers and writes them to the **system.log** file. You can deactivate the **Default** appender, but it cannot be deleted.

If you want a log that isolates messages for a specific area of TeamConnect, you must define an appender for the corresponding logger.

- For instructions on defining a system appender, see [Defining System Appenders](#).
- For a description of the Default system log file, see [Pattern Example](#).
- **XML Appender**—Logs messages regarding XML requests, according to the level selected for the **XML** logger, and writes them to the **xml.log** file.
- **Default** (for audit logging)—Logs messages from all active audit loggers and writes them to the **audit.log** file. You can deactivate the **Default** appender, but it cannot be deleted.

**Note:** *If Collaborati Spend Management is installed on your system, the **Collaborati Appender** is installed by default.*

#### 3.1.2.6.2 Viewing Logs Generated by File Appenders

The system and audit logs generated by file appenders are stored on the web server in a location that is specified when TeamConnect is installed. If you have access to the web server, you can directly view the logs on the server.

If you do not have access to the web server, you can view the logs directly in TeamConnect using the **View Log** hyperlink that is provided for each file appender.

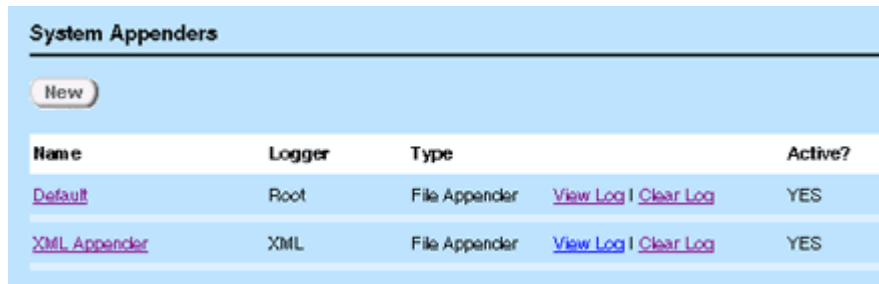
**Tip:** *To find out where your logs are stored, see the path displayed in the **Application Logs Folder** field in a file appender screen.*

The log location is specified using the app.logFolder parameter in the Web Application Deployment Descriptor when TeamConnect is installed.

#### To view a log file

1. On the **Admin** menu bar in the **System Settings** drop-down list, select either **System Logging** or **Audit Logging**, depending on the type of log you want to view.

The **Logging** screen opens, with the corresponding tab displayed by default.



System Appenders				
<a href="#">New</a>				
Name	Logger	Type	Active?	
<a href="#">Default</a>	Root	File Appender	<a href="#">View Log</a>   <a href="#">Clear Log</a>	YES
<a href="#">XML Appender</a>	XML	File Appender	<a href="#">View Log</a>   <a href="#">Clear Log</a>	YES

#### Views or Control Logs

2. In the **System Appenders** (or **Audit Appenders**) section of the tab, locate the name of the log you want to view and click the corresponding **View Log** hyperlink.

The current log file for the log you selected is displayed in a new browser window. The most recent log events are listed at the bottom of the log file.

3. To view the latest log messages, press F5 or click your browser's **Refresh** button.

#### 3.1.2.6.3 Clearing Logs Generated by File Appenders

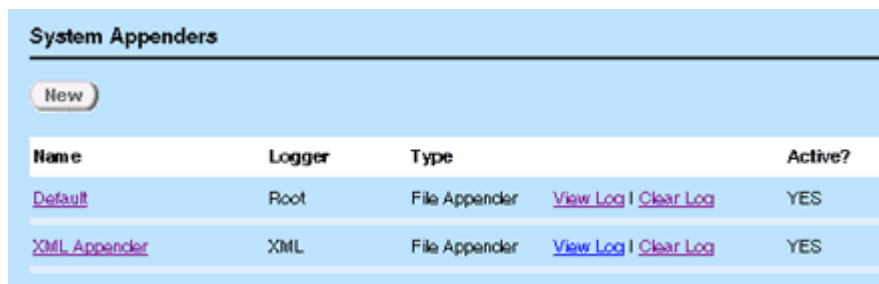
If the **Rollover Policy** field of an active file appender is set to **Rollover on schedule**, you can manually trigger the removal of the current log entries in a log file and start logging again in an empty file.

#### To clear a log file

1. On the **Admin** menu bar in the **System Settings** drop-down list, select either **System Logging** or **Audit Logging**, depending on the type of log you want to clear.

The **Logging** screen opens, with the corresponding tab displayed by default.

2. In the **System Appenders** (or **Audit Appenders**) section of the tab, click the **Clear Log** hyperlink next to the file appender log you want to clear.



System Appenders				
<a href="#">New</a>				
Name	Logger	Type	Active?	
<a href="#">Default</a>	Root	File Appender	<a href="#">View Log</a>   <a href="#">Clear Log</a>	YES
<a href="#">XML Appender</a>	XML	File Appender	<a href="#">View Log</a>   <a href="#">Clear Log</a>	YES

#### Clearing a Log File

The current log entries in the log file are removed and logging begins in the empty file.

3. To view any new log messages, click the corresponding **View Log** hyperlink.

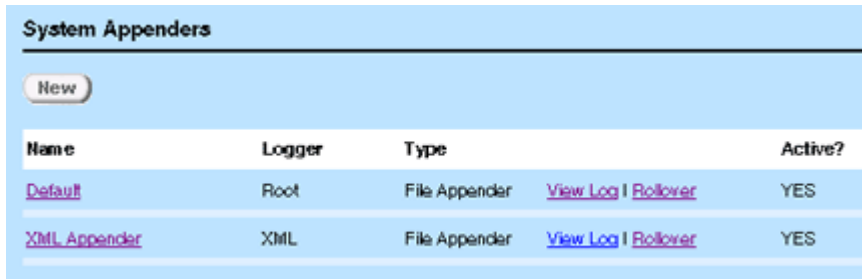
### 3.1.2.6.4 Rolling Over Logs Generated by File Appenders

If the **Rollover Policy** field of an active file appender is set to **Rollover on size**, you can manually trigger the rollover of the current log file and start logging again in new file.

#### To clear a log file

1. On the **Admin** menu bar in the **System Settings** drop-down list, select either **System Logging** or **Audit Logging**, depending on the type of log you want to clear.

The **Logging** screen opens, with the corresponding tab displayed by default.



Name	Logger	Type	Active?
<a href="#">Default</a>	Root	File Appender	YES
<a href="#">XML Appender</a>	XML	File Appender	YES

**System Appenders Section**

2. In the **System Appenders** (or **Audit Appenders**) section of the tab, click the **Rollover** hyperlink corresponding to the file appender log you want to roll over.

The current log file is rolled over and logging begins in an empty file.

3. To view any new log messages, click the corresponding **View Log** hyperlink.

### 3.1.2.6.5 File Appenders

File appenders define logs that capture messages in a text file as they are generated by the system. The log file is stored in the location that was specified when TeamConnect was installed.

Whether you are defining a file appender for system logging or audit logging, the settings that are specific to file appenders are the same.

For general instructions on defining appenders, see one of the following sections:

- For system logging, see [Defining System Appenders](#).
- For audit logging, see [Defining Audit Appenders](#).

**File Appender Information**

Application Logs Folder: [C:\bea\user\\_projects\Wtritech\ClientGuard\logs](#)

\* File Name:

Rollover Policy:

\* Max. Backup Index:

\* Max. File Size (MB):

Immediate Flush

Layout:

\* Pattern:

File Appender Information Section

The following table describes the file appender settings.

File Appender Information

Field or Button	Description
<b>Application Logs Folder</b>	<p>Indicates the location of the log files on the application server as a hyperlink. This location is specified when TeamConnect is first installed.</p> <p><b>Note:</b> You can click the hyperlink to view the contents of the folder if you are using the same machine where TeamConnect is installed. Otherwise, navigate to the folder on the application server.</p>
<b>File Name</b>	<p>Type the name that you want to use as the name of the log file, including its file extension.</p> <p>If you are using HTML or XML layout for the log file, use the corresponding extension.</p> <p>If you want to specify a subdirectory of the logs folder on the application server, you can type the directory location in this field, relative to the logs folder indicated in the <b>Application Logs Folder</b> field. For example, if you want the <b>RulesLog.log</b> file to be stored in a subdirectory called <b>Custom_Rule_Logging</b>, then you would type Custom_Rule_Logging\RulesLog.log as the file name.</p> <p><b>Important:</b> To organize your log files, you must manually create the corresponding subdirectories directly on the application server.</p>
<b>Rollover Policy</b>	<p>Determines when you want the system to stop writing to a file, add a timestamp to the file name, and begin writing to a new file. Select</p>

	<p>one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Rollover on Schedule</b>—Allows the log file to be rolled over daily, weekly, or monthly, as specified by the value of the <b>Backup Schedule</b> setting.</li> </ul> <p><i>Important: If you select <b>Rollover on Schedule</b>, previously rolled log files are not automatically replaced by new files. You have to periodically clean up your logs directory to delete old log files.</i></p> <ul style="list-style-type: none"> <li>• <b>Rollover on Size</b>—Allows you specify a file size at which the log file is rolled over and how many backup files to store before replacing them. This is the default rollover policy.</li> </ul>
<b>Max. Backup Index</b>	<p>The maximum number of backup files to be stored before backup files are replaced by new files. The oldest file is replaced each time a new file is rolled over. By default, the maximum backup index is 5.</p> <p><i>Important: Specifying a larger number, such as 10, may negatively impact performance.</i></p> <p>This field is only displayed if <b>Rollover on Size</b> is selected as the rollover policy.</p>
<b>Max. File Size (MB)</b>	<p>The maximum file size, in megabytes, that the log file should reach before rolling over and starting a new file. By default, the maximum file size is 10 MB.</p> <p>This field is only displayed if <b>Rollover on Size</b> is selected as the rollover policy.</p>
<b>Backup Schedule</b>	<p>Select how often the log file should be rolled over and timestamped and a new file should be created—<b>Daily</b>, <b>Weekly</b>, or <b>Monthly</b>.</p> <p>This field is only displayed if <b>Rollover on Schedule</b> is selected for the rollover policy.</p>
<b>Immediate Flush</b>	<p>Select this check-box to have each write of a logging event followed by a flush operation. This helps prevent all or part of a log message from being lost should the application exit abruptly. This check-box is selected by default.</p> <p>If this check-box is not selected, logging throughput is improved by about 15%, but it is more likely that if the application exits abruptly, a logging message that was written might not be completely captured in the log.</p>



<b>Layout</b>	Select whether you want to use a <b>Pattern</b> , <b>HTML</b> , or <b>XML</b> layout for the contents being written to the log file. <b>Pattern Layout</b> is selected by default.  For more details, see <a href="#">File Appender Layouts</a> .
<b>Pattern</b>	This field becomes available when <b>Pattern Layout</b> is selected in the <b>Layout</b> field.  Type the pattern you want to use to display the logging messages in the log file. For more details, see <a href="#">Pattern Layout</a> .

### 3.1.2.6.5.1 File Appender Layouts

When you define a file appender, you must specify how you want the contents of the file to be organized and formatted. Log4j® provides several types of layouts for file appenders, including Pattern, HTML, and XML layouts. Each of these layouts has the following advantages and disadvantages:

#### Comparison of File Appender Layouts

Layout	Advantages	Disadvantages
<a href="#">Pattern Layout</a>	Provides easy control over content and organization of log file	<ul style="list-style-type: none"> <li>Limited control over output</li> <li>Difficult to parse for specific information</li> </ul>
<a href="#">HTML Layout</a>	<ul style="list-style-type: none"> <li>Requires no configuration or formatting</li> <li>Output is clean, organized and easy to read</li> </ul>	<ul style="list-style-type: none"> <li>Fixed format</li> <li>Difficult to parse for specific information</li> </ul>
<a href="#">XML Layout</a>	Easiest output to parse for specific information using other software	<ul style="list-style-type: none"> <li>Might require more configuration and design files to be created and implemented</li> <li>More difficult to implement</li> </ul>

The pattern layout is a flexible layout approach for file appenders. You specify what information you want to include in the log file by providing a string of characters in the **Pattern** field of a file appender. These instructions are interpreted by the system, so that the resulting log file contains the information you need.

The pattern for a file appender is constructed using conversion specifiers and static characters you might want to include, as demonstrated in the [Pattern Example](#).

## Conversion Specifiers

Conversion specifiers indicate what information to write to the log. The following table describes all possible conversion specifiers for pattern layouts.

#### Log4j® Pattern Layout Conversion Specifiers

Conversion specifier	Output in log file
%c	<p>The name of the logger with which the logging event is associated.</p> <p>If desired, you can specify the desired number of right-most components of the logger name to be printed in the log by following the %c with a precision specifier (a decimal constant within braces). By default, the logger name is printed in full.</p> <p>For an example of how the Default system log uses a precision specifier with this conversion specifier, see <a href="#">Pattern Example</a>.</p>
%C	<p>The fully qualified class name of the calling object issuing the logging request.</p> <p>If desired, you can specify the desired number of right-most components of the class name to be printed in the log by following the %C with a precision specifier (a decimal constant within braces).</p> <p>See <a href="#">Pattern Example</a> for an example of how the Default system log uses a precision specifier with this conversion specifier.</p> <p><b>Caution:</b> <i>Generating this information for a log is extremely slow.</i></p>
%d	<p>The date of the logging event.</p> <p>If desired, you can follow the %d with a date format specifier within braces. For example, %d{HH:mm:ss,SSS} and %d{dd MMM yyyy HH:mm:ss} are acceptable. By default, ISO8601 format is used.</p> <p>Log4j® has its own date formatters. These can be specified using one of the following:</p> <ul style="list-style-type: none"> <li>• ABSOLUTE</li> <li>• DATE</li> <li>• ISO8601</li> </ul> <p>See the <a href="#">Pattern Example</a> for an example of how the Default system log uses a date format specifier with this conversion specifier.</p>
%F	<p>The file name where the logging request was issued.</p> <p><b>Caution:</b> <i>Generating this information for the log is extremely slow.</i></p>

%l	The location information of the calling object which generated the logging event.  Location information includes the file name where the logging request was issued, the line number in the file, the class name, and the method name from which the message was logged.  <b>Caution:</b> <i>Generating this information for the log is extremely slow.</i>
%L	The line number of the file where the logging request was issued.  <b>Caution:</b> <i>Generating this information for the log is extremely slow.</i>
%m	The message associated with the logging event.
%M	The method name where the logging request was issued.  <b>Caution:</b> <i>Generating this information for the log is extremely slow.</i>
%n	The platform-dependent line separator character or characters.
%p	The level of the logging event.
%r	The number of milliseconds elapsed from the start of TeamConnect until the creation of the logging event.
%t	The name of the thread that generated the logging event.
%%	A single percent sign.

## Pattern Example

The following pattern shows the use of several conversion specifiers that write information to the log. This is the pattern used by the **Default** system log:

```
%d{ABSOLUTE} [%t] [%p] [%c{1}] %C{1}.%M - %m%n
```

This can be roughly translated as the following:

```
date [thread that generated the logging event] [level of the logging event] [logger of the logging event] class.method - message followed by a line return
```

For example, an entry in the Default system log might be:

```
11:20:44,005 [jessica] [DEBUG] [sql] TCLog$SQLLogWriter.write - UnitOfWork(2420720)
—end unit of work commit
```

Notice that the spaces, brackets, hyphen and period in this example are printed in the generated line of text in the log as static text.

The HTML layout for file appenders writes logging events in a clean-looking table. Each logging event written to the log is added as a row in the table. The columns in this table are described as follows:

#### File Appender HTML Layout Columns

Time	Thread	Level	Category	Message
Milliseconds elapsed since the TeamConnect was started until	Name of the thread that generated the event	Level of the event	Logger associated with the event	Message associated with the event

**Tip:** In HTML layout, the time of an event is displayed relative to the start time of the application. To track events logged over long periods of time, specify frequent roll over.

The following table is an example of a log file in HTML layout:

Time	Thread	Level	Category	Message
8732	ExecuteThread: '10' for queue: 'default'	DEBUG	syslog.basic.sql	ServerSession(7603551)-client acquired
8742	ExecuteThread: '10' for queue: 'default'	DEBUG	syslog.basic.sql	ServerSession(7603551)-Connection(8749163)-SELECT t1.MODIFIED_ON, t1.PRIMARY_KEY, t1.PROPERTY_MAP, t1.CUSTOM_CLASS_NAME, t1.THRESHOLD_ID, t1.VERSION, t1.CREATED_ON, t1.NAME, t1.IS_ACTIVE, t1.CREATED_BY_ID, t1.LOGGER_ID, t1.MODIFIED_BY_ID FROM Y_LOGGER t0, U_LOG_APPENDER t1 WHERE (t0.PRIMARY_KEY = 2) AND (t0.PRIMARY_KEY = t1.LOGGER_ID)
8752	ExecuteThread: '10' for queue: 'default'	DEBUG	syslog.basic.sql	ClientSession(1142426)-client released
8752	ExecuteThread: '10' for queue: 'default'	DEBUG	syslog.basic.sql	ServerSession(7603551)-client acquired
8762	ExecuteThread: '10' for queue: 'default'	DEBUG	syslog.basic.sql	ServerSession(7603551)-Connection(3908029)-SELECT t1.MODIFIED_ON, t1.PRIMARY_KEY, t1.PROPERTY_MAP, t1.CUSTOM_CLASS_NAME, t1.THRESHOLD_ID, t1.VERSION, t1.CREATED_ON, t1.NAME, t1.IS_ACTIVE, t1.CREATED_BY_ID, t1.LOGGER_ID, t1.MODIFIED_BY_ID FROM Y_LOGGER t0, U_LOG_APPENDER t1 WHERE (t0.PRIMARY_KEY = 3) AND (t0.PRIMARY_KEY = t1.LOGGER_ID)
8772	ExecuteThread: '10' for queue: 'default'	DEBUG	syslog.basic.sql	ClientSession(8954781)-client released
8772	ExecuteThread: '10' for queue: 'default'	DEBUG	syslog.basic.sql	ServerSession(7603551)-client acquired
8772	ExecuteThread: '10' for queue: 'default'	DEBUG	syslog.basic.sql	ServerSession(7603551)-Connection(2709039)-SELECT t1.MODIFIED_ON, t1.PRIMARY_KEY, t1.PROPERTY_MAP, t1.CUSTOM_CLASS_NAME, t1.THRESHOLD_ID, t1.VERSION, t1.CREATED_ON, t1.NAME, t1.IS_ACTIVE, t1.CREATED_BY_ID, t1.LOGGER_ID, t1.MODIFIED_BY_ID FROM Y_LOGGER t0, U_LOG_APPENDER t1 WHERE (t0.PRIMARY_KEY = 4) AND (t0.PRIMARY_KEY = t1.LOGGER_ID)
8792	ExecuteThread: '10' for queue: 'default'	DEBUG	syslog.basic.sql	ClientSession(7008338)-client released
	ExecuteThread:			

File Appender HTML Layout Example

The XML layout writes logging events in a fixed format. This layout is implemented without more configuration for TeamConnect. Therefore, it does not generate a completely well-formed XML file. It is designed to be included as an external entity in a separate file to form a correct XML file. This allows you to determine how the logging events should be displayed in the log.

The XML layout can be useful for creating output that can be parsed by other software to create a report of specific information. For example, to create an audit report containing a summary of methods accessed by users, you can more easily parse XML than pattern text or HTML.

The following is an example output of a logging event in XML layout:

```
<log4j:event logger="syslog.basic.sql" timestamp="1063916251658" level="DEBUG"
thread="ExecuteThread: '10' for queue: 'default'">
<log4j:message><![CDATA[ServerSession(4688271)--client acquired
]]></log4j:message>
</log4j:event>
```

To utilize the XML layout, you need to create the necessary files to use the XML output as content that is pulled in and formatted correctly.

### 3.1.2.6.6 SMTP Appenders

TeamConnect's SMTP appender collects the logger messages in a buffer with a specified size. When the buffer receives a message with a severity of **Error** or higher, the SMTP appender sends an email message with the log events currently in the buffer to the specified email addresses.

#### To display the SMTP Appender Information

1. On the Admin menu in the **System Settings** drop-down list, select **System Logging** or **Audit Logging**, depending on which type of logging you need to define.
2. In the **System Appenders** section, click **New**.  
A new appender screen opens.
3. Select **SMTP Appender** from the **Appender Type** drop-down list.

The **SMTP Appender Information** section is displayed.

For information on how to define appenders, see [Defining System Appenders](#) or [Defining Audit Appenders](#). Whether you are defining an SMTP appender for system or audit logging, the SMTP appender settings are the same. The following image shows a set of sample settings:

**SMTP Appender Information**

\* To: tcadmin@clientguard.com

Buffer Size: 512

Evaluator Class: SMTPAppender.class

Include Location Information

Layout: Pattern Layout

\* Pattern: [%t] [%p] [%c(1)] %C(1).%M - %m%n

**SMTP Appender Information**

In addition to the fields described in the following table, you must configure the **Outgoing Mail Server Settings** section of the **Email** page under **Admin Settings**. For more details, see [the Admin Settings. Email Page Field Descriptions table](#).

#### SMTP Appender Information

Field	Description
<b>To</b>	The email addresses to which the log should be sent when a triggering log event occurs. Use commas to separate each address.
<b>Buffer Size</b>	The maximum size, in kilobytes, of the buffer that collects logging events. When the maximum buffer size is reached, the oldest events are deleted as new events are added to the buffer.  The default buffer size is 512 kilobytes.
<b>Evaluator Class</b>	To <a href="#">specify a log level</a> other than Error (the default) to trigger email messages, you must upload a class that modifies the triggering level. Enter the name of the class file in this field.
<b>Include Location Information</b>	Select this option to include the location information in the logging events placed in the buffer. For a description of the location information, see <a href="#">the Log4j® Pattern Layout Conversion Specifiers table</a> .  If this option is not selected, the location information is replaced with a question mark (?) in the log. However, it can help reduce the generation time of the logging events.
<b>Layout</b>	Select whether you want to use a <b>Pattern</b> , <b>HTML</b> , or <b>XML</b> layout for the contents being written to the buffer. <b>Pattern Layout</b> is selected by default.  The layout functionality is identical to that of the file appender. For more details, see <a href="#">File Appender Layouts</a> .
<b>Pattern</b>	This field becomes available when <b>Pattern Layout</b> is selected in the <b>Layout</b> field.  Type the pattern you want to use to display the logging messages in the log file. For more details, see <a href="#">Pattern Layout</a> .

#### 3.1.2.6.7 Socket Appenders

A socket appender transmits logger messages to a port on a remote host. On the remote host, the messages can then be logged as if generated locally. Using a socket appender allows you to send

logs from various systems to a single server where all logs for your organization are monitored or stored.

To use a socket appender, you must define a socket server to receive the log messages. Log4j® includes socket server examples as part of its distribution.

Whether you are defining a socket appender for system logging or audit logging, the settings shown in the following image are the same.

**Socket Appender Information**

The following table describes the fields in the **Socket Appender Information** section of the Appender screen.

**Socket Appender Information**

Field	Description
<b>Remote Host</b>	The name or IP address of the server to which the logging events should be sent.
<b>Port</b>	The port number on the remote host to which the logging events should be sent.
<b>Reconnection Delay</b>	The number of milliseconds to wait between each failed attempt to connect to the server using the specified port. The default is 30000 milliseconds (30 seconds).  If you set this field to zero (0), the socket appender attempts to connect to the server only once. If the connection fails, TeamConnect does not try to reconnect.
<b>Include Location Information</b>	Select this option to include the location information in the logging events placed in the buffer.  Location information includes the file name where the logging request was issued, the line number in the file, the class name, and the method name from which the message was logged.  If this option is not selected, then the location information is replaced with a question mark (?) in the log. However, it can help reduce the generation time of the logging events.

For general instructions on defining all types of appenders, see one of the following:

- For system logging, see [Defining System Appenders](#).
- For audit logging, see [Defining Audit Appenders](#).

#### 3.1.2.6.8 Defining System Appenders

Although several default system appenders are provided, you can define new system appenders as needed. When you define a system appender, you are defining a log that you want generated for either one specific system logger or for all of the system loggers.

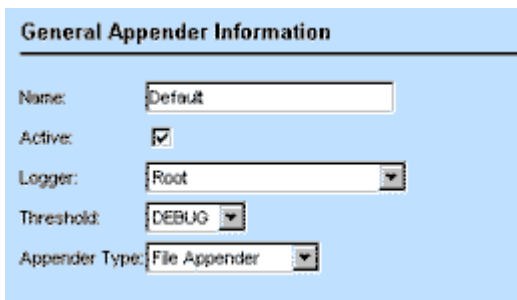
For example, you might want to define a log that only captures messages regarding custom rules that are defined for your implementation of TeamConnect. You can do this by defining an appender for the **Rule** logger.

**Note:** For descriptions of the system loggers, see [System Loggers](#).

#### To define a system appender

1. On the **Admin** menu in the **System Settings** drop-down list, select **System Logging**.
2. If the **Audit** tab is displayed in the **Logging** screen, select the **System** tab.
3. In the **System Appenders** section, click **New**.

A new appender screen opens.



General Appender Information for System Logging

4. In the **General Appender Information** section, enter the desired information as described in [the General Appender Information for System Logging table](#).
5. Depending on which type of appender you selected in the **Appender Type** field, enter the desired information in the specific fields for file, SMTP, or socket appenders as described in the corresponding tables:
  - The [File Appender Information table](#).
  - The [SMTP Appender Information table](#).
  - The [Socket Appender Information table](#).
6. Click **Save**.



- Restart the TeamConnect application on the application server.

The general information that must be filled out for all system appenders, regardless of their type, is shown in [the General Appender Information for System Logging image](#).

The following table describes the fields in the **General Appender Information** section:

**General Appender Information for System Logging**

Field	Description
<b>Name</b>	The name of the system appender as you want it to be identified in the list of system appenders in the <b>System Logging</b> screen.
<b>Active</b>	Select this check-box to activate the appender. Clear to deactivate the appender. Your change takes effect the next time the TeamConnect application is restarted.
<b>Logger</b>	Select whether you want the system appender to be defined for all loggers or one specific logger. The appender logs all messages associated with the logger you select. If you want the appender to be defined for all loggers, select <b>Root</b> .  For more details about the system loggers, see <a href="#">System Loggers</a> .
<b>Threshold</b>	Select the threshold for the system appender.  Allows you to limit the severity of messages to be captured in a particular log. The threshold is appender-specific, whereas the logger levels themselves are system-wide.  For example, if you want a certain log to capture only Error and Fatal messages, but you want another log to capture <a href="#">all levels of messages</a> , you can define an appender with the threshold set to <b>Error</b> , even though the logger level itself is set to <b>Debug</b> .
<b>Appender Type</b>	Select the type of appender you want to define. For details about each appender type, see the following: <ul style="list-style-type: none"> <li><a href="#">File Appenders</a></li> <li><a href="#">SMTP Appenders</a></li> <li><a href="#">Socket Appenders</a></li> </ul>

#### 3.1.2.6.9 Defining Audit Appenders

Appenders for audit logging differ from system appenders in the following ways:

- Because you do not have to set the levels of the audit loggers, your audit log appender does not need a threshold for which levels to log.

- You cannot define an appender for a single audit logger. Messages from all audit loggers that are turned on are logged in an audit appender.

For descriptions of the audit loggers, see [Audit Loggers](#).

### To define an audit appender

- On the **Admin** menu in the **System Settings** drop-down list, select **Audit Logging**.
- If the System tab is displayed in the **Logging** screen, select the **Audit** tab.
- In the **Audit Appendors** section, click **New**.

A new appender screen is displayed.

**General Appender Information for Audit Logging**

- In the **General Appender Information** section, enter the desired information as described in [the General Appender Information for Audit Logging table](#).
- Depending on which type of appender you selected in the **Appender Type** field, enter the desired information in the specific fields for file, SMTP, or socket appenders as described in the corresponding table:
  - The [File Appender Information table](#).
  - The [SMTP Appender Information table](#).
  - The [Socket Appender Information table](#).
- Click **Save**.
- Restart the TeamConnect application on the application server.

After you restart TeamConnect, messages are written to the log that you defined.

The following table describes the general appender information that must be filled out for all audit appenders, regardless of their type.

**General Appender Information for Audit Logging**

Field	Description
<b>Name</b>	The name of the audit appender as you want it to be identified in the list of audit appenders in the <b>Audit Logging</b> screen.
<b>Active</b>	Select this check-box to activate the appender. Clear to deactivate the appender.

	Your change takes effect the next time the TeamConnect application is restarted.
<b>Appender Type</b>	<p>Select the type of appender you want to define. For details about each appender type, see the following:</p> <ul style="list-style-type: none"> <li>• <a href="#">File Appendends</a></li> <li>• <a href="#">SMTP Appendends</a></li> <li>• <a href="#">Socket Appendends</a></li> </ul>

### 3.1.2.6.10 Loggers

*Loggers* categorize the messages that can be captured in a log according to the area of TeamConnect where that message is generated. Loggers are predefined according to area. You cannot create new loggers.

When you need to log messages from a certain area of TeamConnect, you define an appender for the logger that is associated with that area. For example, you can create a log for the portal panes area by defining an appender for the **Portal** logger. You can define a log for the user interface components by defining an appender for the **UI Components** logger.

This section includes the following topics:

- [System Loggers](#)
- [Audit Loggers](#)

#### 3.1.2.6.10.1 System Loggers

System loggers categorize messages in the code according to the area of TeamConnect that generates them, so that you can specify which area's messages you want to capture in a particular log. You can define multiple logs that capture messages from different TeamConnect components or all TeamConnect components.

For each log that you have defined, you can specify an overall threshold for what severity of messages you want written to it, so that the logs contain only the most pertinent information. For more details about the threshold for a log (or appender), see [the General Appender Information for System Loggers table](#).

For each available TeamConnect component, you can specify what severity (level) of messages you want written to the logs regarding this component, so that the logs contain only the most pertinent information about the component.

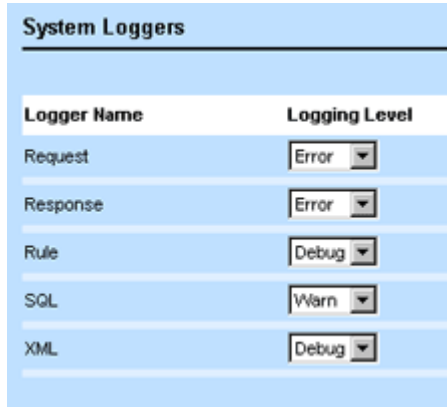
The loggers that are available for system logging are listed on the **System** tab of the **Logging** screen, as shown in the [System Loggers](#) and [Detailed System Loggers](#) images.

All of the system loggers have **Root** as their parent logger. This hierarchy allows appenders defined for the Root logger to be available to descendant loggers. For example, this means that the **Default** appender, which has been defined for the Root logger, receives logging statements from all of the loggers in the hierarchy.

**Tip:** Because the Default appender captures messages from all loggers in the system, you do not need to define another appender for the Root logger.

You can define an appender to be the target of a single logger in the hierarchy. This appender only receives logging statements from the logger for which it is defined. For example, if you need a text file that logs events for the batch display component, you define an appender for the **Batch Display Component** logger.

The following image shows the **System Loggers** section of the **Logging** screen:



System Loggers

The following table describes the **System Loggers** section of the **Logging** screen:

System Loggers Table

Logger	Description
<b>Request</b>	Logs HTTP request information including headers, cookies, and parameters.  This logger can automatically be set to the Debug level each time TeamConnect application is restarted. This depends on the value of the syslog.basic.request parameter in the Web Application Deployment Descriptor.
<b>Response</b>	Logs HTTP response header information.
<b>Rule</b>	Logs messages generated by rules that have been defined for your implementation of TeamConnect.
<b>SQL</b>	Logs EclipseLink-generated SQL statements.  This logger can automatically be set to the Debug level each time TeamConnect application is restarted. This depends on the value of the syslog.basic.sql parameter in the Web Application Deployment Descriptor. For more details about this parameter and other web application parameters, see the Installation Guide.

**XML**

Logs messages regarding XML requests.

The following image shows the **Detailed System Loggers** section of the **Logging** screen:

Logger Name	Logging Level
Authentication	Warn
Batch Display Component	Warn
Custom Blocks	Warn
Deletion	Warn
IMAP	Warn
Portal	Warn
Reflection	Warn
Search	Warn
Stores	Warn
Sync Server	Warn
Tools	Warn
UI Components	Warn
WebDAV	Warn

**Detailed System Loggers**

The following table describes the Detailed System Loggers section of the Logging screen:

**Detailed System Loggers Table**

Logger	Description
Authentication	Logs all messages regarding user authentication.
Batch Display Component	Logs all messages regarding the batch display component.
Custom Blocks	Logs all messages regarding custom blocks.
Deletion	Logs all messages regarding record deletion.

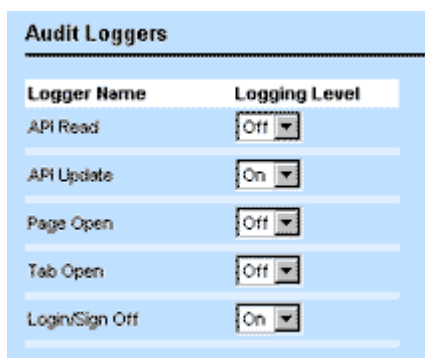
IMAP	Logs events related to the TeamConnect IMAP Server and Email Viewer.
Portal	Logs all messages regarding portal panes, including custom content.
Reflection	Logs all messages generated by TeamConnect's reflection factory.
Search	Logs all messages regarding search views and searching for records.
Stores	Logs all messages regarding TeamConnect stores.
Sync Server	Logs all messages regarding cache synchronization.
Tools	Logs all messages regarding both system and custom tools.
UI Components	Logs all messages regarding rendering of TeamConnect screens.
WebDAV	Logs all messages related to WebDAV document management.

### 3.1.2.6.10.2 Audit Loggers

Audit loggers send their messages to logs defined by audit appenders. Unlike system loggers, audit loggers have the following characteristics:

- Audit loggers are turned on or off system-wide. You do not have to set levels for audit loggers.
- When you define an appender for audit logging, all audit loggers that are turned on send their messages to the log. You cannot define separate logs for the various audit loggers.
- When you define an appender for audit logging, you do not have to specify a threshold. For details about defining audit appenders, see [Defining Audit Appendors](#).

The following image shows the available audit loggers.



**Audit Loggers**

The following table describes the **Audit Loggers** section:

**Audit Loggers Table**

Logger	Description
<b>API Read</b>	Turn on this logger to log read calls to the API.
<b>API Update</b>	Turn on this logger to log update calls to the API.
<b>Page Open</b>	Turn on this logger to log the opening of a page by a user.
<b>Tab Open</b>	Turn on this logger to log the clicking of a tab by a user.
<b>Login/Sign Off</b>	Turn on this logger to log user sign on/off events.

### 3.1.3 About Objects

*Objects* in TeamConnect represent business entities and concepts. Each object has an object definition that determines how object *records* that store your organization's information are created, displayed, and maintained.

When you customize TeamConnect, most of your work involves modifying or creating object definitions to meet the needs of your organization. For example, you might modify the Account *system object* that comes with TeamConnect or create a new "Property" *custom object*.

Object definitions determine the basic appearance of the records created by end users, each record's name and number, required fields, workflow, and so on. You can also create object definitions that dynamically pre-populate the record with certain data or automatically create any necessary related records.

Based on your business requirements, you can specify the following aspects of object definitions:

- **Properties**—Visual and functional aspects, such as custom fields and rules that require them to be populated before users can save the record. For example, you could create the "Case Caption" custom field for the Dispute object. Then, you could create a rule so that Dispute records cannot be saved if this field is empty.
- **Relationships**—Relationships to other objects, such as a dispute record that has several involved parties, which are related to contact records.

#### 3.1.3.1 System Objects

System objects represent business objects that are common to most businesses. TeamConnect comes with the following system objects:

- **Account**—Used to allocate and track various financial transactions.

- **Appointment**—Used to schedule events, engagements, or meetings for a particular purpose, place, and time with specific attendees (users) and resources.
- **Contact**—Information about people and companies, including rates charged for services, relations to projects and other contacts, and so on. You must create a contact record for a user before you can create their user account.
- **Contact Group**—Allows users to organize contact records in various Designer contact groups, which are called "Address Books" in the end-user interface.
- **Document**—Allows [users to upload](#) and manage files "attached" to records or directly uploaded to TeamConnect's **Documents** area. Documents are automatically available as a system block in all object views.

You can also upload files used to customize TeamConnect, such as XML and CLASS files used for custom blocks, custom actions, and so on.

- **Expense**—Allows users to track internal costs of doing business in their organization. Expenses are available as a system block in object views for custom objects.
- **Group**—Allows administrators to [define access rights](#), home pages, and object views per user group and [assign users to one or more groups](#).
- **History**—Allows users to make chronological entries and track events associated with a specific record. History is a related object; that is, history records can only exist when related to other records. History is available as a system block in object views.
- **Invoice**—Allows end users to track invoices or bills sent by vendors who provide their organization with goods or services. Invoices are available as a system block in object views for custom objects.
- **Legal Setting**—Used to hold settings for TeamConnect Legal Matter Management and the budget settings records after the settings for budgets for specific years.
- **Line Item (Invoice)**—Allows end users to list the goods or services specified in a vendor invoice. Users must enter the details of each line item, such as the type (fee or expense), price, quantity, and dates in the line items screen of the corresponding invoice. Line items are available as a system block in object views for custom objects.

Some of the object definition properties of Line Item, such as how to customize the display of specific fields, are controlled by System Setting screens. For more information, see [Setting up Invoice Line Items](#).

- **Involved**—Allows end users to keep track of various involved parties associated with a Project. An Involved record is automatically available as a system Block in Object Views for Custom Objects. This System Object definition is always related to a Custom Object.
- **Task**—Allows end users to track internal assignments they have to do, usually while working on Projects, for example preparing reports, following up on cases, reviewing documentation, and so on. Tasks are automatically available as a system Block in Object Views for Custom Objects.
- **User**—An account created for a specific person who uses TeamConnect. Users can be members of Groups, and can have Rights assigned to them or from a group in which they are a member. Each user has their own password to log in to TeamConnect, which is first specified in their user account (accessible by the Administrator only) together with the username and rights.

The corresponding object records are available in the end-user interface—except for user and group account records, which are only available to users with administrative rights.



The Involved object record is available only through the parent custom object records. For details, see [Unique Custom Object Types](#).

You can customize certain properties of system objects, such as their appearance, custom fields, and search views. However, you cannot modify the Group and User administrative system objects.

You cannot delete system object definitions or create new ones—except for Involved whose object definitions can be deleted from their parent custom object definitions.

System objects are represented by their own main tables in TeamConnect's object model, such as **TAccount** or **TTask**. The exceptions are User and Group objects, which are represented by the **YUser** and **YGroup** tables respectively. For details, see [Object Model: Read This First](#) plus the additional reference tables mentioned in that topic.

### 3.1.3.2 Custom Objects

You can create and define custom objects to meet the business requirements of your organization. For example, you might create a "Policy" object so that end users can create policy records to track each policy in your organization.

When you create a custom object, TeamConnect automatically creates an Involved object for it. For details, see [Unique Custom Object Types](#).

You can also delete custom object definitions during the customization process. However, once your TeamConnect system is in production, deleting custom objects or their components results in permanent loss of the data associated with them.

Custom object records are often referred to as "projects" within this documentation. After you define custom objects, they appear in the user interface with the names that you specify.

Custom objects are represented in TeamConnect's object model by the **TProject** table and a number of "administrative" tables such as **WObjdProjInfo**. For details, see [Object Model: Read This First](#) plus the additional reference tables this reference points you to.

### 3.1.3.3 Unique Custom Object Types

The unique custom object type in TeamConnect is **Involved**.

Involved records allow users to store information about involved parties (people or companies) who are involved in a project. Involved parties have roles such as witness, outside counsel firm, opposing party, and so on.

This objects is unique because its system object definition can only exist within custom object definitions. Likewise, these records can be viewed and edited only from within a custom object record.

### 3.1.3.4 Related Objects

Object records are often associated or related to other records. For example, each dispute record may have multiple related account records. Objects that have relationships with other objects are referred to as related objects.

End users can access related records from hyperlinks, pages, or blocks within a record to which they are related. For example, by default, each custom object is related to certain system objects. These system objects can be accessed from the following pages inside a custom object record:

- Accounts
- Appointments
- Documents
- Expenses
- Histories
- Involved
- Tasks

#### 3.1.3.4.1 Relationship Types

Objects in TeamConnect can have the following types of relationships:

- **Between system objects**—Most system objects are associated with the History and Document system objects. The exceptions are the objects themselves, the Contact Group (Address Book) object, and Group and User account objects.
- **Between system and custom objects**—All custom objects are associated with certain system objects, such as Account, Appointment, Expense, Task, Document, and History objects.
- **Between custom objects**—You can add custom fields of type Custom Object to custom object definitions so that end users can establish relationships between custom object records. For details, see [Custom Object Field Type](#).
- **Contact-centric relationships**—Certain custom objects are centered on a specific contact. For details, see [Contact-Centric Custom Objects](#).
- **Parent-child relations**—Parent (or "main") objects can have multiple child objects. Each child can have only one parent object. For example, a dispute record can have multiple negotiation records, but each negotiation can have only one dispute. This type of relationship is possible between custom object definitions and between account records. For more details, see [Parent-Child Relations between Custom Objects](#).

Parent-child relationships also exist between custom objects and their Involved objects, and between custom objects and embedded custom objects.

The following table lists objects and their default relationships in TeamConnect.

Related Object List

Object	Related Objects											
	Account	Appointment	Contact	Expense	History	Invoice	Line Item	Involved	Task	Document	Custom object	Embedded custom object
Account	x				x					x		

<b>Appoi ntmen t</b>					x					x		
<b>Conta ct</b>			x		x		x			x		
<b>Conta ct Group (Addr ess Book)</b>			x									
<b>Docu ment</b>					x							
<b>Expen se</b>					x					x		
<b>Group</b>					x							
<b>Histor y</b>										x		
<b>Invoic e</b>					x		x			x		
<b>(Invoi ce) Line Item</b>			x			x					x	
<b>Involv ed</b>	x				x					x		
<b>Task</b>					x					x		
<b>User</b>			x		x							
<b>Custo m object</b>	x	x		x	x			x	x	x	x	x

Embe dded custo m object																			
--------------------------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

## Points To Remember

In summary, *related objects* refer to:

- Child objects.
- Embedded objects
- Involved objects.
- Custom and system objects related to each other by default or through custom fields of type Custom Object.

The records of related objects can be accessed both independently from **All Services** (except for Involved, History, and embedded objects), and from hyperlinks, tabs, or blocks within the objects to which they are related.

### 3.1.3.4.2 Parent-Child Relations between Custom Objects

Because of their nature in the business world, certain custom objects cannot exist independently and always require a parent. For example, in the insurance industry, claims can only be filed against existing policies.

You can establish a parent-child relationship in the following ways:

- Create an embedded custom object within a custom object. For details, see [Embedded Custom Objects](#).
- Create two custom objects and specify one to use the other as a parent.

On the **General** tab of the child object definition, you select the parent object from a drop-down list and select a check box to confirm the object requires a parent. For details, see [General Custom Object Information](#).

## Results in End-User interface

Whenever you specify a parent custom object (for example, Policy for Claim) in a custom object definition, the relationship is represented in the following areas of the end-user interface:

- **General block of child records**—A field labeled **Parent *objectName*** is automatically created. With this field, the user can select the appropriate parent record. The record cannot be saved until a parent record is selected.

As soon as the user selects a parent record and saves the child record, the field displays the name of the selected parent record as a hyperlink, for example:

Parent Specialized Invoice: Correli Industries Specialized Invoice

- **Parent record screens**—A new block labeled with the plural form of the child custom object, for example, **Line Items**, is created, as shown in the following image. This block has a search field and provides quick access to the child records.

Line Items [Edit](#) | [Adjust](#)

View:  [Show Search Options](#)

Line Item 1 - 1 of 1

Item Type	Date	TK	Task/Exp	Act	Units	Rate	Disc	Adj	Amount	
1 Expense	07/21/2008		Expense			10.00	30.00	0.00	0.00	300.00
<b>Project:</b> <b>Activity:</b> <b>Description:</b>										

Line Items per page

**Example: Child Records Object in End-User Interface**

This block is automatically added to the system view of the parent object. However, if you are planning to create your own custom view for the parent object, you have to add this block to the object view manually. For details on object views, see [Object Views](#).

- **All Services**—When you define parent and child objects, you can check this box to hide them on the **All Services** drop-down list and the **All Services** page. If this box is checked, the object is also excluded from lists in the Global Search area.

If you choose to omit child objects from the **Go to** and **Create a new** drop-down lists, end users must go through the respective records of the parent object definition to create or access child custom object records.

**Important:** You must assign [user group rights](#) to both parent and child custom objects. Otherwise, users are not able to work with them.

## Results in Designer User Interface

Whenever you specify a parent custom object (for example, Tort for Dispute), the following changes happen in the object definition screens in the Designer interface:

- **Object Views tab**—A system block named after the child object in the plural is automatically created and become available for adding to the custom object views of the parent object.
- **Object Navigator window**—The **parent->** attribute appears in all of the child object definition's **TProject** tables, and you can use it to build attribute paths. For more details, see [Using Object Navigator](#).

### 3.1.3.4.3 Contact-Centric Custom Objects

Certain custom objects are centered on one specific Contact record. Such custom objects have contact-centric relationships.

For example, a loan cannot exist without an applicant. In this case, you need all Loan records to be linked to the Contact record of the individual or company applying for it. You can ensure this relationship when you create a Loan custom object. When you configure the object's general


properties, you select a check box to designate it as contact-centric. Then, you specify the role of the related Contact. For details, see [General Custom Object Information](#).

**Note:** Only custom objects (including embedded custom objects) can be contact-centric.

## Results in End-User Interface

When you create a contact-centric custom object (for example, Loan), the relationship appears in the following areas of the end-user interface:

- **General block of object records**—Displays a contact search module field labeled according to the label you specify when you define the object as contact-centric (in this example, Applicant). End-users are required to select the object's main contact in this field.



**Awards**

Award Type: Permanent Total \* Awarded To: Mariscal, Steven

Amount Awarded: \$500,000.00 Description: For damage to office caused by downstairs tenant

Date Awarded: 7/21/2008

Add Clear

<input type="checkbox"/>	Line	Award Type	Amount Awarded	Date Awarded	Awarded To	Description
	1 of 0					

Remove

Version 3.0 Build 0316  
Copyright © 1999-2008 [Mitratech Holdings, Inc.](#) All Rights Reserved.

**Example: Contact-Centric Record (Edit Mode)**

After the record is saved (and no longer in edit mode), the contact name appears as a link, for example, [Mariscal, Steven](#).

- **Involvement tab**—Displays a hyperlink to the object record after the end user selects the contact in the object record.

## Results in Designer User Interface

Whenever you create a contact-centric custom object definition (for example, Loan), the relationship appears in the following areas of the Designer interface:

- **Object Navigator window**—The contact-> attribute appears in all of the object definition's TProject tables, and you can use it to build attribute paths. For more details, see [Using Object Navigator](#).
- **Unique ID tab**—Displays the contact attribute in the list of attributes (if you are adding attributes to a naming pattern). For details, see [Adding Object Attributes](#).

### 3.1.3.5 Sub-Objects

There are certain items that contain information that pertains only to a certain record, and therefore exist only within that record. Because they are "owned" by a record, they cannot be accessed outside of it. In TeamConnect, these items are referred to as sub-objects. Categories, task and project assignees, contact addresses, emails, and skills are all examples of sub-objects.

Both system and custom objects can have sub-objects, and different objects have different sets of sub-objects. For example, custom objects have the sub-objects Categories, Assignees, and Relations. The sub-objects of the system object Appointment are Categories, Attendees, and Resources. The following table provides a complete list of the sub-objects that are available for TeamConnect objects.

**Sub-object List**

Object	Sub-objects			
	Categories	Assignees	Relations	Other
<b>Account</b>	x			
<b>Appointment</b>	x			<b>Attendee</b> <b>Resource</b>
<b>Contact</b>	x		x	<b>Address</b> <b>Email</b> <b>Fax</b> <b>Internet Address</b> <b>Phone</b> <b>Rate</b> <b>Skill</b> <b>Territory</b>
<b>Document</b>	x			
<b>Expense</b>	x			
<b>History</b>	x			
<b>Invoice</b>	x			<b>Line Item</b>
<b>Task</b>	x	x		

<b>Custom Object</b>	<b>x</b>	<b>x</b>	<b>x</b>	
<b>Involved</b>	<b>x</b>		<b>x</b>	
<b>Embedded Custom Object</b>	<b>x</b>			

Each sub-object has its own value set, or group of values, that must be defined in order to add that sub-object to a record. For example, to add a project assignee, the following values must be specified: an assignee (or user), a role, status (active or inactive), and the date of assignment.

Some values, such as status and date of assignment, are set automatically by the system. Others, such as assignee and role, must be entered by the end-user. In the end-user interface, sub-objects are added to a system or custom object record through a batch screen, where the data entry fields in each row indicate the values that are necessary for the sub-object. For details on using batch screens, see [Working with Batch Entries](#).

If your design requires templates and wizards, it is important to know a sub-object's value set and which values have to be set manually in the template. If any required manual values are not included in a template, the sub-object will not be added by the wizard. For more details, see [Adding Sub-objects to Templates](#).

#### 3.1.3.5.1 Embedded Custom Objects

An *embedded object* is a simplified custom object created within (or embedded in) another custom object. They are similar to child custom objects, but have fewer capabilities.

A child object is a custom object in a hierarchical relationship with another custom object. It has all the functionality of a custom object (such as wizards, rules, assignees, and phases), and has a dynamic life cycle and workflow.

An embedded object has a hierarchical relationship with a custom object, but does not have workflow or a dynamic life cycle. Embedded object records are best suited for static information—information that is not likely to change as the parent record moves from phase to phase.

Embedded objects do not have a workflow or dynamic life cycle. As outlined in [the Properties of Embedded and Child Object table](#), you cannot define phases, assignees, or rules in embedded objects. They are best suited for information that is not likely to change as the parent record moves from phase to phase. Because the information is owned by the parent record, it is not relevant to other records.

The following table compares embedded objects and child objects in key functional areas.

**Properties of Embedded and Child Objects**

<b>Functional area or property</b>	<b>Embedded objects</b>	<b>Child objects</b>
Accessible through <b>All Services</b>	No	Yes
Accessible through parent record	Yes	Yes



Can be contact-centric	Yes	Yes
Can have child objects	No	Yes
Unique ID	Yes	Yes
Auto naming patterns	Yes	Yes
Phases and phase transitions	No	Yes
Assignees	No	Yes
Categories	Yes	Yes
Custom fields	Yes	Yes
Blocks	No	Yes
Object views	No	Yes
Search views	Yes	Yes
Rules	No	Yes
Templates	No	Yes
Wizards	No	Yes
Related objects	No	Yes
Involved parties	No	Yes

### 3.1.4 Creating and Defining Objects

Creating and defining objects is the core procedure of customization and developing a business solution in TeamConnect. The properties you define for each object are consistently applied to all of the object's records, which are then seen in page form by end users.

For example, you can arrange data fields to be displayed with the same layout in each record. You can also configure an object to automatically name records with a pattern of data. This pattern might include the date on which the record was created, the record's category, and the name of user who created it.

Alternatively, you might also configure an object to automatically number records with a pattern of sequential numbers, the date the record was created, and even the record name. As you build it, step by step, each object displays, requests, and arranges the information you want for the end user, in the way you want it.

## Defining Objects

Defining objects is a multi-step procedure. Depending on your design, some steps may not be necessary. However, no matter how many steps you use to implement your design, the task of defining an object always requires:

- Thorough understanding of your organization's business requirements and processes
- Clear vision of the final goal you want to achieve

If an entity-relation diagram was created during the solution design stage, it may be very useful when defining objects. For more information, see [Related Objects](#).

- Knowledge of TeamConnect terminology

As your solution is developed, you need to understand basic terms used in this documentation. Terms are introduced throughout this document, and detailed descriptions can be found in the [Glossary](#).

### 3.1.4.1 About Object Definitions

An *object definition* is a specification of all the properties of an object. When fully defined, an object definition contains all of the details about an object, including its name, the icon that represents it in record screens, its categories, its display, and the custom fields that contain or request information specific to your organization.

Object definitions determine the appearance of records and search screens, how wizards are used to create records, naming and numbering patterns for (custom object) records, and business rules triggered when users are working with object records.

Each type of object definition screen has a specific set of tabs that allow you to manage different properties of the object:

General | Categories | Custom Fields | Forms | Blocks | Object Views | Search Views | Custom Messages | Rules | Templates | Wizards | Notifications | History |

#### System Object Definition Tabs

General | Search Views |  
**Contact Group  
 (Address Book), Group,  
 and User System Object  
 Definition Tabs**

General | Unique ID | Name | Phases | Phase Transitions | Assignee Roles | Categories | Custom Fields | Forms | Blocks | Object Views | Search Views | Custom Messages | Rules | Templates | Wizards | Embedded Objects | Notifications | History | Security |

#### Custom Object Definition Tabs

General | Unique ID | Name | Categories | Custom Fields | Forms | Custom Messages | Rules | List Display |

#### Embedded Object Definition Tabs

Keep in mind that the end-user interface pages display:

- Record information and fields in the main or right pane
- Objects associated with the current record as links and collections (search views) in the left pane

**Note:** *Designer refers to Contact Groups, but the end-user interface calls the same objects Address Books.*

The following table provides a summary of the properties that can be defined for different objects through these tabs. Tabs are listed in alphabetical order and may appear in a different order in the interface.

**System and Custom Object Definition Screen Tabs**

Tab	Object type	Description
<b>Assignee Roles</b>	Custom	Allows you to create new assignee roles or view the existing assignee roles created for the selected object. See <a href="#">Defining Assignee Roles for Custom Objects</a> .
<b>Blocks</b>	System and Custom	Allows you to create new custom blocks or view a list of all existing custom blocks defined for the object. See <a href="#">Blocks</a> . (Does not apply to embedded object definitions.)
<b>Categories</b>	System and Custom	Allows you to define new categories or view the existing categories created for the selected object. See <a href="#">Using Categories</a> .
<b>Conditions</b>	System and Custom	Allows you to create an object that you can use in routes to base the stops on conditional expressions. See <a href="#">Conditional Expressions</a> .
<b>Custom Fields</b>	System and Custom	Allows you to create new custom fields or view the existing organization-specific custom fields created for the object. See <a href="#">Custom Fields</a> .
<b>Document Types</b>	System (Document only)	Allows you to specify which document file types can be uploaded to TeamConnect.
<b>Embedded Objects</b>	Custom	Allows you to add new embedded objects or view a list of all existing embedded objects created for the selected parent object. See <a href="#">Embedded Custom Objects</a> .

<b>General</b>	System and Custom	Allows you to specify general object information for custom object definitions. This information includes the object name and icon file. For details, see: <ul style="list-style-type: none"> <li>• <a href="#">Creating Custom Objects</a></li> <li>• <a href="#">General System Object Information</a></li> <li>• <a href="#">Creating Custom Objects</a></li> <li>• <a href="#">Creating Embedded Objects</a></li> </ul>
<b>History</b>	System and Custom	Allows you to access a search screen where you can search for the history entries associated with the selected object definition.
<b>List Display</b>	Custom (Embedded only)	Offers options for displaying embedded records in the parent object records.
<b>Name</b>	Custom	Displays two options for naming for records of the selected custom object based on a pattern of attributes. See <a href="#">Naming Patterns for Custom Object Records</a> .
<b>Non-US Tax Categories</b>	System	Define non-US Tax Categories, including Codes and Descriptions.
<b>Notifications</b>	System and Custom	Allows you to customize which notification templates are sent at the object level. These settings will override the System Notifications settings in the TeamConnect Admin Settings. See <a href="#">Customizing Notifications for Object Definitions</a> .
<b>Object Views</b>	System and Custom	Allows you to create new object views or view a list of all existing object views defined for the object to meet your organization's specific needs. See <a href="#">Object Views</a> . (Does not apply to embedded object definitions.)
<b>Phase Transitions</b>	Custom	Allows you to create new transitions or view the existing transitions between the phases created on the <b>Phases</b> tab, that is the sequence in which the project phases follow each other. See <a href="#">Defining Phase Transitions for Custom Objects</a> .
<b>Phases</b>	Custom	Allows you to define new phases or view the existing phases created for the project life cycle of the selected custom object. See <a href="#">Creating Phases in Custom Objects</a> .

<b>Rules</b>	System and Custom	Allows you to create new rules or view the existing rules defined for the selected object. See <a href="#">Using Rules</a> .
<b>Search Views</b>	System and Custom	Allows you to create new search views or view a list of all search views created for the object. See <a href="#">Creating Search Views</a> .
<b>Security</b>	Custom	Allows you to set how <a href="#">record-level security</a> and restrictions of the custom objects' records should apply to their related records.
<b>Templates</b>	System and Custom	Allows you to create new templates or view a list of all existing templates created for the object. See <a href="#">Using Templates</a> .
<b>Unique ID</b>	Custom	Displays several options for setting a unique identifier for records of the selected custom object, for example, auto numbering patterns. See <a href="#">Unique Identifiers for Custom Object Records</a> .
<b>Wizards</b>	System and Custom	Allows you to create new wizards or view a list of all existing wizards created for the object. See the <a href="#">Creating Wizards</a> .

### 3.1.4.2 Viewing Object Definitions

You may access system object definitions and existing custom object definitions in the **Object Definition List** screen. The **Object Definition List** screen ([Object Definition List image](#)) provides a snapshot of the existing object definitions. This screen summarizes their general information and relationships to each other.

You may sort the object definition list into ascending or descending order by clicking the **Object Definition** label.

#### To open an existing object definition

1. From the **Go to** drop-down list click **Object Definitions**.

The **Object Definition List** screen appears.

Object Definition	Unique Code	Contact-Centric
Account	ACCT	
Appointment	APPT	
Batch Invoice	BINC	
Contact	CONT	
Contact Group	CTGR	
Document	DOCU	
Expense	EXPE	
History	HIST	
Policy	PLCY	Policyholder
Claim	CLAI	Insured
Claim Involved Party	CLCL	
Claim Milestone	CLML	
Litigation	LITI	Defendant
Medical Report	MDRP	Patient
Recorded Statement	RCST	Witness
Rental	RENT	
Vehicle	VHCL	Operator
Policy Involved Party	PLIN	

Object Definition List

- Click the hyperlink of the object definition you want to open.

The **General** tab of the selected object definition appears.




Click the tab where you want to view, add, or modify the object information. For details, see [About Object Definitions](#).

Although Involved as well as embedded object definitions appear in the list, they can only be created from within their parent custom object definition. For more details on creating custom objects, see [Creating Custom Objects](#).

The following table describes the items in the object definition list.

Object Definition List Screen

User interface element	Description
<b>New</b>	Displays a new custom object definition screen with empty fields where you can define a new custom object. For more details on the definition procedure, see <a href="#">Creating Custom Objects</a> .

	<p><b>Note:</b> You cannot use this button to create system objects, nor embedded custom objects. Embedded custom objects, Milestone and Involved objects can be created only from within their parent object definitions.</p>
<b>Refresh</b>	Click to refresh the list after adding object definitions or deleting existing ones.
<b>Object Definition</b>	<ul style="list-style-type: none"> <li>Displays a list of all available object definitions, including MilestoneInvolved, and embedded object definitions, in a hierarchical tree structure. The display structure and functionality is similar to that of the Windows® Explorer® where you can collapse and expand parent nodes that have sub-nodes within them.</li> </ul> <p><i>Note: The hierarchical relationships are reflected only for custom object definitions.</i></p> <ul style="list-style-type: none"> <li>Click hyperlinks to access object definition screens where you can view and modify the necessary properties through the respective tabs. For more details on the tabs and associated properties, see <a href="#">the System and Custom Object Definition Screen Tabs table</a>.</li> <li>The types of the displayed object definitions are indicated by the following icons: <ul style="list-style-type: none"> <li>—System object definition</li> <li>—Custom object definition</li> <li>—Embedded custom object definition</li> </ul> </li> <li>Click the column heading to sort the list alphabetically, according to the first, top-level nodes in the tree structure.</li> </ul>
<b>Unique Code</b>	Displays the unique codes of the object definitions. Click the column heading to sort the list alphabetically.
<b>Contact-Centric</b>	<p>Indicates which custom object definitions are contact-centric:</p> <ul style="list-style-type: none"> <li>If a custom object definition is contact-centric, the role of the main contact appears.</li> <li>If not, the column remains empty.</li> </ul> <p>This column is not sortable.</p>

### 3.1.4.3 Configuring System Objects

By default, some properties in system objects are already defined. For instance, most system objects are named and have unique codes. For the most part, defining these objects is a matter of configuring general system object information and defining categories, search views, and rules. For details, see [Customization Sequence](#) and [General System Object Information](#).

If your design includes invoices, you may need to configure line item views according to the organization's requirements. For details, see [Customizing Invoice Line Item Views](#).

## Defining Contact Group, User Account, and Group Account Objects

There are three TeamConnect system objects, which unlike other system objects, have a simplified object definition procedure. They are Contact Group (called Address Book in the end-user interface), User, and Group objects. Because of their functionality, they do not require categories, custom fields, rules, or wizards. This functionality is reflected in the set of tabs they have in their object definition screens.

**General | Search Views |  
Contact Group  
(Address Book), Group,  
and User System Object  
Definition Tabs**

### To define the simplified object definition objects

1. Select an **Icon Image** file on the **General** tab of the selected object. See [the General Tab on Custom Object Definition Screens table](#).
2. Customize the default **Search Views** provided with the TeamConnect application or create your own. For details, see [Creating Search Views](#).

The Contact Group (Address Book), User, or Group object is defined.

#### 3.1.4.3.1 General System Object Information

For system objects, the **General** tab of the object definition screen includes identification information, such as object name in the singular and plural forms, unique code, and an icon image file. Depending on the system object, some of these details may not be modified.



**Object Definition Information**

[Claim Involved Party](#) [Claim Milestone](#)

\* Name:

\* Name In Plural:

Icon Image File:

Unique Code:

Order:

Default Selection Mechanism:

Parent Custom Object:

Parent Custom Object Required

Do not show this child object definition in the Main menu

Contact-centric

Role:

\* Required fields are indicated by an asterisk.

**General Tab on Custom Object Definition Screens**













See [the General Tab on Custom Object Definition Screens table](#) for details about the **General** tab of object definition screens. In addition to the General page fields in the table, the **Expense**, **Invoice**, and **Task** system objects include the following check-box:

**Allow edits to custom fields after posting**—Place a checkmark in this check-box for the **Expense**, **Invoice**, or **Task** object to allow users to edit custom fields after posting a record for that object.

See [Customization Sequence](#) for customizing a system object.

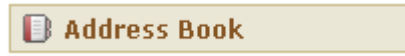
3.1.4.3.2 Default Object Icons

By default, each record in the TeamConnect end-user interface appears with a certain icon that identifies the object to which it belongs. The images of these default icons are automatically generated from the respective Java class files and are assigned to each object. The following are the default images that are used for each object:

<b>Account</b> 	<b>Contact</b> 	<b>Invoice/Line Item</b> 	<b>User</b> 	<b>Involved</b> 
<b>Appointment</b> 	<b>Document</b> 	<b>Expense</b> 	<b>User Group</b> 	<b>Milestone</b> 
<b>Approval</b> 	<b>History</b> No default icon	<b>Task</b> 	<b>Project (Custom Object)</b>	<b>Contact Group (Address Book)</b>



These icons appear at the top of the record window, for example:



If you want to use your own images, upload the appropriate image files to the [icons folder](#) located in the **Top Level** directory in the Documents area.

#### 3.1.4.4 Creating Custom Objects

The process of defining custom objects starts with creating a custom object, defining its general properties, and saving the object. Note that this process may be a complex and lengthy procedure.

**Note:** Before users can view custom objects, you must [assign the corresponding object rights](#) to their group. Each Custom Object has its own set of rights.

There are two approaches to creating a custom object: Copy an existing object definition, or create an object definition from scratch.

If the object to be created will have a definition similar to an existing object definition, you can copy the definition of the existing object, then modify the copy to reflect the exact properties you want in the new object definition. This may save time compared to creating the entire new object definition from scratch.

##### To create a custom object by copying an existing definition

1. In the Designer, go to **Object Definitions**.
2. Click on the existing object definition name.
3. In the resulting page, click **Create a Copy**.
4. A new page appears, showing general information for the new definition. You should immediately change the following field values:
  - Name
  - Name in Plural
  - Unique Code (the present value is the same as the one from the existing definition, which would cause an error if not changed)
5. Enter other appropriate general information in the fields as described in [the General Tab on Custom Object Definition Screens table](#).
6. **Save** the object definition.

##### To create a custom object from scratch

1. To create a new custom object definition, select one of the following options:

- Click **New** in the **Object Definition List** screen.
  - In the Designer window, click **Create a new**, and then select **Object Definition**.
  - A new object definition screen appears with a blank **General** tab displayed by default.
2. Enter the appropriate general information in the fields as described in [the General Tab on Custom Object Definition Screens table](#).
  3. Save the object definition.

## Continue defining object properties

Regardless of which approach you used to create the custom object, you should now continue defining its properties as explained in [Customization Sequence](#).

Once the object properties match your solution design, the object is defined and displayed on the **Object Definition List** screen. For details, see [Viewing Object Definitions](#).

## Resulting User Interface and Other Changes

When you save the custom object information, the following events take place:

- You see the **Create Involved** and **Create Milestone** buttons at the top of the **General** tab.

The screenshot shows the 'Object Definition Information' form. At the top, there are two buttons: 'create Involved' and 'create Milestone'. Below these are several input fields and dropdown menus:

- \* Name: Loan
- \* Name In Plural: Loans
- Icon Image File: (Select) [dropdown]
- Unique Code: LOAN
- Order: 0
- Default Selection Mechanism: Search Module [dropdown]
- Parent Custom Object: (None Selected) [dropdown]
- Parent Custom Object Required
- Contact-centric
- Role: Applicant

**Create Involved and Create Milestone Buttons**

- Two [document folders](#) with the object's name are automatically created in the Documents area and added to the following directories:
  - **Top Level/System/Object Definitions/Object Name**  
This folder stores all documents related to the design of the object definition, such as XML files for custom blocks, Involved and Milestone object definition files, document templates, and rules.
  - **Top Level/Attachments/Object Name**  
This folder stores all of the files that are uploaded to the object records.


**Important:** For each object record, this folder is created only when the record is saved and end users access the **Documents** block of that record for the first time.

For more details on document folders, see the *Administration Guide*.

For a complete overall procedure of defining custom objects, see [Creating Custom Objects](#).

#### 3.1.4.4.1 General Custom Object Information

The **General** tab of a custom object definition screen displays general information about the object as described in [the General Tab on Custom Object Definition Screens table](#).



**General Tab on Custom Object Definition Screens**

To add and save a custom object you must define its name in the singular and plural forms and specify its unique code. You may also specify the rest of the general information according to your design. For instructions on how to define a custom object, see [Customization Sequence](#).

The following table describes the items on the **General** tab of custom object definitions.

**General Tab on Custom Object Definition Screens**

Item	Description
<p><b>customObject Involved Party</b></p> <p><b>Milestone</b></p>	<p>These hyperlinks provide access to the custom object's Involved and Milestone object definitions. They are only displayed once you define these objects.</p> <p>When you add a custom object, you cannot create Involved or Milestone objects for it until you save the custom object definition. When you save the custom object definition for the first time, you see the <b>Create Involved</b> and <b>Create Milestone</b> buttons (<a href="#">Create Involved and Create Milestone Buttons image</a>).</p>

<b>Name</b>	<p>Type a name for the custom object. The maximum length is 50 alphanumeric characters.</p> <p><b>Important:</b> <i>The name must not start with a number and cannot contain special characters.</i></p> <p>This name appears in the end-user interface and names the corresponding Document area folder created for the object definition (see <a href="#">Resulting User Interface and Other Changes</a>).</p> <p>Any special characters entered in this field are automatically converted to hyphens in Document area folder names. For example, <b>Matter:Issue</b> appears as <b>Matter-Issue</b>.</p>
<b>Name in Plural</b>	<p>Type the plural form of the custom object name. The maximum length is 50 alphanumeric characters. The plural name appears in the end-user interface where appropriate.</p>
<b>Icon Image File</b>	<p>Select the image file to appear in the following locations when users access records of this object:</p> <ul style="list-style-type: none"> <li>• Top of the record screen, next to the record name</li> <li>• On the object's task bar buttons</li> <li>• In the object's search screens</li> </ul> <p>You must upload all custom image files to the <a href="#">System&gt;Icons folder</a> at the <b>Top Level</b> level in the <b>Documents</b> area. Otherwise, the default image is used. All default images are generated from the class files and are not located in the <b>Icons</b> folder. For more details, see <a href="#">Default Object Icons</a>.</p>
<b>Unique Code</b>	<p>Enter a four-character alphanumeric combination to uniquely identify this custom object. You can type the unique code in upper or lowercase. However, it is saved and displayed in uppercase. When you save the custom object, the unique code is always displayed as read-only and you cannot change it.</p> <p>The code must be unique for all objects, including Involved and Milestone objects, which are created from within custom objects.</p> <p>The unique code is used as the tree position for the root category of the custom object in the custom block XML files, XML layer, API rules, and Document Generator.</p>
<b>Order</b>	<p>Type an integer to specify where you want the name of this custom object to appear in ordered lists in the end-user interface.</p>

	Each level in the parent-child hierarchies is sorted separately according to the order specified. The objects with the same display order are sorted alphabetically.
<b>Default Selection Mechanism</b>	<p>Specify how the end user will select the object's records when they are referenced in other object records. For example, in the <b>Parent</b> field on the <b>General</b> tab of child projects, or in custom fields of type Custom Object.</p> <p>The available options include:</p> <ul style="list-style-type: none"> <li>• <b>Drop-down list</b>—Recommended for listing a small number of records.</li> <li>• <b>Search Module</b>—Recommended for a large number of records, as it may take a long time to load the list.</li> </ul>
<b>Parent Custom Object</b>	<p>Select a custom object that must be set as a parent for the object you are defining.</p> <p><i><b>Important:</b> This is not a mandatory selection, as it depends on your business model.</i></p> <p>For details, see <a href="#">Parent-Child Relations between Custom Objects</a>.</p>
<b>Parent Custom Object Required</b>	<p>Select this check-box if having a parent custom object is mandatory for the custom object you are creating.</p> <p>See also <a href="#">Parent-Child Relations between Custom Objects</a>.</p>
<b>Do not show this child object definition in the Main menu</b>	<p>This field is available only for custom object definitions that require a parent.</p> <p>Select this check-box if you do not want users to create and access the records of this object from the drop-down lists in the end-user interface, or to avoid unnecessary clutter in the menus.</p> <p>This option does not affect user group rights to create or read these records. They can still do so from the respective parent records.</p>
<b>Do not show New button in related object block</b>	<p>Select this check-box if you want to hide the <b>New</b> button that provides a drop-down list of wizards in the related object block.</p> <p>For more information about the drop-down lists of wizards, see <a href="#">Testing and Troubleshooting Wizards</a>.</p>
<b>Contact-centric</b>	Select this check-box to make the object contact-centric. When you select this check-box, the Role field appears.

	<p><b>Note:</b> This setting is optional and should be based on your organization's needs.</p> <p>For details, see <a href="#">Contact-Centric Custom Objects</a>.</p>
<b>Role</b>	<p>Enter the role of the contact on which the custom object is dependent. This becomes a field label on the General tab of all object records in the end-user interface. For details see <a href="#">Contact-Centric Custom Objects</a>.</p> <p>This field appears only if the Contact-centric check-box is selected.</p>
<b>Single Object Instance Only</b>	<p>Select this check-box to specify that the custom object can have only one record in the end-user interface.</p> <p>When the user clicks the link for the custom object, TeamConnect automatically opens the only record. If the record does not exist, the user must create a record first.</p>

#### 3.1.4.4.2 Creating Custom Objects

The Milestone and Involved custom objects are system objects that can be created and accessed only from within Custom Objects in Designer. For more information, see [Unique Custom Object Types](#).

Whenever you add a new custom object, you see the **Create Involved** and **Create Milestone** buttons at the top of the **General** tab.

**Create Involved and Create Milestone Buttons**

#### To create a Milestone or Involved object

1. Open the **General** tab of the custom object definition where you want to create the appropriate object.
2. Click either the **Create Involved** or **Create Milestone** button.

The corresponding object definition screen of the selected object appears with the **General** tab displayed by default. The **Name** and **Name in Plural** fields are automatically populated:

**General Tab on Involved Party  
Object Definition Screen**

3. (Optional) Type the desired name in the singular and plural for the object.
4. (Optional) From the **Icon Image File** drop-down list, select the image file that appear when the user accesses the object records. Otherwise, the default image is used instead.  
  
All image files are located in the **Icons** folder at the **Top Level** level in the **Documents** area.
5. In the **Unique Code** field, enter a four-character alphanumeric combination to uniquely identify this object.  
  
This code is unique for each of these objects, regardless of the custom object it belongs to.
6. Save the object definition and continue defining its properties as described in [Configuring System Objects](#).
7. When finished defining the object's properties, save the definition and click the **Refresh** button at the top of the **Object Definition** page.

The Milestone or Involved object is defined.

## Resulting user interface and Other Changes

When you save Involved or Milestone object information, the following actions take place for each object:

- The object's name appears as a hyperlink at the top of the **General** tab of the respective custom object.
- Two [document folders](#) with the object's name are automatically created in the Documents area and added to the following directories:
  - **Root/System/Object Definitions/*customObjectName*/*milestoneName***
  - or
  - **Top Level/System/Object Definitions/*customObjectName*/*involvedName***



This is the folder where all documents related to the design of the object definition are stored, for example, XML files for custom blocks, appropriate object definition files, document templates, and rules.

- **Root/Attachment/customObjectName/recordUniqueID/MilestoneList**

or

**Top Level/Attachment/customObjectName/recordUniqueID/InvolvedList**

This is the folder where all files uploaded to the object records are stored.

**Important:** For each Involved and Milestone record, this folder is created only when the record is saved and end users access the **Documents** block of that record for the first time.

Milestone and Involved objects have the same set of object definition tabs as system objects and their object definitions can be customized like those of system objects. See [Configuring System Objects](#).

General | Categories | Custom Fields | Forms | Blocks | Object Views | Search Views |  
Custom Messages | Rules | Templates | Wizards | Notifications | History |

#### System Object Definition Tabs

The roles of the involved parties are defined through the **Categories** tab of the Involved object definition screen. See [Using Categories](#).

Group rights to the Milestone and Involved objects are identical to those of their parent object. For example, if a group lacks access rights to the parent object, it cannot access that object's Milestone and Involved records. Likewise, search results for such records are not displayed.

#### 3.1.4.4.3 Unique Identifiers for Custom Object Records

*Unique identifiers* are unique combinations of alphanumeric characters assigned to each custom object record when it is created. Users can view unique IDs in the end-user interface if the identifier is set to be displayed.

You can assign unique identifiers for custom object records:

- By using auto numbering. See [Creating Auto Numbering Patterns](#).
- By using object attributes. See [Creating Unique Identifier Patterns from Object Attributes](#).
- By using default numbering

You can select the desired options for the project identifiers through the **Unique ID** tab of the respective custom object definition.

**Unique ID**

---

How would you like to make individual Loans unique?

By using automatic numbering

By using a combination of the object attributes  
(Unique ID won't be displayed)

By using manual numbering

By using default numbering (a default unique ID will be generated but will be hidden from the user)

**Unique Identifier Options**

The following table explains each available option in detail.

**Unique Identifier Options**

Option	Description
<b>By using automatic numbering</b>	<p>Select this option to have TeamConnect automatically generate numbers for object records according to the specified pattern.</p> <p>When a new record is being created, the <b>Number</b> field will have the word <b>Auto</b> displayed in parentheses. When the record is saved, the <b>Number</b> field displays the assigned sequential number. See <a href="#">Creating Auto Numbering Patterns</a>.</p>
<b>By using a combination of the object attributes</b>	<p>Select this option to create a pattern consisting of the object attributes that, taken together, will uniquely identify the individual object records of the selected custom object.</p> <p>No numbering will be displayed in the end user interface.</p> <p>See <a href="#">Creating Unique Identifier Patterns from Object Attributes</a>.</p>
<b>By using manual numbering</b>	<p>Select this option to allow users to enter their record numbers manually.</p> <p>When a new record is being created, the <b>Number</b> field will be displayed as an empty text field where the user can enter the desired number.</p> <p><i>Tip: Remember to select the desired display format for the project hyperlink. See <a href="#">Project Link Display Format</a>.</i></p>
<b>By using default numbering</b>	<p>Select this option to have primary keys automatically assigned to each project and stored in the database.</p> <p>No numbering will be displayed in the end-user interface.</p>

Depending on the option you choose at the top of the **Unique ID** tab, the bottom part of this screen may display an additional section where you can specify the desired properties.

### 3.1.4.4.3.1 Creating Auto Numbering Patterns

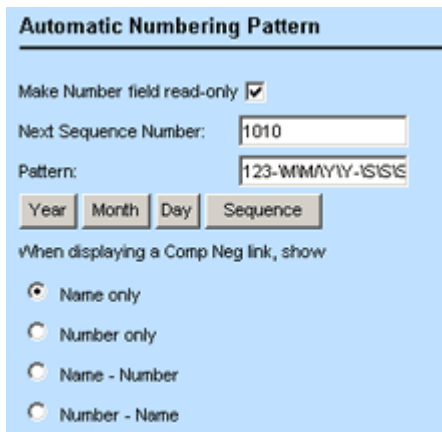
You can specify an automatic numbering pattern for generating individual record IDs for custom objects. A pattern can consist of a combination of the following elements:

- The date when the project is created
- A sequential number, which differentiates a particular project from the other projects created on the same day
- Any static information, such as a department name or special characters used as format separators

#### To set an auto numbering pattern for projects

1. Open the **Unique ID** tab of the appropriate custom object definition.
2. Select the **By using automatic numbering** option.

The **Auto Numbering Pattern** section appears at the bottom of the screen.



The screenshot shows a form titled "Automatic Numbering Pattern". It includes a checkbox for "Make Number field read-only" which is checked. Below this are two input fields: "Next Sequence Number" with the value "1010" and "Pattern" with the value "123-WMMYY-ISIS". There are four buttons labeled "Year", "Month", "Day", and "Sequence". Below these buttons is a section titled "When displaying a Comp Neg link, show" with four radio button options: "Name only" (selected), "Number only", "Name - Number", and "Number - Name".

**Auto Numbering Pattern Settings**

3. Fill in the appropriate fields as described in [the Auto Numbering Pattern Settings table](#).
4. Select the format for displaying the project hyperlink.
5. Click **Save**.

TeamConnect uses the new auto numbering pattern to assign a number to each new custom object record.

If you change the auto numbering pattern of an object definition, any existing records keep the previous unique identifier conventions until users update the records, at which point the new auto numbering pattern is used.

Updating the auto numbering pattern does not count as updating the object definition, so record locking is not enforced. However, updating other fields of the object definition causes an optimistic locking error.

The following table describes the auto numbering pattern settings.

#### Auto Numbering Pattern Settings

Field or control	Description
<b>Make Number field read-only</b>	<p>Select this check-box to make the project number uneditable.</p> <p><i><b>Important:</b> If the number is editable, the user will have the ability to modify the number. These manual changes will take precedence over the automatic pattern.</i></p>
<b>Next Sequence Number</b>	<p>Enter a starting number for the sequential part of the auto numbering pattern. By default it is set to 0 or, if auto numbering has already been enabled, the next number in the sequence.</p>
<b>Pattern</b>	<p>Enter a pattern for the number, using any of the following elements:</p> <ul style="list-style-type: none"> <li>• Alphanumeric characters</li> <li>• Special characters, such as \$, &amp;, *, ~.</li> <li>• Spaces</li> <li>• Dates</li> <li>• Sequence characters (see the buttons below)</li> </ul>
<b>Year</b>	<p>With the cursor in the <b>Pattern</b> field, click this button to add a two- or four-digit year to the <b>Pattern</b>, for example, \Y\Y.</p>
<b>Month</b>	<p>With the cursor in the <b>Pattern</b> field, click this button to add a two-digit month to the <b>Pattern</b>, for example, \M\M.</p>
<b>Day</b>	<p>With the cursor in the <b>Pattern</b> field, click this button to add a two-digit day to the <b>Pattern</b>, for example, \D\D.</p>
<b>Sequence</b>	<p>With the cursor in the <b>Pattern</b> field, click this button to add a multi-digit sequential number to the <b>Pattern</b>, for example, \S\S\S\S\S\S.</p> <p>It is a good idea to base the number of digits that you include in the sequence on the number of records that are created in a day. For example, if your organization creates 100 claims per day, you might want to consider making the sequence consisting of three digits (\S\S\S).</p> <p><i><b>Important:</b> You must include the sequence part into the <b>Pattern</b>, to make sure that the project number is unique.</i></p>
<b>When displaying a</b>	<p>Select the appropriate choice button to specify which format to use when displaying record hyperlinks of the object in other object records. See <a href="#">Project Link Display Format</a>.</p>

<b>CustomObject link, show</b>	
------------------------------------	--

#### 3.1.4.4.3.2 Project Link Display Format

Whenever a custom object record is referred to in other records or appears, for example, in search screens, it appears as a hyperlink. This hyperlink appears in one of the four available formats:

##### Project Link Display Options

Option	Example
<b>Name Only (Default)</b>	Johnson v. Matthews
<b>Number Only</b>	2001-1018-003
<b>Name - Number</b>	Johnson vs. Matthews-2001-1018-003
<b>Number - Name</b>	2001-1018-Johnson vs. Matthews

You may select the desired format on the **Unique ID** tab of the custom object for the automatic and manual numbering options. Existing records keep the previous settings only until a user makes changes or updates them. When a user updates an existing record, the new setting you specified takes effect.

#### 3.1.4.4.3.3 Creating Unique Identifier Patterns from Object Attributes

You can specify a sequence of custom object attributes to uniquely identify individual custom object records. This can be useful to ensure that no duplicate records are created. For example, if you want to prevent duplicate matter records from being created for the same person involved in a dispute, you can create a unique identifier pattern that consists of the person's name, the type of dispute, and an incident date. In this way, you ensure that no two matter records with the same details are created for the same dispute.

For more information on attributes, see [Using Object Navigator](#) and [Object Model: Read This First](#) plus the additional reference tables this reference points you to.

##### To set a unique identifier pattern for projects

1. Open the **Unique ID** tab of the appropriate custom object definition.
2. Select the [By using a combination of the object attributes](#) option.

The **Object Attributes** section appears.

Object Attributes in Unique Identifiers

3. Select the **Number of entries you would like to add** from the drop-down list.
4. From each drop-down list, select the desired object attribute.
5. If you want to add more attributes, click **add more**. Otherwise, click **Save**.

The object attribute pattern you defined for the custom object takes effect immediately. Existing records keep the previous unique identifier conventions until a user updates them.

## Points To Remember

Keep the following points in mind when creating unique identifier patterns from object attributes:

- Object attribute patterns are stored in the database but they are not displayed in the end user interface.
- All object attributes you select for a pattern are added alphabetically one after another, no formatting is used.
- The attributes that you can select for the unique identifier patterns are automatically preselected by the system for you. They often include a few system fields, such as **contact** (applicable for contact-centric object definitions), **name**, **nameUpper**, **defaultCategory**, **numberString**, **numberStringUpper**, and all custom fields under the Root category.
- Even though taken in isolation certain attributes may never constitute a unique pattern (for example, **name**), taken together, they do.
- All object attributes you select for a pattern automatically become required fields for the end user.
- If you add custom fields to the pattern, remember to always include those fields in the custom pages you create.

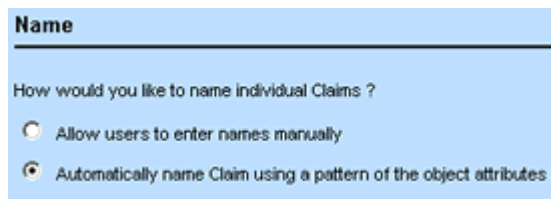
### 3.1.4.4.4 Naming Patterns for Custom Object Records

Unlike unique identifiers, such as record numbers and object attribute patterns, that serve to uniquely identify each record in the database, names in custom object records are intended to provide a description of each record that may or may not be unique.

Depending on your organization's naming conventions, you can assign names to custom object records manually or automatically. Using an automatic naming pattern for project record names is useful when:

- Your organization has project naming conventions. For example, each dispute record must be named after the contact of the person or company that your organization is representing, the contact of the opposing party, and the type of dispute.
- Naming patterns can be identified in your organization's project names, even if there are no specific naming requirements or conventions.
- Spelling, or other errors and inconsistencies in record names may severely affect the business process.

You can set the record naming options for a custom object through the **Name** tab of its object definition.



**Project Naming Options on Name Tab**

The following table describes the naming options.

#### Project Naming Options

Option	Description
<b>Allow users to enter names manually</b>	Select this option to have the user manually enter names for project records.  When a new object record is being created, an editable <b>Name</b> field will be displayed for the user.
<b>Automatically name customObject using a pattern of the object attributes</b>	Select this option to define a naming pattern that will be automatically assigned to all custom object records.  The name is automatically displayed as a read-only field after the project is saved.

With the auto naming feature for custom object records, you can construct the desired pattern according to the standards and conventions accepted in your organization.

### Example

Consider an automatic naming pattern for a custom object called Loan where each record must be named after the contact who applied for the loan, the type of loan applied for, and the amount of money requested. The format for displaying this information would be: Applicant's LoanType - LoanAmount.

To create a pattern like this, do the following actions:

- The name of the applicant from one of the following elements:
  - A custom field of type Involved that contains the applicant's name
  - The contact attribute that contains the main contact's (applicant's) name, which is available only in contact-centric custom objects

By default, all persons' names appear in the following format: Last Name, First Name.

- The default category of the custom object that contains the loan type
- The custom field that contains the loan amount information
- The following **literals**:
  - Apostrophe "s"
  - Hyphen
  - Three spaces between the attributes and literals

***Tip:** Whenever you have spaces around literals, add these spaces together with their literals, for example, a space after the 's, and two spaces before and after the hyphen. They are added even if you do not see them in the item list.*

When you add the information to the name pattern of the custom object, TeamConnect automatically creates the name for each record when the user saves it. This way, all records are named correctly and consistently with your organization standards, based on the attributes you select.

## Points To Remember

Consider the following points when creating naming patterns for objects:

- The naming pattern of a record will reflect only those attributes that have values. If an attribute in a naming pattern has no value, it will not be displayed in the record name.

***Tip:** To ensure that all record names are properly displayed, you can make the attributes required. If you specify any custom fields, you must include them in your custom pages.*

- Naming patterns must contain at least one attribute. They cannot consist only of literals.
- Project names support a maximum of 250 characters. If the data stored in the selected attributes and literals exceed 250 characters, the name of the project is truncated.

***Tip:** If you have added long strings to the naming pattern, consider abbreviating them.*

- To have a person's name displayed in the naming pattern, specify a custom field of type Involved, or the contact attribute for contact-centric objects.
- By default, all person contacts' names will be displayed in the format Last Name, First Name. If you do not want to display the comma between the names or you want to change their order, you can use two attributes.

For example, instead of selecting the **contact** attribute for a contact-centric object, you have to add it twice and traverse further to select an attribute for each of the names:



- `contact.firstName`
- `contact.name`

You may define naming patterns by selecting the **Automatically name custom object using a pattern of the object attributes** option button on the **Name** tab of the custom object definition. For instructions, see [Creating Auto Naming Patterns](#).

#### 3.1.4.4.4.1 Adding Object Attributes

Attributes are the building blocks of objects and contain the actual data you enter into TeamConnect. For example, you may use attributes in naming patterns for custom object records to ensure that certain values are consistently used.

Naming patterns can include such values as:

- Name of an involved party
- Specific dates
- Categories
- Names of other objects, such as parent objects or other custom objects

You can select the attributes you want to use in your naming patterns from the **Object Attributes** section of the **Name** tab of the custom object definition. Before you begin defining naming patterns for custom objects, you must have an understanding of how to use Object Navigator.

For more information, see [Using Object Navigator](#) and [Object Model: Read This First](#) plus the additional reference tables this reference points you to.

Item	Format	Order
1 Item type: <input type="text" value="Attribute"/> <input type="text"/> <input type="button" value="reset"/> N/A		<input type="text"/>
1 <input type="checkbox"/> .contact		1
2 <input type="checkbox"/> defaultCategory		3
3 <input type="checkbox"/> detailList(Loan).detailNumValueList(LoanAmount).detailValue	##0###(2)	5


[Check All](#) - [Uncheck All](#)

Object Attributes on Name Tab

The following table describes the items in the **Object Attributes** section of the **Name** tab.

#### Object Attributes on Name Tab

Field	Description
Item type	Provides the following options: <ul style="list-style-type: none"> <li>• Attribute</li> <li>• Literal</li> </ul>

	<ul style="list-style-type: none"> <li>• Character space</li> </ul> <p>Select <b>Attribute</b> to define attributes for the naming pattern.</p>
<b>Object Navigator</b>	<p>Click the <b>Object Navigator</b>  icon. In the pop-up window, navigate to the desired attribute that contains the value you need for the pattern and click <b>ok</b>. The path to the selected attribute appears in the <b>Item</b> field.</p> <p>See <a href="#">Using Object Navigator</a>.</p>
<b>Reset</b>	<p>Click to clear the contents of the <b>Item</b> field.</p>
<b>Format</b>	<p>Select the appropriate format for the selected attribute.</p> <p>For different attribute types, different <b>Format</b> data entry fields appear. For more information, see <a href="#">Attribute Format Fields</a>.</p>
<b>Order</b>	<p>Enter an integer to indicate the <b>Order</b> of the attributes selected.</p> <p>The integer reflects exactly where in the naming pattern you want the attribute to be displayed (beginning, end, or middle of the name).</p> <p>Items with the same display order are sorted and displayed alphabetically according to the first letter of the attribute selected from Object Navigator (for example, "c" in contact, not "S" in Smith).</p>
<b>Item List</b>	<p>Displays the object attributes (and literals) that are already selected to be used in the project name pattern.</p> <p>Select the check-box of the desired attribute and click either the <b>edit</b> or <b>delete</b> buttons.</p>

#### 3.1.4.4.2 Attribute Format Fields

Depending on the type of attribute you select using Object Navigator, you can apply various display formatting options. For example, if you want the value of an attribute of type String to be displayed in upper or lowercase in each object record name, you select the appropriate format.

You can specify the following attribute format types in naming patterns:

- Boolean
- Date
- Number
- String

The following table describes the formatting options. For each of the object attribute types, two **Format** data entry fields appear.

**Note:** If you decide to select no formatting for the attributes, the default settings are used instead as indicated in the table.

#### Format Fields for Object Attributes on Name Tab

Attribute type	Format option	Description
<b>Boolean</b>	<b>True value</b>	Enter true or false values in the appropriate data entry fields for <b>Boolean</b> values, which are most commonly displayed as check-boxes or choice buttons.
	<b>False value</b>	For example, if "isRegistered" custom field is the selected <b>Boolean</b> attribute, you may want to enter "Yes" for the true value and "No" for the false value.  <b>Default Value:</b> False.
<b>Date</b>	<b>Pattern</b>	Select month, day, and year format for the date attribute.  <b>Default Format:</b> The settings selected on the <b>Preferences</b> tab of the <b>System Settings</b> screen.
	<b>Separator</b>	Select the symbol you would like to act as a separator between the month, day, and year of the date attribute.  <b>Default Format:</b> The settings selected on the <b>Preferences</b> tab of the <b>System Settings</b> screen.
<b>Number</b>	<b>Pattern</b>	Select the format for the number attribute.  The display of the numbers is based on the locale. For example, in Europe the number 1.234,56 has a period (.) as a grouping symbol and a comma (,) as the decimal point. In the United States, the number 1,234.56 has a comma (,) as a grouping symbol and a period (.) as a decimal point.  <b>Default Format:</b> The US standard, where the comma is a grouping symbol.
	<b>Decimal Places</b>	Select the number of digits you would like to appear after the decimal point of the number attribute.  <b>Default Format:</b> The settings at the time when the attribute value was saved in the database.
<b>String</b>	<b>Case</b>	Select either <b>UPPERCASE</b> or <b>lowercase</b> to format the string attribute accordingly.  <b>Default Format:</b> The settings at the time when the attribute value was saved in the database.

	<b>Abbr</b>	<p>If you want to abbreviate a string, select the desired abbreviation option. Whichever option you select, a blank data entry field appears for you to enter an integer. Depending on the option, the integer you enter specifies the following information:</p> <ul style="list-style-type: none"> <li>• When <b>First X Characters</b> is selected—the integer specifies the first number of characters from the beginning of the string.</li> </ul> <p>For example, if the string attribute consists of a contact name, such as Smith, John, and integer is <b>3</b>, the string will be formatted as SMI.</p> <ul style="list-style-type: none"> <li>• When <b>Acronym</b> is selected—the integer you enter specifies the first number of characters from each word in the string.</li> </ul> <p>For example, if the string attribute consists of a contact name, such as Smith, John, and integer is <b>3</b>, the string will be formatted as SMIJOH.</p> <p><b>Note:</b> <i>If an integer greater than the number of characters in the string is entered, all of the characters will be included in the abbreviation.</i></p>
--	-------------	--

#### 3.1.4.4.3 Adding Literals

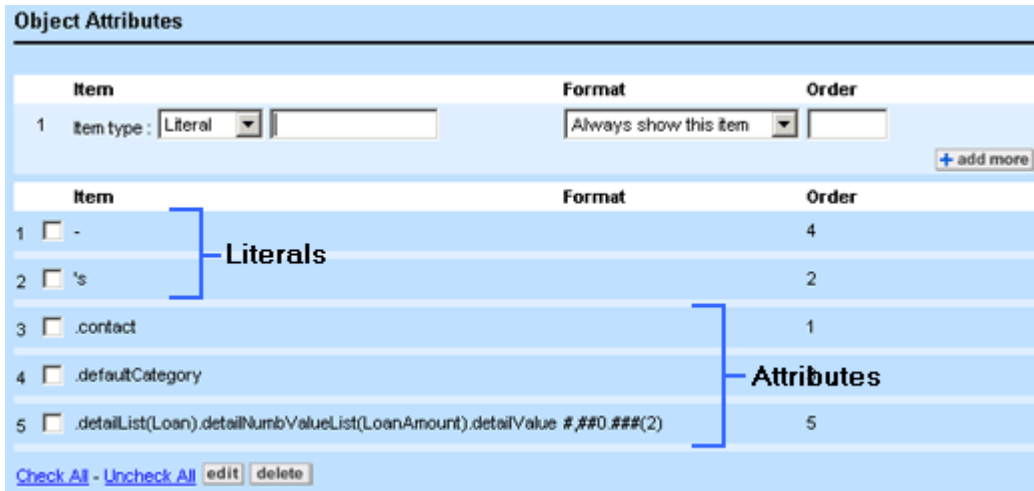
A literal is an alphanumeric or special character that is used as a separator between two attributes, or as a prefix or suffix to an attribute.

For example, if a user selected **Smith, John** as the applicant in a loan record, set **Loan** as the default category, and entered 20,000 as the loan amount, and the three attributes containing these values are added to the Loan naming pattern, the name of the project would be saved as Smith, JohnLoan20,000. Adding literals to the three attributes would create a clearer name for your users.

For example, you can add an apostrophe "s" with a space after it ('s ) as a literal after the contact attribute and a hyphen with two spaces around it. After adding these literals, the name of the project would be saved as Smith, John's Loan - 20,000. Otherwise, you can add a space as a separate character.

**Important:** *A name cannot consist of literals only, it must have attributes.*

You can select the literals you want to use in your naming patterns from the **Object Attributes** section of the **Name** tab of the custom object definition.



Literals and Object Attributes on Name Tab

The following table describes the items in the **Object Attributes** section of the **Name** tab that you use to specify literals.

Literals and Object Attributes on Name Tab

Field	Description
<b>Item type</b>	Select <b>Literal</b> to add a literal to the record naming pattern.
<b>Literal text field</b>	Enter the appropriate characters to create either a separator between two attributes, or a prefix or suffix for an attribute.  Tip: To add spaces between literals and attributes, type the space together with the literal, for example, a space before and after a slash. To add a space between two attributes, you must add it as a separate literal called <b>space character</b> .
<b>Format</b>	Select the format for the literal by selecting one of the following options: <ul style="list-style-type: none"> <li>• <b>Always show this item</b>—The literal always displays as part of the name when the user saves the record. For example, if creating a literal "record," you may want the word "record" to always appear in the name.</li> <li>• <b>Don't show if the previous item is null</b>—The literal will not display if the object attribute before it has no value or is non-existent. For example, if creating a literal "'s," you may not want the "'s" to appear if there is no name in front of it.</li> </ul>
<b>Order</b>	Enter an integer to indicate the display <b>Order</b> of the literals.  The integer you enter reflects exactly where in the naming pattern you want the literal to be displayed (at the beginning or end of an attribute, or between two attributes).  Items with the same display order are sorted and displayed alphabetically.

<b>Item List</b>	Displays the object attributes and literals that are used in name pattern for projects.  Select the check-box of the desired attribute or literal and click either the <b>edit</b> or <b>delete</b> buttons to make changes.
------------------	--

#### 3.1.4.4.4 Creating Auto Naming Patterns

##### To create an auto naming pattern for a custom object

1. Review the [Points To Remember](#).
2. Open the **Name** tab of the appropriate custom object definition where you want to create a record naming pattern.
3. Select the **Automatically name custom object using a pattern of the object attributes** option.

The Object Attributes section appears at the bottom of the screen.

4. Depending on your needs, do of the following actions:
  - Specify the object **Attributes** whose values you want to appear in the object records' names, as described in [the Object Attributes on Name Tab table](#).
  - Specify the **Literals** that you want to add to the naming pattern of the custom object, as described in [the Literals and object Attributes on Name Tab table](#).

**Caution:** *If the data stored in the attributes and literals you select have a combined amount of over 250 characters, the name of the project is truncated, cutting off the information that appears after 250 characters.*

5. If you want to continue adding items, click **add more**. Otherwise, click **Save** on the toolbar.

The record naming pattern you defined for the custom object takes effect immediately. Existing records keep the previous naming conventions only until a user makes changes or updates an existing record. When a user updates an existing record, the new naming convention you specified takes effect.

When the naming pattern is set, the **Name** field in all records of the selected custom object appears as read-only.

#### 3.1.4.4.5 Creating Phases in Custom Objects

All custom object records have their own life cycle consisting of a set of phases. The number of phases and their transitions varies depending on the custom object and your organization's business practices.

You may define the phases for a custom object on the **Phases** tab of its object definition and specify the transition from one phase to another on the **Phase Transitions** tab. For information about setting phase transitions, see [Defining Phase Transitions for Custom Objects](#).

You may create and view all project phases for custom objects on the **Phases** tab. To access this tab, select **Object Definitions** from the **Go to** drop-down list, and then choose the appropriate custom object definition.

**Phases**

Number of entries you would like to add:

	Phase Name	Display Order	Unique Code
1	<input type="text" value="Pre-Trial"/>	<input type="text" value="30"/>	<input type="text" value="pret"/>
2	<input type="text"/>	<input type="text"/>	<input type="text"/>

[+ add more](#)

Displaying records 1-10 of 11. [Next](#)

	Phase Name	Display Order	Unique Code
1	<input type="checkbox"/> Appeal	35	APPL
2	<input type="checkbox"/> Arbitration	25	ARBT
3	<input type="checkbox"/> Close	40	CLOS
4	<input type="checkbox"/> Depositions	0	DEPO
5	<input type="checkbox"/> Discovery	10	DISC
6	<input type="checkbox"/> Interrogatories	15	INTR
7	<input type="checkbox"/> Mediation	0	MEDI
8	<input type="checkbox"/> <input type="text" value="Open"/>	<input type="text" value="1"/>	OPEN
9	<input type="checkbox"/> Settlement	20	STTL
10	<input type="checkbox"/> Suit Filed	5	SUIT

[Check All](#) - [Uncheck All](#) [edit](#) [delete](#)

**Phases Tab**

By default the **Phases** tab displays two phases, **Open** and **Close**. They are automatically set as the Initial and Final phases for the project life cycle on the **Phase Transitions** tab. You can always delete these two phases or rename them after you have added other phases.

**Phases Tab**

Field or control	Description
<b>Phase Name</b>	Type the name of the phase you are adding. The maximum length is 250 alphanumeric characters.  This name will be displayed in two places in the end-user interface: the <b>Phases</b> tab and on the <b>General</b> tab (as the current phase) of the record.
<b>Order</b>	Enter an integer to indicate the order in which you want the name of this phase to be displayed in the drop-down lists on the <b>Phase Transitions</b> tab. The phases with the same <b>Order</b> are sorted alphabetically and displayed accordingly.

	<i>Tip: It may be helpful to enter orders in increments of five. If you ever have to add more phases, you will have some leeway as to where in the drop-down list you can place the additional phases.</i>
<b>Unique Code</b>	Type a 4-character alphanumeric combination to uniquely identify this phase.  <i>Note: You can type the unique code in uppercase or lowercase, however in the database, it is saved in uppercase.</i>
<b>Phase List</b>	Displays a list of phases added to the selected custom object. Select or clear the corresponding check-boxes on the left to delete or edit the respective phases.

#### 3.1.4.4.5.1 Adding Phases

Before adding phases, write a complete list of phases you would like to add to a custom object. This helps you ensure that you add all necessary phases and facilitates the process of defining transitions between them.

#### To create a project phase for a custom object

1. Open the **Phases** tab of the appropriate custom object definition.
2. Select the **Number of entries you would like to add** from the drop-down list.
3. For each data entry row, fill in the information as described in [the Phases Tab table](#).
4. If you want to continue adding phases, click **add more**. Otherwise, click **Save** on the toolbar.
5. Define the desired phase transitions. See [Defining Phase Transitions for Custom Objects](#).

#### 3.1.4.4.5.2 Editing Phases

If you need to change either the name of a phase or its order, you can do so by editing the phase. However, if you need to change its tree position, you must delete the phase first.

**Caution:** *If possible, phases should be deleted during the design stage when there are no existing records in the database. Otherwise, loss of data occurs.*

#### To modify a project phase

1. Open the **Phases** tab of the appropriate custom object definition.



2. Select the check-boxes next to the names of the phases that you want to edit.
3. Click **edit**.

The phase details that you can modify become editable.

**Phases**

Number of entries you would like to add:

	Phase Name	Display Order	Unique Code
1	<input type="text" value="Pre-Trial"/>	<input type="text" value="30"/>	<input type="text" value="pret"/>
2	<input type="text"/>	<input type="text"/>	<input type="text"/>

[+ add more](#)

Displaying records 1-10 of 11. [Next](#)

	Phase Name	Display Order	Unique Code
1	<input type="checkbox"/> Appeal	35	APPL
2	<input type="checkbox"/> Arbitration	25	ARBT
3	<input type="checkbox"/> Close	40	CLOS
4	<input type="checkbox"/> Depositions	0	DEPO
5	<input type="checkbox"/> Discovery	10	DISC
6	<input type="checkbox"/> Interrogatories	15	INTR
7	<input type="checkbox"/> Mediation	0	MEDI
8	<input type="checkbox"/> <input type="text" value="Open"/>	<input type="text" value="1"/>	OPEN
9	<input type="checkbox"/> Settlement	20	STTL
10	<input type="checkbox"/> Suit Filed	5	SUIT

[Check All](#) - [Uncheck All](#)

#### Editable Phases

4. Make the necessary changes to the **Phase Name** or **Order** of the selected items.
5. Click **Save**.

The changes you made become effective immediately upon saving.

#### 3.1.4.4.5.3 Deleting Phases

If you need to delete a phase or change its tree position, you can do so by deleting the phase.

**Caution:** *If possible during the design stage, try to delete phases while there are still no existing records in the database. If you delete a phase that is used by records, the phase information is permanently deleted from the records and the phase transition sequence is broken.*

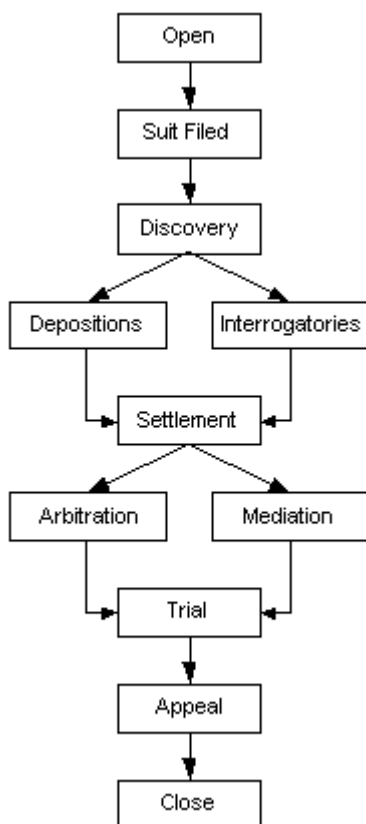
#### To delete a project phase

1. Open the **Phases** tab of the appropriate custom object definition.
2. Select the check-boxes next to the names of the phases that you want to delete.
3. Click **delete**.
4. When asked to confirm your action, click **OK**.

The phase information is deleted from the system.

#### 3.1.4.4.6 Defining Phase Transitions for Custom Objects

After you have created the phases necessary for the project life cycle of a custom object, you must specify the way these phases follow each other. Phases may branch into several alternative phases at the same level. For example, the following flowchart shows how the phases shown in [the Phase Transitions Tab image](#) may follow each other.



**Phase Transition Example**

The process of creating a phase transition consists of:

- Naming the phase transition
- Specifying the phase from which the transition occurs
- Specifying the phase to which the transition occurs

- Determining whether the transition is visible to the end user

In the case of a linear transition, you specify the phases as you want them to follow each other, for example, from Open to Suit Filed and from Suit Filed to Discovery.

Even if you have only two phases, such as **Open** and **Close**, you must create a phase transition. In this case, select **Open** from the **Initial Phase** field and **Close** from the **Final Phase** field on the **Phase Transitions** tab.

When the phases branch, you must create multiple sequences, such as:

- From Discovery to Depositions
- From Discovery to Interrogatories
- From Depositions to Settlement
- From Interrogatories to Settlement

To access the **Phase Transitions** tab of a custom object definition, select **Object Definitions** from the **Go to** drop-down list, and then choose the appropriate custom object.

The following image shows the **Phase Transitions** tab of a custom object definition. Its phase transitions correspond to those shown in the diagram.

**Phase Transitions**

Initial Phase:

Final Phase:

Number of entries you would like to add:

	Name	From Phase	To Phase
1	<input type="text" value="Pre-Trial"/>	<input type="text" value="Mediation"/>	<input type="text" value="(Select)"/>
2	<input type="text"/>	<input type="text" value="(Select)"/>	<input type="text" value="(Select)"/>

[+ add more](#)

Displaying records 1-10 of 12. [Next](#)

	Name	From Phase	To Phase
1	<input type="checkbox"/> Appeal	Trial	Appeal
2	<input type="checkbox"/> Arbitration	Settlement	Arbitration
3	<input type="checkbox"/> Closed	Appeal	Close
4	<input type="checkbox"/> Depositions	Discovery	Depositions
5	<input type="checkbox"/> Discovery	Suit Filed	Discovery
6	<input type="checkbox"/> Interrogatories	Discovery	Interrogatories
7	<input type="checkbox"/> Mediation	Settlement	Mediation
8	<input type="checkbox"/> Settlement	Depositions	Settlement
9	<input type="checkbox"/> Settlement	Interrogatories	Settlement
10	<input type="checkbox"/> <input type="text" value="Suit Filed"/>	Open	Suit Filed

[Check All - Uncheck All](#)

**Phase Transitions Tab**

The following table describes the items on the **Phase Transitions** tab.

## Phase Transitions Tab

Field or control	Description
<b>Initial Phase</b>	<p>Select the phase you want the project life cycle to begin with. The <b>Initial</b> phase is automatically linked to the <b>Opened On</b> date field in the Project's <b>General</b> tab in the end-user interface.</p> <p><i>Note: By default, the <b>Open</b> phase is set as the <b>Initial</b> phase for all projects, and you can change it only when other phases are added on the <b>Phases</b> tab.</i></p>
<b>Final Phase</b>	<p>Select the phase you want the project life cycle to end with. The <b>Final</b> phase is automatically linked to the <b>Closed On</b> date field in the Project's <b>General</b> tab in the end-user interface.</p> <p><i>Note: By default, the <b>Close</b> phase is set as the <b>Final</b> phase for all projects, and you can change it only when other phases are added on the <b>Phases</b> tab.</i></p>
<b>Name</b>	<p>Type the name of the phase transition as you want it to be displayed for the end user in the <b>(actions)</b> drop-down list in the top right-hand corner of the of the Project screen. The maximum length is 250 alphanumeric characters.</p> <p><i>Tip: Type the name of the phase to which you want to go.</i></p>
<b>From Phase</b>	<p>Select the phase from which you want the transition to occur.</p>
<b>To Phase</b>	<p>Select the phase to which you want the transition to occur.</p>
<b>Visible</b>	<p>In this check-box, enter a check to make the transition visible to the end user, or clear the box to make the transition non-visible. Non-visible transitions are useful when the transition is always going to be accomplished through rule actions or other automated means, not through manual selection by an end user. By default, all phase transitions are visible.</p>
<b>Phase Transition List</b>	<p>Displays a list of phase transitions set for the selected custom object. Select or clear the corresponding check-boxes on the left to delete or edit the respective phase transitions.</p>

#### 3.1.4.4.6.1 Creating Phase Transitions

Before creating phase transitions in TeamConnect, draw a diagram outlining the sequence in which you want the phases to follow each other. For an example, see [Defining Phase Transitions for Custom Objects](#).

##### To create a project phase transition

1. Make sure all the necessary phases for the project life cycle are created on the **Phases** tab of the custom object. See [Creating Phases in Custom Objects](#).
2. Open the **Phase Transitions** tab of the appropriate custom object definition.
3. Select the appropriate initial and final phases from the corresponding drop-down lists as described in [the Phase Transitions Tab table](#).
4. Select the **Number of entries you would like to add** from the list.
5. For each data entry row, fill in the information as described in [the Phase Transitions Tab table](#).
6. If you want to continue adding phase transitions, click **add more**. Otherwise, click **Save** on the toolbar.

The newly added phase transitions appear in the list of phase transitions and the available transitions appear in the actions drop-down list of object records.

7. Test the created phase transitions to ensure no infinite loops or broken sequences were created.

If the test is successful, the phase transition was successfully created.

#### 3.1.4.4.6.2 Editing Phase Transitions

You can edit the name of a phase transition, but you cannot change its **From** and **To** phases. However, you can delete the transition and then recreate it.

**Caution:** *If possible, phase transitions should be deleted during the design stage when there are no existing records in the database. Otherwise, data will be lost.*

##### To modify a project phase transition

1. Click the **Phase Transitions** tab of the appropriate custom object definition.
2. Select the check-boxes next to the phase transition that you want to edit.
3. Click **edit**.

The phase details that you can modify become editable.

**Phase Transitions**

Initial Phase:

Final Phase:

Number of entries you would like to add:

	Name	From Phase	To Phase
1	<input type="text" value="Pre-Trial"/>	<input type="text" value="Mediation"/>	<input type="text" value="(Select)"/>
2	<input type="text"/>	<input type="text" value="(Select)"/>	<input type="text" value="(Select)"/>

[+ add more](#)

Displaying records 1-10 of 12. [Next](#)

	Name	From Phase	To Phase
1	<input type="checkbox"/> Appeal	Trial	Appeal
2	<input type="checkbox"/> Arbitration	Settlement	Arbitration
3	<input type="checkbox"/> Closed	Appeal	Close
4	<input type="checkbox"/> Depositions	Discovery	Depositions
5	<input type="checkbox"/> Discovery	Suit Filed	Discovery
6	<input type="checkbox"/> Interrogatories	Discovery	Interrogatories
7	<input type="checkbox"/> Mediation	Settlement	Mediation
8	<input type="checkbox"/> Settlement	Depositions	Settlement
9	<input type="checkbox"/> Settlement	Interrogatories	Settlement
10	<input type="checkbox"/> <input type="text" value="Suite Filed"/>	Open	Suit Filed

[Check All](#) - [Uncheck All](#)

Phase Transitions Tab

4. Make the necessary changes to the phase transition **Name** of the selected items.
5. In the **Visible** check-box, enter a check to make the transition visible to the end user, or clear the box to make the transition non-visible. Non-visible transitions are useful when the transition is always going to be accomplished through rule actions or other automated means, not through manual selection by an end user.
6. To change the **Initial Phase** or **Final Phase**, select the desired phase from the drop-down lists at the top of the screen.
7. Click **Save**.
8. Test the phase transitions to ensure no infinite loops or broken sequences exist.
 

Once the testing is complete, the phase transition is modified.

#### 3.1.4.4.6.3 Deleting Phase Transitions

If you no longer need a phase transition, you can delete the phase.

**Caution:** *If possible, delete phase transitions during the design stage when there are no existing records in the database. If you delete a phase transition that is used by existing records, the information is deleted from the records and the phase transition sequence is broken.*

#### To delete a project phase transition

1. Click the **Phase Transitions** tab of the appropriate custom object definition.
2. Select the check-boxes next the phase transition that you want to delete.
3. Click **delete**.
4. When asked to confirm your action, click **OK**.

The phase transition information is deleted from the system.

5. Test the remaining phase transitions to ensure that no infinite loops or broken sequences exist.

The phase transition is deleted.

#### 3.1.4.4.7 Defining Assignee Roles for Custom Objects

*Assignee roles* are job functions you define for users when they are assigned to a project in the end-user interface. Depending on the custom object, assignee roles may include, for example, associates, supervisors, adjusters for claims, defense attorneys, paralegals, and secretaries for legal matters. In the end-user interface, [assignee roles appear](#) in the **Role** drop-down list of the **Assignees** block of project record.

Assignee roles are lookup table items that pertain only to custom object definitions and are defined within them. They have tree positions that identify them in the database. For more details, see [Using Categories and Lookup Tables](#). As a TeamConnect solution developer or administrator, you can insert the desired assignee role in the appropriate table or delete them if necessary through the **Assignee Roles** tab of the respective custom object definition.

#### 3.1.4.4.7.1 Adding Assignee Roles

##### To add an assignee role to an object

1. Open the **Assignee Roles** tab of the appropriate custom object definition.

**Assignee Roles**

Show roles in node:

Number of entries you would like to add:

	Role Name	Order	Tree Position
1	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>
2	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>

[+ add more](#)

	Role Name	Order	Tree Position
1	<input type="checkbox"/> Specialist	0	SPCL
2	<input type="checkbox"/> Adjuster	0	ADJT
3	<input type="checkbox"/> <input type="text" value="Underwriter"/>	<input type="text" value="0"/>	UNWR <input type="button" value="ok"/> <input type="button" value="cancel"/>
4	<input type="checkbox"/> Claim Associate	0	ASST
5	<input type="checkbox"/> Clerk	0	CLRK
6	<input type="checkbox"/> Supervisor	0	SPRV

[Check All](#) - [Uncheck All](#)

Assignee Roles Tab

- In the **Show roles in node** drop-down list, select the desired hierarchy level where you want to add assignee roles.
- Select the **Number of entries you would like to add** from the list.
- For each data entry row, enter the appropriate information in the fields as described in [the Assignee Roles Tab table](#).
- If you want to continue adding assignee roles, click **add more**. Otherwise, click **Save** on the toolbar.

The newly added roles appear in the list below the data entry fields and are immediately available in the end-user interface.

The following table describes the items on the **Assignee Roles** tab.

Assignee Roles Tab

Field or control	Description
<b>Show roles in node</b>	Select the desired node (hierarchy level), where you want to add assignee roles.
<b>Role Name</b>	Enter the desired unique name for the role. The maximum length is 50 alphanumeric characters.
<b>Order</b>	Enter an integer to indicate the order in which you want the name of this category to be displayed in the <b>Role</b> drop-down list in the end-user



	<p>interface. Roles with the same <b>Order</b> are sorted alphabetically and displayed accordingly.</p> <p><i><b>Tip:</b> It may be helpful to enter orders in increments of five. If you ever have to add more roles, you will have some leeway as to where in the drop-down list you can place the additional roles.</i></p>
<b>Tree Position</b>	<p>Enter a 4-character alphanumeric combination to identify the role. See <a href="#">Hierarchical Tree Structure</a> for more details.</p> <p><i><b>Note:</b> You can enter the tree position in uppercase or lowercase, however, in the database, the tree position is saved in uppercase.</i></p>
<b>Role List</b>	<p>Displays a list of roles created for the object. Select or clear the corresponding check-boxes on the left to delete or edit the respective roles.</p>

#### 3.1.4.4.7.2 Editing Assignee Roles

If you need to change the order in which assignee roles appear in the drop-down list in the end-user interface, or their names, you can do so by editing these assignee roles. However, if you need to change their tree positions, you will need to delete the roles first.

**Caution:** Roles should be deleted during the design stage when there are no existing records in the database. Otherwise, data will be lost.

#### To modify an assignee role

1. Open the **Assignee Roles** tab of the appropriate custom object definition.
2. Select the appropriate node (hierarchy level) where you want to modify assignee roles.
3. Select the check-boxes next to the assignee roles you would like to modify.
4. Click **edit**.

The corresponding fields of the selected items become editable.

**Assignee Roles**

Show roles in node:

Number of entries you would like to add:

	Role Name	Order	Tree Position
1	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>
2	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>
<a href="#">+ add more</a>			
	Role Name	Order	Tree Position
1	<input type="checkbox"/> Specialist	0	SPCL
2	<input type="checkbox"/> Adjuster	0	ADJT
3	<input type="checkbox"/> <input type="text" value="Underwriter"/>	<input type="text" value="0"/>	UNWR <input type="button" value="ok"/> <input type="button" value="cancel"/>
4	<input type="checkbox"/> Claim Associate	0	ASST
5	<input type="checkbox"/> Clerk	0	CLRK
6	<input type="checkbox"/> Supervisor	0	SPRV

[Check All](#) - [Uncheck All](#)

Assignee Roles Tab

5. Make the necessary changes to the **Role Name** or **Order** of the selected items.
6. Click **Save**.

The assignee role is modified.

#### 3.1.4.4.7.3 Deleting Assignee Roles

If, at the design stage when there are no existing records in the database, you need to delete an assignee role, or change its tree position, you can do so by deleting the assignee role.

**Caution:** *If the roles you have selected are used in any records, the data entries in these fields will be deleted completely. Also, if you delete a node, all of its sub-levels and their data entries are deleted too.*

#### To delete an assignee role

1. Open the **Assignee Role** tab of the appropriate custom object definition.
2. Select the appropriate node (hierarchy level) where you want to modify assignee roles.
3. In the role list, select the check-box next to the item you want to delete.
4. Click **delete**.
5. When asked to confirm your action, click **OK**.

6. Click **Save**.

The assignee role is deleted.

#### 3.1.4.4.8 Customizing Notifications for Object Definitions

Notifications are sent by email to users who have opted to receive them whenever certain actions take place in TeamConnect. The default and custom system notifications can be replaced at the object level with custom notification templates. Customizing notification templates for an object definition as detailed below will supersede any [custom notifications](#) in the TeamConnect Admin Settings.

#### To customize a notification for an object definition

1. Open the **Notifications** tab of the appropriate system or custom object definition.
2. Select the desired notification template from the one of the following drop-down menu options:
  - **Set as an assignee**—The notification that users receive when assigned to a record or task.
  - **Workflow approval request**—The notification that users receive when assigned as the approver of a request.
  - **Workflow approval notice**—The notification that users who complete an action that created an approval request receive when another user approves the request.
  - **Workflow rejection notice**—The notification that users who complete an action that created an approval request receive when another user rejects the request.
  - **Workflow review request**—The notification that users receive when assigned as a reviewer of a request.
  - **Workflow expiration notice**—The notification that approvers receive to remind them that an approval request is about to expire.
  - **Workflow review completion notice**—The notification that users receive when a review they requested is complete.
  - **Set as an attendee**—The notification that users receive when added as an attendee to an appointment.
3. Click **Save**.

#### 3.1.4.5 List Displays in Embedded Objects

List display is a type of screen that is used to display embedded object records within their parent object record. For example, if Relation is an embedded object in the Dispute custom object definition, the list display you select determines how Relation records appear within the block or section in each Dispute record.

TeamConnect always uses the editable list style to display embedded object records. See the non-editable hyperlinks of the **Project** column in the following image.

Relations <a href="#">Edit</a>		
Project	Relation	Project
DISP-000000-Kremer vs. State of California	is a cross claim for	<a href="#">ADCO-00000001</a>
DISP-000000-Kremer vs. State of California	is an addendum of	<a href="#">ADCO-00000004</a>

**Example: Embedded Object as Editable List Block in End-User Interface**

The user clicks either **Edit** (next to **Relations**) or the **Edit** button in the toolbar, and the editable display of the block appears, that is, the list changes from view mode to edit mode. See the object links of the **Project** column in the following image.

Relations					
1 - 3 of 3					
<input type="checkbox"/>	Line	Project	Relation	Project	Action
<input type="checkbox"/>	1	DISP-000000-Kremer vs. State of California	is a cross claim for	<a href="#">ADCO-00000001</a>	- +
<input type="checkbox"/>	2	DISP-000000-Kremer vs. State of California	is an addendum of	<a href="#">ADCO-00000004</a>	- +
<input type="checkbox"/>	3	DISP-000000-Kremer vs. State of California	(Select) <input type="text"/>	<input type="text"/>	↔ - +

[Remove](#) [Add New Item](#)

**Example: Embedded Object List Block in Edit Mode**

#### 3.1.4.5.1 Displaying Embedded Records

Embedded object records can be displayed on a tab or within a block in the parent object record. They cannot be displayed in the **All Services** list. When you create embedded objects, you configure their records to be displayed either as search pages with different search views, or as editable lists as shown in the following image.

**Assignees**

1 - 1 of 1

<input type="checkbox"/>	Line	Primary	Status	Assignee	Role	Action
<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	Active	(Select)	(Select)	- +

**Remove** **Add New Item** **Unassign** **Reassign**

**Involved Parties**

**New Involved**

Involved Parties 0 - 0 of 0 **View:** Default **Advanced Involved Search**

Role	Object Link
Nothing found to display.	

Involved Parties per page 30

**Example of Embedded Object List Display**

You may specify which embedded object record fields appear in the editable list. Users can then edit field values directly from the list and save the changes. That way, if a field value has to be edited in multiple records, users do not have to open each individual record to perform such edits.

#### 3.1.4.5.2 Creating Embedded Objects

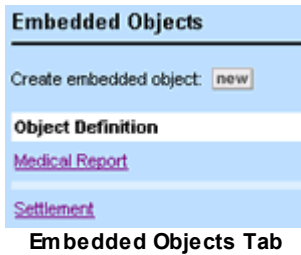
All embedded objects for a custom object can be created through the **Embedded Objects** tab of the respective custom object definition.

**Tip:** *Creating an embedded object definition is similar to creating a custom object definition. For details, see [Creating Custom Objects](#) and [the General Tab on Custom Object Definition Screens table](#).*

**Note:** *You can convert an embedded object definition to a child object definition if your design requires it. There are several cautions and considerations in doing so. See [Converting an Embedded Object Definition to a Child Object Definition](#)*

#### To create an embedded custom object

1. Open the **Embedded Objects** tab of the appropriate parent custom object definition.



**Embedded Objects Tab**

2. Click **new**.

The **General** tab of a new embedded object definition screen appears with empty fields.

3. Enter the following information:
  - Name of embedded object and its plural form
  - Unique code
  - Icon image file

**General Tab on Embedded Object Definition Screens**

4. If the embedded object definition has a main contact, set it as **Contact-centric** and specify the **Role** for that contact.
5. Click **save**.
6. Click the **List Display** tab and define the list's columns of information in the **Editable List** section of the **List Display** tab. For more information about defining editable lists, see [Creating Results Display Columns](#).
7. Continue defining the embedded object definition's properties as described in [Creating Custom Objects](#).
8. Click save.

You can access the new embedded object definition either from the parent object definition, or directly from the [Object Definition List](#) screen as described in [Viewing Object Definitions](#).

## Resulting user interface Changes

Whenever you create an embedded object for a custom object, the following takes place in the corresponding object screens in the Designer:

- A block named after the embedded object is created and becomes available for adding to the custom object views of the parent custom object definition. To add the embedded object's block, open **General** tab of the applicable object view and select it from the **Add the block** field.
- The embedded object becomes available in the **Related Object** drop-down list on the **Filter Display** and **Results Display** tabs of parent object search views.
- The embedded object becomes available on the **Records** tab of the parent object's templates.
- The embedded object becomes available as a related object in all applicable instances of Object Navigator. For details, see [Using Object Navigator](#) and [Object Model: Read This First](#) plus the additional reference tables this reference points you to.
- The name of the embedded object is added to the list of custom objects on the **Rights** tab of all user and group accounts. This means that in order for the end users to be able to use this object, the rights must be assigned accordingly.

## Points To Remember

When creating an embedded object within a custom object, consider the following information:

- You can create multiple embedded objects for a single custom object.

For example, the Dispute object could have Allegation, Corrective Action, and Damage as its embedded objects.

- In the end-user interface, embedded object records are accessible through a separate page in the record page of the parent record. This page has the same name as the embedded object in the plural.

The embedded object's block is automatically added to the system view of the parent object. To display embedded object records, you must add the block to the object view used to display the parent object. For more details, see [Creating Custom Pages](#).

- You can set the block for embedded object records to be displayed either as a search screen with different search views, similar to child object records, or as an editable list similar to batch screens used for sub-objects.
- When creating an embedded object, you must assign the [appropriate rights](#) for the embedded object to the users and groups, like you do to any custom object.
- Embedded object definitions can be created only from within their respective parent custom object definitions. You cannot create embedded object definitions from the following locations:
  - **Create a new** drop-down list.
  - [Object Definition List](#) screen displayed when you choose **Object Definitions** in the **Go to** drop-down list.

You can, however, see and access the existing embedded object definitions from the [Object Definition List](#).

- End users cannot save an embedded object record unless they save its parent object record.

## 3.1.4.5.3 Defining Display for Embedded Object Records

Defining the editable list display for an embedded object is similar to defining the Results display for a search view.

### Points To Remember

- If you want users to access records directly from an editable list, include a column with the **Object Link** system display key. When the user clicks the record hyperlink, it displays the record screen of the selected record. See [Defining Object Links](#).
- When adding or editing a record, the column with the object link has no data entry fields or displays a non-editable hyperlink.
- Make sure to add columns for all fields required to create records of the selected embedded object, such as the **Name** field. For details on required fields, see [Required Fields](#).
- You can include custom fields only from the Root category.

### To define how embedded object records are displayed within their parent record

1. Open the **List Display** tab of the appropriate embedded object definition.

**Object Definition: Award**

General | Unique ID | Name | Categories | Custom Fields | Forms | Blocks | Object Views | Rules | **List Display**

**Editable List**

Number of entries you would like to add: 1

Column Name	Display Key	Order	Display Width
1	<input type="radio"/> System (Select) <input type="radio"/> Custom (Root only) (Select)		
1	<input type="checkbox"/> Amount Awarded	AmountGrantedAwardedAW	30
2	<input type="checkbox"/> Awarded To	contact	50
3	<input type="checkbox"/> Award Type	TypeAW	20
4	<input type="checkbox"/> Date Awarded	DateGrantedAwardedAW	40
5	<input type="checkbox"/> Description	DescriptionAW	60

Check All - Uncheck All

#### Editable List Settings for Embedded Objects

2. Define the desired batch screen for the editable list in the same way that you would define the Results display for a search view, using the instructions in [Search View Results Display](#).



3. Save the embedded object definition.
4. Test the created display in the end-user interface.

#### 3.1.4.5.4 Converting Embedded Objects to Child Objects

TeamConnect lets you convert embedded objects to child objects. Child objects and embedded objects are very different, so the conversion changes the way that the converted object displays and performs.

An embedded object is not independent of its parent object and does not have tabs on which to enter information. Embedded objects are only accessible through the parent object.

A child object, though not fully independent of its parent object, has its own set of tabs, such as **Phases**, **Phase Transitions**, **Templates**, and **Security**. When you convert an embedded object to a child object, all of the tabs that are typical to a child object are now available.

### Before You Convert

Before you convert an embedded object to a child object, ensure that you have Object Definition Update rights.

### What is Not Retained?

Access rights are not retained. You must assign rights to the object as if it were being newly created.

### What Are the Prerequisites for Conversion?

You must perform the conversion at a time when no ordinary users are logged in to TeamConnect.

#### To convert an embedded object definition to a child object definition

1. In the Designer, click **Go to**, and then click **Object Definitions**.
2. In the Object Definitions list, expand the parent or child object definition that contains the embedded object that you want to convert.
3. Click the embedded object.  
The embedded object opens to its **General** tab.
4. Click **Convert to Child object definition**.
5. A confirmation message appears. If there are existing records for the embedded object definition, the record count is shown. If there are a large number of records, you are advised that the confirmation process may require an extended period of time.
6. Click **OK** on the confirmation message if you are sure that you want to convert this object. Wait for the process to complete.

The former embedded object is now a new child object and displays the tabs that apply to all child object definitions.

7. Click any tab to enter additional information.
8. Save or Save and Close the record.

The new child object is displayed under its parent object in the Object Definition list view.

**Note:** *If you convert an object, and then view it before you save it, you may still see properties that applied to the object before you converted it. To ensure a complete conversion, save the converted object before you view it.*

## Conversion Results

**Note:** *If conversion fails due to an application server crash or other major error, it can be re-run later, and will resume with any records that were not completed in the previous run.*

After the conversion, a status message is added to the system log. The new child object now has the following properties:

- A default assignee role named Default with tree key DEFA
- An "All <name of embedded object, such as "~All matters">" search view is set as search across objects
- A default search module with result columns (Opened On and Main Assignee) and criteria (name and number string)
- A templates folders under Root\System\Object Definitions\<name of embedded object, such as "matter">\
- The "Create a Copy" group rights
- All project rule types (user invoke custom action, audit, etc.)
- The **Parent required** and **Do not display in menu** check-boxes are checked on the **General** tab of the object definition
- In the Designer, the following tabs are now available for the new child object:
  - Phases
  - Phase Transitions
  - Assignee Roles
  - Search Views
  - Templates
  - Wizards
  - Embedded Objects
  - History

- Security
- In the TeamConnect user interface, the following pages are now available for the new child object:
  - Phases
  - Assignees
  - Relations
  - Documents
  - Involved
  - Tasks
  - Appointments
  - Expenses
  - Accounts
  - Workflow
  - History

#### 3.1.4.6 Deleting a Custom Object Definition

You must take some precautions when you are about to delete a custom object definition.

Instances of custom objects become rows in the T\_PROJECT database table. If your database has such instances in existence, you must delete those rows before deleting the object definition. Locate the Y\_OBJECT\_DEFINITION table row that corresponds to the object definition that you wish to delete, and note the value of column APPLICATION\_ID in that row. Then run a manual SQL query that deletes T\_PROJECT rows where column APPLICATION\_ID contains that value.

In addition, if you are running Data Warehouse, ensure that the Data Warehouse synchronization job runs after you have deleted the T\_PROJECT rows, and before you actually delete the object definition itself. Data Warehouse tracks any deletions that are done against T\_PROJECT and it cannot properly process those deletions if the matching object definition is already gone at the time that the synchronization job runs. After synchronization is complete, you may delete the object definition.

### 3.1.5 Using Categories and Lookup Tables

*Categories* and *lookup tables* allow you to organize information that displays on the end-user interface page.

#### Categories

Categories are hierarchical list items that allow you to organize custom fields of a record into blocks or sections in a given end-user interface page. By adding or deleting categories in a record, end users may display or hide associated blocks of details (custom fields) in that record. Categories also help users to organize records by certain types, for example, employee contact records as opposed to non-employee contact records, or domestic and international accounts.

For another example, you may set up a block so that, if a user adds a **External : Attorney** category to a contact record, a **Contact Vendor Details** block also appears.

You must specify categories when defining:

- Custom Fields
- Blocks
- Access to object records at the category level

## Lookup Tables

Lookup tables are database tables that organize, store and quickly access multiple items, such as contacts' roles, types of addresses, phone, fax, and other contact information. In addition to adding, renaming, inactivating, and deleting items in system lookup tables, you may define new custom lookup tables and their contents.

In the end-user interface, lookup tables appear either as drop-down lists (most common) or radio buttons that allow the user to look up necessary information and make an appropriate selection.

TeamConnect's user interface uses several types of list fields that reference lookup table items. For example, users can select the country in which a contact lives from a drop-down list that references the **Country Item** system lookup table.

### 3.1.5.1 Using Categories

Categories are list items used to organize custom fields into blocks of fields in records. By adding or deleting categories in a record, users display or hide the associated blocks of fields. You may also use categories to organize records by certain types, such as employee contact records as opposed to non-employee contact records.

In custom pages, all custom fields, and custom blocks are tied directly to categories, that is, they are created and displayed by category only. For details, see [Custom Fields](#).

***Note:** The word **details** is sometimes used to signify a block of custom fields, in the end-user interface, that are specific to one category. Since many categories may be assigned to a single record, it is also possible for **details** to represent a collection of many such blocks.*

When granting user groups access to information on custom pages, you must give them the [rights to the categories](#) for which the custom fields were created.

## Points To Remember

Consider the following points when working with categories:

- Based on categories, you may create and display custom fields, and blocks.
- When users delete categories from records, any data in the associated fields is lost.
- If you delete a category from an object definition, the fields associated with the category and any data in the fields are lost.
- Access to all categories in the end-user interface is controlled by the rights assigned to user group accounts.

- The categories for the Involved object are called **Involved Roles** and appear in the **Involved** block of each custom object record.
- When you create a category, you specify a four-character alphanumeric combination to uniquely identify it in the TeamConnect database. For more details, see [Category Tree Positions](#).
- A category name must be unique within its object if it has any [custom fields included in Data Warehouse](#).

#### 3.1.5.1.1 Viewing Categories

You can view lists of categories through the **Categories** tab in a specific object definition.

In the end-user interface, categories appear in the **Categories** page of each object record (except for Tasks and Expenses), where users can select them from the **Category** hierarchical list.

#### To view an object's categories

1. In the Designer window, select **Object Definitions** from the **Go to** drop-down list.  
The **Object Definition List** appears.
2. Click the appropriate object definition.  
The **General** tab of the object definition appears
3. Click the **Categories** tab.

### Categories

Show items in node:

Number of entries you would like to add:

Item Name	Order	Tree Position	Is Active
1	<input type="text" value="0"/>	<input type="text" value=""/>	YES
2	<input type="text" value="0"/>	<input type="text" value=""/>	YES

[+ add more](#)

Item Name	Order	Tree Position	Is Active
1 <input type="checkbox"/> InHouse	1	INH5	YES
2 <input type="checkbox"/> Attorney	2	ATTY	YES
3 <input type="checkbox"/> Paralegal	2	PARA	YES
4 <input type="checkbox"/> Investigator	2	NVGS	NO

[Check All](#) - [Uncheck All](#)

Categories Tab

4. In the **Show items in node** list, select the appropriate node where you want to view categories.
5. Do one of the following actions:
  - a. To add categories, see [Adding Object Categories](#).
  - b. To edit categories, see [Editing Object Categories](#).
  - c. To inactivate categories, see [Inactivating Object Categories](#).
  - d. To delete categories, see [Deleting Object Categories](#).

The following table describes the items in the **Categories** tab.

**Categories Tab**

Field or control	Description
<b>Show items in node</b>	Select the appropriate node (hierarchy level) where you want to add the category.
<b>Item Name</b>	Enter a name for the category. The maximum length is 50 alphanumeric characters.
<b>Order</b>	Enter an integer to indicate the order in which the category appears in the <b>Category</b> drop-down list in the end-user interface. Categories with the same <b>Order</b> appear in alphabetical order.  <i>Tip: It may be helpful to enter orders in increments of five. If you ever have to add more categories, you have leeway as to where you insert them.</i>
<b>Tree Position</b>	Enter a four-character alphanumeric combination to identify the category. This combination is the category's partial tree position. The full tree position of a category must include the partial tree positions of its parent categories all the way up to the root category. For more details, see <a href="#">Points To Remember</a> . Although you can enter the tree position in uppercase or lowercase, it is saved in uppercase letters in the database.
<b>Category List</b>	Displays a list of categories created for the object. Select or clear the corresponding check-boxes on the left to delete or edit categories.
<b>edit</b>	Click to change the name of the selected category.

	For more details, see <a href="#">Editing Object Categories</a> .
<b>delete</b>	<p>Click to delete the selected category.</p> <p><b>Caution:</b> <i>Deleting categories can cause the loss of associated data. If you delete a node, all of its sub-levels and their data entries are also removed. If your system is in production, consider inactivating categories instead of deleting them.</i></p> <p>For more details, see <a href="#">Deleting Object Categories</a>.</p>
<b>inactivate/ activate</b>	<p>Click to inactivate or activate the selected category.</p> <p>When you inactivate a category, users can see the item in all fields that reference the category, but they cannot select it. Inactivated categories appear differently than active categories. For more information, see <a href="#">Inactivating Object Categories</a>.</p>

#### 3.1.5.1.1.1 Locations of Categories in End-User Interface

##### Object Category Locations in the End User Interface

Category name	Object type	Section in end-user interface	Field name
<b>Account</b>	System	Categories	Category
<b>Appointment</b>	System	Categories	Category
		General	Default category
<b>Contact</b>	System	Categories	Category
<b>Document</b>	System	Categories	Subject
<b>Expense</b>	System	General	Category
<b>History</b>	System	General	Default category
		Categories	Category
<b>Invoice</b>	System	Categories	Category
<b>Line Item</b>	System	General tab of Invoice > Line Items	Category

<b>Involved (Role)</b>	System	Project > Involved	Default role
		Involved > Roles	Role
<b>Project</b>	Custom	Categories	Category
<b>Task</b>	System	Contact > Rates	Task
		Task > General	Default category

### 3.1.5.1.1.2 Category Tree Positions

When you create a category on the **Categories** tab of the an object definition, you specify a four-character alphanumeric combination to uniquely identify it in the database. This is its partial tree position in the category hierarchy.

The *full tree position* of a category is a combination of partial tree positions assigned to the category and all of its parent categories, starting with the root category and separated by underscores.

For example, if category B with partial tree position BBBB is created under category A with partial tree position AAAA in the Account object definition, the full tree position of category B is ACCT\_AAAA\_BBBB.

On the **Categories** tab of the object definition, the full tree position is indicated by the selections you make in the **Show items in the node** drop-down list.

The top-level node is called the Root category. Its full tree position is the same as the "unique code" of the object definition, which are listed in [the System Object Unique Codes table](#) and appear in the following locations:

- **General** tab of the respective object definition
- **Object Definition List** screen

You may need a category's full tree position when creating custom blocks, automated actions and qualifiers, document generator templates, XML requests, and so on.

Each Involved and embedded object has its own root category that is independent of the custom object to which it belongs. Its full tree position is the unique code of the respective Involved or embedded object.

The following table lists the unique code (or Root category) of all system objects:

**System Object Unique Codes**

Object	Unique Code
<b>Account</b>	ACCT
<b>Appointment</b>	APPT



<b>Contact</b>	CONT
<b>Contact Group (Address Book)</b>	CTGR
<b>Document</b>	DOCU
<b>Expense</b>	EXPE
<b>Fee</b>	FEE
<b>History</b>	HIST
<b>Invoice</b>	INVC
<b>Task</b>	TASK
<b>User Account</b>	USER
<b>Group Account</b>	GROU

Likewise, the Root category of custom objects is the same as the object's unique code. For example, if the Dispute object has a unique code DISP, its top-level category would also be named Dispute, and its tree position would be DISP.

#### 3.1.5.1.1.3 System Object Categories

By default the **Categories** tab of system objects displays the following information:

- **Root** as the only available node (hierarchy level), which you cannot delete or inactivate because it is needed to see other categories.

***Important:** Root is a concept only. The name of the root node never actually appears as **Root** in the interface. The root node adopts the name of the object itself. Thus the name of the root node for the Contact object is **Contact**, as shown in the figure below.*

The tree position of the Root category is the same as the unique code assigned to the system object on the General tab of its object definition.

- **Default** category, which you can delete after you add other categories.

As you add categories to a system object, they are added to the **Show items in node** drop-down list and displayed hierarchically. Subcategories are marked by the appropriate indentation. Children of the **Root** category appear at the same level as the **Root** category.



**Example: System Object Categories in End-User Interface**

#### 3.1.5.1.1.4 Custom Object Categories

By default, the **Categories** tab displays the name of the custom object as the only available node (hierarchy level), which you cannot delete or inactivate because it is needed to see other categories.

The tree position of the Root category is the same as the unique code assigned to the system object on the **General** tab of its object definition.

As you add categories to a custom object, they are added to the **Show items in node** drop-down list and displayed hierarchically. As in system objects, subcategories are marked by the appropriate indentation in the hierarchy. Children of the root category appear at the same level as the root category.

For example, Claim is the custom object name and root category in the following figure.



**Custom Object Categories, as Displayed in End-User Interface**

### 3.1.5.1.2 Adding Object Categories

You define new categories on the **Categories** tab of the respective object definition.

#### To add a category to an object

1. Open the **Categories** tab of the desired system or custom object definition.
2. In the **Show items in node** list, select the appropriate node where you want to add categories.
3. Select the **Number of entries you would like to add** from the corresponding drop-down list.
4. For each data entry row, fill in the appropriate fields as described in [the Categories Tab table](#).
5. If you want to continue adding categories, click **add more**. Otherwise, click **Save** on the toolbar.

The newly added categories appear in the list below the data entry fields. The categories are available immediately in the end-user interface.

### 3.1.5.1.3 Editing Object Categories

You can change category names and the order in which they appear in category lists. You cannot change the tree positions of a category unless you first delete the category and then recreate it.

#### To modify an object category

1. Open the **Categories** tab of the desired system or custom object definition.
2. In the **Show items in node** list, select the appropriate node where you want to modify object categories.
3. Select the check-boxes next to the categories you would like to modify.
4. Click **edit**.  
The corresponding fields of the selected categories become editable.
5. Make the necessary changes to the **Item Name** or **Order** of the selected categories.
6. Click **Save**.

The changes you made become effective immediately upon saving.

### 3.1.5.1.4 Inactivating Object Categories

You can inactivate categories so that users are prevented from selecting them from the corresponding records in the user interface. Inactivating categories allows you to prevent the use of unwanted values yet maintain existing data in records that include inactive categories.

**Important:** Before inactivating categories, read [Points To Remember](#).

### To inactivate an object category

1. Open the **Categories** tab of the desired system or custom object definition.
2. In the **Show items in node** list, select the appropriate node where you want to inactivate object categories.
3. Select the check-box next to each category you would like to inactivate.
4. Click **inactivate**.

If the selected categories do not have any child categories, they are inactivated, users cannot select them, and you are finished with this procedure.

If any of the selected categories have at least one child category, a message box with the following question appears:

**Are you sure you want to inactivate these items?**

5. Do one of the following actions:
  - a. If you want to inactivate the items and all of their child items, select the check-box next to **Include all child items** and click **ok**.
  - b. If you want to inactivate just the items within the selected node but not any of the child items, click **ok**.
  - c. If you decide not to inactivate any items, click **cancel**.

Inactivated categories are indicated in the **Active** column of the **Categories** list of the object definition.

**Important:** Update the status of all inactivated categories so that this information is available if you ever need to convert your TeamConnect data.

### Points To Remember

Consider the following points before inactivating categories:

- When you inactivate a category, its associated fields will continue to be displayed in records that already had the category assigned.
- Because an inactive category cannot be selected, its associated fields cannot be displayed in records that did not include the category prior to it being inactivated—even if users add an active child category.
- If an inactive category is assigned to a field via templates, rules, or the XML layer, an exception will be thrown to prevent the creation or update of the record. For example, if a template specifies the default category to be set on Create, but the category is deactivated, the record cannot be created.
- You cannot inactivate categories that are specified as the default category or associated with required custom fields. If you attempt to do so, you will receive an error.
- Inactivated categories are visible to end users in records where they were selected prior to being inactivated. In some circumstances, inactivated categories appear in gray. For example, the

**Show details for** field displayed on the **General** tab of some project records will display any inactive categories in gray. If accessibility settings are enabled, **(inactive)** is appended to inactive category names and they are not displayed in gray.

- Users can specify inactivated categories as search criteria and report qualifiers. Inactivated categories appear in gray in search screens or **(inactive)** is appended to the category names.
- Existing records with inactive categories retain the associated custom field values. Users can edit records with previously selected inactivated categories and click **save**.
- If a user deletes an inactive category from a record and then clicks **save**, they cannot revert to the prior value unless a TeamConnect solution developer temporarily activates the category.
- Users can change an inactive category that was specified as the default category of a record (before it was inactivated) to a non-default category. However, once users save the change, they cannot set the inactive category as the default, despite its still being set in the record.
- If you ever need to convert your TeamConnect data, you must first activate all inactive categories and lookup table items. Once the data conversion is completed, you can restore the inactive items manually.
- When you inactivate categories, their status is recorded in the **Active** column of the **Categories** tab in the object definition.
- If a category has parent or child categories, you cannot readily determine their active status without navigating to each level in the hierarchy. Therefore, you should record the status of all inactive categories so that this information is readily available if you ever need to convert your TeamConnect data. Otherwise, you will have to manually search for any inactive categories.
- In the special case of the **Line Item** object definition, you cannot inactivate a category if that category is listed in the **CSM Settings** page, on the pages related to billing codes and to non-US tax codes, as being part of the default set of codes that are authorized for vendors. This includes task, fee, and expense codes. You must first de-authorize the code in that screen and de-authorize it in each vendor that has that code. Then you can inactivate the category.

#### 3.1.5.1.5 Activating Object Categories

You can activate previously inactivated categories at any time so that users can select them from the corresponding object records in the user interface.

#### To activate object categories

1. Open the **Categories** tab of the desired system or custom object definition.
2. In the **Show items in node** list, select the appropriate node where you want to activate object categories.
3. Select the check-box next to each category you would like to activate.
4. Click **activate**.

If the selected categories do not have any child categories, they are activated, users can select them, and you are finished with this procedure.

If any of the selected categories have at least one child category, a message box with the following question appears:

**Are you sure you want to activate these items?**

5. Do one of the following actions:
  - a. If you want to activate the categories and all of their child categories, select the check-box next to **Include all child items** and click **ok**.
  - b. If you want to activate just the categories within the selected node but not any of the child items, click **ok**.
  - c. If you decide not to activate any categories, click **cancel**.

Activated categories are indicated in the **Active** column of the **Categories** list of the object definition.

**Important:** Update the status of all activated categories so that this information is available if you ever need to convert your TeamConnect data.

#### 3.1.5.1.6 Deleting Object Categories

If you need to delete a category or change its tree position, you can do so by deleting the category. Deleting a category removes all of its sub-categories and corresponding data entries, custom fields, and custom blocks.

**Caution:** You should only delete object categories during the design phase, when there are no corresponding object records in the database. Otherwise, loss of data from the associated fields will occur. Alternately, you can inactivate categories.

**To delete an object category**

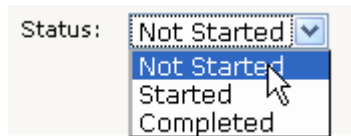
1. Open the **Categories** tab of the desired system or custom object definition.
2. In the **Show items in node** list, select the appropriate node where you want to delete object categories.
3. Select the check-boxes next to the categories you would like to delete.
4. Click **delete**.
5. When asked to confirm your action, click **OK**.
6. Click **Save**.

The selected categories are permanently deleted from the system.

### 3.1.5.2 Using Lookup Tables

Lookup tables store and list items for users to select. Depending on the system settings and an individual user's Preference settings, lookup tables can be displayed in the following types of fields:

- Drop-down list (most common)
- Option (radio) buttons



**Lookup Table Display Example of a Drop-Down List**

- Disable record locking in edit mode
- Enable record locking in edit mode

**Lookup Table Display Example of Option Buttons**

System lookup tables contain a set of default items. For example, the Address Type lookup table has three items: **Home**, **Business**, and **Other**. You can add to, edit, inactivate, or delete them to fit your design.

You cannot create new system lookup tables. However, you can create custom lookup tables that are specific to your organization. For details, see [Creating Custom Pages](#).

#### 3.1.5.2.1 Types of Lookup Tables

TeamConnect lookup tables can be viewed and modified on the **Lookup Tables** screen, which has separate tabs for the following types of lookup tables:

##### **System** | Custom | Multi-currency Lookup Tables Tabs

- [System Lookup Tables](#)—Provided by default with TeamConnect, system lookup tables are used in specific areas of TeamConnect. For example, the **Activity Item** table is included for task records. All fields using system lookup tables appear in system blocks. Although you can add or delete items to system lookup tables, you cannot delete the tables. For a complete list of system lookup tables, see the [System Lookup Tables table](#).
- [Custom Lookup Tables](#)—Referenced by custom fields to store information specific to your business. You can use custom lookup tables in custom fields for more than one object or belonging to different object categories.

For more information about custom fields and custom lookup tables, see [Creating Custom Pages](#).

- **Multi-currency Table**—Stores information about currencies, such as exchange rates and currency symbols. Invoices can be recorded in any currency that is stored in this table.

## 3.1.5.2.2 System Lookup Tables

System lookup tables are provided with TeamConnect by default for use with specific system objects. You can also create custom lookup tables, as described in [Creating Custom Pages](#).

The following table lists TeamConnect's system lookup tables and their location in the end-user interface. It also indicates which tables can have a hierarchical tree structure.

**System Lookup Table Information**

Table Name	Object Record	System Block	Field Name	Tree Structure	Unique Code	Table Name in object model
<b>Activity Item</b>	Task	General	Activity		ACTI	LTaskActivityItem
	Invoice	Line items/ Expanded View	Activity		ACTI	LTaskActivityItem
<b>Address Type</b>	Contact	General	Type		ADDR	LContAddressType
<b>Area Item</b>	Appointment	General	Area	x	AREA	LApptAreaItem
<b>Contact Relation Type</b>	Contact	Relations	Type of relationship		CONR	LContRelationType
	Involved	Relations	Type of relationship		CONR	LContRelationType
<b>Country Item</b>	Contact	General	Country		COUN	LCountryItem
<b>Email Type</b>	Contact	General	Type		MAIL	LContEmailType
<b>Fax Type</b>	Contact	General	Type		FAXX	LContFaxType
<b>Internet Address Type</b>	Contact	General	Type		INET	LContInetAddressType
<b>Integration Message Queue</b>	This lookup table is deprecated. It provides backward compatibility for existing Integration Message Queue rules.				MESQ	LMessageQueueItem



<b>Invoice Rejection Reason</b>	Invoice	My Approvals pop-up screen	Reason for Reject		INRR	LApphRejectReason
<b>Phone Type</b>	Contact	General	Type		PHON	LContPhoneType
<b>Project Relation Type</b>	Project	Relations	Type of relationship		PRJR	LProjRelationType
<b>Resource Type</b>	Appointment	Resources	Resource		RESO	LApptResourceType
<b>Skill Type</b>	Contact	Skills	Type	x	SKIL	LContSkillType
<b>State Type</b>	Contact	General	State/Province	x	STAT	LContState
<b>Territory Type</b>	Contact	Territories	Territory	x	TERR	LContTerritoryType

#### 3.1.5.2.2.1 Viewing System Lookup Tables

You can access system lookup tables from the main Designer screen to add, rename, inactivate, or delete items.

#### To open the system lookup tables screen

1. On main Designer screen, from the **Go to** drop-down list, select **Lookup Tables**.  
The **System Lookup Tables** screen appears, as shown in the following image.

Item Name	Order	Tree Position	Activity Code	Is Active
1 <input type="text"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>	YES
1 <input type="checkbox"/> A100 Activities	0	A100	ACTIVITY	YES
2 <input type="checkbox"/> Default	0	DEFA	ACTIDEFA	YES
3 <input type="checkbox"/> A101 Plan and prepare for	1	A101	A101	YES
4 <input type="checkbox"/> A102 Research	2	A102	A102	YES
5 <input type="checkbox"/> A103 Draft/revise	3	A103	A103	YES

System Lookup Tables Screen

2. Select the appropriate lookup table in the **Show items belonging to** drop-down list.

If the lookup table has a hierarchical tree structure, another drop-down list appears, labeled **Show items in node**. If necessary, select the appropriate node in the **Show items in node** drop-down list to view the items that are listed within that item.

The items that have been added to the lookup table (or to the selected item within the lookup table) appear.

The following table describes the items in the System Lookup Tables screen.

System Lookup Tables Screen

Field	Description
<b>Show items belonging to</b>	Select the appropriate system lookup table containing the items you want to view.
<b>Show items in node</b>	Select the appropriate node (hierarchy level) where items are located. <i>Note: This field appears only in those lookup tables that can have the hierarchical tree structure.</i>
<b>Item Name</b>	Enter a name for the item. This name appears to the end-user. The maximum length is 50 alphanumeric characters.

<b>Order</b>	<p>Enter an integer to indicates the order in which it appears to the user. Items with the same order are sorted alphabetically. For details, see <a href="#">Hierarchical Tree Structure</a>.</p> <p><i>Tip: Enter integers in increments of five so that you will have room to insert additional lookup table items if necessary.</i></p>
<b>Tree Position</b>	<p>Enter a four-character alphanumeric code that is unique among the items in the lookup table. For details, see <a href="#">Hierarchical Tree Structure</a>.</p> <p><i>Note: While you can enter the tree position in uppercase or lowercase, the tree position is saved in uppercase in the database.</i></p>
<b>Lookup Item List</b>	<p>Displays a list of items in the selected node of the lookup table with corresponding check-boxes to delete or edit the item.</p>
<b>edit</b>	<p>Click to change the name of the selected lookup table item.</p> <p>For more details, see <a href="#">Editing System Lookup Tables</a>.</p>
<b>delete</b>	<p>Click to delete the selected lookup table item.</p> <p><i>Caution: Deleting lookup table items can cause the loss of associated data. If you delete a node, all of its sub-levels and their data entries are also removed. If your system is in production, consider inactivating lookup tables instead of deleting them.</i></p> <p>For more details, see <a href="#">Deleting Lookup Table Items</a>.</p>
<b>inactivate/activate</b>	<p>Click to inactivate or activate the selected lookup table item.</p> <p>When you inactivate a lookup table item, users can see the item in all fields that reference the lookup table, but they cannot select it. Inactivated lookup table items appear differently than active items. For more information, see <a href="#">Inactivating Lookup Table Items</a>.</p>

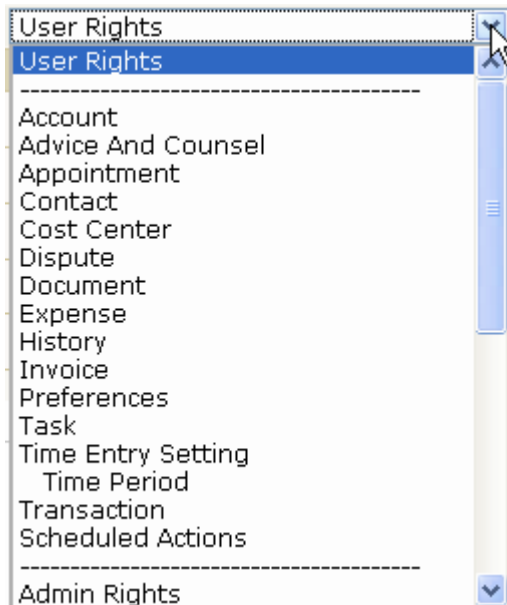
#### 3.1.5.2.2.2 Lookup Table Structure

You may configure the display order of lookup table items and organize them in a hierarchical structure. As you add items to a lookup table, specify their tree positions so they are uniquely identified in the database.

### Hierarchical Tree Structure

Many lookup tables consist of one level, so they appear as a flat list of items. However, lookup tables may also have a hierarchical structure with multiple sub-levels. These lookup tables also

display in a drop-down list and appear in the end-user interface with headings and indents. The names of the sub-levels appear in their appropriate positions in the hierarchy, as shown in the following image.



**Example: System Lookup Table Hierarchy in Drop-Down List**

Levels in the lookup table structure that have their own sub-levels are referred to as Nodes. The tree originates with the Root node, which is the first level and has the following properties:

- Cannot be deleted.
- Not displayed in the end-user interface. Thus, items directly below the Root node appear to end users as first level items.
- In both system and custom lookup tables, the tree position of the Root node is the predefined unique code of the table. For a list of system table unique codes, see [the System Lookup Table Information table](#).

When adding countries and states that you want to appear in the State/Province drop-down lists in the Contact Address block, the following applies:

- If you want a list of states to appear for a country selected in the contact address, be sure that the country is added to the state table with same unique code as the country table and there are states defined as child items of the country item.
- When adding a new country to the Country lookup table, you must also add the new country and its unique code to the state table, and then, if necessary, add the states as child items of the country.

## Tree Positions

All items in lookup tables are assigned a unique key called the tree position. In the **Designer** user interface, the tree position appears as four alphanumeric characters in uppercase, such as BUS1.

When you add items to a lookup table, you specify their tree positions so they can be uniquely identified in the database.

In hierarchical tables, each item in a lookup table is identified by the path or "tree" that leads to it within the table's hierarchy. This tree originates with tree position of the Root and continues to the tree position of an item in the hierarchy.

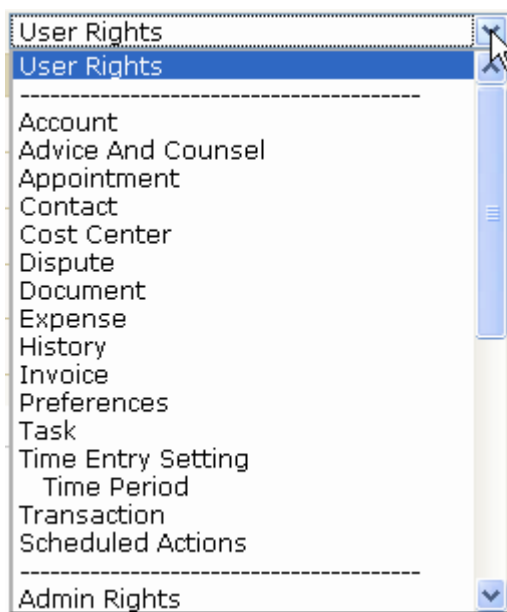
The *full tree position* of a lookup table item is a combination of tree positions assigned to the item and all items above it in the hierarchy, starting with the Root. Each level is separated by underscores.

For example, if an item with tree position CCCC is in the second level node of a lookup table item with the tree position BBBB and the Root's tree position is AAAA, the full tree position of the item is AAAA\_BBBB\_CCCC.

## Order

When you add lookup table items, you can specify an integer to determine the order in which the item appears with respect to the other items in the same hierarchical level (belonging to the same node). Lookup table items with the same order appear alphabetically.

For example, in the following image, the **Order** for **Cost Center** might be 5, because it is the fifth item in its level. **Time Period** might have an **Order** of 1 because it is the first (and only) item in its level.



**Example: System Lookup Table Hierarchy in Drop-Down List**

### 3.1.5.2.3 Adding Lookup Table Items

You can add items to system lookup tables to meet your organization's needs. For example, you could add the **is court of** item to the Contact Relation Type lookup table to support relations to contact records of courts.

#### To add an item to a system lookup table

1. On the **System Lookup Tables** screen, select the appropriate lookup table in the **Show items belonging to** drop-down list.

If the lookup table has a hierarchical tree structure, the **Show items in node** drop-down list appears. Select the appropriate node in the Show items in node drop-down list.

2. Select the **Number of entries you would like to add** from the corresponding drop-down list.
3. For each data entry row, fill in the appropriate fields as described in [the System Lookup Tables Screen table](#), such as its unique tree position.
4. Click **add more**.

The newly added item appears in the lookup item list.

#### 3.1.5.2.4 Editing System Lookup Tables

You can edit system lookup table contents as needed by modifying the name or order of items. For example, you could change the name of the **Lap Top** item in the Resource Type lookup table to **Computer** and change its position in the list so that it appears next to the Projector item.

You cannot edit the tree position of an item, but you can add a new item and inactivate or delete the old item.

If renaming the item changes the item's meaning, you should only rename it on a newly installed system. Otherwise, existing data captured using the tree position of the lookup table item will effectively be altered.

#### To edit a system lookup table item

1. On the **System Lookup Tables** screen, select the appropriate lookup table in the **Show items belonging to** drop-down list.

If the lookup table has a hierarchical tree structure, select the appropriate node in the **Show items in node** drop-down list.

2. Select the check-box next to each item you would like to change.
3. Click **edit**.

The corresponding fields of the selected item or items become editable.

4. Make the necessary changes to the **Item Name** and **Order** of the selected items.
5. Click **ok**.

Your changes are reflected in the lookup item list.

#### 3.1.5.2.5 Inactivating Lookup Table Items

You can inactivate lookup table items so that users are prevented from selecting them from the corresponding fields in the user interface. Inactivating items allows you to prevent the use of unwanted values yet maintain existing data in records that include inactive items.

**Important:** Before inactivating lookup table items, read [Points To Remember](#).

### To inactivate lookup table items

1. On the **System Lookup Tables** screen, select the appropriate lookup table in the **Show items belonging to** drop-down list.

If the lookup table has a hierarchical tree structure, select the appropriate node in the **Show items in node** drop-down list.

2. Select the check-box next to each item you would like to inactivate.
3. Click **inactivate**.

If the items do not have any child lookup table items, the selected items are inactivated, users cannot select them in any fields, and you are finished with this procedure.

If any of the selected items have at least one child lookup table item, a message box with the following question appears:

**Are you sure you want to inactivate these items?**

4. Do one of the following actions:
  - a. If you want to inactivate the items and all of their child items, select the check-box next to **Include all child items** and click **ok**.
  - b. If you want to inactivate just the items within the selected node but not any of the child items, click **ok**.
  - c. If you decide not to inactivate any items, click **cancel**.

Inactivated items are indicated in the **Active** column of the lookup item list.

**Important:** Record the status of all inactive items so that this information is available if you ever need to convert your TeamConnect data.

### Points To Remember

Consider the following points before inactivating lookup table items:

- If an inactive value is assigned to a field via templates, rules, or the XML layer, an exception will be thrown to prevent the creation or update of the record. For example, if a template specifies an item as the default value of a field to be set on Create, but the item is inactivated, the record cannot be created.
- You cannot inactivate lookup table items used by required custom fields or custom fields that set a default value. If you attempt to do so, you will receive an error.
- Inactivated items are visible to end users in records if they were previously added to a record. Depending on the circumstances, they may be displayed in gray. If accessibility settings are enabled, **(inactive)** is appended to inactive item names and they are not displayed in gray.
- Users can specify inactivated items as search criteria and report qualifiers. Inactivated items appear in gray or **(inactive)** is appended to the item names.

- Existing records with inactive items retain the inactive values. Users can edit records with previously selected inactivated items and click **save**.
- If a user changes the value of a field that specified an inactive item and then clicks **save**, they cannot revert to the prior inactive value unless a TeamConnect solution developer temporarily activates the item.
- If you ever need to convert your TeamConnect data, you must first activate all inactive categories and lookup table items. Once the data conversion is completed, you can restore the inactive items manually.
- When you inactivate lookup table items, their status is recorded in the **Active** column of the lookup table screen.
- If an item has parent or child items, you cannot readily determine their active status without navigating to each level in the hierarchy. Therefore, you should record the status of all inactive items so that this information is readily available if you ever need to convert your TeamConnect data. Otherwise, you will have to manually search for any inactive items.

#### 3.1.5.2.6 Activating Lookup Table Items

You can activate previously inactivated lookup table items so that users can select them from the corresponding fields in the user interface.

#### To activate lookup table items

1. On the **System Lookup Tables** screen, select the appropriate lookup table in the **Show items belonging to** drop-down list.

If the lookup table has a hierarchical tree structure, select the appropriate node in the **Show items in node** drop-down list.

2. Select the check-box next to each item you would like to activate.
3. Click **activate**.

If the items do not have any child lookup table items, the selected items are activated, users can select them from corresponding fields, and you are finished with this procedure.

If any of the selected items have at least one child lookup table item, a message box with the following question appears:

#### **Are you sure you want to activate these items?**

4. Do one of the following actions:
  - a. If you want to activate the items and all of their child items, select the check-box next to **Include all child items** and click **ok**.
  - b. If you want to activate just the items within the selected node but not any of the child items, click **ok**.
  - c. If you decide not to activate any items, click **cancel**.

Activated items are indicated in the **Active** column of the lookup item list.



**Important:** Update the status of all activated lookup table items so that this information is available if you ever need to do any kind of conversion of your TeamConnect data.

#### 3.1.5.2.7 Deleting Lookup Table Items

System lookup tables themselves cannot be deleted. However, you can delete items in a system lookup table if they are not needed. TeamConnect automatically checks to make sure that each item is not being referenced by records. If an item is being used, you are prompted to confirm its deletion.

You must consider the impact of deleting lookup table items—especially on an existing system that has records using the item. If the tree position of the item is used by rules or reports, you must consider the impact of deleting it and make any necessary modifications. You may be able to inactivate or rename the item rather than delete it.

**Important:** If you delete an item, all record data referencing it is lost. If you delete a node, all of its sub-levels and their data entries are also removed.

#### To delete an item from a system lookup table

1. On the **System Lookup Tables** screen, select the appropriate lookup table in the **Show items belonging to** drop-down list.

If the lookup table has a hierarchical tree structure, another drop-down list appears, **Show items in node**. Select the appropriate node in the **Show items in node** drop-down list.

2. Select the check-box next to each item in the list that you would like to delete.
3. Click **delete**.

The items you deleted are immediately removed from the lookup item list and all screens where the lookup table is accessible in the end-user interface.

#### 3.1.5.2.8 Custom Lookup Tables

Like system lookup tables, custom lookup tables can be hierarchical in nature, automatically start with a Root node, and appear in the order specified for each item in the table.

Each custom lookup table is identified by its four-character alphanumeric unique code, which is assigned when you create the table.

Each item in a custom lookup table is assigned a *partial tree position*, a four-character alphanumeric combination, when you add it to the table.

To identify the option selected as a value in a custom field of type List when using the API, XML Layer, or other back end functionality, you must use the *full tree position* of the corresponding item in the custom lookup table associated with the field.

It starts with the unique code of the table + ROOT and includes the partial tree position of the parent items (if there are any) and the selected item itself, in UPPERCASE and separated by underscores, for example, OCCU\_ROOT\_HURE\_HRAS.

## 3.1.5.2.8.1 Viewing Custom Lookup Tables

The Custom Lookup Tables screen allows you to view existing custom lookup tables. If you need to create new ones, use this procedure to access the screen, and then see [Adding Custom Lookup Tables](#).

**To open the custom lookup tables screen**

1. In the Designer window, in the **Go to** drop-down list, select **Lookup Tables**.
2. Click the **Custom** tab.

The **Custom Lookup Tables** screen appears.

3. Select the appropriate user-defined lookup table in the **Show items belonging to** drop-down list.
4. Select the appropriate node in the **in node** drop-down list, if necessary.

A list of items available for the selected lookup table or its selected node appears below the drop-down lists.

The following table describes the items in the custom lookup tables screen.

**Custom Lookup Tables Screen**

Field or button	Description
<b>New lookup table name</b>	<p><b>Important:</b> Use this field only if you want to create a lookup table.</p> <p>Enter a name for a new custom lookup table.</p> <p>Maximum length is 50 alphanumeric characters. However, Data Warehouse only supports a maximum of 35 characters.</p>
<b>Unique Code</b>	<p><b>Important:</b> Use this field only if you want to create a lookup table.</p> <p>Enter a four-character alphanumeric code that is unique across all custom lookup tables.</p> <p><b>Note:</b> You can enter the unique code in uppercase or lowercase, however, in the database, it is saved in uppercase.</p>
<b>Add Table</b>	Click to add the new table to the custom lookup tables listed in the <b>Show items belonging to</b> drop-down list.
<b>Show items belonging to</b>	Select the custom lookup table where you want to view, add, edit, or delete items.

<b>in node</b>	Select the node (hierarchy level), if necessary, to locate the appropriate item or items.
<b>Delete Table</b>	Click to delete the selected table in the <b>Show items belonging to</b> drop-down list. See <a href="#">Deleting Custom Lookup Tables</a> .
<b>Item Name</b>	Enter a name for the option as it should be displayed in the field. Maximum length is 50 alphanumeric characters.
<b>Order</b>	Enter an integer, indicating the <b>Order</b> for the item in the list. The items with the same <b>Order</b> are sorted alphabetically and displayed accordingly.  <i><b>Tip:</b> It may be helpful to enter orders in increments of five. If you ever have to add more options, you will have some leeway as to where in the list you can place the additional items.</i>
<b>Tree Position</b>	Enter a four-character alphanumeric combination that uniquely identifies the lookup item within the node in which you are adding it.  <i><b>Note:</b> You can enter the tree position in uppercase or lowercase, however, in the database, it is saved in uppercase.</i>
<b>Is Active</b>	<ul style="list-style-type: none"> <li>• <b>YES</b>—The lookup table item is active</li> <li>• <b>NO</b>—The lookup table item is inactive and cannot be selected by end users. For more details, see <a href="#">Inactivating Custom Lookup Table Items</a>.</li> </ul>
<b>Lookup Item List</b>	Displays a list of lookup items in the selected node of the table.  Select or clear the corresponding check-boxes on the left to delete or edit lookup items.

#### 3.1.5.2.8.2 Adding Custom Lookup Tables

You can add as many custom lookup tables for custom fields of type List as you need. Remember, however, that you can re-use the same table for multiple custom fields, for example, if you need to add a field with the same options under different categories or even in different object definitions.

***Tip:** Always make sure that the names you assign to your custom tables are self-explanatory and indicative of the items they contain or for which fields they are created.*

#### To add a new custom lookup table

1. On the **Custom Lookup Tables** screen, in the **New lookup table name** field, enter a name for the table.  
The table name can be 50 characters maximum in length.
2. In the **Unique Code** field, type four alphanumeric characters for a unique identifier for the table.
3. Click **Add Table**.  
The newly added table appears alphabetically in the **Show items belonging to** drop-down list.
4. Add the desired items to the table. For details, see [Adding Items to Custom Lookup Tables](#).  
The custom lookup table is created and available for use in custom fields of type List.

#### 3.1.5.2.8.3 Deleting Custom Lookup Tables

You can delete custom lookup tables that are no longer necessary for custom fields. However, you will not be able to delete a custom lookup table if it is used by a custom field or if any of its items is used in a project auto naming pattern in custom object definitions.

##### To delete a custom lookup table

1. On the **Custom Lookup Tables** screen, select the custom lookup table you would like to delete in the **Show items belonging to** drop-down list.
2. Click **Delete Table**.
3. When asked to confirm your action, click **OK**.

If the table is already being used in records, you will receive the appropriate message, otherwise the table with all of its items is deleted from the database.

#### 3.1.5.2.8.4 Adding Items to Custom Lookup Tables

You need to add items to custom lookup tables to maintain the appropriate list of selections for your users, according to your business needs.

##### To add new items to a custom lookup table

1. On the **Custom Lookup Tables** screen, in the **Show items belonging to** drop-down list, select the custom lookup table to which you need to add items.
2. In the **in node** drop-down list, select the appropriate level where the items should be.
3. Select the **Number of entries you would like to add**.
4. For each data entry row, enter the following information:
  - Name of the item as it should be displayed in the field.

- Its display order.
  - A four-character alphanumeric combination that is unique in the selected node.
  - If necessary, for more specific details, see [the Custom Lookup Tables Screen table](#).
5. Click **add more**.

The newly added item appears in the lookup item list.

#### 3.1.5.2.8.5 Editing Custom Lookup Table Items

You can edit custom lookup table contents when you want to display them in a different order in the user interface, or if you want to change the name that appears for each item.

You cannot edit the tree position of a lookup item. During the design stage, when no records have been created yet, you can add a new item and then delete the item that has the incorrect tree position.

#### To edit an item in a custom lookup table

1. On the **Custom Lookup Tables** screen, in the **Show items belonging to** drop-down list, select the custom lookup table where you need to edit items.
2. In the **In Node** drop-down list, select the appropriate level where you want to make changes.
3. Select the check-box next to each item you need to change.
4. Click **Edit**.

The corresponding fields of the selected items become editable.

5. Make the necessary changes to the **Item Name** and **Order** of the selected items.
6. Click **OK**.

Your changes are reflected in the lookup item list.

#### 3.1.5.2.8.6 Inactivating Custom Lookup Table Items

You can inactivate lookup table items so that users are prevented from selecting them from the corresponding fields in the user interface. Inactivating items allows you to prevent the use of unwanted values yet maintain existing data in records that include inactive items.

**Important:** Before inactivating lookup table items, read [Points To Remember](#).

#### To inactivate lookup table items

1. On the **Custom Lookup Tables** screen, select the appropriate lookup table in the **Show items belonging to** drop-down list.

If the lookup table has a hierarchical tree structure, select the appropriate node in the **Show items in node** drop-down list.

2. Select the check-box next to each item you would like to deactivate.
3. Click **Inactivate**.

If the items do not have any child lookup table items, the selected items are deactivated, users cannot select them in any fields, and you are finished with this procedure.

If any of the selected items have at least one child lookup table item, a message box with the following question appears:

**Are you sure you want to deactivate this item?**

4. Do one of the following actions:
  - a. If you want to deactivate the items and all of their child items, select the check-box next to **Include all child items** and click **ok**.
  - b. If you want to deactivate just the items within the selected node but not any of the child items, clear the check-box next to **Include all child items** and click **ok**.
  - c. If you decide not to deactivate any items, click **Cancel**.

Inactivated items are indicated in the **Active** column of the lookup item list.

**Important:** Record the status of all inactive items so that this information is available if you ever need to convert your TeamConnect data.

## Points To Remember

Consider the following points before inactivating lookup table items:

- If an inactive value is assigned to a field via templates, rules, or the XML layer, an exception will be thrown to prevent the creation or update of the record. For example, if a template specifies an item as the default value of a field to be set on Create, but the item is inactivated, the record cannot be created.
- You cannot deactivate lookup table items used by required custom fields or custom fields that set a default value. If you attempt to do so, you will receive an error.
- Inactivated items are visible to end users in records if they were previously added to a record. Depending on the circumstances, they may be displayed in gray. If accessibility settings are enabled, **(inactive)** is appended to inactive item names and they are not displayed in gray.
- Users can specify inactivated items as search criteria and report qualifiers. Inactivated items appear in gray or **(inactive)** is appended to the item names.
- Existing records with inactive items retain the inactive values. Users can edit records with previously selected inactivated items and click **save**.
- If a user changes the value of a field that specified an inactive item and then clicks **save**, they cannot revert to the prior inactive value unless a TeamConnect solution developer temporarily activates the item.

- If you ever need to convert your TeamConnect data, you must first activate all inactive categories and lookup table items. Once the data conversion is completed, you can restore the inactive items manually.
- When you inactivate lookup table items, their status is recorded in the **Active** column of the lookup table screen.

If an item has parent or child items, you cannot readily determine their active status without navigating to each level in the hierarchy. Therefore, you should record the status of all inactive items so that this information is readily available if you ever need to convert your TeamConnect data. Otherwise, you will have to manually search for any inactive items.

#### 3.1.5.2.8.7 Activating Custom Lookup Table Items

You can activate previously inactivated lookup table items so that users can select them from the corresponding fields in the user interface.

##### To activate lookup table items

1. On the **Custom Lookup Tables** screen, select the appropriate lookup table in the **Show items belonging to** drop-down list.

If the lookup table has a hierarchical tree structure, select the appropriate node in the **Show items in node** drop-down list.

2. Select the check-box next to each item you want to activate.
3. Click **activate**.

If the items do not have any child lookup table items, the selected items are activated, users can select them from corresponding fields, and you are finished with this procedure.

If any of the selected items has at least one child lookup table item, a message box with the following question appears:

**Are you sure you want to activate this item?**

4. Do one of the following actions:
  - a. If you want to activate the items and all of their child items, select the check-box next to **Include all child items** and click **ok**.
  - b. If you want to activate just the items within the selected node but not any of the child items, clear the check-box next to **Include all child items** and click **ok**.
  - c. If you decide not to activate any items, click **cancel**.

Activated items are indicated in the **Active** column of the lookup item list.

**Important:** Update the status of all activated lookup table items so that this information is available if you ever need to convert your TeamConnect data.

#### 3.1.5.2.8.8 Deleting Custom Lookup Table Items

You can delete items in a custom lookup table when they are no longer needed. For example, in the design stage of implementation, you might need to clean up the contents of a custom lookup table to have the associated field display the appropriate selections.

TeamConnect automatically checks to make sure that each lookup item is not being used in records. If an item is being used as data in a record, you are notified and asked to confirm your decision to delete it.

**Caution:** *Deleting a lookup item also deletes the item's data entries in records, all of its child items, and the child items' data entries.*

#### To delete an item from a custom lookup table

1. On the **Custom Lookup Tables** screen, in the **Show items belonging to** drop-down list, select the custom lookup table where you need to delete items.
2. In the **in node** drop-down list, select the appropriate level where you want to make changes.
3. Select the check-box next to each item you want to delete.
4. Click **delete**.
5. When asked to confirm your action, click **OK**.
6. If the item or one of its child items is being used as data in records, another confirmation is required. Click **OK**.

The item and all of its children, if there are any, are deleted along with their record entries.

### 3.1.6 Creating Custom Pages

You may define the following components of TeamConnect custom pages to determine the appearance of your organization's record pages:

- **Custom fields**—Fields that you create for specific categories of system and custom objects to meet your organization's needs. You specify the label that appears in the end-user interface and what type of value the field holds, such as Memo Text or Check-Box. You may add custom fields to custom blocks.
- **Blocks**—A group of system and custom fields included into a single XML file for custom blocks or Java Server Page (JSP) file for system blocks. You may add blocks to object views and wizard pages.
- **Object views**—Layout of record pages, blocks, and other user interface elements on the page that you may customize and assign to different user groups. You may create multiple object views for each object definition, so different groups see only the information relevant to them.
- **Custom tools**—You may create tools specific to your organization to help automate a task. You may define how the page appears and assign rights to the user groups that need the tool. For details on how to create custom tools, see [Creating Custom Tools](#).

TeamConnect provides system fields and default object views for all system objects. When you create a custom object, it includes a predefined object view. Depending on the needs of your



organization, creating custom blocks and object views may give you more flexibility in page layout and greater control of user access to record information.

Use the Screen Designer tool to create blocks.

**Note:** *The Designer user interface refers to most interface windows in the browser as "screens." The end-user interface refers to its own interface windows as "pages." This document refers to Designer interface windows as "screens" and the end-user equivalents as "pages."*

**Important:** *Keep in mind that the Designer user interface may often refer to end-user interface windows (pages) as "screens."*

This chapter details the following sequence of creating custom pages:

- [Creating Custom Fields](#)
- [Creating Object Views](#)

### 3.1.6.1 Custom and System Views

Each object definition in TeamConnect has its own system fields organized into system blocks and system views. By default, system views display generic information to all user groups—unless you create and assign custom views.

The following table outlines the major differences between custom and system views in object definitions.

**System vs. Custom Views**

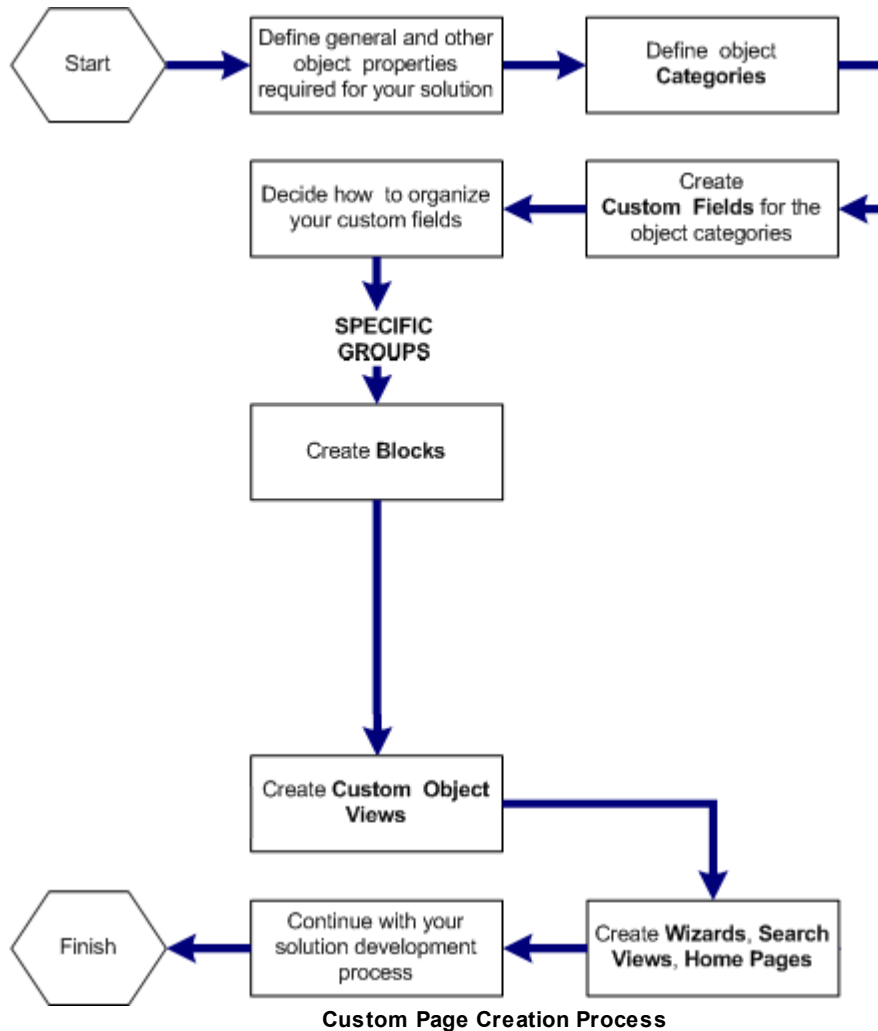
Custom views	System views
You define your own object views.	You accept the predefined object views.
The page layout is alterable. For example, you can reposition pages or blocks.	The page layout is fixed.
You may use custom fields with the organization-specific information to create blocks.	Do not use custom fields with the organization-specific information to create blocks.
You may restrict access to your custom information by creating different object views for different groups using different blocks.	You may restrict access to your custom information only by category, through assigning different rights. Otherwise, all users see the same blocks.

While system views may meet your organization's needs for certain system objects, such as Appointment or Contact, you probably need to create custom views for custom objects.

### 3.1.6.2 Process Overview

The following flowchart is a snippet of the general customization process. It details the steps and decisions you have to make while organizing your custom fields for display in record pages.

When your custom fields are organized, you may proceed to creating the rest of custom pages, such as wizards, search views, and home pages.



### 3.1.6.3 Custom Fields

You may create custom fields to capture information specific to your organization's business needs. The default installation of TeamConnect is configured to capture generic information that most organizations need, such as record name, number, date when created, user who created it, and so on. However, your organization might need to include information that is specific to your industry and business methods.

For example, if your organization shipped packages, you might need to store the number of packages shipped to a particular client, the total weight of the packages, and the type of delivery. The default installation of TeamConnect does not include such fields, but you could create custom fields for the purpose. To define each of these fields, you need to know the following information:

- What kind of information it should contain so that you can label it accordingly, such as **Total Shipping Weight**.
- What field type it must be to store the necessary information properly, such as Number, List, or Text fields.

After you enter this information on the **Custom Fields** tab of the appropriate object definition, the users with the appropriate rights see the fields on their pages, as shown in the following images.

Object Definition: Contact

General | Categories | **Custom Fields** | Forms | Blocks | Object Views | Search Views | Rules | Templates | Wizards | History | Duplicate Contact Manager |

Custom Fields

For Category: Contact

Number of entries you would like to add: 1

Field Name	Label	Field Type	Include in Data Warehouse	Is Required ?	Default Value
1 Affiliation	Affiliation	List	(Select)	(Select)	

+ add more

Custom Fields in the Designer Interface

General Information

Name:

\*Matter ID: (Id will be automatically generated)

Opened On:

Details

\*Time Period: Weekly

\*Total Billable Hours per day: 8

\*Days per week: 5

\*Timekeeper Category:

Task Category: (Select)

There were errors parsing the 'CjbBdcSettingSYS.xml' block. Please see the error log for further information.

Example: Custom Fields in the End-User Interface

To have your custom fields displayed on the page, you have to include them in blocks. These elements provide a layout for custom fields on the page and allow you to add certain formatting attributes to their tags.

### 3.1.6.3.1 Where To Use Custom Fields

You create custom fields to capture and display your organization-specific information and then assign the appropriate object views. The following table lists the areas and features that use custom fields to capture information or to implement the desired workflow. The table also indicates where to find additional information.

**Where Custom Fields May Be Used in TeamConnect**

Feature	Description	For details, see
<b>Wizards</b>	Custom fields may be added as individual components or within blocks to gather initial information when a record is being created	<a href="#">Creating Wizards</a>
<b>Search Views</b>	Custom fields may be used as search criteria and search results display keys	<a href="#">Creating Search Views</a>
<b>Rules</b>	Custom fields may be used as qualifiers in user interface rules, as well as in actions in custom action rules	<a href="#">Using Rules</a>
<b>Auto Generated Letters and Documents</b>	Custom field values may be used by the Document Generator to automatically create letters and other documents	<a href="#">Administrator Help</a>
<b>XML Layer</b>	Custom field values may be used for data conversion and integration with other applications	<i>XML Layer Reference</i>

## Points To Remember

Consider the following points about custom fields:

- Custom fields are category-specific. That is, they can be created and accessed only through their respective categories.
- Access and display of custom fields is controlled by the user's rights to each category and its respective custom fields and whether the category is active.
- If you inactivate categories, users are prevented from selecting them from the corresponding object's records and the associated blocks of custom fields are not displayed. For more information about inactivating categories, see [Inactivating Object Categories](#).
- To reference a custom field, in most cases, you need to know its field name (not label), the full tree position of the category under which it is created, and the field type.
- In the database, object model, XML Layer, and API, tables, tags, and interfaces that deal with custom field definition or value information typically have the word "Detail" as part of their name

(for example, Y\_OBJ\_DETAIL\_FIELD, <Detail/>, or EnterpriseEntity). Hence, you may also come across references to custom fields as "detail fields."

3.1.6.3.2 Custom Fields Tab in Object Definitions

All custom fields for an object may be created, viewed, modified, or deleted using the **Custom Fields** tab of the corresponding object definition screen.

**Custom Fields**

For Category:

Number of entries you would like to add:

	Field Name	Label	Field Type	Include in Data Warehouse	Is Required ?	Default Value
1	<input type="text"/>	<input type="text"/>	(Select) <input type="text"/>	(Select) <input type="text"/>	(Select) <input type="text"/>	
<a href="#">+ add more</a>						
	Field Name	Label	Field Type	Include in Data Warehouse	Is Required ?	Default Value
1	<input type="checkbox"/> Action	Action	List (Actions)	NO	NO	(None)
2	<input type="checkbox"/> CaseAge	Case Age	Number Field	YES	NO	No default value
3	<input type="checkbox"/> Cause	Casue	Text	YES	YES	No default value
4	<input type="checkbox"/> Severity	Severity	List (Severity Level)	YES	NO	(None)

[Check All](#) - [Uncheck All](#)

Custom Fields Tab

The following table describes the fields on the **Custom Fields** tab.

Custom Fields Tab

Field	Description
<b>For Category</b>	<p>Select the category to which you would like to add custom fields.</p> <p>All custom fields are category-specific, which means they can be created and viewed by category only.</p> <p>Categories are organized in a hierarchical way in object definitions. Only the top five levels of the hierarchy (inclusive of the first node) can be used when creating custom fields.</p> <p>In system objects, this node is known as Root, while in custom objects this node has the same name as the object itself—such as Dispute for the Dispute object.</p>
<b>Field Name</b>	<p>Enter an alphanumeric name to uniquely identify the field. This name is used to reference the custom field in blocks, letters, XML Layer, and reports.</p>

	<p>Field names must be unique within each category and use specific naming conventions. See <a href="#">Custom Field Name Requirements</a>.</p> <p>The field name is only editable when a custom field is being defined for the first time. It can never be modified after the field is created.</p> <p><i>Tip: Try to spell out the entire word rather than making abbreviations so that other solution developers can easily understand exactly what the field is created for.</i></p>
<b>Label</b>	<p>Type the field label as you would like it to be displayed on the page. If the field will be included in Data Warehouse, the field label must not exceed 35 characters. Otherwise, the field label can include up to 50 characters.</p> <p><i>Tip: If you need to use different labels for the same field in different blocks (for example, intended for different groups of users), create more custom fields of the type Label Only in the <a href="#">Custom Field Types table</a>.</i></p>
<b>Field Type</b>	<p>Select an option from the drop-down list to specify the type of custom field. For more information about each type, see <a href="#">Field Types</a>.</p>
<b>Include in Data Warehouse</b>	<p>Determines whether a field will be processed by Data Warehouse. Select <b>YES</b> if you are going to run reports and you want to include the information entered in the custom field for your reports. For information on Data Warehouse, see <a href="#">Data Warehouse Requirements for Custom Fields</a> and Using Data Warehouse.</p>
<b>Is Required?</b>	<p>Select <b>YES</b> to make the custom field required for everybody. If you need to make this field required only to certain users, select <b>NO</b> and create the appropriate rules. For more information, see <a href="#">Required Fields</a>.</p> <p>By default, required fields are indicated by an asterisk on the page.</p>
<b>Default Value</b>	<p>(Optional) Enter or select a value to be displayed as default for the custom field on the page. For example, a check-box can be displayed as either checked or unchecked.</p>
<b>Exclude from Custom Search</b>	<p>Choose <b>NO</b> (the default) or <b>YES</b>. If you choose YES, this field will not appear to the end user in custom search screen dropdown lists for criteria or result fields.</p> <p>Specifying Yes can help you simplify searches for the end user by making dropdown lists shorter and less confusing.</p>

	<p><b>Note:</b> A Yes value causes the field to disappear from the dynamically-generated section of the dropdown list that starts below category names. In the uncommon case that you explicitly define this field as a criterion or result field in a custom search template, then the field will still appear in the dropdown list, in the section above the category names, where all the explicitly defined fields are located.</p>
<p><b>Custom Field List</b></p>	<p>Displays the custom fields of the selected object. Select or clear the corresponding check-boxes next to the field and click <b>edit</b> or <b>delete</b> to change or delete the appropriate custom fields.</p>

#### 3.1.6.3.3 Custom Field Name Requirements

Custom field names must uniquely identify their custom fields and are used in a number of back-end operations to reference those fields.

Aside from making a field name, make sure that each field name meets the following conditions:

- A unique alphanumeric combination
- Unique within its specified category
- Begins with a letter
- Does not include spaces, punctuation, or special characters, that is, a field name must be created from A through Z, a through z, and 0 through 9
- Is no more than 30 characters long, unless it is of the type **Involved**, then it must be no more than 26 characters long
- Is not a reserved word in your database server. (There are dozens of such words; refer to your database documentation for a list.)

#### 3.1.6.3.4 Data Warehouse Requirements for Custom Fields

This section explains Data Warehouse's requirements for custom fields.

The **Include in Data Warehouse** option adds a custom field to a Data Warehouse table for use in reports. All the values of custom fields of a single category that are specified to be included in Data Warehouse are stored in one Data Warehouse table. When you design categories and their reportable fields, you must not exceed Data Warehouse's limits.

Data Warehouse requirements vary depending on whether your system uses an MS SQL Server or an Oracle database. Unless stated otherwise, the following requirements apply to both MS SQL Server and Oracle:

- A category name must be unique within its object if it has any custom fields included in Data Warehouse.

If a category name is NOT unique within its object, all the corresponding Data Warehouse tables will be created. However, Business Objects create only one class for all the Data Warehouse

tables that have the same category name. If the objects within that class mix data from the Data Warehouse tables that have the same category name, the report may include incorrect data.

- Field labels must be unique within a category.

If a field label is NOT unique within its category, the corresponding Data Warehouse table will be created. However, Business Objects will add only one column to the corresponding class in the universe for all the non-unique field labels, and the report may include incorrect information because Business Objects use the data from only one of the fields that has a non-unique field label within one category.

- If a custom field is to be included in Data Warehouse, its field label must NOT exceed 35 characters.

If a field label exceeds 35 characters, its Data Warehouse table can be created, but Business Objects may not create the necessary object, so reports cannot include the custom field.

The TeamConnect limit for field labels is 50 characters, regardless of whether the field is included in Data Warehouse.

- For objects of the type **Involved**, the length of custom field names must NOT exceed 26 characters.

TeamConnect limits the length of custom field names to 30 characters. However, Data Warehouse scripts add a four-character suffix to the custom field names of objects of the type Involved, so the additional four characters could cause field names to exceed the 30-character limit. If the limit is exceeded, creation of the Data Warehouse table fails and an error is logged in the Data Warehouse log file.

- Custom field names must NOT equal any of the words that are reserved for use by your database server. Refer to your database documentation regarding column names for a list of those words.

## Oracle Limitations

Data Warehouse tables in Oracle support a maximum of 1000 columns. They do not have a per-row data length limit. That is, in a single category, you can include up to 1000 custom fields in Data Warehouse, regardless of the field type. You must design your categories and reportable fields so that the 1000 column limit is not exceeded.

## MS SQL Server Limitations

In MS SQL Server, Data Warehouse tables support a maximum of 8060 bytes per row. You should design your categories and reportable fields so that the 8060 byte limit cannot be exceeded. The following table lists the bytes allocated per column for each field type. These amounts are subtracted from MS SQL Server's maximum allocation of 8060 bytes per row.

For example, if there are 30 custom fields of the type **Text** within a single category, the Data Warehouse table has 30 columns that use 250 bytes each. Altogether, 30 custom fields of the type **Text** use 7500 bytes of the 8060-byte allocation, which only leaves 560 bytes for other field types in that category.

**Bytes Allocated for each Field Type (MS SQL Only)**

Field type	Bytes allocated
------------	-----------------



<b>Check-Box</b>	4 bytes
<b>Custom object/ Involved</b>	4 bytes
<b>Date</b>	8 bytes
<b>List type</b>	50 bytes
<b>Memo</b>	16 bytes  <i><b>Note:</b> The 16 bytes is for a pointer to the TeamConnect table with the original memo text data, which can contain up to 1 GB in MS SQL Server.</i>
<b>Number</b>	8 bytes
<b>Text</b>	250 bytes

**Important:** When run with MS SQL Server, Data Warehouse has a 250-byte limit for custom fields of the type **Text**. If a user enters data in excess of this 250-byte limitation, the Data Warehouse scripts truncate the text when populating the table.

TeamConnect limits data entry in custom fields of the type **Text** to 2000 characters. However, when the Data Warehouse scripts are run with MS SQL Server, any text that exceeds the 250-byte limitation is truncated.

In addition, some Non-ASCII characters use two bytes per character of the 250-byte data length limitation. To make sure that users do not exceed the 250-byte limit, which results in truncated data in reports, choose one or more of the following options:

- Use memo text fields instead of custom fields of the type **Text** if users might exceed the 250-byte limit.
- Develop rules in TeamConnect to enforce the 250-byte limit for each custom field of the type **Text**.

Data Warehouse tables in MS SQL Server support a maximum of 1024 columns as long as the number of bytes for all fields in the category does not exceed the 8060 byte limit. For example, if a category has 1000 check-boxes and 24 memo text fields included in Data Warehouse, MS SQL Server supports all of the fields because the total allocation is less than 8060:

$$(1000 * 4 \text{ bytes}) + (24 * 16 \text{ bytes}) = 4384 \text{ bytes}$$

MS SQL Server does not limit the creation of Data Warehouse tables with a large number of columns, although it displays a warning in the Data Warehouse log file about the 8060 byte limit. If a user attempts to insert data into a table in excess of the 8060 byte limit, MS SQL Server displays an error, the system cannot access the Data Warehouse table, and the data cannot be transferred to the Data Warehouse table.

For more information on Data Warehouse, see the Data Warehouse Help.

#### 3.1.6.3.5 Required Fields

A field is considered required if the user cannot save a record without entering a value in that field. If a user attempts to save a record without populating a required field, TeamConnect displays a message that the field is required.

By default, the labels of all required system fields and custom fields (set as required when they were created) appear in boldface with an asterisk next to them, in the end-user interface. However, fields are not automatically displayed this way if they are required as a result of rules, or if another field uses its value. It is easier to require such fields when creating them than to modify the way they appear.

**\*Maximum Search results:**

**Example: Required Field in End-User Interface**

## How To Make Fields Required

You may make a field required in one of the following ways:

- By creating a custom field in an object definition and setting it as required for everybody. These fields are automatically displayed as required.
- By creating rules to make system or custom fields required under certain conditions. These fields are not automatically displayed as required.

Fields may also become required (though not marked as such in the user interface) as a result of certain design decisions that you may make. For example:

- If you add an object attribute to the unique identifier (ID) pattern in custom object definitions, the corresponding field becomes required.
- If you make a custom object definition contact-centric, the corresponding contact field becomes required.
- If you set a parent custom object definition as required, the corresponding parent project field in all child projects becomes required.

## Practical Tips

Keep these tips in mind while working with required fields:


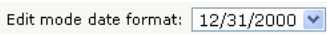
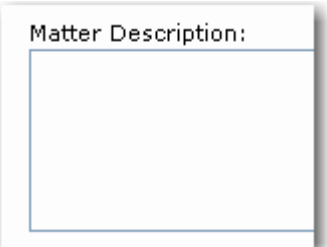
- If you include attributes in a naming pattern for custom object records, consider making the corresponding fields required. This action ensures that these fields are populated and that the resulting descriptions in the **Name** field are more consistent.
- If you make certain fields required in contact records, the users may not be able to add contacts through the contact search module window.
- Use the tag attributes, [notUseRequiredFieldStyle](#) or [irequired](#) to control whether custom field labels appear as required when including them in blocks.

- Make sure you include all required fields into wizards that you may be creating for that object. Otherwise, the user is unable to save the record after using the wizard.


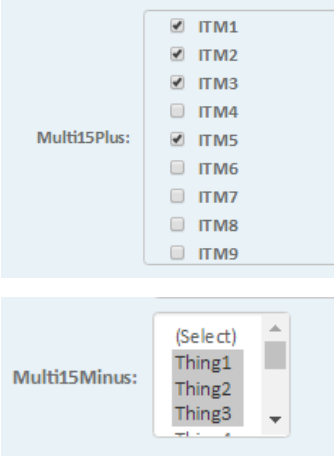

3.1.6.3.6 Field Types

Based on the type of data that each field can store and the field's functionality in the user interface, all system and custom fields in TeamConnect may be one of the seven types listed in the following table.

**Custom Field Types**

Field type	Description	Resulting field in the end-user interface
<b>Check-Box</b>	Stores boolean values, when you need to provide the user with options that might or might not be selected.	
<b>Date</b>	<p>Stores date values that appear in the format specified in system settings and user preferences. By default, each date field provides the user with a pop-up calendar window that appears when the user clicks in the field.</p> <p>The date and time fields are related. To create a date field with a time field, use the appropriate tag attribute with this field when including it in a form or block file. See <a href="#">Date Field Tag Attributes</a>.</p> <p>When creating a custom field of type Date, a checkbox labeled "Time Zone Independent" appears. You should check this box if the field always contains the same value for any user, viewing it in any time zone in the world. For example, a person's birth date would be time-zone-independent, so the box should be checked. However, the date and time of a telephone conference call would be dependent on time zone, so the box should not be checked. Note that you can only make this choice once, when first creating the custom field. You cannot edit the field later and change this attribute.</p>	
<b>Memo Text</b>	<p>Stores large amounts of text, up to:</p> <ul style="list-style-type: none"> <li>• About 1 GB of characters (if using ASCII) in MS SQL Server.</li> <li>• 4 GB of character data in the Oracle database.</li> </ul> <p>To specify the desired display size for a memo text field, use the appropriate tag attribute with this field</p>	

	when including it in a form or block file. See <a href="#">Memo Text Field Tag Attributes</a> .	
<b>Text</b>	<p>One-line field that can store up to 2000 characters in both MSSQL and Oracle databases.</p> <p>To specify the desired display size for a text field or the maximum number of characters it can accept, use the appropriate tag attribute with this field when including it in a form or block file. See <a href="#">Text Field Tag Attributes</a>.</p> <p><i>Tip: For alphanumeric fields, use a Text field.</i></p>	Cookie Domain: <input type="text"/>
<b>Number</b>	<p>Stores fixed and floating-point numbers in the following formats:</p> <ul style="list-style-type: none"> <li>• <b>Decimal</b>—The number of digits allowed after the decimal point depends on the decimal format attribute used. Without the attribute, the field accepts as many digits after the decimal point as needed.</li> <li>• <b>Dollar</b>—The field accepts a maximum of two digits after the decimal point and automatically displays the dollar (\$) sign.</li> <li>• <b>Percent</b>—The field accepts a maximum of three digits before the decimal point and as many digits afterwards as needed. It also automatically displays the percent (%) sign.</li> </ul> <p>Do not enter any special characters, such as \$, %, and so on, as part of default values for this field type. For details on available tag attributes that can be used when including a number field in a form or block file, see <a href="#">Number Field Tag Attributes</a>.</p> <p><i>Tip: For alphanumeric fields, use a Text field.</i></p>	Estimated Hours: <input type="text" value="0"/>
<b>Label Only</b>	<p>The label only field type is available only for custom fields. It allows you to create:</p> <ul style="list-style-type: none"> <li>• <b>Several labels for the same field</b>—Create different labels for the same field and use them in the different blocks. For example, if you need different labels for the same field so that it appears according to different groups of users, you can create a label only field.</li> </ul>	N/A

	<ul style="list-style-type: none"> <li>• <b>Section headings to help organize a page</b> —Create sections for groups of custom fields.</li> </ul> <p>In either case, you are able to change the section heading or field label in one place, by using a label only field, instead of modifying each individual XML file. It is far more efficient and leaves little room for human error.</p> <p>This field is created like any other custom field. For details, see <a href="#">Creating Custom Fields</a>.</p>	
<p><b>List</b></p>	<p>Displays a list of options from which the user can select one, which are provided by a custom lookup table associated with the field.</p> <p>For more information, see <a href="#">List Field Type</a> and <a href="#">Adding Custom Lookup Tables</a>.</p> <p>The example shows a drop-down list. Lists may also be option (radio) buttons.</p>	<p>Status: </p>
<p><b>Multi-value List</b></p>	<p>Displays a list of options from which the user can select one or more, which are provided by a custom lookup table associated with the field.</p> <p>For more information, see <a href="#">List Field Type</a> and <a href="#">Adding Custom Lookup Tables</a>.</p>	
<p><b>Search Module</b></p>	<p>Allows the user to search and select the desired contact or project record. Search modules allow users to open a selected record directly from the field.</p> <p>There are two types of custom fields that appear as search module fields in the user interface, of type:</p> <ul style="list-style-type: none"> <li>• <b>Involved</b>—Available only in custom object definitions. For details, see <a href="#">Involved Field Type</a>.</li> <li>• <b>Custom Object</b>—For details, see <a href="#">Custom Object Field Type</a>.</li> </ul>	<p>Contact: <input type="text" value="Finch, Larry"/> </p>

### 3.1.6.3.6.1 List Field Type

Fields of the type List allow the user to select one out of multiple options. In the user interface, they may be displayed as drop-down lists (default display option) or option (radio) buttons.

**Important:** To create a field of type List, you must first create a custom lookup table to associate with the field.

Custom lookup tables are specific to your design and intended to provide lists of options for custom fields of the type List. Typically, each List field has its own custom lookup table that was created for it. For information about custom lookup tables, see [Adding Custom Lookup Tables](#).

Depending on your design, some lookup tables may be re-used in several custom fields—even in different object definitions. For example, a table providing the **Yes/No/Unknown** options for a list of option buttons can be reused for many different fields.

You can change the contents of a custom lookup table by adding, editing, or deleting items to create useful lists that are customized for your design.

## Points To Remember about List Fields

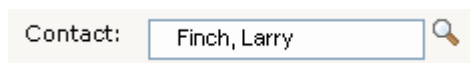
Keep the following points in mind when working with List fields:

- Each List field must be associated with a custom lookup table.
- To create a list of option buttons, create a regular List field and use the `type="radio"` tag attribute when including it in a form or block file. For details, see [List Field Tag Attributes](#).

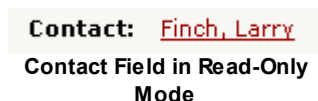
### 3.1.6.3.6.2 Involved Field Type

Create custom fields of the type Involved to allow users to quickly add involved parties, such as claimant, insured, to a project record. Involved field values can also be used in the auto naming patterns for projects.

In the end-user interface, this field appears as a contact search module field in the edit mode, and as a hyperlink in read-only mode. By clicking the hyperlink, the user is linked directly to the selected contact record.



**Contact Field in Edit Mode**



**Contact Field in Read-Only Mode**

## Points To Remember

Keep the following points in mind when working with Involved fields:

- Fields of the type Involved can be created only for custom objects.
- To create a custom field of type Involved, you have to select a specific involved role. This is the role of the contact that the user selects in this field.

When you select this field type on the **Custom Fields** tab of an object definition, another drop-down list appears that lists all roles (categories) from the associated Involved object definition.

- When defining an Involved custom field, you can also associate a specific search view with the field. This search view appears when the user opens the search module for this field to select a contact.
- When the user selects a contact in this field in the end-user interface, the following changes occur:
  - On the **Involved** tab of the project, a new Involved record is automatically added for this contact.
  - On the **Involvement** tab of the selected contact record, an entry is added. This tab lists all projects in which the contact is involved.

### Search Views for Involved Custom Fields

When defining a custom field of type Involved, you have the option of setting a specific search view to be displayed when the user opens the search module for the field. The search views available for you to select are those search views which are marked as available for use in search modules in the Contact Object Definition, including both **Person and Company Search** and **Company Search Only** search views. For details about these settings, see [Defining General Search View Information](#).

The way that the search module for your custom field of type Involved appears to your users depends on your settings as a solution developer:

- If you want users to always use the same search view in the custom field, then select that search view when defining the custom field.
- If you want users to be able to select which search view to use in the search module for the custom field, you must do the following actions:
  - a. Select **None** in the **Search View** field when defining the custom field. This option is selected by default.
  - b. Ensure that each contact search view that you want users to have available in search modules is set accordingly (see [the General Tab on Search View Screens table](#)).

The available search views are listed in the user's search module in the **Current View** drop-down list, sorted according to each search view's **Order**.

- If only one contact search view is available for use in search modules, then the available search view is used automatically in the search module. Also, users do not have the option to select a different search view.

This field is created like any other custom field. For details, see [Creating Custom Fields](#).

#### 3.1.6.3.6.3 Custom Object Field Type

Fields of the type Custom Object allow you to establish a non-hierarchical relationship between the current object definition, which can be system or custom, and another custom object definition. Hence, the name of the field type—Custom Object. This field type allows the user to select a related project record directly in the currently displayed record.

In the end-user interface, Custom Object fields appear as project search module fields or drop-down lists in the edit mode. See [the Custom Field Types table](#) for examples of these field types.

In the read-only mode, Custom Object fields appear as hyperlinks. By clicking the hyperlink, the user is linked directly to the identified record.

**Contact:** [Finch, Larry](#)

**Contact Field with Hyperlink  
in Read-Only Mode**

## Points To Remember

Keep the following points in mind when working with Custom Object fields:

- The Custom Object field type is available when creating custom fields for both system and custom object definitions.
- You cannot select system object definitions or embedded custom object definitions when defining fields of the type Custom Object.
- Whether a Custom Object field appears as a search module or a drop-down list in the end-user interface depends on the **Default Selection Mechanism** specified on the **General** tab of the selected custom object definition.

The display format of the field may also be determined by the display tag attribute in the block file. For details, see [Custom Object Field Tag Attributes](#).

- To limit the available records in a Custom Object field so that a user can select only related records, records in a certain phase, records with a certain default category, and so on, you must add a qualifier in the definition of the field. For more details, see [Qualifiers for Custom Object Fields](#).
- If the field is to be displayed as a search module, you can select a specific search view to be displayed in the search module. Or, you can provide users with the ability to select which search view they want to use in the search module. For more details, see [Search Views for Custom Object Fields](#).
- To specify the desired size for a Custom Object field, you have to use the size tag attribute with this field when including it in a form or block file. For details, see [Custom Object Field Tag Attributes](#).

**Data Entry Fields for Custom Fields of the Type  
Custom Object**

Data Entry Fields for Custom fields of the type Custom Object

Field	Description

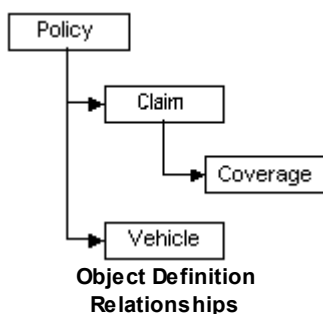


<b>Field Type</b>	Select <b>Custom Object</b> to define a custom field of type Custom Object, which allows users to select a project record belonging to the specified custom object definition.
<b>Custom Object List</b>	Select the custom object definition whose records users must be able to select in this custom field.
<b>Search View</b>	This field is only available if the custom object selected in the Custom Object List has search module set as its default selection mechanism on General tab of its object definition.  Select the search view that you want users to use in the search module for this field. For details about your other options, see <a href="#">Search Views for Custom Object Fields</a> .
<b>Qualifier</b>	If necessary, define an expression in this field to filter the records available to users. You can define the qualifier by doing one of the following actions: <ul style="list-style-type: none"> <li>Type a complete qualifier expression in this field, as described in <a href="#">Qualifiers for Custom Object Fields</a>.</li> <li>Use Object Navigator to help you create the path on each side of the expression, as described in <a href="#">Creating Qualifiers Using Object Navigator</a>.</li> </ul> <p>This field is limited to 256 characters.</p>
<b>Add Path using Object Navigator</b>	Select this check-box to add paths to your qualifier expression with the help of Object Navigator.  When this check-box is selected, several additional fields and buttons appear. For details, see <a href="#">Creating Qualifiers Using Object Navigator</a> .

### To define a custom field of type Custom Object

- In the **Field Type** column on the **Custom Fields** tab of the appropriate object definition, select **Custom Object**.

Several additional fields appear, as shown in the following image.



2. In the drop-down list which lists custom objects, select the custom object whose records the user must be able to select in the custom field.
3. If you want users to always use the same search view in the custom field, select that search view in the Search View drop-down list.
4. If you want users to be able to select which search view to use in the search module window for the custom field, then you must do the following actions:
  - a. Select **None** in the **Search View** field when defining the custom field. This option is selected by default.
  - b. Ensure that each search view that you want users to have available in search modules where they select this type of project is set to be available in search modules (see [the General Tab on Search View Screens table](#)).

The available search views are listed in the user's search module in the **Current View** drop-down list, sorted according to each search view's **Order**.
5. If necessary, create a qualifier to filter the available records from which the user can make a selection. You can create the qualifier in one of two ways:
  - In the **Qualifier** field, manually type the appropriate code to create a complete expression. For more details, see [Qualifiers for Custom Object Fields](#).
  - Select the **Add Path using Object Navigator** check-box. This gives you the option to use Object Navigator for each side of the expression. You will have to manually type the operator in the middle of the expression. For more details and instructions, see [Creating Qualifiers Using Object Navigator](#).

For more details on creating custom fields, see [Creating Custom Fields](#).

If the default selection mechanism of the custom object selected in the definition of your custom field of type Custom Object is search module, you can select a specific search view to be displayed to users in the search module.

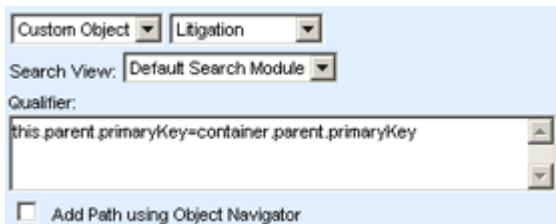
The search views available for you to select are those search views which are marked as available for use in search modules in the Object Definition for the selected custom object. This is an option in the definition of the search view (see [Defining General Search View Information](#)).

The search views available in the search module for your custom field of type Custom Object depend on your settings as a solution developer. You can do either of the following operations:

- Select a specific search view that is always displayed in the search module window.
- Allow users to select from multiple search views in the Current View drop-down list, as they typically do in search pages.

If only one search view for the selected custom object is available for use in search modules, then the available search view is used automatically in the search module. Also, users will not have the option to select a different search view.

When creating custom fields of the type Custom Object, you can define qualifiers to filter records of the custom object definition selected for the field. For example, you might want to limit the user's options to records in a certain phase, with parent records of a specific type, and so on. This is done by writing a qualifier expression in the **Qualifier** field on the **Custom Fields** tab of an object definition, as shown in the following image.



**Data Entry Fields for Custom Fields of the Type Custom Object**

Essentially, in a qualifier expression, you specify a condition to be met in order for records of the selected custom object to be displayed in the custom field. This condition is determined by the business needs of your organization.

Creating a qualifier for a Custom Object custom field is optional; however, it can narrow down the list of records available for selection, which can save users time and prevent invalid selections.

**Important:** *Creating qualifiers for custom fields of the type Custom Object requires a good understanding of the TeamConnect object model and its attributes. Familiarity with the logic of creating qualifiers in search views and rules is also helpful.*

## Points To Remember

The following are the basic points about qualifiers for custom fields of the type Custom Object:

- The code that you provide in the **Qualifier** field must be a complete expression, consisting of two sides that each identify a data value, separated by an operator (such as =). The expression compares two values. For details about the available operators, see [Data Types](#).
- You can create the qualifier either by typing it manually in the **Qualifier** field or with the help of Object Navigator. If you use Object Navigator, you still need to put the various parts together correctly, as described in [Creating Qualifiers Using Object Navigator](#).
- You can use multiple qualifier expressions in the **Qualifier** field by separating them by semicolons (;). They are treated as logical AND statements. In other words, only records that meet the conditions specified by all of the expressions in the **Qualifier** field are returned and available for the user to select.
- When you create qualifiers for custom fields of the type Custom Object in *embedded* object definitions, in the end-user interface, the condition or conditions specified in the qualifier only take effect when the user adds the embedded project and saves its parent first.
- The more complex your qualifier is, the longer it may take for the available records to be displayed when a user wants to make a selection in this custom field.

Whether you decide to create your qualifier by typing it manually or with the help of Object Navigator, you must familiarize yourself with the information in the following sections:

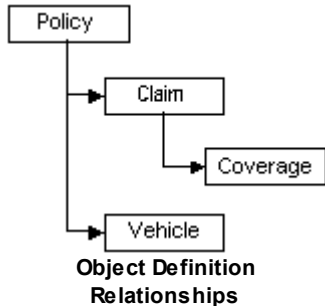
- [Qualifier Syntax and Data Types](#)

- [Creating Qualifiers Using Object Navigator](#)
- [Examples of Qualifiers for Custom Fields of the Type Custom Object](#)

## Example

Let's have a look at a specific example of a qualifier in a field of type Custom Object:

Suppose your design consists of several custom object definitions: Policy, Claim, Coverage, and Vehicle. They have the following relationships: Policy is a parent of Claim and Vehicle, Claim is a parent of Coverage (see the following figure).



In the Coverage object definition, you create a field of type Custom Object and select the Vehicle object in the custom object list ([the Data Entry Fields for Custom Field of the Type Custom Object image](#)). In the end-user interface, this means that Coverage records will have a **Vehicle** field where the user selects a Vehicle record.

If you do not write qualifiers for the **Vehicle** field, all Vehicle records in the database are available for selection in this field. You want to limit the choice of Vehicle records only to those records that have the same Policy parent record, as the Claim record, which is the parent of the Coverage record that has the **Vehicle** field.

The following qualifier limits the choice of Vehicle records as desired, by comparing the primary keys of the parent records:

```
this.parent.primaryKey=container.parent.parent.primaryKey
```

In this qualifier expression:

- Keyword **this** refers to the Vehicle record in the Vehicle field.
- Because Policy is a parent of Vehicle, the parent attribute on the left is the parent Policy record of the Vehicle record.
- Keyword **container** refers to the Coverage record where the Vehicle field is located.
- Because Claim is the parent of Coverage, and Policy is the parent of Claim, the first parent attribute on the right is the parent Claim record of the Coverage record that contains the field, and the second parent attribute is the parent Policy record of the Claim record.

If the parent record of a Vehicle record is the same as the "grandparent" record of the Coverage record, the Vehicle record is available for selection in the Vehicle field.

There are certain conventions which must be followed in order to construct a valid, complete qualifier expression for a custom field of type Custom Object. In order to create your own qualifier, you must understand both the syntax and the possible data types for an expression.

## Syntax

The following points describe the syntax of expressions for qualifiers in custom fields of the type Custom Object:

- The syntax for paths is the same as in the fields where paths are created with Object Navigator, such as rule qualifiers added through the user interface. Namely, the expression you write in the **Qualifier** field essentially consists of object attributes, separated by periods.
- The **Qualifier** field is case-sensitive. Object attributes must be typed exactly as they appear in Object Navigator (see [Using Object Navigator](#)) or using the reference tables in [Object Model: Read This First](#) plus the tables it points you to. Also, custom field names must be typed exactly as they are defined in the **Custom Fields** tab of the object definition.
- There are two keywords that can be used when writing qualifiers. Always start a path with one of these keywords:

- `this`

Refers to the project the user selects in the custom field of type Custom Object. The left side of your expression always starts with this.

- `container`

Refers to the record where the custom field of type Custom Object is located. Typically, the right side of your expression starts with container, unless you need to type in a literal value rather than identify an object attribute.

- System fields are identified with the use of a path that identifies the object attribute, like this:

```
this.parent.parent.primaryKey
```

For complete examples of system fields in paths, see the [Examples of Qualifiers for Custom Fields of the Type Custom Object table](#).

- Custom fields are identified with the use of a path that includes the category tree position and custom field name in parentheses, like this:

```
this.detailList(FullTreePosition).detailNumbValueList(FieldName).detailValue
```

where FullTreePosition is the full tree position of the category under which the custom field is created, and FieldName is the name (not label) of the custom field itself.

**Important:** Make sure to type the full [category tree position](#), NOT the name, of the category.

- When identifying custom field values with a path, you must use the correct path based on the field type, as shown in [the Syntax for Identifying Custom Field Values in Paths table](#). This is because the path changes slightly depending on the type of value you are identifying (number, date, list, and so on), as indicated in **bold** in the table.

For detailed examples of custom fields in paths, see the [Examples of Qualifiers for Custom Fields of the Type Custom Object table](#).

### Syntax for Identifying Custom Field Values in Paths

Custom field type	Path syntax to identify custom field value
<b>Text</b>	<code>detailList (FullTreePosition) .detailTextValueList (FieldName) .detailValue</code>
<b>Memo Text</b>	<code>detailList (FullTreePosition) .detailMemoValueList (FieldName) .detailValue</code>
<b>Number</b>	<code>detailList (FullTreePosition) .detailNumbValueList (FieldName) .detailValue</code>
<b>Date</b>	<code>detailList (FullTreePosition) .detailDateValueList (FieldName) .detailValue</code>
<b>List (lookup items)</b>	<code>detailList (FullTreePosition) .detailObjValueList (FieldName) .detailValue</code>
<b>Custom Object</b>	<code>detailList (FullTreePosition) .detailObjfValueList (FieldName) .detailValueFID</code>  <i><b>Important:</b> detailValueFID is the primary key of the project record selected in the identified custom field.</i>
<b>Check-Box</b>	<code>detailList (FullTreePosition) .detailNumbValueList (FieldName) .detailValue</code>  <i><b>Important:</b> Values for custom fields of the type check-box are stored as numbers, where checked=1 and not checked=0.</i>
<b>Involved</b>	<code>detailList (FullTreePosition) .detailInvlValueList (FieldName) .detailValue.contact.primaryKey</code>  <i><b>Important:</b> This path identifies the primary key of the contact selected in the Involved custom field.</i>

## Data Types

When constructing an expression for a qualifier, you must ensure that the type of data identified by the path on each side of the expression is the same. For example, you cannot compare a name and a number or a primary key, a string and a decimal, a date and a complex type, or object.

The [Data Types and Operators for Custom Fields of the Type Custom Object table](#) lists the types of data that you can use in the **Qualifier** field of a custom field of type Custom Object. It also lists the operators that you can use in the expression for each data type. To find out the data type of a specific attribute, see [Object Model: Read This First](#) plus the additional reference tables this reference points you to.

It is important to understand that each side of your expression must identify one of these simple types of data values. Specifically, you cannot compare attributes marked as "bridges" in [Object Model: Read This First](#) (and the additional reference tables it points to). These values identify other records, rather than simple data values. For example, parent, createdBy, mainAssignee, and project are all bridges rather than simple data values.

Therefore, when you need to compare records, you must use one of the attributes of the record that is a simple data type and that uniquely identifies it. For example, if you want to compare a parent record to another parent or grandparent record, compare their primary keys. Similarly, if you want to compare a user to another user, compare the username (a text value). For other examples, see the [Examples of Qualifiers for Custom Fields of the Type Custom Object table](#).

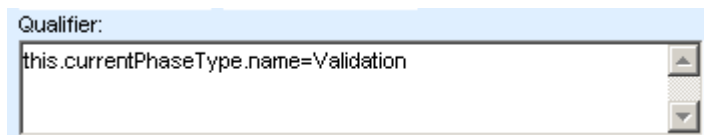
#### Data Types and Operators for Custom fields of the type Custom Object

Data type	Operator	Values	Examples
<b>Text</b>	=	<ul style="list-style-type: none"> <li>Path that identifies a text value</li> <li>A literal text value that you type manually in the <b>Qualifier</b> field</li> </ul>	<pre>this.parent.name=container.name this.currentPhaseType.uniqueCode=LITI</pre>
<b>Number</b>	= < > <= >=	<ul style="list-style-type: none"> <li>Path that identifies a numeric value</li> <li>A literal numeric value that you type manually in the <b>Qualifier</b> field</li> </ul>	<pre>this.parent.primaryKey=container.parent.parent.primaryKey this.parent.primaryKey=container.detailList(CLAM_AUTO).detailObjfValueList(Vehicle).detailValueFID this.detailList(RESE).detailNumbValueList(AppraisalAmt).detailValue&gt;=500000</pre>
<b>Date</b>		<ul style="list-style-type: none"> <li>Path that identifies a date value</li> <li>TODAY (typed manually in the <b>Qualifier</b> field)</li> <li>TODAYX</li> </ul> <p>For dates, &gt; means after today and &lt;</p>	<pre>this.createdOn&gt;=container.createdOn this.closedOn=container.detailList(CLAM_AUTO).detailDateValueList(CompletionDate).detailValue this.createdOn&lt;=TODAY30</pre>

		means before today.	
<b>Lookup table item (from a system or custom lookup table)</b>	=	<ul style="list-style-type: none"> <li>Path that identifies a lookup item selected in a field</li> <li>A full tree position of a lookup item that you type manually in the <b>Qualifier</b> field</li> </ul> <p>For an explanation of how tree positions of lookup items are formed for custom lookup tables, see <a href="#">Custom Lookup Tables</a>.</p>	<code>thisdetailList (CLAM_AUTO).detailObjValueList (LossCode).detailValue=LOSS_ROOT_VDRE</code>
<b>Boolean (true/false values)</b>	=	<ul style="list-style-type: none"> <li>Path that identifies a boolean value</li> <li><code>true</code> or <code>false</code> typed manually in the <b>Qualifier</b> field</li> </ul>	<code>this.isClosed=false</code>

You can use Object Navigator to assist you in constructing the paths on either side of your qualifier expression for a custom field of type Custom Object. This allows you to rely on the internal structure of TeamConnect to guide you in identifying attributes, rather than having to type them.

While Object Navigator helps you construct a path, you must also make the correct selections and type literal text in the **Qualifier** field where appropriate. For example, the following qualifier is constructed using Object Navigator where necessary (`currentPhaseType`), while a specific value also had to be typed into the field (`name.Validation`):



**Qualifier Example**


To take advantage of this option, select the **Add Path using Object Navigator** check-box when defining the custom field, as shown in the following image.



Qualifier:

this.detailList(LIT).detailDateValueList  
 (ServiceDate).detailValue<=container.detailList  
 (CLAM).detailDateValueList(ServiceDate).detailValue

Add Path using Object Navigator

this  reset add


#### Fields for Adding Paths Using Object Navigator

Even if you decide to use Object Navigator to assist you, you must understand the concepts outlined in the following sections:

- [Qualifiers for Custom Object Fields](#)
- [Qualifier Syntax and Data Types](#)
- [Examples of Qualifiers for Custom Fields of the Type Custom Object](#)

The following table describes the fields for adding paths to define data entry fields for custom fields of the type custom object.

#### Fields for Adding Paths Using Object Navigator

Field or button	Description
<b>Add Path using Object Navigator</b>	Select this check-box to add paths to your qualifier expression with the help of Object Navigator. When this check-box is selected, several additional fields and buttons appear.
<b>this/container</b>	Select the appropriate keyword to begin the path you are creating:  <b>this</b> —Select when creating the path on the left side of the expression.  <b>container</b> —Select when creating the path in the right side of the expression, unless you need to type the literal value you are looking for.  These keywords are also described in <a href="#">Syntax</a> .
<b>Object Navigator icon and field</b> 	Click the icon to open Object Navigator and build a path. For details on how to use Object Navigator, see <a href="#">Using Object Navigator</a> .  Once you have created a path and clicked <b>ok</b> in the Object Navigator window, the path appears in the <b>Object Navigator</b> field on the <b>Custom Fields</b> tab.
<b>reset</b>	Click to clear your selections in the fields specific to creating a path with Object Navigator.
<b>add</b>	Click to add the path you have defined using Object Navigator to the <b>Qualifier</b> field.  If you have already added any parts of the expression to the <b>Qualifier</b> field, the path is added to the end of the string.

## Points To Remember

The following points are important to understand about using Object Navigator for defining qualifiers:

- You must begin paths by first selecting either **this** or **container**, depending on from which record your path must originate (see [the Fields for Adding Paths Using Object Navigator table](#)).
- You may not need to use Object Navigator for both sides of your expression. Often, you need to type the particular text or numeric value you may be looking for, such as a phase name or the tree position of a lookup item, on the right side of the expression.
- If both sides of the expression you need to write require you to build a path, you may need to click the **Object Navigator** icon twice—first to build one path and again to build the other path, one on each side of the operator.
- You must manually type an operator after adding the left side of the expression and before adding the right side of the expression. For details, see [Data Types](#).
- When you build a path and click **add**, the path that you have created is added to the **Qualifier** field.

### To build an expression using Object Navigator

1. After specifying the type of the custom field and selecting the desired custom object, select **Add Path using Object Navigator**.  
  
The screen refreshes to display new fields, as shown in [the Fields for Adding Paths Using Object Navigator image](#).
2. Select **this**.
3. Click the **Object Navigator** icon.  
  
Object Navigator opens in a new browser window.
4. Navigate to the appropriate object attribute and click **ok**. For details about using Object Navigator, see [Using Object Navigator](#).  
  
The path you created appears in the Object Navigator field.
5. Click **add**.  
  
The path, starting with `this.`, is added to the **Qualifier** field.
6. Place your cursor at the end of the path in the **Qualifier** field and type the appropriate operator. For details, see [Data Types](#).
7. If the right side of your expression needs to be a literal value (such as the name of a phase or the tree position of a lookup item), type this literal value after the operator.  
  
Your qualifier has been defined. You can continue to set the other values to define the custom field, as described in [the Custom Fields Tab table](#).
8. If instead, the right side of your expression needs to identify an object attribute:
  - a. Select **container**.
  - b. Repeat steps 3 and 4.

- c. Click **add**.

Your qualifier has been defined. You can continue to set the other values to define the custom field, as described in [the Custom Fields Tab table](#).

The following examples are typical expressions used in a Qualifier field for a custom field of type Custom Object. You can use these as models for your own qualifiers.

**Examples of Qualifiers for Custom fields of the type Custom Object**

What you want to do	Qualifier expression	What this expression says
Ensure that projects available for selection in the custom field and the record that contains the custom field have the same parent.	<code>this.parent.primaryKey=container.parent.primaryKey</code>	The primary key of the parent of the project record selected in the field equals the primary key of the parent of the record where the field is located.
Filter project records available in the custom field by their phase. Only those records whose current phase is Open can be selected in the field.	<code>this.currentPhaseType.name=Open</code>	The name of the current phase of the project record selected in the field equals Open.
In a record where there are two custom fields of the type Custom Object, and the records selected in these fields have a parent-child relationship, filter project records available in one field to only show children of the record selected in the other field.	<code>this.parent.primaryKey=container.detailList(PAYM).detailObjfValueList(Feature).detailValueFID</code>	The primary key of the parent record of the project record selected in the field equals the primary key of the project record selected in another field of type Custom Object in the same record.
Ensure that the projects available for selection in the custom field have the same person in a custom field of type Involved as the person identified in the contact-centric field of the record that contains the custom field.	<code>this.detailList(CLAM).detailInvlValueList(ReportedBy).detailValue.contact.primaryKey=container.contact.primaryKey</code>	The primary key of the contact who is identified in the ReportedBy custom field in the project record selected in the field equals the primary key of the contact who is selected in the contact-centric field in the record where this field is located.
Ensure that only projects with the specified default category are available for	<code>this.defaultCategory.treePosition=CLAM_LIAB</code>	The tree position of the default category of the project record selected in the field is CLAM_LIAB.

selection in the custom field.		
Ensure that only projects with a particular value selected in a custom field of type List are available for selection in the custom field.	<code>this.detailList (CLAM_AUTO) .detailObjValueList (LossCode) .detailValue=LOSS_ROOT_VDRE</code>	The tree position of the item selected in a custom field in the record selected in the field is LOSS_ROOT_VDRE.
Ensure that projects available for selection in the custom field were created within the past 30 days.	<code>this.createdOn&lt;=TODAY30</code>	The createdOn date of the record selected in the field is within 30 days of today or equal to today's date.
<p>Ensure that projects available for selection in the custom field meet three conditions:</p> <ul style="list-style-type: none"> <li>• The parent of the project is the same as the parent of the record that contains the custom field.</li> <li>• The current phase is Litigation.</li> <li>• The default category is Auto.</li> </ul>	<code>this.parent.primaryKey=container.parent.primaryKey;this.currentPhaseType.name=Litigation;this.defaultCategory.treePosition=CLAM_AUTO</code>	This qualifier contains three expressions, separated by semicolons (;). See the descriptions for similar expressions in this table.

### 3.1.6.3.7 Creating Custom Fields

To use blocks for your custom pages, you must first create the necessary custom fields.

#### To create desired custom fields

1. Analyze your organization's specific information that needs to be captured in the selected object, in addition to the default fields provided within TeamConnect.
2. Decide which custom fields you need to create to capture this information.

**Tip:** Make a list of the labels for these custom fields and their field types.

3. Make sure the custom lookup tables are created for the custom fields of the type List.
4. Because all custom fields are category-specific, decide to which categories in the object definition you want to add your custom fields.

5. Make sure the desired object categories are added to the object definition.
6. Add your custom fields to the appropriate category in the object definition. See [Adding Custom Fields](#).
7. Include your custom fields into the blocks, depending on your needs and design.

### To add a custom field to an object definition

1. In the Designer window, from the **Go to** drop-down list select **Object Definitions**.
2. Select the appropriate object definition from the displayed list.
3. Open the **Custom Fields** tab of the displayed object definition.
4. Select the **Number of entries you would like to add** from the drop-down list.
5. For each data entry row, enter the values in the appropriate fields as described in [the Custom Fields Tab table](#).
6. If you want to continue adding custom fields, click **add more**. Otherwise, click **Save** on the toolbar.

TeamConnect automatically saves the custom field information and adds the fields to the system block called **Details**.

7. Create the appropriate components for the object view—that is, custom blocks.

### Points To Remember

Custom fields can be created and displayed by category only. Because of this category dependency, you must carefully plan for which categories to create the desired custom fields, taking into consideration the following:

- For all categories and custom fields you add in an object definition, associated access rights are automatically created. They must be assigned to the appropriate users and user groups in order for the users to be able to view the categories and their custom fields.
- The root-level category in both custom and system objects is always automatically added to the object records to provide access to other categories (unless you do not intend to use categories and custom fields in your design at all). This means that all custom fields created for this category are automatically displayed in all records for all users.
- Always arrange your custom fields so that you do not have to replicate the same fields for different categories.

This improves performance and also prevents possible confusion if the custom fields of two categories are used in the same object record.

- A category name must be unique within its object if it has any custom fields included in Data Warehouse. For more information, see [Data Warehouse Requirements for Custom Fields](#).

### 3.1.6.4 Blocks

*Blocks* are sets of object information represented by system and custom fields organized in a single XML or JSP file. Blocks typically display as individual sections on the page, as shown in the following image.

<b>Personal Information</b> <a href="#">Edit</a>	
<b>Name:</b> Joe Attorney	
<b>Job Title:</b>	
<b>Company:</b> Mitrastech Holdings, Inc.	
<b>Address Information</b> <a href="#">Edit</a>	
<b>Phone Numbers</b> <a href="#">Edit</a>	<b>Fax Numbers</b> <a href="#">Edit</a>
<b>Business:</b> (800) 555-1212 (Primary)	
<b>Mobile:</b> (310) 555-1212	
<b>Home:</b> (512) 555-1212	

System Blocks

System and custom blocks display in the following different ways:

- *System blocks* are included by default in the system object views of TeamConnect. You cannot change, delete, or add new system blocks. For example, on the **General** page of a contact record, **Personal Information**, **Address Information**, and **Phone Numbers** are all system blocks.
- *Custom blocks* are XML files that might refer to Java classes and display sets of system or custom fields organized in sections on record pages or wizard pages.

### Custom Blocks

You can use custom blocks to do the following operations:

- Display different system or custom fields for each group of TeamConnect users.
- Design new pages for a record or wizard page.
- Display multiple sets of fields on one record page or wizard page.
- Display custom fields associated with several categories in the same section.

**Important:** *You must assign the applicable rights for the appropriate categories and details (custom fields) in order for users to be able to view custom blocks that include custom fields.*

Custom blocks in records and wizards are different in the following ways:

#### Differences between Record and Wizard Custom Blocks

	In records	In wizards

<b>Type of fields you can include in custom blocks</b>	System fields, Custom fields	System fields, Custom fields, <a href="#">Wizard parameters</a>
<b>How users view custom blocks</b>	Add the blocks to object views and assign appropriate rights.	Add the blocks as components to wizard pages.

## Button styling

Buttons tags will not have any styling by default. Assign a class (e.g., btn-primary) for styling these objects.

## Custom Java Blocks

When you create a screen with custom actions, you may need to create a custom Java block (CJB). The CJB uses the TeamConnect API to specify actions and properties for the custom block. The XML file, which includes the structure, layout, and contents of the custom block, references the Java class. If you are not creating custom actions for the block, you only need to create the XML file without the CJB.

**Note:** Do not use a CJB to construct arbitrarily complex JavaScript and HTML code for the custom block.

### 3.1.6.4.1 Creating Blocks

When you create a block, you create an XML file using regular HTML tags and formatting in addition to specific TeamConnect tags and attributes that display custom fields, wizard parameters, and their labels on the page. You may also use style sheets, if necessary.

You can edit an XML file for a block manually, using every available TeamConnect block tag. For blocks that are to become part of object views, you can also use the Screen Designer, which has a drag-and-drop GUI that lets you build blocks quickly. The set of block tags available to the Screen Designer is just a subset of all the available block tags in TeamConnect.

See the *Screen Designer Help* for information on creating custom blocks using the Screen Designer tool.

**Note:** As a time-saving feature you can start a block's XML file using the Screen Designer, then later edit that file manually to add extra block tags.

### To plan and create a custom block using manual editing

1. Identify the category whose custom fields you want to include in the block.

2. For that category, identify what custom fields each user group has access to.
3. When creating a block for a wizard page, identify the parameters you intend to include in the block and make sure they are defined and added to the appropriate wizard page or pages. For details, see [Defining Parameters](#) and [Defining Page Components in Wizards](#).
4. Verify that the desired object category is added to the object definition and the appropriate custom fields are created for the category.
5. Decide how you want to arrange the custom fields (and wizard parameters, if there are any) in the block:
  - Where the fields should be displayed on the page.
  - What the background color of the page should be, if you do not want the system settings or users' preferences to apply.
6. Create the appropriate XML file that contains the necessary fields and structure as explained in [Understanding XML Files](#).
7. Upload the XML file to TeamConnect.
8. Upload any support files, such as scripts, images, or style sheets, to the **Top Level/ HTTPRoot** folder in the Documents area of TeamConnect.
9. Associate the XML file with the appropriate object on the **Blocks** tab of the object as explained in [Defining Blocks](#).
10. If you are creating an object view, include the block in the desired custom object view of the appropriate object. For more information on object views, see [Defining Object Views](#).
11. If you are creating a wizard, add the block to the desired wizard page. For details, see [Defining Page Components in Wizards](#).

### Further editing for custom categories

At runtime, a block that was created using Screen Designer may display an error message if all of the following are true:

- The block is part of a custom object definition.
- The category chosen for the block is a custom category.
- The record currently being displayed has not had that custom category assigned to it.

To guard against this possibility, you can do the following:

- Finish laying out the block using Screen Designer.
- Manually edit the resulting XML file, adding a **tc:section** tag that encapsulates the block and specifies the custom category used by the block.

**Note:** After this manual edit, the XML file cannot be reopened in Screen Designer, since Screen Designer is unable to parse the **tc:section** tag.

### Page Element Reference



The table below explains Page Elements, and their properties, in more detail. Where the documentation refers to "expression", the syntax used is Java Expression Language (JEXL) unless otherwise stated.

#### Page Elements

Element	Property	Notes
<b>Field Column Layout</b>		See the detailed explanation of a layout in step 4.
<b>If Statement</b>		(tc:if) Conditions the display of any elements inside the <b>if</b> by evaluating an expression.
	test	The expression to be evaluated.
	negate	If YES, the expression must evaluate false for the contents inside the <b>if</b> to be displayed. If NO, the expression must evaluate true for the contents inside the <b>if</b> to be displayed. If this property is omitted, it is assumed to be NO.
<b>If-Else Statement</b>		When dragged to the layout, produces two <b>If</b> elements, the second of which uses property <b>negate</b> =YES. The same value of property <b>test</b> should then be entered in each of the two <b>If</b> elements.
<b>Java Class</b>		(tc:useClass)
	id	Must be unique within the block. Can be referenced by CSS or Javascript
	name	Required. The name of the Java class. The class file must be located in the same folder that contains the block XML file.
<b>Message</b>		(tc:message) Places text output on the page outside the boundaries of a Field Column layout. Unlike the similar <b>out</b> element, a <b>message</b> element is localized.
	key	The full i18n key of the message text to be displayed. For information about i18n keys, see <a href="#">Creating Custom Messages</a> .
<b>Message Parameter</b>		This element can only be dragged into an existing <b>Message</b> element. Some <b>Message</b> elements accept substitution parameters at runtime. For each such parameter in a <b>Message</b> , you need a <b>Message Parameter</b> element.

	value	Text or expression to be placed in the body of the <b>Message</b> element.
<b>Out</b>		(tc:out) Places text output on the page outside the boundaries of a Field Column layout. Unlike the similar <b>Message</b> element, an <b>Out</b> element cannot be localized.
	value	Text or expression to be rendered in the block.

#### 3.1.6.4.2 Creating Block XML Files

To create an XML file for a custom block, you need to know the following information:

- Basic HTML tags and structure and XML requirements.
- The following custom field and category information if you want to include custom fields in the block:
  - Field name—Field names uniquely identify the fields within a category. You can locate them on the **Custom Fields** tab of the object definition for the block.
  - The full tree position of the category for the custom field—The full tree position of a category is a combination of tree positions assigned to the category and all of its parent categories, starting with the root category and separated by underscores.

For example, if category B with partial tree position BBBB is created under category A with partial tree position AAAA in the Account object definition, the full tree position of category B is ACCT\_AAAA\_BBBB.

- Wizard parameter names if you are creating the block for a wizard and want to include parameters. The name of a wizard parameter uniquely identifies the parameter within a wizard. You can locate parameter names on the **Page Components** tab of the wizard page. [Define parameters](#) for the page that includes the new block.

***Tip:** To avoid duplicating parameters on the same page, select the **Display in blocks only** check-box for the parameters that you include in your blocks.*

- The appropriate [block tags](#) and formatting.
- The tc:useClass tag with the name of the Java class file and an alias in the `id=` field to refer to the Java class.

***Note:** When creating an XML file, you should not hard code background and foreground colors or font type and size. Also minimize the use of JavaScript.*

#### To create an XML file for a block

1. Open the text editor of your choice.

Microsoft Notepad is a common choice, but other text editors can be used to create XML files.

2. Type the following lines at the beginning of the document:

```
<?xml version="1.0" encoding="UTF-8"?>
<tc:transform version="1.0" xmlns:tc="http://www.w3.org/1999/XSL/Transform">
```

3. Refer to a Java class using the tc:useClass tag, as shown in the following line:

```
<tc:useClass id="cjb_alias" name="Java_class_name"/>
```

For `Java_class_name`, enter the name of the Java class. For `cjb_alias`, enter the name of the alias that you use in the XML file to refer to the Java class.

4. Type the contents of your block.

For a sample of the tags you should include, see [Sample XML File](#).

5. Include the closing tag at the end of the document:

```
</tc:transform>
```

6. Save this new file on your hard drive, with an .xml extension (for example, salesInvolved.xml).
7. Upload the XML file and the Java class to the **Screens** directory of the object for which you are creating the block. See [Uploading XML Files](#).

#### 3.1.6.4.2.1 Block Tags

Custom block files use several TeamConnect tags that allow you to display and organize custom fields and parameters within a block.

##### Team Connect Block Tags

Tag	Description
<b>tc:include</b>	Displays a custom block in a custom block.
<b>tc:label</b>	Displays custom field labels in a block.
<b>tc:field</b>	Displays custom fields in a block.
<b>tc:blockTemplate</b>	Displays the title of a block.
<b>tc:wizardParameter</b>	Displays wizard parameter fields in a block created for wizards.
<b>tc:wizardParameterLabel</b>	Displays wizard parameter labels in a block created for wizards.
<b>tc:message</b>	Displays localized messages in a custom block.
<b>tc:messageParam</b>	Displays message parameter values.

For additional block tags that are related to reports, see in [Using Tags to Embed Reports in Blocks](#).

In cases where this topic refers to an "expression" inside a tag, Java Expression Language (JEXL) is the syntax.

**Caution:** *Do not type field labels or other field information directly in your block files. The formatting specified in the system settings and user preferences do not applied to them. You also have to manually update the labels in all files if the labels change.*

## tc:include

This tag displays a custom block within a custom block.

The <tc:include> tag:

- Must always be in lowercase.
- Must use the following attribute:
  - The block attribute that references the block file in the current object definition.
- Uses the following syntax:

```
<tc:include block="BlockName" />
```

- BlockName—The file name of the block you want to include. If the BlockName attribute does not have a file extension, TeamConnect looks for BlockName.xml, and then for BlockName.scr.xml.

For example, the following code would include and render a block with file name SampleFile.xml or SampleFile.scr.xml:

```
<tc:include name="SampleFile" />
```

## tc:label

This tag displays the label of a custom field. You may use it in blocks created for object views and wizards. Insert tc:label before the tc:field tag to identify the field on the page. For certain field types, such as check-boxes, the label typically displays after the field.

The <tc:label> tag:

- Must always be in lowercase.
- Must use two attributes:
  - The select attribute that references the field name of the appropriate custom field.
  - The category attribute that references the tree position of the field's category.
- Uses the following syntax:

```
<tc:label name="FieldName" category="FullTreePosition" />
```

- FieldName—The name (not label) of the field you want to include.
- FullTreePosition—The full tree position of the category for the field.

For example, the following code displays the label "Incident Date & Time" of the custom field with the name "LossDate," created for the root category of the Claim object definition with the unique code CLAM:

```
<tc:label name="LossDate" category="CLAM" />
```

For attributes that you can use with field labels, see [Label Tags](#).

## tc:field

This tag displays the field, either blank or with its default value. You may use it in blocks created for object views and wizards.

The <tc:field> tag:

- Must always be in lowercase.
- Must use two attributes:
  - The name attribute that references the field name of the appropriate custom field
  - The category attribute that references the tree position of the field's category.
- Uses the following syntax for system fields:

```
<tc:field name="applicationEntity.FieldName" appropriate attributes />
```

Uses the following syntax for custom fields:

```
<tc:field name="applicationEntity.FieldName" category="FullTreePosition" appropriate attributes />
```

- **FieldName**—The name (not label) of the field you want to include.
- **FullTreePosition**—The full tree position of the category for the field.
- **appropriate attributes**—The appropriate formatting attributes, if necessary.

For example, the following code displays two date fields, ClaimMade and LossDate, both created for the root category of the Claim object definition with the unique code CLAM. The LossDate field, however, appears with a time field next to it because of the showTime formatting attribute.

```
<tc:field name="ClaimMade" category="CLAM" />
<tc:field name="LossDate" category="CLAM" showTime="true" />
```

For attributes that you can use with tc:field, see [Using Tag Attributes](#).

## tc:wizardParameter

This tag displays the parameter field, either blank or with the default value specified for it. You can only use this tag in wizard blocks.

The <tc:wizardParameter> tag:

- Must always use the name attribute that references the parameter name.

- Should not be inside the <tc:section> tag, unless you want to display parameters only if the same section includes the custom fields for the category during record creation.
- Uses the following syntax:

```
<tc:wizardParameter name="ParameterName" appropriate attributes />
```

- ParameterName—the name (not label) of the parameter you want to include.
- appropriate attributes—the appropriate formatting attributes, if necessary.

For example, the following code displays two number fields, InjuredPersons and ClaimAmount. The ClaimAmount field, however, displays values with the dollar sign (\$) before them because of the format attribute used with it:

```
<tc:wizardParameter name="InjuredPersons" />  
<tc:wizardParameter name="ClaimAmount" format="DOLLAR" />
```

For attributes that you can use with tc:wizardParameter, see [Using Tag Attributes](#).

### tc:wizardParameterLabel

This tag displays the label for a parameter field. Insert tc:wizardParameterLabel before the tc:wizardParameter tag to identify the field on the page. For certain field types, such as check-boxes, the label typically displays after the field. If the wizard uses text prompts, you may not need to include the label. You can only use this tag in wizard blocks.

The <tc:wizardParameterLabel> tag:

- Must always use the name attribute that references the parameter name.
- Uses the following syntax:

```
<tc:wizardParameterLabel name="ParameterName" />
```

- ParameterName—The name (not label!) of the field you want to include.

For example, the following code displays the label "Method used to report the claim" of the parameter of type List with the name "HowReported":

```
<tc:wizardParameterLabel name="HowReported" />
```

For attributes that may be used with field labels, see [Label Tags](#).

### tc:blockTemplate

This tag displays the title of a custom block. TeamConnect defines the titles of system blocks.

The <tc:blockTemplate> tag:

- Must appear exactly as shown.
- Uses the following syntax:

```
<tc:blockTemplate blockTitle="title_text" />
```

## tc:message

This tag allows localized messages to appear in custom blocks.

The <tc:message> tag:

- Uses the following syntax:

```
<tc:message key="messageKey" />
```

- messageKey—Identification key for the corresponding value.

For example, the following code displays the value of a common custom key for welcome message text:

```
<tc:message key="custom.common.welcomeMessage" />
```

**Note:** Keys are defined in the Designer on the [Custom Messages tab](#).

## tc:messageParam

This tag, used with the <tc:message> tag, supplies the value for a parameter.

The <tc:messageParam> tag:

- Must be nested inside a <tc:message> tag.
- Replaces parameters in the order of the <tc:messageParam> tag.
- May appear multiple times.
- Uses the following syntax:

```
<tc:message key="messageKey">  
    <tc:messageParam value="value" />  
</tc:message>
```

- value—Argument used to replace the parameter.

For example, if the defined custom.common value is welcome {0}, the code may appear as follows:

```
<tc:message key="custom.common.welcomeMessageWithUsername">  
    <tc:messageParam value="{cjb.currentUsername}" />  
</tc:message>
```

**Note:** The Designer defines keys on the [Custom Messages tab](#).

## 3.1.6.4.2.2 Tags to Embed Reports in Blocks

You can use block tags to embed reports or links to reports within a block.

### tc:report

This tag renders the report's chart output within the block. It does not render tabular results.

The <tc:report> tag:

- Must always be in lowercase.
- Must use two attributes:
  - The path attribute (required) that references the location and name of the appropriate report
  - The width and height attributes (optional) that define how much visual space in the block to use for report output. Their default value is 500px.
- Can contain one or more <tc:parameter> tags nested inside the <tc:report> tag.
- Uses the following syntax:

```
<tc:report path="path to report" width="500px" height="500px" />
```

- path to report—The location and name of the report you want to render. The path is based on the root folder for Reports in TeamConnect. Folders are delimited with the forward slash (/) character.

Even though each path starts with the name of the root folder, "Reports," the tag also supports omitting the string "Reports" from the beginning of the path and using just a forward slash to imply the root. This shortened form of designating the root folder is valid for all paths in all report tags.

- 500px—The separate sizes for the width and height of the space for rendering chart output. 500 pixels is the default for both. The suffix "px" or "%" is required at the end of the value to determine the units of measurement. The suffix "px" means that the size is measured in pixels. The suffix "%" means that the size is measured as a percentage of the size of the block itself.

For example, the following code sample renders chart output for report **OC\_Spending**, which is found in subfolder **ManagementReports** beneath the root report folder. The default width and height for the chart output have been overridden with specific values:

```
<tc:report path="/ManagementReports/OC_Spending" width="650px" height="400px"/>
```

### tc:reportLink

This tag renders a link within the block. When clicked, the link executes a report. The report appears in a new page, with a link on that page to return to the block that you had been viewing. The user interface of the new page is almost identical to that presented by the **Reports** page when you click on the name of one of the listed reports.

The <tc:reportLink> tag:

- Must always be in mixed case, as shown.



- Uses the following attribute:
  - The path attribute (required) that references the location and name of the appropriate report.
- Can contain one or more `<tc:parameter>` tags nested inside the `<tc:reportLink>` tag.
- Uses the following syntax:

```
<tc:reportLink path="path to report"/>
```

- path to report—The location and name of the report you want to render. The path is based on the root folder for Reports in TeamConnect. Folders are delimited with the forward slash (/) character.

For example, the following code sample renders a link to report **OC\_Spending**, which is found in subfolder **ManagementReports** beneath the root report folder:

```
<tc:reportLink path="/ManagementReports/OC_Spending"/>
```

### **tc:reportFolderLink**

This tag renders a link within the block. When clicked, the link opens a page as if the user had navigated to the **Reports** tab of the application and to the path specified in this tag. The section is equivalent to the section described in [Using Report Folders](#). No reports run as a result of this link, but the user may then select one of the reports within the specified folder.

The `<tc:reportFolderLink>` tag:

- Must always be in mixed case, as shown.
- Uses the following attribute:

The path attribute (required) that references the location and name of the report folder.

- Uses the following syntax:

```
<tc:reportFolderLink path="path to folder"/>
```

- path to folder—The location and name of the report folder you want to display. The path is based on the root folder for Reports in TeamConnect. Folders are delimited with the forward slash (/) character.

For example, the following code sample renders a link to subfolder **ManagementReports** beneath the root Reports folder:

```
<tc:reportFolderLink path="/ManagementReports"/>
```

To link to the root Reports folder itself, use either syntax:

```
<tc:reportFolderLink path="/" />
```

```
<tc:reportFolderLink path="Reports" />
```

### **tc:parameter**

This tag supplies a value to a parameter that a report uses. This tag is only valid within the body of a `<tc:report>` tag or a `<tc:reportLink>` tag.

**Note:** If a report uses parameters, and a parameter is not specified within the `<tc:report>` tag or `<tc:reportLink>` tag, the parameter runs with the default value.

The `<tc:parameter>` tag:

- Must always be in lowercase.
- Uses the following attributes:
  - The name attribute (required) corresponds to the parameter name that the report designer defines. If the name value is not resolved at report runtime, an error message appears.
  - The value attribute (required) specifies the value associated with this parameter during the execution of the report.

The value attribute can contain a list of multiple values, separated by semicolons.

If the value(s) for this attribute are not resolved at report runtime, an error message appears.

Permissible values are dependent on the datatype of the parameter. Most datatypes allow the value of the parameter to be expressed as a string literal, as long as that string conforms to the datatype. For example, a date datatype accepts the string "2011-04-30" and a category datatype accepts the string "EXPE\_ABCD". Datatypes that do not permit string literals include **User**, **Project**, **Contact**, **Involved**, and **Account**. For those five datatypes, you must use an attribute path expression.

Parameters of datatype Enum accept a string literal composed of one or more Java constant names. Multiple names are delimited by semicolons (;). For more information about Enum (enumerations) see [Enumerations](#).

All datatypes, whether or not they permit string literals, accept an attribute path expression that references an object compatible with the datatype. Examples of attribute path expressions are `#{cjb.startedOn}` for a date parameter and `#{applicationEntity.treePosition}` for a lookup table parameter.

- Uses the following syntax (shown within a containing `tc:report` tag):

```
<tc:report path="path to report" width="500px" height="500px">
  <tc:parameter name="name" value="value"/>
  <tc:parameter name="name" value="value"/>
</tc:report>
```

- path to report—The location and name of the report you want to render. The path is based on the root folder for Reports in TeamConnect. Folders are delimited with the forward slash (/) character.
- 500px—The separate sizes for the width and height of the space for rendering chart output. 500 pixels is the default for both. The suffix "px" or "%" is required at the end of the value to determine the units of measurement. The suffix "px" means that the size is measured in pixels. The suffix "%" means that the size is measured as a percentage of the size of the block itself.
- name—The name of the parameter that the report designer knows.

- value—The object or literal that contains a value to be associated with the parameter for this execution of the report.

For example, the following code sample renders chart output for report **RejectedInvoices**, which is found in subfolder **ManagementReports** beneath the root report folder. The default width and height for the chart output are overridden with specific values. A report parameter named "pastXdays" is assigned a value contained in the attribute path "limitDays":

```
<tc:report path="ManagementReports/RejectedInvoices" width="650px"
height="400px">
    <tc:parameter name="pastXdays" value="{cjb.limitDays}"/>
</tc:report>
```

### 3.1.6.4.2.3 XML File Sample

This topic contains an XML code sample that results in the **General Subpoena** block for the following wizard page.

The screenshot shows a wizard window titled "My Wizard" with a "Step 1 of 2" indicator. The main content area is titled "Wizard Custom Block Example" and contains the following form fields:

- \*Matter Name:** A text input field.
- Matter Description:** A large text area.
- Attorney's Fee Awarded:** A text input field with a dollar sign (\$) prefix.
- Matter Security:** A dropdown menu with the selected value "Legal Dept Access Only".
- Create Budget Now:** A dropdown menu with the selected value "No".

A note at the bottom of the form states: "\*Required fields are noted by an asterisk".

**Note:** You can customize the tabbing order by adding a `tabIndex` attribute to each of the `<tc:field>` and `<tc:wizardParameter>` tags within an XML file. For details, see [tabIndex](#).

The XML sample in this topic demonstrates how to include the following examples in an XML file:

- Using the `<tc:useClass>` tag to refer to the `MyCustomBlock.class` file.
- Using the `<tc:blockTemplate>` tag with the optional `blockTitle` attribute.
- Adding custom fields to a block. This custom field in this code sample are text, number, and lookup table fields.
- Using formatting tag attributes for the number field, such as `format` and `size`.
- Adding wizard parameters to a block. Because this XML sample contains parameters, you can only use it in wizards. To use the code sample in an object view, remove or comment out the parameters from the code.

- Specifying a custom JavaScript file to add styles to the CJB.

The [custom block image](#) shows the resulting block on a wizard page.

## Code Sample

```
<?xml version="1.0" ?>
<tc:transform version="4.0" xmlns:tc="http://www.mitrates.com/schemas/2008/
custom">
  <tc:section category="DISP">
    <tc:blockTemplate blockTitle="Wizard Custom Block Example">
      <!-- Use a custom java class -->
      <tc:useClass id="cjb" name="MyCustomBlock"/>
      <div class="bg_blockcolor">
        <table cellpadding="10" width="100%">
          <tr>
            <td width="50%" align="left" valign="top">
              <table cellpadding="3" width="96%">
                <!-- System field -->
                <tr valign="top">
                  <td align="right" class="tdlabel2">
                    <tc:label colon="true" required="true"
                    key="custom.common.legal.matterName"></
                    tc:label>
                  </td>
                  <td>
                    <!-- Default value of field generated by
                    custom java class -->
                    <tc:field maxLength="250"
                    name="enterpriseEntity.name" size="77"
                    value="{cjb.generateName}"></tc:field>
                  </td>
                </tr>
                <!-- Custom field of type text -->
                <tr valign="top">
                  <td align="right" class="tdlabel2">
                    <tc:label colon="true"
                    name="MatterDescriptionDI" category="DISP"></
                    tc:label>
                  </td>
                  <td>
                    <tc:note cols="80" category="DISP"
                    name="MatterDescriptionDI" rows="5"></tc:note>
                  </td>
                </tr>
                <!-- Custom field of type number -->
```

```
<tr valign="top">
  <td align="right" class="tdlabel2">
    <tc:label colon="true"
      name="AttorneysFeeAwardedDI"
      category="DISP"></tc:label>
  </td>
  <td>
    <!-- Number value formatted as currency -->
    <tc:number name="AttorneysFeeAwardedDI"
      category="DISP" format="CURRENCY" size="12"></tc:number>
  </td>
</tr>
<!-- Custom field of type lookup -->
<tr valign="top">
  <td align="right" class="tdlabel2">
    <tc:label colon="true" name="MatterSecurityDI"
      category="DISP"></tc:label>
  </td>
  <td>
    <tc:lookup nullValueDisplayString="(None
      Selected)" category="DISP"
      name="MatterSecurityDI" tableCode="SECE"
      allowNullValue="true"></tc:lookup>
  </td>
</tr>
<!-- Wizard parameter -->
<tr valign="top">
  <td align="right" class="tdlabel2">
    <tc:wizardParameterLabel
      name="CreateBudgetNow" colon="true"></tc:wizardParameterLabel>
  </td>
  <td>
    <tc:wizardParameter name="CreateBudgetNow"
      wizardUniqueKey="DNDI"></tc:wizardParameter>
  </td>
</tr>
</table>
</td>
</tr>
</table>
</div>
<!-- Include custom javascript uploaded to Top Level > System > HTTPRoot
-->
<script src="./httproot/myjs.js" type="text/javascript"> </script>
```

```
</tc:blockTemplate>
</tc:section>
</tc:transform>
```


#### 3.1.6.4.3 Uploading Custom Block Files

After you have created the required XML file and Java class you must upload them to the **Screens** directory of TeamConnect.

##### To upload CJB files to TeamConnect

1. Log in to TeamConnect and Designer, then in the **Go To** drop-down list select **Documents**.
2. Browse to the **Screens** directory of the required custom or system object for which you are creating the block.

For example, if the required object was named Claim, you would browse to **Top Level \System\Object Definitions\Claim\Screens**.

3. Click **Upload Document**  on the toolbar.
4. From the **Upload New File** screen, choose one of the files to upload. Enter the **File Name** and **File Type** and click **upload file**.

You can do the same for any other files.

The name of the file now appears on the **File Name** drop-down list on the **Blocks** tab of the system or custom object. To define the block in TeamConnect, see [Defining Blocks](#).

#### 3.1.6.4.4 Defining Blocks

After you have written the XML file that contains the structure and content of the desired block, you need to define it as a block for TeamConnect. You can do this by selecting the XML file on the **Blocks** tab of the appropriate object definition.

##### To add a block to an object definition

1. In the Designer window, from the **Go to** drop-down list select **Object Definitions**.
2. Select the appropriate object definition and then select the **Blocks** tab.

**Blocks**

[Go to Screens folder](#)

Number of entries you would like to add:

	File Name	Friendly Name
1	(Select)	<input type="text"/>
2	(Select)	<input type="text"/>

[+ add more](#)

Displaying records 1-10 of 28. [Next](#)

	File Name	Friendly Name
1	<input type="checkbox"/> OB_CitationsCL	Citations - Claimant
2	<input type="checkbox"/> OB_CitationsIN	Citations - Insured
3	<input type="checkbox"/> OB_DistractionsCL	Distractions - Claimant
4	<input type="checkbox"/> OB_DistractionsIN	Distractions - Insured
5	<input type="checkbox"/> OB_DrugsCL	Drugs - Claimant
6	<input type="checkbox"/> OB_DrugsIN	Drugs - Insured
7	<input type="checkbox"/> OB_ImproprMntnsCL	Improper Maintenance - Claimant
8	<input type="checkbox"/> OB_ImproprMntnsIN	Improper Maintenance - Insured
9	<input type="checkbox"/> OB_InvolvedParties	Involved Parties
10	<input type="checkbox"/> OB_ObstructedViewCL	Obstructed View - Claimant

[Check All](#) - [Uncheck All](#)

Custom Blocks Tab

- Enter the appropriate values as described in [the Blocks Tab table](#).
- If you want to continue adding custom blocks, click **add more**. Otherwise, click **Save** on the toolbar.

The blocks are now available for inclusion in custom object views or wizards for the selected object definition. To view a block on the screen, you must include it in an object view (for details, see [Creating Object Views](#)), or if it contains parameters, you must add it as a page component in a wizard (for details, see [Defining Page Components in Wizards](#)).

The following table describes the items in the **Blocks** tab.

Blocks Tab

Item	Description
<b>Go to Screens folder</b>	Click this hyperlink to go to the <b>Screens</b> directory of the selected object. This directory is where you store all your XML files (blocks) for the selected object.
<b>File Name</b>	Select the XML file that contains the block information.

	<b>Note:</b> This drop-down list reflects the contents of the Screens directory of the selected object.
<b>Friendly Name</b>	Enter a name (or title) to identify the block. This name is used to reference the block when adding it to an object view or a wizard page, and it is best to reflect the contents of the blocks. The maximum length is 250 alphanumeric characters.  <b>Tip:</b> Make the friendly name as descriptive as possible so other solution developers can easily understand what the block is created for.
<b>Block List</b>	Displays the blocks for the selected category. Select or clear the corresponding check-boxes on the left to change or delete the appropriate blocks.

## Friendly Names

Friendly names are distinguished by objects, so you cannot use the same name twice for the same object, but you can use the same name for other objects. For example, you can use the friendly name "Assignees" in all objects, except for custom objects and the Task object because they already use the Assignees name.

The following table includes the words that you cannot use as friendly names for custom blocks.

### Friendly Names You Cannot Use

Objects	Words You Cannot Use
All Objects	<ul style="list-style-type: none"> <li>• General</li> <li>• Categories</li> <li>• Detail Forms</li> <li>• Documents</li> <li>• Security</li> <li>• Access Info</li> <li>• History</li> <li>• Group Rights</li> <li>• Workflow</li> </ul>
Accounts	<ul style="list-style-type: none"> <li>• Amount</li> <li>• Posting Criteria</li> <li>• Posting Criteria Task</li> </ul>



	<ul style="list-style-type: none"><li>• Posting Criteria Expense</li><li>• Posting Criteria Involved</li><li>• Posting Criteria Vendor</li><li>• Posting Criteria Invoice</li><li>• Transactions</li><li>• Deposit/Withdraw</li><li>• Transfer</li><li>• Child Accounts</li></ul>
Appointments	<ul style="list-style-type: none"><li>• Attendees</li><li>• Resources</li></ul>
Contacts	<ul style="list-style-type: none"><li>• Address</li><li>• Phone</li><li>• Email</li><li>• Fax</li><li>• Internet Address</li><li>• Skills</li><li>• Rates</li><li>• Employee</li><li>• Relations</li><li>• Territories</li><li>• Involved</li></ul>
Documents	<ul style="list-style-type: none"><li>• Version</li></ul>
Invoices	<ul style="list-style-type: none"><li>• Line Items</li><li>• Transaction</li></ul>
Involved Parties	<ul style="list-style-type: none"><li>• Account</li><li>• Relations</li></ul>
Projects	<ul style="list-style-type: none"><li>• Phase Histories</li><li>• Assignees</li></ul>

	<ul style="list-style-type: none"> <li>• Relations</li> <li>• Milestone</li> <li>• Task</li> <li>• Appointment</li> <li>• Expense</li> <li>• Account</li> <li>• Involved</li> </ul>
Tasks	<ul style="list-style-type: none"> <li>• Assignees</li> <li>• Billing Information</li> <li>• Transaction</li> </ul>

#### 3.1.6.4.5 Troubleshooting Custom Blocks

##### Custom Block Troubleshooting Tips

Possible problem example	Possible solutions
The time fields do not appear in date fields.	Make sure you are using the correct formatting tags and set their values properly. See <a href="#">Using Tag Attributes</a> .
Numbers do not display with correct currency symbols, percentages, or decimal points.	
Option buttons and search modules appear as drop-down lists.	
Fields are too wide or too narrow.	
Fields and labels do not appear on the page.	Check the following: <ul style="list-style-type: none"> <li>• The correct spelling of custom field and parameter names.</li> <li>• The full category tree positions.</li> <li>• Correctly created sections for each category.</li> <li>• The correct spelling of field and label tags.</li> <li>• The rights to all categories and their custom fields.</li> </ul>

	<ul style="list-style-type: none"> <li>The categories, with fields you do not see, added to the record you are testing.</li> </ul>
The layout of the entire block does not display properly.	Make sure the <code>colspan</code> for each table row is set correctly if the tables use the <code>colspan</code> attribute.
Nested tables do not display properly. The background color is broken.	Make sure each nested table is enclosed between the <code>&lt;tr&gt;&lt;td&gt;</code> and <code>&lt;/tr&gt;&lt;/td&gt;</code> tags in the XML file.
The system displays a message about parsing errors.	Troubleshoot the XML file. See <a href="#">Understanding XML Files</a> .
The system displays a message about block errors and refers to an error log.	Check the error log for more information.
The block does not reflect changes to an XML file.	Make sure you have checked out and checked in the XML file properly in the <b>Documents</b> area.
The uploaded XML file fails to appear on the page.	<p>Check for the following:</p> <ul style="list-style-type: none"> <li>The uploaded file defined as a custom block in the object definition.</li> <li>The custom block part of a custom view that is assigned to the whole system or to the default group in your user account.</li> <li>The block added as a page component to the wizard.</li> <li>The <b>.xml</b> extension of the file in lowercase.</li> </ul>
Required custom fields do not display in red (or another specified color).	<p>Check for the following:</p> <ul style="list-style-type: none"> <li>In the XML file, fields use the <code>tc:label</code> tag and parameters use the <code>tc:wizardParameterLabel</code> tag.</li> <li>A different color is not hard coded in the XML file.</li> </ul>

### 3.1.6.5 Object Views

*Object views* are pages created to display object records. These views consist of sets of tabs providing complete information about a record. Within TeamConnect we distinguish between two types of object views, system and custom.

The Designer interface uses "tab" to signify a page in the end-user interface. However, switching between pages in that interface is accomplished by clicking on links in the left pane, not by clicking on tabs. In general in this section, a tab corresponds to a left-pane link.

*System object views* are the predefined record pages created for TeamConnect objects (both system and custom). You can see what each system view includes, but you cannot change, delete or add new system views.

*Custom object views* are the pages that you, as the TeamConnect solution developer, create for your organization to meet its specific needs or individual preferences. In other words, you can create your own custom views, change, and delete the existing custom views, as necessary.

If different groups are going to see different object views, [assign each view to the appropriate groups](#) with their **Group Account** pages.

If the rest of the TeamConnect users who do not have a default group set in their user accounts are going to see the same custom view, assign it globally in the **System Settings** screen.

By default, all object views that appear to the TeamConnect users are set on the **Default Object Views** tab of the **System Settings** screen, where all objects have their system views automatically selected as their default views. To change these settings, you have to first create your desired custom views for the necessary objects and then assign them as default one for the whole system.

*Note: If you decide to keep the system object views, you do not have to create or define object views. If you decide to create and define custom object views, continue through the sections that follow.*

For all objects, you have the option of either using the system views or creating your own custom views as follows:

- [Defining Object Views](#)
- [Creating Object Views](#)
- [Assigning Object Views](#)

## Object View Names

There are certain object view naming conventions that are followed when system views are created for all objects. This makes it easier for you to distinguish between the system and custom views of the system and custom objects.

- All *system views* of the *system objects* have the same name as the objects themselves, for example account, contact, and so on.
- All *system views* of the *custom objects* have the name of the object with the word "Project" after it, for example Claim Project, Policy Project, Litigation Project, and so on.
- All *custom views* of *all objects* have the names you, as the solution developer, assigned to them when creating them. See [Creating Object Views](#).

The following image shows a Designer example of system object views as they appear on the **Default Object Views** tab in the **System Settings** screen.



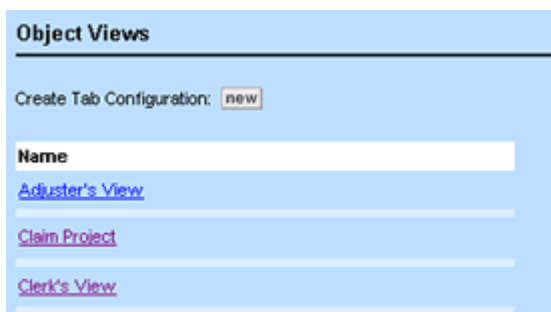
Account	Account
Appointment	Appointment
Contact	Contact
Document	Document
Expense	Expense
History	History
Invoice	Invoice
Task	Task
Claim	Claim Project
. Medical Report	Medical Report Project
. Settlement	Settlement Project
Comp Neg	Comp Neg Project
Comp Neg Involved	Comp Neg Involved Party
Comp Neg Milestone	Comp Neg Milestone
Contract	Contract Project

Default Object Views Tab on System Settings Screen

#### To open an existing object view

1. In the Designer window, from the **Go to** drop-down list, select **Object Definitions**.
2. Select the appropriate object definition and then select the **Object Views** tab.

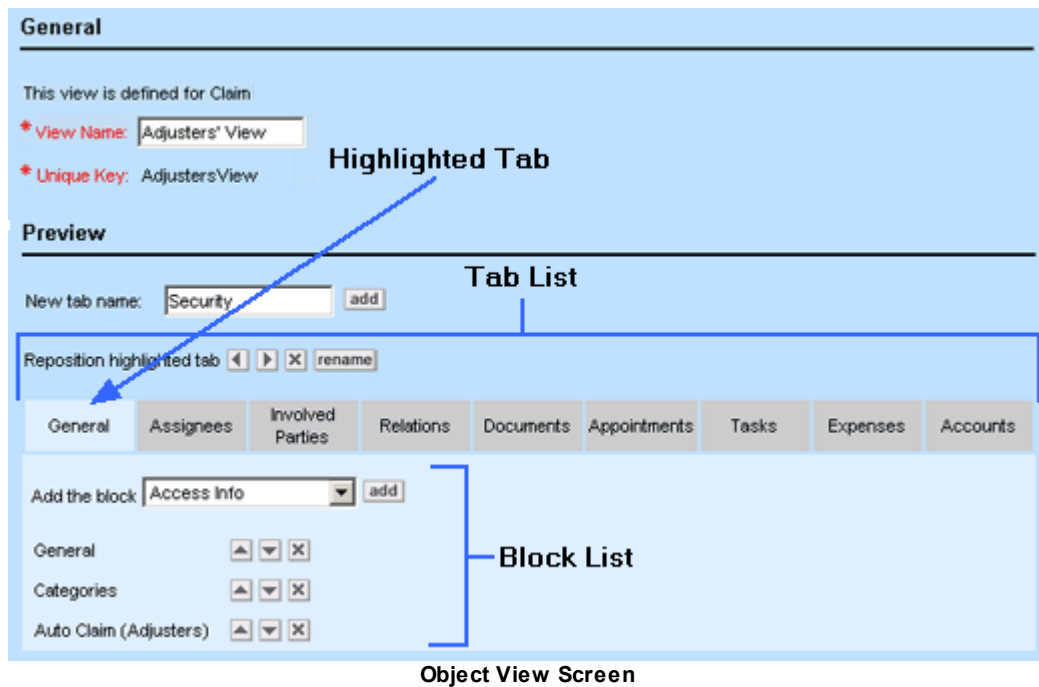
The corresponding **Object Views** screen appears.



Name
<a href="#">Adjuster's View</a>
<a href="#">Claim Project</a>
<a href="#">Clerk's View</a>

Object Views Tab (Adjuster's View is the Custom View and Clerk's View is the System View)

3. Click the required object view to open its Setup screen.





Object View Screen

The **Object Views** screen is divided into two sections:

- **General**, which displays the name of the view, the object to which the view is assigned, and the object view's unique key.
- **Preview**, which displays the tabs in the object view and the order of the tabs, and allows you to add blocks to the highlighted tab.

#### Object View Screen

Field or control	Description
<b>View Name</b>	<p>Enter a name to identify the object view. This name should specify the group for which this view is intended. Maximum 250 characters long.</p> <p><b>Note:</b> When first naming a view, you must <b>Save</b> the view to display other sections of the screen.</p>
<b>Unique Key</b>	<p>Enter an alphanumeric combination to identify the object view within the object definition. The unique key must be 50 or fewer characters in length, cannot contain spaces, underscores, commas, punctuation marks, or any special characters.</p> <p>Once you save the object view the first time, you cannot change its unique key.</p>

		Object views within the same object definition cannot have the same unique key, but object views belonging to different object definitions can have the same key.
<b>New tab name</b>		Enter the name of a new page you would like to add to the object view. Click <b>add</b> to add the page.
<b>Reposition highlighted tab</b>		Click to reposition the selected tab to the left or right of its current position.  <i><b>Note:</b> Click a tab to select it. The selected tab does not have a border color; tabs that are not selected have a gray border around them.</i>
		Click to delete the selected tab, then click <b>OK</b> in the confirmation dialog box.
	<b>rename</b>	Click, enter a name in the pop-up window to rename the selected tab, and then click <b>OK</b> .
<b>Tab List</b>		Lists all tabs present in the object view, and their current order.  <i><b>Note:</b> Click a tab to select it. The selected tab does not have a border color; tabs that are not selected have a gray border around them.</i>
<b>Add the block</b>		Select a block to add.  <i><b>Note:</b> The options in this drop-down list include all system blocks and all custom blocks defined for the selected object.</i>
<b>Block List</b>		Lists all blocks present on the selected tab, in the current order.  <i><b>Note:</b> Click the <b>up</b> and <b>down</b> arrows next to the block name to reposition the block above or below its current position, or click <b>delete</b> next to the block name.</i>

### 3.1.6.5.1 Defining Object Views

Defining custom object views is the last step in the process of customizing what information different users or groups see in TeamConnect.

Custom object views typically consist of the system blocks and custom blocks. Before creating and assigning object views, you should consider the following questions:

- Have you defined all categories necessary for the object?

All custom fields and blocks are category-specific, which means they can be created and displayed by category only. For information on defining categories, see [Using Categories](#).

- Have you created all custom fields necessary for the object?

Custom fields are the central component of object views, as they are used to store information specific to your organization, and are used in blocks. For information on defining custom fields, see [Creating Custom Fields](#).

- If you are going to use custom blocks, have you created all custom blocks you need for the object view?

Blocks are sets of information that are included into different object views created for different groups of users. For information on defining blocks, see the Screen Designer Help.

If you answered yes to all questions that apply to your needs, you are ready to create your own custom object views. After you have defined your object views, you must assign them to the appropriate groups or the entire system.

#### 3.1.6.5.1.1 Creating Object Views

To create an object view, you need to name the view, add all necessary tabs to it, and lay out the blocks on each tab in the proper order.

***Tip:** Make a list of tabs you would like to include in the object view you are creating and a list of blocks that you want to include on each tab.*

#### To create a custom object view

1. Open the **Object Views** tab of the appropriate object definition.
2. Click **New**.

You can also create a copy of an existing object view by opening the appropriate object view of the displayed object definition, and then clicking Create a Copy.

The **General** section of the new object view appears.

3. In the **View Name** field, type a name for the new object view.
4. In the **Unique Key** field, type a name or value that will uniquely identify the view within the object definition.

An object view's unique key cannot contain spaces, underscores, punctuation marks, or any special characters.

5. Click **Save**.



The **Preview** section of the new object view appears.

6. On the **New tab name** field, type the name the tab that you would like to create in your custom object view, and click **add**.



7. In the **Add the block** drop-down list, select a block that you would like to add to the selected tab, and click **add**.

The selected block appears under the selected tab in the **Preview** section.

8. Repeat step 7 for as many blocks as you would like to add to the selected tab.
9. Position the blocks you have added in the desired order by clicking the up and down arrows (  ) next to each block.
10. Repeat steps 6 and 9 for all tabs you would like to add to your custom object view.
11. Position the tabs you have added in the desired order by selecting the appropriate tab and clicking the right and left **Reposition highlighted tab** arrows (  ).
12. Click **Save**.

To see your new object views in the displayed list, click **Refresh**.

To allow users to view and use your object views you must assign them to the designated groups or set the view as default in the system settings. See [Assigning Object Views](#).

## Points To Remember

Consider the following when creating custom object views:

- The more blocks you add to a tab, the longer it takes to open it.
- Each tab that you create appears in the end-user interface only when at least one block is added to the tab.
- Blocks for related objects are not displayed in the record until the user saves it first.
- The **General** block must be always added to the custom object view, as it typically contains the required system fields. Make sure to include all required fields in every custom object view.
- Because all custom components of an object view are category-specific, make sure you add the **Categories** block to the object view. Otherwise, the users are unable to add categories and consequently see their custom fields.

Task and Expense do not have the **Categories** block. Instead, there is a field called **Default Category** in their **General** block.

The Contact object presents its category-related custom field values in the **Details** page, not in the **General** page.

- If you want child or embedded object records to be displayed in the selected object's records, you must add the corresponding blocks to a tab in the object view and name it accordingly.

These blocks are automatically created when the child or embedded custom objects are created.

For example, if you want each parent Policy object record to have its child Claim records and its embedded object Coverage records displayed on the same tab, you must create this tab (for example, called Claims and Coverages) in the custom object view in the Policy object definition and add the blocks called Claim and Coverage to that tab.

**Important:** If you want system object records, such as appointments, accounts, tasks, and expenses, to be displayed in a project, you must create tabs or sections with the appropriate names and add the corresponding system blocks.

#### 3.1.6.5.1.2 Assigning Object Views

By default, all users see the system views for all objects as specified on the **Default Object Views** tab of the **System Settings** screen. After a custom object view is created, you must assign it either to a group or to the entire system before users are able to use that object view.

The object views that are set in the **System Settings** screen are going to be displayed for a user who:

- Does not belong to a user group
- Has no default group selected in his or her user account
- Has the **(Select a View)** option displayed next to the custom object in default **Object Views** tab of his or her default group.

**Note:** The default object views set in group accounts always take precedence over the system settings.

#### To assign object views to the system:

1. In the Designer window bar, in the **System Settings** drop-down list select **Default Object Views**.
2. Scroll to the object to which you want to assign a view and do one of the following actions:
  - If you want to set a certain custom view created for the object as default for the system, select the name of that view in the drop-down list next to the object.
  - If you want to set the system view of the object as default for the system, select that view in the drop-down list next to the object. For details on how to identify a system view in the list, see [Object View Names](#).
3. Repeat step 2 for all objects to which you want to change default views.
4. Click **Save**.

The selected settings take effect immediately, and the users who do not have a default group set in their user accounts see the object views assigned to the system.

#### 3.1.6.5.2 Troubleshooting Object Views

The following table presents common object view problems and their solutions.

##### Custom Object View Troubleshooting Tips

Possible problems	Possible solutions
-------------------	--------------------

The custom view that I created is not displayed.	<p>Check the following information:</p> <ul style="list-style-type: none"> <li>• The view is assigned either to the whole system or to the default group in your user account.</li> <li>• If you have assigned the view to a group, make sure the user account you are using for testing is a member of the group, and the group is set as default in this account.</li> </ul>
My custom blocks are not displayed in my object view.	<p>Check the following information:</p> <ul style="list-style-type: none"> <li>• Make sure the blocks are defined and added to the tabs in your view.</li> <li>• Make sure that the <b>Categories</b> block is added to the view.</li> <li>• Troubleshoot your custom blocks. See <a href="#">Troubleshooting Custom Blocks</a>.</li> <li>• Clear the cache. See <a href="#">Clearing Custom Blocks Cache</a>.</li> </ul> <p>See also <a href="#">Points To Remember</a>.</p>
Not all of my tabs appear on the page.	Make sure that each missing tab has at least one block added to it.
The blocks for related objects are not displayed in the record.	Save the record first and then check again.

### 3.1.6.5.3 Clearing Custom Blocks Cache

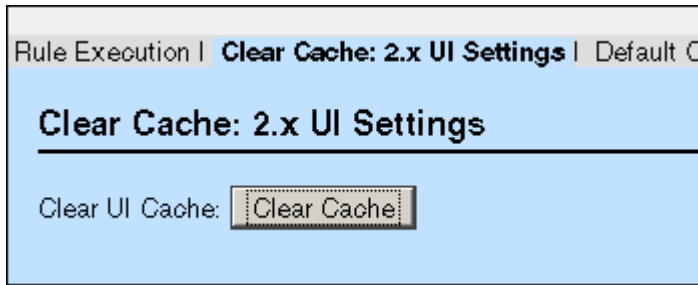
TeamConnect has a user interface caching mechanism that stores page designs and other user interface data accessed by the application. You should clear the cache whenever you make changes to blocks and object views, and after making other changes to the **Main** user interface.

**Important:** *If you make a change that requires clearing the cache but you do not clear the cache at that time, users do not see the change.*

#### To clear the custom blocks cache

1. In the Designer window, from the **System Settings** drop-down list, select **Clear Cache 2.x UI Settings**.
2. On the displayed tab, click the **Clear Cache** button.

The cache has been cleared and you can close the **System Settings** screen and continue with your work.



System User Interface Settings

#### 3.1.6.5.4 Customizing Invoice Line Item Views

For invoice records, there is an additional capability of defining two views for [invoice line items](#). Unlike other custom views, this capability is limited to a number of system fields that will be displayed in a predetermined layout, which means that you do not have to create any XML files or define custom object views. All you can do is to select the fields for each view, specify their default values, if any, and allow their values to be copied to new line items.

**Note:** *If your organization has a custom tool for creating additional custom fields for invoice line items, you may have additional capabilities that are out of scope of this documentation.*

You can customize the appearance of the **Line Items** block which can be placed in any Invoice Object View's tab. Users can also customize the location of the Line Item Block inside the Invoice Object View through the end-user interface in the Invoice Object Definition View. Here, users can create a new tab for the line item block or place the line item block in the **General** tab.

You can customize the fields that appear in the **Line Items** block on the user interface. Users have the ability to set field defaults and can make the view as similar to their business practice as necessary.

The following table describes the items in the **Line Items** customization tab.

**Line Items Customization Tab**

Field or column	Description
<b>Field</b>	The name of the field that appears on the <b>Line Items</b> tab in an invoice.
<b>Required</b>	Specifies whether the field is required.  Select the check-box for the <b>Activity</b> and <b>Timekeeper</b> fields to make these fields required for Tasks.  Important: If you have upgraded from a previous version of TeamConnect, leave the <b>Timekeeper</b> check-box selected to ensure backward compatibility with existing invoice records.
<b>Line Item Entry</b>	For each field, select whether to show it or hide it in the <b>Line Item Entry</b> screen.

<b>Table Display</b>	For each field, select whether to show it or hide it in the <b>Table Display</b> screen.
<b>Task Adjust</b>	For each field, select whether to show it or hide it in the <b>Batch Adjust</b> screen. For information on hiding certain fields in the In-Line & Bulk Adjustment screen, see <a href="#">Customizing Invoice Line Item Views for In-Line &amp; Bulk Adjustment</a> .
<b>Expense Adjust</b>	For each field, select whether to show it or hide it in the <b>Batch Adjust</b> screen. For information on hiding certain fields in the In-Line & Bulk Adjustment screen, see <a href="#">Customizing Invoice Line Item Views for In-Line &amp; Bulk Adjustment</a> .
<b>Default Value</b>	Select a default value for each field.
<b>Copy Capability</b>	For each field, select the check-box to indicate that the value of the field is copied from the previous line item to the new line item when a user clicks the <b>Copy</b> icon in the <b>Action</b> column.

## Customizing Invoice Line Item Views for In-Line & Bulk Adjustment

The behavior of line item views is different for [the In-Line & Bulk Adjustment screen](#).

In the In-Line & Bulk Adjustment screen, the **Date** field will not honor the Line Items Customization settings and will always display.

Additionally, the **Type**, **Category**, and **Timekeeper** fields will display in the In-Line & Bulk Adjustment screen unless both **Task Adjust** and **Expense Adjust** are set to **Hide** for each respective field.

### Search Views

There are two search views of the **Line Items** block that you can use to [view line item details](#)—*All Line Items* and *Line Item Warnings*. Users have the ability to toggle between these two views, set one as default, and can make the two views as similar or different as necessary. For more information on line item views, see the *User Guide*.

#### 3.1.7 Creating Search Views

To help users find records, you can create search views that define different combinations of search criteria fields and how the results appear.

Users search for records by entering search criteria in the resulting search screens. Lists of matching records appear with various columns of information to help users recognize and access the desired record. For example, on the **Tasks** search screen, users may select various search views from links in the left navigation pane to help them find their own pending tasks or others' overdue tasks.

TeamConnect contains many predefined search views for objects installed by default. It also allows you to customize the search views for each object to include a variety of fields, providing users with choices of how they want to search for and display records.

## Uses for Search Views

A custom-defined search view can be designated as a collection. When you create such a collection, the collection name appears as a link in the left navigation pane of the end-user interface. Clicking on that link runs the search and displays the search results in the work area of the user interface. In this way a search with complex filtering criteria can be reused quickly, saving the end user the trouble of entering those criteria each time. You can also define, and then assign search views to groups of users.

You may use search views not only for finding records, but also to create a summary of or report on records, which you may print directly from the search screen. For example, you may use a search view for creating a report of all expenses posted during the last 30 days.

When you define a search view, you must also specify where it may be used or accessed in any of the following ways: as a related object, through a portal pane, as a collection link for the object, and through a search module. This flexibility allows you to design different search views to meet the needs of users depending on where they need to search for records.

## Searching with Related Objects

When a search view is used with an object that may have other kinds of objects related to it, the search results are limited to records that actually do have related objects present. For example, this combination might happen if a custom object has a category associated with it, which specifies a relation with Involved objects.

If a specific record of that custom object type has the specified category assigned to it, but does not have any relationships with Involved records currently, that record is omitted from the search results.

### 3.1.7.1 Designing Search Views

Creating search views is one of the last steps in the process of defining objects in TeamConnect as described in [Customization Sequence](#).

The procedure for creating a search view is described in detail in [Creating Search Views](#).

When you are ready to create search views for an object, first determine the following characteristics for each search view you want to create:

- Where the search view should be available, for example, from multiple user-interface locations or a separate search view for each location. The available locations are:
  - Related objects
  - Portal panes
  - Search modules
  - Links in the left pane (for search views that are designated as collections)

In addition, one search view can be set to display Global Search results for the object.

- Whether to display search results automatically when the user selects the search view (Auto Search).
- Which fields are available as search criteria.
- Which qualifiers are invisible, so that records are filtered by certain invisible criteria in addition to the criteria entered by the user.
- Which section titles are needed for the **Filter** tab (the search criteria section).
- Which fields appear on the **Results** tab (the displayed records section for records found by the search) in addition to the object link.
- If the search view is for a contact or appointment, whether you want to use the predefined Card or Calendar display types.

All search views are created within the Designer object definition area for their objects in TeamConnect. Search views may be created, viewed, and modified in the **Search Views** screen, which contains the following tabs:

**General** | Filter Display | Results Display |  
**Search Screen Tabs**

- **General**—Contains general information about the search view, such as the name, view type, locations in TeamConnect where it is available, and whether it is set to auto search. For more details, see [Defining General Search View Information](#).
- **Filter Display**—Contains the qualifiers defined for the search view with which users can conduct a search. This tab also includes invisible qualifiers that automatically limit all search results, to help you control which records users can access. For more details, see [Search View Filter Display](#).
- **Results Display**—Contains the settings that determine how the search results appear for the user in the displayed records section or the **Results** tab of the search screen. For more details, see [Search View Results Display](#).

#### 3.1.7.1.1 Accessing the Search View Screen

##### To open a search view screen

1. Open the appropriate object definition, and then click the **Search Views** tab.

A list of all existing search views for the object appears, as shown in the following image.  
*This example is on the Contact Object.*

Search Views

Create Search View: [new](#)

View Name	Order	Used For
<input type="checkbox"/> <a href="#">AAContact Search</a>	10	Search Module(Person and company search)
<input type="checkbox"/> <a href="#">ALL Contact all fields</a>	10	
<input type="checkbox"/> <a href="#">All Contacts</a>	1	Object Collection
<input type="checkbox"/> <a href="#">Company Search Module</a>	1	Search Module(Company only search)
<input type="checkbox"/> <a href="#">custom fields hover over</a>	10	
<input type="checkbox"/> <a href="#">custom field sub category</a>	10	Custom Search
<input type="checkbox"/> <a href="#">Custom Search</a>	0	Custom Search
<input type="checkbox"/> <a href="#">Default Search Module</a>	0	Search Module(Person and company search)
<input type="checkbox"/> <a href="#">Demo Search view</a>	10	
<input type="checkbox"/> <a href="#">Global Search Default</a>	10	
<input type="checkbox"/> <a href="#">Global Search View</a>	10	Global Search
<input type="checkbox"/> <a href="#">komal hover over</a>	10	
<input type="checkbox"/> <a href="#">Law Firms</a>	4	Related Object, Object Collection
<input type="checkbox"/> <a href="#">Legal Department Staff</a>	5	Related Object, Object Collection
<input type="checkbox"/> <a href="#">Outside Counsel Attorneys</a>	6	Related Object, Object Collection
<input type="checkbox"/> <a href="#">Recently Created</a>	2	Object Collection
<input type="checkbox"/> <a href="#">Recently Modified</a>	3	Object Collection
<input type="checkbox"/> <a href="#">Search for Companies</a>	0	Portal Pane, Search Module(Company only search)
<input type="checkbox"/> <a href="#">Search for Contacts</a>	0	Related Object, Search Module(Person and company search)
<input type="checkbox"/> <a href="#">Search for Court</a>	13	Portal Pane, Related Object, Search Module(Company only search)
<input type="checkbox"/> <a href="#">Search for Judge</a>	10	Portal Pane, Related Object, Search Module(Person and company search)
<input type="checkbox"/> <a href="#">Search for Outside Counsel Attorney</a>	11	Portal Pane, Related Object, Search Module(Person and company search)
<input type="checkbox"/> <a href="#">Search for Staff</a>	12	Portal Pane, Related Object, Search Module(Person and company search)
<input type="checkbox"/> <a href="#">Search for Vendor</a>	14	Portal Pane, Related Object, Search Module(Company only search)
<input type="checkbox"/> <a href="#">subcategory hover over</a>	10	Custom Search
<input type="checkbox"/> <a href="#">system fields hover over</a>	10	Custom Search

[Check All](#) - [Uncheck All](#) [delete](#)

Use this search view's results display for global search:

Use this search view's results display for contact groups and recently viewed:

Use this search view for workflow approvals:

Use this search view for workflow processes:

Use this search view for hover over display popups:

Sample Search Views Tab Screen

- Click the hyperlink of the search view you want to open, or click **new** to create a new search view for the selected object definition.

The corresponding search view screen appears with its **General** tab (see [the General Tab on Search View Screens image](#)) displayed by default.



- Click the tab you need to open.

The following table describes the items in the **Search Views** tab.

**Search Views Tab Screen Elements**

Field or column	Description
<b>Create Search View:</b>	Click <b>new</b> to create a new search view. This displays a new <b>Search View</b> screen.
<b>View Name</b>	Displays the name of each search view as a hyperlink. Click the name of the search view to open it.
<b>Order</b>	<p>Displays the display order in which all search views for this object appear in the <b>Current View</b> drop-down list on the search screen.</p> <p>The first search view in the list is the search view that appears by default when a user first accesses a search screen.</p> <p>For details about setting the display order, see <a href="#">the General Tab on Search View Screens table</a>.</p>
<b>Used For</b>	Displays all areas where the search view is available, such as search modules, custom search, and portal panes. Not all objects allow search views to be used in all areas. For details about where search views can be used, see <a href="#">the General Tab on Search View Screens table</a> .
<b>Check All - Uncheck All</b>	<p>To delete a search view, select a single check-box next to the search view or click <b>Check All</b> to select all the search views on the page, and click <b>delete</b>.</p> <p>Click <b>Uncheck All</b> to deselect all the search views.</p>
<b>Use this search view's results display for search across objects</b>	<p>Select a search view from the drop-down list to display Global Search results.</p> <p>For system objects, the default search view is <b>Global Search Default</b>. For custom objects, TeamConnect uses the search view with the lowest display order. If more than one search view shares the same sort order, TeamConnect uses the first search view in alphabetical order.</p>
<b>Use this search view's results display for static object collections and recently viewed</b>	<p>Select a search view from the drop-down list as the default to display static object collections. (See <b>Object Collection</b> in <a href="#">the General Tab on Search View Screens table</a>)</p> <p>This search view's results display is also the display used for Recently Viewed records for this object. Recently Viewed is one of</p>

	<p>the available collections in the left navigation pane of the user interface.</p> <p><i>When working with search views for the Contact object, the label for this check-box will instead say "Use this search view's results for contact groups (Address Books) and recently viewed." A Contact Group is a form of static object collection.</i></p>
<b>Use this search view for workflow approvals</b>	<p>This field is visible for all objects that can have approval rules.</p> <p>Select a search view from the drop-down list to display on the <b>Requests to Approve</b> page of <b>My Approvals</b> and the <b>Active Requests</b> page of <b>My Requests</b>.</p> <p>The fields under the <b>Filter Display</b> tab of the search view display as the fields you use to enter search criteria. The fields under <b>Results Display</b> of the search view display as the columns in the table with the search results.</p>
<b>Use this search view for workflow processes</b>	<p>This field is visible for all objects that can have approval rules.</p> <p>Select a search view from the drop-down list to display on the <b>Attention Required</b> and <b>In Progress</b> pages of <b>My Workflow Processes</b>.</p> <p>The fields under <b>Filter Display</b> of the search view display as the fields you use to enter search criteria. The fields under <b>Results Display</b> of the search view display as the columns in the table with the search results.</p>
<b>Use this search view for hover over display popups</b>	<p>Select a search view here to display when items within the object are hovered over with the mouse.</p> <p>For more information on this field and setting up custom hover pop-ups, please see <a href="#">Custom Hover Popups</a>.</p>

### 3.1.7.1.2 Creating Search Views

The following instructions for creating a search view provide a general overview of the entire process. This process includes several general steps that are broken down into greater detail in other instructions, as indicated in the appropriate step.

#### To create a search view

1. Open the appropriate object definition, and then click the **Search Views** tab.
2. Click **New**.

The **General** tab appears.

3. Enter all general information about the search view on the **General** tab.  
For details, see [Defining General Search View Information](#).
4. Click the **Results Display** tab and do the following actions:
  - a. If you want users to be able to access records directly from the search results, create the **Object Link**.  
For details, see [Defining Object Links](#).
  - b. Design the other display columns for the results screen of the search view.  
For details, see [Creating Results Display Columns](#).
5. Click **Save** to save the search view. This allows you to add qualifiers.
6. Click the **Filter Display** tab and do the following actions:
  - a. Create all desired section titles for the search view.  
For details, see [Creating Section Titles](#).
  - b. Create all desired search qualifiers for the search view.  
For details, see [Creating Search View Qualifiers](#).
7. Click **Save**.

The search view is now available in the appropriate screens in the **Current View** drop-down list. After you save a search view, you may edit it as necessary.

#### 3.1.7.1.2.1 Creating Search Views for Home Pages

You can use a search view that you have created to display useful information to users in their TeamConnect **Home** page, such as the tasks to which they are assigned, their daily appointments, and projects for which they are the main assignee.

Search views that are created for use in home pages have several requirements:

- The **Portal Pane** check-box on the search views **General** tab must be selected. If this check-box is not selected, then the search view is not available in the **Content** section of the **Portal Pane** screen.
- Qualifiers with predefined values are required in order for the search view to display properly in the portal pane. If no search qualifiers are defined on the **Filter Display** tab of the search view, then the portal pane lists all available records in the database for that object. If desired, the qualifiers can be invisible.

For example, you might want the portal pane to list the current user's tasks to which they are assigned, rather than the tasks for all users.

- You should also take into consideration which extra qualifiers you want to have available when the user clicks **Edit Filter** in the portal pane, and make these qualifiers visible qualifiers.

For example, you might want to give the user the ability to further limit the Tasks that appear according to the date which they are due.

- You must ensure that each search view has an object link display key in order for users to have direct access to records from the search view.

**Card** and **Calendar** display types for the Contact and Appointment objects are not available for search views in home pages.

The following steps describe how to create a search view that you can use in a portal pane. To learn more about using search views in portal panes, see [Adding Search Views](#).

#### To create a search view that can be used for home pages

1. Open the appropriate object definition, and then click the **Search Views** tab.
2. Click new.  
The **General** tab appears.
3. On the **General** tab, do the following actions:
  - a. In the **Used For** list of check-boxes, select **Portal Pane**.
  - b. Enter all other general information about the search view on the **General** tab.  
For details, see [Defining General Search View Information](#).
4. Click the **Results Display** tab and do the following actions:
  - a. Create the **Object Link**.  
This is a necessary element for search views in portal panes in order for users to be able to access records displayed in the portal pane. For details, see [Defining Object Links](#).  
Do not confuse this with the **Object Link** type of portal pane content.
  - b. Design the other display columns for the results screen of the search view.  
For details, see [Creating Results Display Columns](#).
5. Click **Save** to save the search view. This allows you to define qualifiers.
6. Click the **Filter Display** tab and do the following actions:
  - a. **Create all desired section titles for the search view. The section titles appear when the user clicks Edit Filter in the Home page.**  
For details, see [Creating Section Titles](#).
  - b. Create all desired invisible qualifiers with predefined qualifier values for the search view. Invisible qualifiers automatically filter the records that appear on the user's Home page in the portal pane.  
For details, see [Visible and Invisible Qualifiers](#).
  - c. Create all desired visible qualifiers for the search view. Visible qualifiers are available when the user clicks **Edit Filter** in the Portal Pane.  
For details, see [Visible and Invisible Qualifiers](#).
7. Click **Save**.

The search view that you have created for the current object definition is now available for use when creating portal panes. For more details, see [Adding Search View Object Links](#).

#### 3.1.7.1.2.2 Custom Hover Popups

TeamConnect designers are able to create custom pop-up views for items on an object when hovered over. When creating these custom views, it is important to note that the hover is built entirely from the **Results Display** tab. Please see the walkthrough below for further detail.

<input type="checkbox"/>		Holden, Cody		
<input type="checkbox"/>		Inactive Co	Cody Holden	
<input type="checkbox"/>		Invoice, Inv	Driver's License: 0012345678	
<input type="checkbox"/>		Johnson, Ju	Employer: AfiaVendor Law Firm	
<input type="checkbox"/>		Judge, Mike	Preferred Phone: 111111111	

Customized Hover Pop-Up

### Creating a Custom Hover Over

**Note:** Existing search views can be used for custom hovers.

1. Click the **Setup** link in the upper-right corner of TeamConnect.
2. From the **Go to...** drop-down list, select **Object Definitions**.
3. Click the desired object for the custom hover. The custom hover can be applied to any system or custom object. Select the **Search Views** tab.
4. Create a new Search View.
5. From the General tab, ensure that none of the checkboxes under **Used for:** are selected. Enter in the desired name and unique key.
6. Click the **Results Display** tab. Add the desired system or custom fields to be viewed by entering column name, selecting the appropriate item from the dropdown, and adjusting the order as desired. More details can be found on [Search View Results Display](#). Further sorting details can be adjusted from the **Sort by:** dropdowns at the bottom of the screen.

General | Filter Display | Results Display |

Search Results

Number of entries you would like to add: 1 ▼

Column Name	Display Key	Format	Order	Span?	Column Width (%)
1 Preferred Phone	<input checked="" type="radio"/> System Default Phone <input type="radio"/> Custom (Select) (Select) ▼ <input type="radio"/> Related Object (Select) (Select) ▼	<input type="text"/> <input type="button" value="reset"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
1 Driver's License	driverLicense		0	NO	0
2 Employer	company		1	NO	0

-

Default Sort Order

Sort By: Driver's License ▼
  Ascending  
 Descending

Then Sort By: (None Selected) ▼
  Ascending  
 Descending

Custom Hover Results Display

7. After saving, navigate to the **Search Views** page. Select the created view for the **Use this search view for hover over display popups**.

Use this search view's results display for global search: Global Search View ▼

Use this search view's results display for contact groups and recently viewed: All Contacts ▼

Use this search view for workflow approvals: (None Selected) ▼

Use this search view for workflow processes: (None Selected) ▼

Use this search view for hover over display popups: Custom Hover Example ▼

Applying Selected Custom Hover

### 3.1.7.1.2.3 Specifying Search Views Used by Global Search

The *Global Search* feature in TeamConnect's user interface allows users to search for specific object types, **All** object types or across **All Projects**. [Global Search](#) searches all of the applicable fields for the user's search string, so you do not have to configure any search criteria for Global Search.

However, Global Search uses the **Results Display** configured for each object's search view.

When searching across **All** or **All Projects**, Global Search uses the search view specified in the following drop-down list on the object's **Search Views** tab to display its results:

### Use this search view's results display for search across objects

For system objects, the default global search view is **Global Search Default**. For custom objects, Global Search uses the search view with the lowest display order. If more than one search view shares the same sort order, Global Search uses the first search view in alphabetical order.

---

Global Search does not use the search view's **Filter Display**. Instead, it searches all of the objects' fields that are supported by Global Search. Memo text fields are also searched. For full-text searching, see [Full-Text Search Requirements](#).

**To specify a search view for Global Search**

1. Open the appropriate object definition, and then click the **Search Views** tab.

A list of all existing search views for the object appears.

Search Views

Create Search View: [new](#)

View Name	Order	Used For
<input type="checkbox"/> <a href="#">AAContact Search</a>	10	Search Module(Person and company search)
<input type="checkbox"/> <a href="#">ALL Contact all fields</a>	10	
<input type="checkbox"/> <a href="#">All Contacts</a>	1	Object Collection
<input type="checkbox"/> <a href="#">Company Search Module</a>	1	Search Module(Company only search)
<input type="checkbox"/> <a href="#">custom fields hover over</a>	10	
<input type="checkbox"/> <a href="#">custom field sub category</a>	10	Custom Search
<input type="checkbox"/> <a href="#">Custom Search</a>	0	Custom Search
<input type="checkbox"/> <a href="#">Default Search Module</a>	0	Search Module(Person and company search)
<input type="checkbox"/> <a href="#">Demo Search view</a>	10	
<input type="checkbox"/> <a href="#">Global Search Default</a>	10	
<input type="checkbox"/> <a href="#">Global Search View</a>	10	Global Search
<input type="checkbox"/> <a href="#">komal hover over</a>	10	
<input type="checkbox"/> <a href="#">Law Firms</a>	4	Related Object, Object Collection
<input type="checkbox"/> <a href="#">Legal Department Staff</a>	5	Related Object, Object Collection
<input type="checkbox"/> <a href="#">Outside Counsel Attorneys</a>	6	Related Object, Object Collection
<input type="checkbox"/> <a href="#">Recently Created</a>	2	Object Collection
<input type="checkbox"/> <a href="#">Recently Modified</a>	3	Object Collection
<input type="checkbox"/> <a href="#">Search for Companies</a>	0	Portal Pane, Search Module(Company only search)
<input type="checkbox"/> <a href="#">Search for Contacts</a>	0	Related Object, Search Module(Person and company search)
<input type="checkbox"/> <a href="#">Search for Court</a>	13	Portal Pane, Related Object, Search Module(Company only search)
<input type="checkbox"/> <a href="#">Search for Judge</a>	10	Portal Pane, Related Object, Search Module(Person and company search)
<input type="checkbox"/> <a href="#">Search for Outside Counsel Attorney</a>	11	Portal Pane, Related Object, Search Module(Person and company search)
<input type="checkbox"/> <a href="#">Search for Staff</a>	12	Portal Pane, Related Object, Search Module(Person and company search)
<input type="checkbox"/> <a href="#">Search for Vendor</a>	14	Portal Pane, Related Object, Search Module(Company only search)
<input type="checkbox"/> <a href="#">subcategory hover over</a>	10	Custom Search
<input type="checkbox"/> <a href="#">system fields hover over</a>	10	Custom Search

[Check All](#) - [Uncheck All](#) [delete](#)

Use this search view's results display for global search:

Use this search view's results display for contact groups and recently viewed:

Use this search view for workflow approvals:

Use this search view for workflow processes:

Use this search view for hover over display popups:

Sample Search Views Tab Screen

- Select the desired search view from the **Use this search view's results display for search across objects** drop-down list.



You can use a search view that you have created to add a search filter to the following pages in TeamConnect:

- **Requests to Approve** of **My Approvals**
- **Active Requests** of **My Requests**
- **Attention Required** of **My Workflow Processes**
- **In Progress** of **My Workflow Processes**

When you select a search view for these pages, the fields under the **Filter Display** tab of the search view display as the fields you use to enter search criteria. The fields under the **Results Display** tab of the search view display as the columns in the table with the search results.



**Requests to Approve**

Record: Invoice Default: All Records **Select** Hide Filter

Field	Operator	Value
Workflow Status	Is	(Any) Approved Cancelled Pending

**Search** Clear Values | Reset to Default

Requests to Approve 0 - 0 of 0

Record	Vendor	Invoice Date	Invoice Total	Request	Requester	Due Date	Last Approver	Action
No records available.								

Requests to Approve Page with a Search Filter

### To create search views for Approvals, Requests, and Workflow Process Pages

1. Open the appropriate object definition.
2. Click the **Search Views** tab.

A list of all existing search views for the object appears (see [the Sample Search Views Tab Screen image](#)).

3. Depending of the pages where you want to add the search view, select a search view from one of the following drop-down lists:

Page in TeamConnect	Search Views Drop-Down List
<ul style="list-style-type: none"> <li>• <b>Requests to Approve of My Approvals</b></li> <li>• <b>Active Requests of My Requests</b></li> </ul>	Use this search view for workflow approvals
<ul style="list-style-type: none"> <li>• <b>Attention Required of My Workflow Processes</b></li> <li>• <b>In Progress of My Workflow Processes</b></li> </ul>	Use this search view for workflow processes

#### 3.1.7.1.2.4 Search View Requirements for Full-Text Searching

Full-text searching allow users to find records even if they lack the name or number of a record. For example, the name of a city might be mentioned in the notes of the record. To find the record, the user would search by the content of the corresponding field or use Global Search.

To find a document that has been uploaded to the Documents area, users can search for specific text strings, such as a person's name that is mentioned in the document.

To allow users to search the contents of the following areas of TeamConnect, you must add the corresponding search qualifiers to your search views:

- **Notes** system field in system object records
- Custom fields of type Memo Text
- **Description** system field in History records
- Content of text-based documents that are uploaded to TeamConnect, such as:
  - Microsoft Word®, Excel®, PowerPoint®
  - Novell/Corel WordPerfect®
  - Adobe Acrobat® (PDF)
  - ASCII text files
  - HTML files
  - Other file types supported for text queries by Oracle Text or SQL Server Full-Text. For a detailed list, see your database server documentation.

**Note:** *Global Search does not support document searches. For more details, see [Specifying Search Views Used by Global Search](#).*

### Full-Text Search Requirements

To provide content searching capability to users, the following requirements must be met:

- You must configure and enable Oracle Text (Oracle interMedia) or SQL Server Full-Text Search on your TeamConnect installation. This ordinarily happens by default during installation.
- You must add the corresponding fields in search views as qualifiers, as explained in [Adding Document Content Searching to Search Views](#) and [Adding Field Content Searching to Search Views](#).

If you want to give users the ability to perform document content searching, you must provide a search view with this feature.

By default, a search view for the Document object definition, called **Content Search**, is provided with TeamConnect. However, you can also add the capability of searching by document content to your own custom search views, using the system field called **Document Body** as a search qualifier.

#### To add content search qualifiers to a document search view

1. Open the **Document** object definition and click the **Search Views** tab.
2. Do one of the following actions:
  - To change an existing search view, click the name of the search view to open it.
  - To create a search view, click **new** and do the following actions:
    - i. Complete the **General** tab.  
For details, see [Defining General Search View Information](#).
    - ii. On the **Results Display** tab, add at least one column to the results display.  
This allows you to save the search view. For details, see [Search View Results Display](#).
    - iii. Click **Save**.  
After the search view is saved, you can define search qualifiers.
3. Click the **Filter Display** tab and create a search qualifier using the system field **Document Body**.  
  
This field appears on the **Filter** tab or the search criteria section of the object's search screen so that users can enter the text content that they are seeking. For details, see [Creating Search View Qualifiers](#).
4. If you are using Oracle as your database server and want the **Score** field to be displayed on the **Results** tab or the displayed records section of the object's search screen, click the **Results Display** tab and select the system field **Document Body**.
5. Click **Save**.  
  
The content search qualifier is now available in the [search view](#).

To allow users to search according to the field contents, you can specify the following fields as search qualifiers on the Filter Display tab of a search view:

- **Description** field in history records
- **Notes** field in system records
- Custom fields of type Memo Text

Content search fields will be displayed in the search criteria section of the search screen so that users can type the text that they are seeking. A question mark (?) hyperlink displayed next to such fields allows users to select search operators.

You can also add the content search fields on the **Results Display** tab so that users can see the entire contents of the fields in the results. For details about field types and their storage sizes, see [Field Types](#).

#### To add a search qualifier for record content searching

1. Open the appropriate object definition and click the **Search Views** tab.

2. Do one of the following actions:
  - To change an existing search view, click its name to open it.
  - To create a search view, click **new** and do the following actions:
    - i. Complete the **General** tab.
    - ii. On the **Results Display** tab, add at least one column to the results display.
    - iii. Click **Save**.

Once you save the search view, you can define its search qualifiers.

3. Click the **Filter Display** tab and create a search qualifier using the following:
  - **Notes** system field in a system object
  - **Description** system field in the History object
  - Custom fields of the type Memo Text in a system or custom object

To verify that a custom field is a Memo Text field, locate it on the **Custom Fields** tab of the object definition and check its field type.

4. (Optional) Click the **Results Display** tab and add the field that you selected as the search qualifier.
5. Click **Save**.

The content search qualifier is now available in the search view.

#### 3.1.7.1.3 Defining General Search View Information

General search view information includes the name of the search view, the object for which it is defined, and other general details. These settings are found on the **General** tab of search view screens and determine which search view for an object appears by default in various screens.

#### To define general search view information

1. Open the appropriate object definition, and then click the **Search Views** tab.
2. Select the appropriate search view, or click **new** to create a search view.

The **General** tab of the search view appears.

**Search View Information**

Entity Name: Performance

\* View Name: Performance Search

\* Unique Key: per1

Used for:

- Custom Search
- Calendaring Application
- Integration
- Related Object
- Portal Pane
- Search Module
- Object Collection

Order: 10

Auto Search

(All Users) (All Users) Elite Workflow Process Manager

Subscribe | Unsubscribe

\* Required fields are indicated by an asterisk.

General Tab on Search View Screens

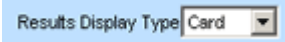
- Complete the fields in the **General** tab as described in [the General Tab on Search View Screens table](#).

You cannot save the search view until you create a results display, unless **Card** or **Calendar** is selected in the **Results Display Type** drop-down list for a contact or appointment search view.

Once you specify the general information for the search view, you can create the results display. See [Search View Results Display](#).

General Tab on Search View Screens

Field	Description
<b>Entity Name</b>	Automatically displays the name of the object for which this search view is defined.
<b>View Name</b>	<p>Enter a descriptive name for the search view as you want it to appear in the left pane of the corresponding search screen (maximum 250 characters).</p> <p><b>Note:</b> It is recommended not to create a name that includes special characters. For example: ?, \, /,  , :, &lt;, &gt;, *, %, #, &amp;, (, ), {, }, [, ], _ ". If the view is enabled for IMAP synchronization (IMAP Search), then special</p>

	<p><i>characters will be replaced by hyphens in the IMAP Client folder names. See <a href="#">Enabling IMAP Search for Custom Object Search Views</a>.</i></p>
<p><b>Unique Key</b></p>	<p>Enter an alphanumeric combination to uniquely identify the search view. Search views within the same object definition cannot have the same unique key, but search views belonging to different object definitions can have the same key.</p> <p>The unique key must be 50 or fewer characters in length and cannot contain spaces, underscores, commas, punctuation marks, or any special characters.</p> <p>After you save the search view, the unique key appears as read-only text and you cannot change it.</p>
<p><b>Results Display Type</b></p> 	<p>(Contact and appointment search views only)</p> <p>Select the type of results display for the search screen. The results display type only affects the way the search results appear, not search criteria.</p> <p>Most search views display the search results in the form of a list (also known as a tabular display), with a <b>Results</b> tab or a displayed records section that contains the list of results and a <b>Filter</b> tab or search criteria section that contains the search criteria. Contact and appointment search views support the following additional display types:</p> <ul style="list-style-type: none"> <li>• <b>Card</b> (for the Contact object)—All results appear in a grid format rather than a list. Each contact's information appears in one cell, instead of in a row across the screen.</li> <li>• <b>Calendar</b> (for the Appointment object)—Search results appear in the form of a calendar, with appointments displayed as links to the appointment records.</li> </ul> <p>The Appointment object, when invoked from the tab bar in the end user interface, will automatically determine which of its search views uses the Calendar results display type. It will open the Appointments page using that search view.</p> <p>You cannot design display columns for the <b>Card</b> or <b>Calendar</b> views. However, you can <a href="#">specify how many contact cards appear</a> on a page of results as a system-wide setting.</p>
<p><b>Used For</b></p>	<p><b>Custom Search</b></p> <p>Select this to make the search view a template for a custom search. In a custom search, the user determines which fields (out of the fields available in the search view) are used as criteria. There are several special considerations for a custom search template - for details, see <a href="#">Custom Search Templates</a>.</p>

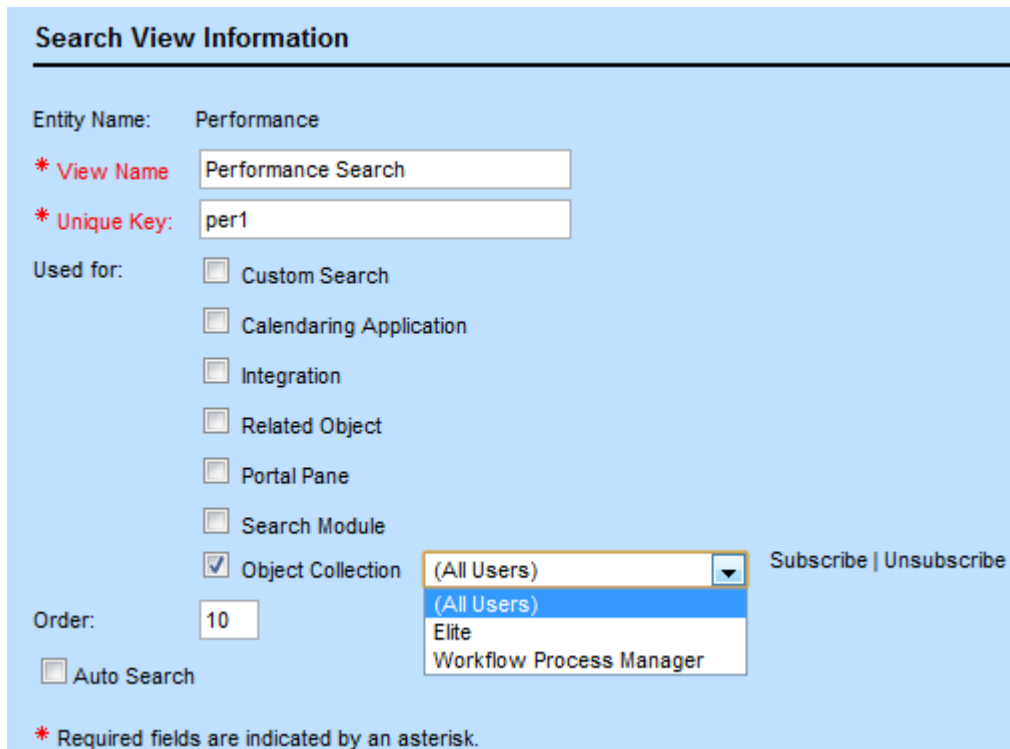
<b>Calendaring Application</b>	The use of calendaring applications is not currently supported in TeamConnect.
<b>Integration</b>	Select to make the search view visible in another application, such as Microsoft Outlook, allowing the user to select an individual record from the list of search results.
<b>Related Object</b>	<p>Select to make the search view available for searching from within a related object record. In these situations, the object for which you are creating the search view is considered to be a related object. This option is available in search views for all objects except for Contact.</p> <p>If your Filter Display tab uses the search criterion <b>Parent Project</b>, you cannot use this <b>Related Object</b> feature in the same search view.</p> <p><i>Tip: Create a search view with only <b>Related Object</b> selected and set the search view to <b>Auto Search</b>. This allows users to quickly see records related to the record they are working with, as the search is performed when they first access the block or tab.</i></p>
<b>Portal Pane</b>	<p>Select to make the search view available to be included in portal panes within home pages. For details, see <a href="#">Creating Search Views for Home Pages</a>.</p> <p><b>Card</b> and <b>Calendar</b> results display types cannot be displayed in portal panes. Therefore, when one of these is selected as the <b>Results Display Type</b> in a contact or appointment search view, the <b>Portal Pane</b> option is removed from the <b>General</b> tab.</p>
<b>Search Module</b>	<p>(Account, Contact, Project, and User search views only)</p> <p>Select to make the search view available in the search module for the object. For more details, see <a href="#">About Search Views in Search Modules</a>.</p> <p>Contact search views display two additional options when you select <b>Search Module</b>:</p> <ul style="list-style-type: none"> <li>• <b>Person and Company Search</b>—The search view is available to search for Person and Company contacts in all contact search module system fields except for the Company field in contacts. It is also available in custom fields of type Involved.</li> <li>• <b>Company Search Only</b>—The search view is available to search only for Company contacts from the <b>Company</b> field</li> </ul>

		of contacts and in custom fields of type Involved, but not from any other contact search module system fields.
<b>Used For</b>	<b>Object Collection</b>	<p>Select to make the search view available as one of the <b>Collection</b> links in the left navigation pane of the application's user interface. When regular users are looking at collections of records, this search view will be one of the links available to them. Clicking on the link will launch this search view.</p> <p>Users do not ordinarily get access to a collection unless they manually subscribe to the collection. However, you can give users access to an object collection as explained in the process <a href="#">Subscribing and Unsubscribing for Collections</a>.</p>
<b>Order</b>		<p>Type an integer to specify the display Order in which you want this search view to appear in the <b>Current View</b> drop-down list on the search screen, relative to other search views defined for the selected object.</p> <p>The search view with the lowest order is the first search view in the list, and appears when the user first opens the search screen. This is also the search view used by Global Searches for specific object records.</p> <p>If search views have the same display <b>Order</b> (for example, they are all set to 0), they are sorted alphabetically in the <b>Current View</b> drop-down list.</p>
<b>Auto Search</b>		Select this check-box to automatically do a search and display search results based on predefined criteria on the <b>Filter Display</b> tab. For more details, see <a href="#">About Auto Search</a> .

#### 3.1.7.1.3.1 Subscribing and Unsubscribing for Collections

When a search view's **Used for** field value is **Object Collection**, you may subscribe users to that collection. As shown in [the General Tab on Search View Screens image](#), when subscribing, you can choose to subscribe all users, or to subscribe users from a specific user group from a drop-down list.





**Search View Information**

Entity Name: Performance

\* View Name: Performance Search

\* Unique Key: per1

Used for:

- Custom Search
- Calendaring Application
- Integration
- Related Object
- Portal Pane
- Search Module
- Object Collection (All Users) Subscribe | Unsubscribe

Order: 10

Auto Search

\* Required fields are indicated by an asterisk.

General Tab on Search View Screens

Click the **Subscribe** link to apply your selection. You may also click the **Unsubscribe** link to remove the selected groups from that collection's subscription.

You can select multiple groups at the same time in the drop-down list when you are subscribing or unsubscribing. Additionally, if you add new members to a group, they are automatically subscribed to all assigned search views.

When you select **All Users** and then click a link, you get a warning that subscribing or unsubscribing all users might take a long time. Click **OK** in the warning box to continue the subscription process.

#### 3.1.7.1.3.2 About Auto Search

**Auto Search** automatically displays results when the user selects the search view. If the search view is the first in the **Current View** drop-down list (according to the display order of the object's search views), the search results appear as soon as the user accesses the search screen.

**Note:** Auto search has no effect on search views that have Used For values equal to Custom Search. Such search views will always pause for user edits and wait for the user to click the Search button. An end user, when saving a custom search, can choose whether or not to automatically execute the saved search when it is retrieved.

Auto search is useful for creating search views that always use the same qualifiers. For example, when a user is working with a project record and wants to see all related expense records, it is helpful to provide a search view that automatically displays the Expense records.

**Tip:** Using auto search is not recommended if there is typically a large number of results for the type of search, because the user cannot take any further action until the search is complete.

Auto Search can be useful in the following areas:

- For the default search view of an embedded object.
- For the default search view of a child object.
- For a search view that appears in a portal pane on a home page.
- For a search view that appears in a search module window.

#### 3.1.7.1.3.3 About Search Views in Search Modules

You can set a search view for Contact, Account, User, or any Custom Object (except for embedded Custom Objects) to be available in search module. These are the only objects whose records can be selected in search module fields. Making the appropriate search views available in a search module provides the user with the most helpful options for finding the necessary record.

To make a search view available to the user in search module fields, select the **Search Module** check-box on the **General** tab of the search view.

The following are important points about search views in search modules:

- Search views that you use in search modules should be designed with the needs of the user in mind when he or she is selecting a record in those particular fields. For example, in custom fields of type Involved (which display Contact search views), it may be very important that the user is able to search by contact categories.
- Certain fields must be added as search qualifiers on the **Filter Display** tab when a search view is made available in the search module. This is because these fields are used when a user performs a quick search directly from the search module field.

The following text strings are the search qualifiers required for search views in each type of search module:

- Contact Person and Company Search: **First Name Upper, Last/Company Name**
- Contact Company Only Search: **Company Name**
- Project: **Name Upper, Number**
- Account: **Name**
- User: **First Name, Last Name** (of contact)
- You cannot nest search modules within search modules. Search views in search modules must not have qualifiers which appear as search module fields, because TeamConnect does not display a search module from within another search module.

For example, you cannot add the **Vendor** field to the **Filter** tab or search criteria section of an account search view in a search module, because it is a contact search module field.

**Tip:** To verify that your search view is not displaying any search module fields, open the search module and select your search view, and look for any fields with the magnifying glass and folder icons next to them. These are search module fields.

- In search views for Contact that are available in the search module, you can display the **add person** and **add company** buttons, which allow the user to add contacts directly through the search module. For details, see [Search Module Actions in Contact Filter Display](#).
- The user search module displays search views that you define for the User system object, not the Contact object.
- System fields that appear as search module fields, such as the **Contact** field in an Involved record, display available search views to the user in the following manner:
  - If you make multiple search views available for use in the search module for an object, they become available in the **Current View** drop-down list of the search module for the user. The search views are sorted according to their **Order**. The first search view in the list appears in the search module by default.
  - If you only make one search view for an object available in the search module, then it appears automatically in the search module window where users access it through system fields.
  - You cannot specify a particular search view to be displayed for each individual system field.
- In custom fields that appear as search module fields, you have the following options:
  - You can specify that only one specific search view appears.
  - You can allow the user to select from all available search views in the **Current View** drop-down list, if desired.

For details about using search views in custom fields, see:

- [Search Views for Involved Custom Fields](#)
- [Search Views for Custom Object Fields](#)

#### 3.1.7.1.3.4 Enabling IMAP Search for Custom Object Search Views

Email messages can be shared between IMAP clients and TeamConnect Documents folders for custom object records (such as Dispute attachment folders in the Documents area). You must enable an IMAP Search option for a custom object definition's search views before corresponding TeamConnect-IMAP Mail folders for that search view's results become available in the IMAP client.

### Tips

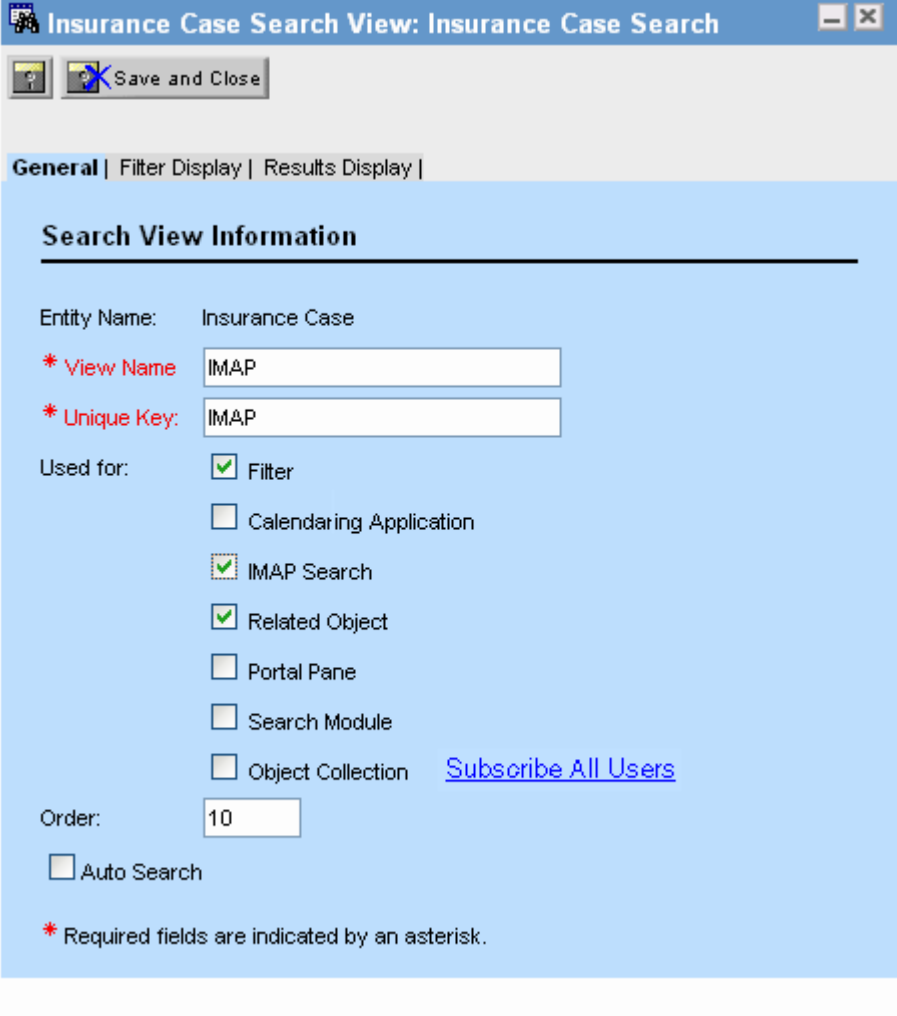
Keep the following tips in mind when enabling IMAP search for custom object search views:

- Only custom object search views may be configured to display IMAP folders.
- If a custom object has no search views with IMAP Search enabled, the corresponding custom records will not display as TeamConnect-IMAP Email folders from the IMAP client or TeamConnect views.
- If a custom object search view with IMAP enabled returns no record results, the record's TeamConnect-IMAP Email folder still displays.
- The current user only sees results from a custom object search view corresponding to records that the user has rights to see.

- Special characters (?, \, /, |, :, <, >, \*, %, #, &, (, ), {, }, [, ], \_, ") in custom object search view names or **Record Number** fields are replaced with hyphens (-) in the IMAP client view when IMAP Search is enabled.
- When using Lotus Notes as an IMAP client, do not to use underscores (\_) in search view names, project numbers, or IMAP subfolders because Lotus Notes removes the first underscore from the folder names.

#### To enable IMAP Search for Custom Object search views

1. From the **Go to** drop-down list, select **Object Definitions**.
2. Click the custom object (for example, Dispute).
3. Select the **Search Views** tab.
4. Click the Search View for which to create an IMAP folder.
5. From the **General** tab, under **Used for**, check **IMAP Search**.



Insurance Case Search View: Insurance Case Search

Save and Close

General | Filter Display | Results Display |

### Search View Information

Entity Name: Insurance Case

\* View Name: IMAP

\* Unique Key: IMAP

Used for:

- Filter
- Calendaring Application
- IMAP Search
- Related Object
- Portal Pane
- Search Module
- Object Collection [Subscribe All Users](#)

Order: 10

Auto Search

\* Required fields are indicated by an asterisk.

Creating an IMAP Search View

6. Click **Save**.

The result from the IMAP client view is that users who have rights to view records for that custom object are able to view IMAP folders for the record's search views in the following format:

```
\AccountName  
|---\ObjectType  
|---\ViewName  
|---\RecordNumber:PrimaryKey
```

Where:

- *AccountName* is the TeamConnect IMAP account name.
- *ObjectType* is a custom object type, for example Disputes.
- *ViewName* is the name of a Search View, for example MyDisputes.
- *RecordNumber:PrimaryKey* is the format of the record ID, for example DISP\_0007:57 or DISP\_0010:58.

#### 3.1.7.1.3.5 Projects and Third-party Calendars

Information about appointments and projects can be shared between TeamConnect and third-party calendaring applications, for example, Microsoft Outlook. When a user creates a TeamConnect appointment from a third-party calendaring application, the appointment first needs to be associated with a TeamConnect project.

From the TeamConnect project object definition, you must enable a search view type, **Calendaring Application**. Afterward, the project becomes available for selection from the third-party calendaring custom form for TeamConnect appointments.

#### To publish TeamConnect projects to Third-party Calendaring Applications

1. From the **Go to** drop-down list, select **Object Definitions**.
2. Click the custom object (for example, Dispute).
3. Select the **Search Views** tab.
4. Click the Search View to use to filter available projects for association with appointments created in 3rd party calendaring applications.
5. From the **General** tab, under **Used for**, check **Calendaring Application**.
6. Click **Save**.
7. The result from Microsoft Outlook is when users create appointments using the custom TeamConnect form, they are able to select TeamConnect projects to associate with the appointments.

### 3.1.7.1.3.6 Custom Search Templates

Any search view can be designated as a custom search template by setting its **Used For** value to **Custom Search**. This allows an end user to retrieve the template's information and use it to design an individualized search at runtime.

TeamConnect automatically creates a custom search template at the moment that a newly created object definition is saved. This default template contains most system fields, plus most of the custom fields that belong to the object definition's default category. If you change an object definition later to include more custom fields, you should revise the custom search template to contain the new fields that you want to allow to be searched.

It is strongly recommended that there be only one custom search template in existence at one time for a given object definition. When more than one template exists, and a user requests a custom search, the template that was created earliest will be chosen. If you wish to designate a search view as a template, and there is already an existing template, it is recommended that you first edit the search view that corresponds to the existing template, and uncheck every option in the "Used for" field of the **General** tab. That search view then will not be visible to custom search or any other operations. Then you may designate the new search view as a custom search template.

When a search view is used as a template, it is required that all of the "Used for" options, except **Custom Search**, be unchecked.

Columns that are defined in the **Filter** tab have an extra option, the **Include by Default** check-box. If checked, this means the column will appear as one of the choices that may be used as a criterion in a custom search. The end user may override this setting at runtime.

Columns that are defined in the **Results Display** tab have an extra option, the **Include by Default** check-box. If checked, this means the column will appear in the results of a custom search. The end user may override this setting at runtime.

A search view used as a template does not need, and will not use, Section Title information that is entered in the **Filter** tab of the search view definition.

**Note:** For Custom Search templates, it is important that each column in the Filter tab, and each column in the Results Display tab, has a unique value in its "order" attribute. For example, if you give two Filter columns the same value in "order", only one of those columns will appear at runtime to the end user.

### 3.1.7.1.4 Search View Results Display

The **Results Display** tab of the Search View screen allows you to specify how an object's search results appear to users. The displayed records section or **Results** tab can display the content of fields from each record that meets the search criteria. For example, search results could display each record's name, number, and the main assignee, as shown in [the Displayed Records Example image](#).

You can set search results to display as vertical columns or as horizontal columns that span the width of the results display. You cannot set all columns to span - at least one column must remain vertical.

**Note:** The **Results Display** tab is not visible for search views whose **Used For** selections (in the Filter Display tab) are limited only to "Filter".

Search results provide links to the listed records. The link that users click to open each record is called the object link. One record is listed on each line of results.

The object link always appears automatically in the search results. On the **Results Display** tab of the search view, you specify the layout of the other columns (not rows) that display the information for each record, as shown in the following image

General | Filter Display | **Results Display**

**Search Results**

Number of entries you would like to add:

Column Name	Display Key	Format	Order	Column Width (%)
1 <input type="text"/>	<input checked="" type="radio"/> System <input type="text" value="(Select)"/> <input type="radio"/> Custom <input type="text" value="(Select)"/> <input type="text" value="(Select)"/>		<input type="text"/>	<input type="text"/>
<a href="#">+ add more</a>				
Column Name	Display Key	Format	Order	Column Width (%)
1 <input type="checkbox"/> Invoice Date	invoiceDate	Date only	3	10
2 <input type="checkbox"/> Invoice Number	Object Link		1	40
3 <input type="checkbox"/> Invoice Total	netInvoiceTotal	Currency	4	10
4 <input type="checkbox"/> Status	postingStatusIIDString		5	10
5 <input type="checkbox"/> Vendor	vendor		2	30

[Check All](#) - [Uncheck All](#) [edit](#) [delete](#)

**Results Display Tab with a Display Columns List under Display Key**


**Note:** If the search view is for the Contact or Appointment object and you have selected **Card** or **Calendar** in the **Results Display Type** drop-down list on the **General** tab, you cannot modify the way results appear.

The appearance of search results is also affected by [system-wide settings](#) and [each individual user's Preferences](#) settings.

The following table describes the items on the **Results Display** tab.

#### Results Display Tab

Field or column	Description
<b>Column Header</b>	Type a label (maximum 250 characters) that describes to the user the record information that appears. This label appears at the top of the column in the displayed records section or the <b>Results</b> tab of the object's search screen.
<b>Display Key</b>	<p>Select system or custom fields as display keys for the results of the search screen.</p> <p>Display keys help users identify records returned by the search. The values in these fields are pulled from the database and appear in the search results on the object's search screen, giving users a summary or "preview" of each record.</p> <p>To specify a display key, select one of the following elements:</p>

	<ul style="list-style-type: none"> <li>• <b>System</b>—Allows you to select a system field from the drop-down list. <ul style="list-style-type: none"> <li>○ When you select a system field that links to another object table in the object model, an Object Navigator field appears. You can use it to add attributes of related objects and sub-objects as display keys. For example, in an Invoice search view, you can include the <b>Service Date</b> field from a line item. For more details about using Object Navigator, see <a href="#">Using Object Navigator</a>.</li> <li>○ When you select a system field that can return multiple values for a record, another drop-down list appears. You can use this drop-down list to be more specific about the values that display in the search results.</li> </ul> </li> <li>• <b>Custom</b>—Allows you to select a category and a custom field name from two drop-down lists. <p>When you select a custom field of type Custom Object or Involved, an Object Navigator field appears. You can use it to add custom fields from other custom objects or Involved records as display keys. For example, if Claim is a child of Policy, you can use Policy custom fields from any category as display keys in a Claim search view.</p> </li> <li>• <b>Related Object</b>—Allows you to select a child or embedded object from the drop-down list. <p>After you select an object, click the Object Navigator  icon, and select an attribute of that object to be a display key.</p> <p>If you want users to access records directly from the search results, one of the display keys must be defined as an object link. See <a href="#">Defining Object Links</a>.</p> </li> </ul>
<b>Format</b>	<p>Specify a format for number, category, or date fields that appear in the search results.</p> <p>When selecting a date field, you can select <b>Date and time</b> or <b>Date only</b>. If you do not select a date format, the date appears in date only format.</p> <p>When selecting the <b>Default Category</b>, <b>Categories</b>, <b>Default Role</b>, or <b>Roles</b> field, <b>Category Description</b> is the default format, which means the category or role name appears in the search results (for example, L110). If you specify <b>Category Full Hierarchy</b>, the full hierarchy of the category or role appears in the search results (for example, Outside Counsel Fees &gt; Litigation Code Set &gt; L100 &gt; L110).</p> <p>When selecting a field that displays a number, you can specify <b>Currency</b>, <b>Decimal</b>, or <b>Percent</b>. If you select <b>None</b> or make no selection, the number appears without any formatting.</p>



	<p><b>Important:</b> The <b>Currency</b> format uses the symbol for the system default currency to display the number values in the search results, even if a particular numeric value is actually in a different currency.</p>
<b>Order</b>	<p>Type an integer for the order in which this column is to be displayed, lowest to highest being displayed from left to right, in the search results.</p> <p><b>Warning:</b> If multiple columns have the same display order value, only one will appear at runtime. Be sure to use unique values for <b>Order</b>.</p> <p><b>Tip:</b> If you set the display <b>Order</b> for your columns in increments of 5, you can then easily make slight modifications to the order when necessary without having to recreate the order structure.</p>
<b>Span?</b>	<p>If checked, the selected column is displayed as a horizontal field that spans the width of the results display. The appearance of spanned columns depends on the information in the column.</p> <p>The span feature applies to the search result pages for the following:</p> <ul style="list-style-type: none"> <li>• Global Search</li> <li>• Recently Viewed Items</li> <li>• Workflow Approvals</li> <li>• Custom Search</li> <li>• Related Objects</li> <li>• Portal Panes</li> <li>• Object Collections</li> </ul> <p><b>Note:</b> You cannot set all columns to span—at least one column must remain vertical. Additionally, if you need to sort information in a results display, spanned columns are not included in the sort.</p>
<b>Column Width (%)</b>	<p>Enter the percentage for this column out of the total width of the results display.</p> <p>If the total of all column widths of the defined results display columns does not equal 100 percent, then you no longer have control over the column widths. They are resized automatically based on their contents, browser behavior, and other factors.</p>

	To allow the columns to be sized automatically according to the contents, leave this field blank when defining all columns in the results display.
<b>Display Columns List</b>	<p>Displays a list of all columns that have been defined for the search view results.</p> <p>To edit or delete display column definitions, select the check-boxes next to the display column definitions and click <b>edit</b> or <b>delete</b>.</p> <p>Items in the <b>Display Key</b> column cannot be edited.</p> <p>If the qualifier was selected with Object Navigator, the first 90 characters of the Object Navigator path appear in the Attribute column. For details on paths, see <a href="#">Using Object Navigator</a>.</p>

### 3.1.7.1.4.1 Displayed Records Section Examples

## Object Search Screen

The display column definitions shown in [the Results Display Tab with a Display Column List under Display Key image](#) produce the displayed records section of the object's search screen, as shown in the following image. The Results Display tab also contributes to the following parts of the image:

- The column headers and display order of the columns.
- The object links that allow the user to link directly to each record that matches the search criteria.
- Display keys that translate into specific information about each record that helps the user to identify it.

Invoice Number	Vendor	Invoice Date	Invoice Total	Status
<a href="#">0624_01</a>	<a href="#">The Bakery</a>	06/24/2008	\$40.00	Not Posted
<a href="#">06302008</a>	<a href="#">The Bakery</a>	06/30/2008	\$1,240.00	Not Posted
<a href="#">inv0617_01</a>	<a href="#">Holt, Nan</a>	06/18/2008	\$28.00	Not Posted
<a href="#">inv0619_01</a>	<a href="#">The Bakery</a>	06/19/2008	\$170.00	Not Posted
<a href="#">inv0620_01</a>	<a href="#">The Bakery</a>	06/20/2008	\$80.00	Not Posted
<a href="#">inv0620_02</a>	<a href="#">The Bakery</a>	06/20/2008	\$80.00	Not Posted
<a href="#">inv0620_03</a>	<a href="#">The Bakery</a>	06/20/2008	\$42.00	Not Posted
<a href="#">inv0620_05</a>	<a href="#">Artifacts R Us</a>	06/20/2008	\$28.00	Not Posted
<a href="#">inv0624_02</a>	<a href="#">The Bakery</a>	06/24/2008	\$195.00	Not Posted
<a href="#">inv0624_03</a>	<a href="#">The Bakery</a>	06/24/2008	\$147.00	Not Posted

Displayed Records Example

## Results Display with Spanned Columns

Date	Type	Category	TK	Rate	Units	Disc	Adj	Amt
Mar 1, 2011	Fee	1310 Written Discovery	Lee, Deanne	£300.00	0.20	£0.00	£0.00	£100.00
<b>Description:</b> Attention to e-mails and correspondence regarding discovery, new trial date.								
Mar 11, 2011	Fee	1320 Analysis/Strategy	Lee, Deanne	£335.00	0.60	£0.00	£0.00	£201.00
<b>Description:</b> Email correspondence with persons in charge regarding positive results from grand jury; email correspondence regarding same; telephone conference with								
Mar 11, 2011	Fee	1310 Written Discovery	Lee, Brendon	£335.00	0.20	£0.00	£0.00	£67.00
<b>Description:</b> Reviewed verifications received; drafted and revised letter to Plaintiffs' counsel enclosing verifications.								
Mar 14, 2011	Fee	1360 Settlement/Non-binding ADR	Lee, Brendon	£500.00	0.40	£0.00	£0.00	£200.00
<b>Description:</b> Attention to e-mails regarding proposed mediators, scheduling issues.								

Spanned Columns Example

### 3.1.7.1.4.2 Defining Object Links

The *object link* is an essential component of search views that is designed to provide users access to records. It provides a hyperlink on the displayed records section that allows users to link directly to records that are retrieved by the search. Without an object link, records appear on the displayed records section of the object's search screen, but users cannot access them.

However, object links are not necessary in search views that are designed for the purpose of creating reports. For example, you might want to use a search view for creating a list of all currently open Matter records, and you do not need a link to records in the resulting list.

Object links appear as the name, description, or number for the record, depending on the object. See the following table for a complete list of which field appears as the object link for each object.

Object Links in Search Views

Object		Field displayed as object link
Account		Name
Appointment		Subject and Location
Contact	Person	Last name, First name
	Company	Company Name
Contact Group (Address Book)		Group name
Document		Name
Expense		Description
Group		Group Account

<b>History</b>	Description
<b>Invoice</b>	Invoice number
<b>Project (custom object)</b>	Name, Number, or both, depending on the setting of the <b>Auto Numbering Pattern</b> on the <b>Unique ID</b> tab of the object definition.
<b>Task</b>	Subject
<b>User</b>	Last name, First name [of contact record]

#### 3.1.7.1.4.3 Creating Object Links

##### To create an object link for a search view

1. Open the appropriate object definition, and then click the **Search Views** tab.
2. Select the appropriate search view.
3. Open the **Results Display** tab.
4. Select **System**.
5. In the drop-down list, select **Object Link**.
6. Complete the rest of the fields as indicated in [the Results Display Tab table](#).
7. Click **add more**.

The object link is added to the **Display Columns List**.

8. Click **Save** to save the object link for the search view.

The next time a user opens the search view in a search screen, this object link appears on the displayed records section or the **Results** tab of the object's search screen.

#### 3.1.7.1.4.4 Creating Results Display Columns

##### To create display columns for a search view

1. Open the appropriate object definition, and then click the **Search Views** tab.
2. Select the appropriate search view.
3. Open the **Results Display** tab.

4. Select the **Number of entries you would like to add** from the drop-down list.
5. For each data entry row, complete all appropriate fields as described in [the Results Display table](#).
6. Click **add more** to add the columns to the list and continue adding more display columns. Otherwise, click **Save**.

The next time a user opens this search view, the display columns appear on the displayed records section or the **Results** tab of the search screen.

#### 3.1.7.1.5 Search View Filter Display

Search qualifiers appear as criteria on the **Custom Search** page to allow the user to enter search criteria when conducting a search. You need to create a qualifier for each field you want to include in the search view's search criteria. The more qualifiers you add when creating a search view, the more types of information users are able to search by to find the records they need.

You can define search qualifiers and the appearance of the **Custom Search** page on the **Filter Display** tab of the search view.

You can add the following items to the search view on the **Filter Display** tab:

- Visible qualifiers (see [Visible and Invisible Qualifiers](#))
- Invisible qualifiers (see [Visible and Invisible Qualifiers](#))
- Section Titles (see [Creating Section Titles](#))

**Note:** *If no qualifiers are specified in the search view, then no records are filtered out of the search results. Instead, all records for the object are listed.*

#### 3.1.7.1.5.1 Visible and Invisible Qualifiers

Search qualifiers are usually displayed as search criteria fields on the **Filter Display** tab of the search view. These qualifiers are called visible qualifiers, and are available to the user. For example, in a search view for Invoice records, a user could search by **Invoice Date**, **Posting Status**, **Invoice Number**, and the **Service Date** of a line item. To make a qualifier visible, select the **Visible** check-box when defining the qualifier on the **Filter Display** tab of the search view.

However, in certain search screens, you might want to limit the scope of the user's search results using qualifiers that are not available on the **Custom Search** page. This is done by creating invisible qualifiers. Invisible qualifiers cannot be accessed by the user, because they are not displayed in the search criteria. An invisible qualifier is applied to all searches performed with this search view.

For example, you might want users who search for Matter records to see only records they are assigned to in the search results. This would require creating an invisible qualifier that has the field name assigneeList\_User and the condition value Current User (see the search qualifier listed in [the Filter Display Tab on Search View Screens image](#)). Only records that are assigned to the user who is currently performing the search appear in the search results.

The qualifier with the label **Created this month** in [the Filter Display Tab on Search View Screens image](#) is another example of an invisible qualifier. This qualifier automatically limits the search results

so that only records that have been created during the last 30 days appear on the **Results** page of the user's search.

To make a qualifier invisible, clear the **Visible** check-box when defining the qualifier on the **Filter Display** tab of the search view.

**Important:** If no condition value is entered, an invisible qualifier has no effect.

### 3.1.7.1.5.2 Qualifier Conditions

Qualifier conditions are composed of two parts: an operator and a value.

- The operator lets the user specify what kind of limit to use when searching. For example, operators for a date field include **More Than X Days Ago** and **Between Dates**.
- The value is used with the operator. For example, you can enter **5** for the number of days in a date field search.

When creating a search view, the **Condition** fields on the **Filter Display** tab give you the ability to select which operator should appear by default when the user accesses the search criteria of the object's search screen. You might use this function, for example, when you want the **Contains** operator to be displayed by default for a Text custom field.

If needed, you can also select a default value for the operator, such as the name of a user, a number of days, or a dollar amount. Typically, you need to select a condition value only when creating an invisible qualifier. However, you can provide default selections in a visible qualifier as well. If you select a default operator or value for a visible qualifier, then the [user is able to make different selections](#) if desired. The following image shows examples of conditions.

Label	Field	Condition	Row Order	Column Visible
1	Intensive Tasks	Greater Than 50	1	YES
1	Date Served	Between X and Y Days From Now	6	YES
2	Docket Number	Contains	3	YES
3	Due Date	Between X and Y Days From Now	5	YES
4	Issuing Court	Contains	2	YES
6	Party Serving Subpoena	Contains	8	YES
6	Related Dispute Matter	SECURED	7	YES
7	Responded Date	Between X and Y Days From Now	9	YES
8	Subpoena Category	Subpoena	0	NO
9	Subpoena Matter Name	Contains	1	YES
10	Type of Subpoena	(Select) Order to appear Order to produce documents	4	YES

Filter Display Tab on Search View Screens

### Filter Display Tab on Search View Screens

Field or column	Description
<p><b>Label</b></p>	<p>Type a name (maximum 250 characters). This name appears as a label for the field on the <b>Filter</b> tab or search criteria section of the search screen. It indicates to the user what information to enter in the field for the search criteria.</p> <p><i>Note: A colon (:) is automatically placed after the label text in the search view.</i></p>
<p><b>Field</b></p>	<p>Do one of the following to select a field that will appear on the <b>Filter</b> tab or search criteria section of the search screen as a search qualifier:</p> <ul style="list-style-type: none"> <li>• <b>System</b>—Allows you to select a system field from the drop-down list.</li> </ul> <p>When you select a system field that links to another object table in the TeamConnect object model, an Object Navigator icon and field appear. You can use these to add attributes of related and sub-objects as qualifiers. For example, in an account search view, you can include the company of the involved party who can post transactions to the account. For more details, see <a href="#">Using Object Navigator</a>.</p> <p>You can also select <b>System</b> to create a section title that appears on the <b>Filter</b> tab or the search criteria section of the search screen. For details, see <a href="#">Creating Section Titles</a>. Upon clicking <b>System</b>, you are given the option of how to view the field on the <b>Filter</b> tab or search criteria section of the search screen:</p> <ul style="list-style-type: none"> <li>○ Drop-down list</li> <li>○ Multi-select list</li> </ul> <p>Choose the option that fits best with your user interface.</p> <p><b>Custom</b>—Allows you to select a category and a custom field name from two drop-down lists.</p> <p>When you select a custom field of type Custom Object or Involved, an Object Navigator field appears. You can use it to add custom fields from other custom objects or Involved records as qualifiers. For instance, if Claim is a child of Policy, you can use Policy custom fields from any category as qualifiers in a Claim search view.</p> <p>For many fields, you are given the option of how to view the field on the <b>Filter</b> tab or search criteria section of the search screen:</p> <ul style="list-style-type: none"> <li>○ Drop-down list</li> <li>○ Multi-select list</li> </ul>

	<ul style="list-style-type: none"> <li>○ Search field</li> <li>○ Multi-select search field</li> </ul> <p>Choose the option that fits best with your user interface. The "search field" options use auto-suggest; after the user types two or more characters in the field, a list of matching records (up to a maximum of 10) appears next to the field and the user can click on a list entry to select it. The user can also click the search icon to launch a separate page on which they will select a value for the field.</p> <p><b>Related Object</b>—Allows you to select a related, child, embedded, or sub-object from the drop-down list. Then, use Object Navigator to select a field from that object as a qualifier.</p> <p><b>Caution:</b> <i>If the search view is set to be available in the search module, do NOT add search qualifiers that also use the search module, such as certain custom fields of type Custom Object or Involved. These fields will not be displayed properly in the search module window. For more details, see <a href="#">About Search Views in Search Modules</a>.</i></p>
<b>Condition</b>	<p>If desired, create a default condition for the qualifier. For most fields, you can select an operator only, or an operator with a value.</p> <p>If you do not specify an operator or value in the condition, and the <b>Visible</b> check-box is selected for the qualifier, then the default operator appears in the search criteria of the object's search screen. The user is able to set the operator and value as desired.</p>
<b>Row Order</b>	<p>Type an integer for the <b>Row Order</b> in which the qualifiers appear. Qualifiers are listed vertically in the search criteria of the object's search screen, using the order you specify. The display order should be useful for users when they are entering search criteria.</p> <p><b>Warning:</b> <i>If multiple qualifiers have the same row order, only one will appear at runtime. Be sure to use unique values for Row Order.</i></p>
<b>Column</b>	<p>Select the column in which the search qualifier will be listed in the search criteria of the object's search screen. Choose Column 1 to display the qualifier in the left column of the screen; choose Column 2 to display the qualifier in the right column of the screen. The ability to choose which column the search qualifier filter will reside in gives you greater control of the screen elements and offers the ability to manipulate the layout of the object's search screen.</p>
<b>Visible</b>	<p>Select whether the qualifier for the search view is visible. If the qualifier is not visible, users do not see the field and are unable to enter criteria</p>

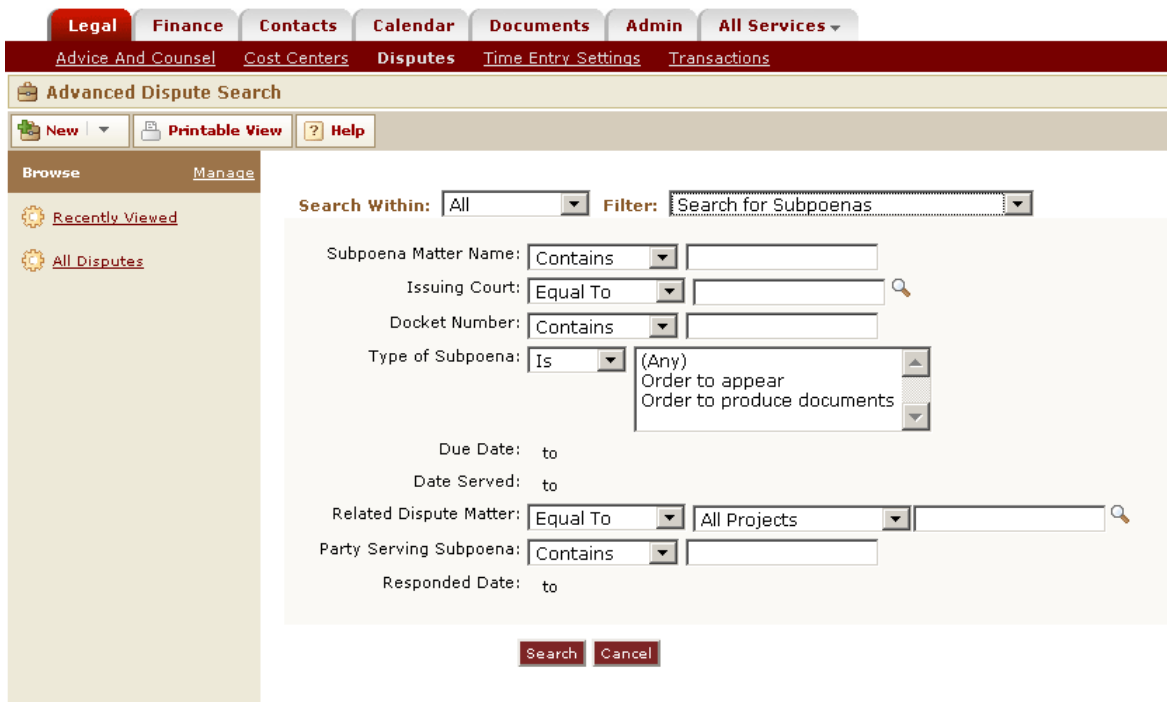


	<p>in this field on the <b>Filter</b> tab or the search criteria section of the object's search screen.</p> <p><b>Important:</b> Invisible qualifiers must have a condition value.</p> <p>For details, see <a href="#">Visible and Invisible Qualifiers</a>.</p>
<p><b>Search Qualifiers List</b></p>	<p>Displays a list of existing qualifiers for the search view. Select the check-box next to a search qualifier and click <b>edit</b> or <b>delete</b> to modify any of its items—except for the <b>Field</b> column, which cannot be edited.</p> <p>If the qualifier was selected with Object Navigator, the first 90 characters of the Object Navigator path appear in the Attribute column. For more details on paths, see <a href="#">Using Object Navigator</a>.</p>

3.1.7.1.5.3 Filter Tab Example

The search qualifiers shown in [the Filter Display Tab on Search View Screens](#) produce the **Filter** tab or the search criteria section of the object's search screen, as shown in the following image. Notice the section titles and the vertical display order of the search qualifiers. Also notice that the invisible qualifiers are not displayed.

When list fields that are associated with a lookup table are added as search qualifiers, the fields appear as combo list boxes to allow for multiple selections.



Filter Tab Example with Visible Search Qualifiers

## 3.1.7.1.5.4 Search Module Actions in Contact Filter Display

The contact search module provides users the ability to add a new contact directly through the search module window. If you want users to create contacts this way, you can indicate this by using the settings in the **Search Module Actions** section on the **Filter Display** tab of the contact search view, as shown in the following image.



The **Actions** section is only displayed if you have indicated on the **General** tab that the search view should be available in the search module. It appears below the **Search Qualifiers** section (as shown in [the Filter Display Tab on Search View Screens image](#)) and provides the options for displaying the **add person** and the **add company** buttons.

If **Company Search Only** is selected on the **General** tab of the search view, then you do not have the option of displaying the **add person** button to the user. The search view is available to users only through the **Company** field of a contact record, which is intended for finding and selecting only company contacts.

If you provide the capability to add a contact through the search view in the search module, consider the following information:

- The **add person** and **add company** buttons appear for the user only when the search view appears in the search module. If the search view is set to be available in other areas, such as search modules or portal panes, the buttons do not appear when the search view appears in those areas.
- When a user adds a contact through [the search module](#), only certain values that the user can provide on the **Filter** tab or the search criteria section of the object's search screen are saved in the new contact record, even if additional fields are available on the **Filter** tab or search criteria section. For details on adding contacts from search modules, see the User Guide.
- If there are any required custom fields for contacts, users will not be able to add contacts through the search module when the category for those custom fields is indicated as the default category of the contact they want to add.
- Any rules that are triggered on the creation of a contact record will also be triggered when the user clicks **add person** or **add company**.

The following table describes the items in the **Search Module Actions** section on the **Filter Display** tab of the contact search view.

Search Module Actions Section of Contact Search View

Field		Description
Allow user actions	Add Person	Select to add the <b>Add Person</b> button to the Contact search screen. This button allows users to add a person's contact record using the values entered on the

		<p><b>Filter</b> tab or the search criteria section of the search screen.</p> <p>If <b>Company Search Only</b> is selected on the <b>General</b> tab of the search view, then this option is not available on the <b>Filter</b> tab or the search criteria section of the search module.</p>
	<p><b>Add Company</b></p>	<p>Select to add the <b>New Company</b> button to the Contact search screen. This button allows users to add a Company contact record using the values entered on the <b>Filter</b> tab or search criteria section of the search module.</p>

#### 3.1.7.1.5.5 Creating Search View Qualifiers

Before creating search view qualifiers for your search screen, make sure to plan which qualifiers you want to include in the search view, including visible qualifiers, invisible qualifiers, and section titles. This helps you set the correct display order for each qualifier.

Search views return all subcategories in the hierarchy of a category search. Because child categories contain their parent's category, a search that returns a parent category shows all of its child items as well.

#### To create search view qualifiers

1. Open the appropriate object definition, and then click the **Search Views** tab.
2. Do one of the following actions:
  - To change an existing search view, click the name of the search view to open it.
  - To create a search view, click **new** and do the following actions:
    - i. Complete the **General** tab. For details, see [Defining General Search View Information](#).
    - ii. On the **Results Display** tab, add at least one column to the results display. This allows you to save the search view. For details, see [Search View Results Display](#).
    - iii. Click **Save** to save the search view.
 

After the search view is saved, you can define search qualifiers.
3. Click the **Filter Display** tab.
4. In the data entry rows, enter the appropriate information as described in [the Filter Display Tab on Search Views Screens table](#).
5. Click **add more** to continue adding qualifiers. Otherwise, click **Save** to save the qualifier as part of the search view.

You have added search qualifiers to the filter display of the search view. The next time a user opens the search view in a search screen, these search qualifiers appear on the **Filter** tab and/or the search criteria section of the object's search screen.

### 3.1.7.1.5.6 Creating Section Titles

You can use section titles to label groups of the search qualifiers in your search views. For example, if you have a lot of qualifiers for date fields in the search view, you can organize your qualifiers into two sections entitled "General Record Information" and "Matter Record Dates," as shown in the following image.

The screenshot displays the 'Advanced Dispute Search' interface. At the top, there are navigation tabs for 'Legal', 'Finance', 'Contacts', 'Calendar', 'Documents', 'Admin', and 'All Services'. Below these are sub-tabs for 'Advice And Counsel', 'Cost Centers', 'Disputes', 'Time Entry Settings', and 'Transactions'. The main title is 'Advanced Dispute Search'. There are buttons for 'New', 'Printable View', and 'Help'. A left sidebar shows 'Browse' and 'Manage' options, with 'Recently Viewed' and 'All Disputes' listed. The main search area has a 'Search Within' dropdown set to 'All' and a 'Filter' dropdown set to 'Search for Subpoenas'. Below this are several search criteria fields: 'Subpoena Matter Name' (Contains), 'Issuing Court' (Equal To), 'Docket Number' (Contains), 'Type of Subpoena' (Is), 'Due Date' (to), 'Date Served' (to), 'Related Dispute Matter' (Equal To), 'Party Serving Subpoena' (Contains), and 'Responded Date' (to). A 'Search' button and a 'Cancel' button are at the bottom.

**Filter Tab Example with Visible Search Qualifiers**

**Note:** Section titles are ignored for search views that are used as custom search templates.

By default, section titles appear in size 10 black font, but users can set their formatting preferences in their **Appearance** settings.

#### To create a section title in the filter display

1. Open the appropriate object definition, and then click the Search Views tab.
2. Do one of the following actions:
  - To change an existing search view, click its hyperlink to open it.
  - To create a search view, click **new** and do the following actions:

- i. Complete the **General** tab. For details, see [Defining General Search View Information](#).
- ii. On the **Results Display** tab, add at least one column to the results display. This allows you to save the search view. For details, see [Search View Results Display](#).
- iii. Click **Save** to save the search view.

After the search view is saved, you can create a section title.

3. Click the **Filter Display** tab.
4. In the data entry row, in the **Label** field, enter the text of the section title.
5. From the **System** drop-down list, select **(Section Title)**.
6. In the **Row Order** field, enter the display order of the section title.

*Tip: Place the section title in the correct order in relation to the fields for which it is a heading. For example, if the **Row Order** is set to 5, then the qualifiers you want displayed below it must have display orders 6, 7, 8, and so on. For an example, see [the Filter Display Tab on Search View Screens tab image](#).*

7. Under Column, check the column (or screen location) where the search qualifier will reside in the **Filter** tab or the search criteria section of the object's search screen:
  - Choose Column 1 to display the qualifier in the left column of the screen.
  - Choose Column 2 to display the qualifier in the right column of the screen.
8. Check the **Visible** check-box so that the section title appears in the search screen.
9. Click **add more** to continue adding search qualifiers or click **Save** to save the section title as part of the search view.

The next time a user opens the search view in a search screen, this section title appears on the **Filter** tab or the search criteria section.

#### 3.1.7.1.5.7 Tips for Working with Qualifiers

This section provides more information about how selected search view qualifiers behave in TeamConnect.

### Parent Project

When working on a search view for a child object, the qualifier **Parent Project** is available, except when the **Used for** general option in the search view includes the **Related Object** choice.

### Assigned On, Unassigned On, and Main Assignee Assigned On

These three qualifiers appear in the **System** drop-down list in the **Filter Display** tab of a search view definition. You can use these qualifiers when building search views against objects that have lists of assignees.

These qualifiers accept values that are date ranges. The behavior of date ranges for **Assigned On** and **Unassigned On** varies, depending on whether the search request also specifies an assignee user ID:

- If you select a specific assignee, then the date ranges will only look at assigned and unassigned dates for that specific assignee.
- If you do not select a specific assignee, then any assignee whose assigned and/or unassigned dates match your date ranges will cause the record to qualify the search results. (If nine out of ten assignees do not qualify based on the date ranges, but the tenth one does qualify, the record will qualify for the search results.)
- If a single assignee appears twice in a record's assignee list, in two different roles, both roles' dates will be examined to see whether either of them qualify to be included in the search results.

The behavior of the date range for **Main Assignee Assigned On** varies, depending on whether the search request also specifies a main assignee user ID:

- If you select a specific main assignee, only records matching that main assignee will be considered, and the date range will only look at assigned dates for those records.
- If you do not select a specific main assignee, then any main assignee whose assigned date matches your date range will cause the record to qualify for the search results.

It is important to remember that TeamConnect only checks dates for the current main assignee. If there was a different main assignee in the past, that information is not known to the search filter. Also, the assignment date for the main assignee is the date that the person was first added to a record. If you added someone as a regular assignee at first, then promoted them to main assignee one month later, the **Main Assignee Assigned On Date** will be considered the earlier date, not the one a month later.

As an example, if you specify a main assignee of "Margaret Thatcher", then use an **Assigned On** date range of August 1st through August 31st, and leave the regular assignee drop-down list set to **ANY**, you will get all records for which Margaret Thatcher is the main assignee, and for which anyone was assigned during August. This does not necessarily mean that Margaret Thatcher was assigned during August-she could be assigned to a record in June and, if someone else was assigned during August, the record would qualify for inclusion. If you want to make your filter more specific, and find records where Margaret Thatcher is main assignee and she was assigned in August, you must use the **Main Assignee Assigned On** date range, not the **Assigned On** date range.

## Searching the Documents of an Individual Project (Custom Object) Record

When using the **Search** button from within a specific record's **Document** screen, the qualifier **Search in Folder** presents several choices in a drop-down list. The only choice that is valid for a search inside a specific record is **Current Folder**.

If you are already within a subfolder of a record's documents when you launch such a search, the search results will be limited to that subfolder.

## Searching History Records of a Specific Project or a Specific Custom Object Type

Qualifier **Parent Project**, a choice in the **System** drop-down list in the **Filter Display** tab, can restrict history searches to a specific custom object type and/or a specific project name or number.

- When you add the **Parent Project** qualifier to a new or existing search view, the **Condition** drop-down list will show a list of all custom object definitions to which you have "read" authority.
- If you want to restrict this history search to a specific custom object type, choose that type from the drop-down list. Otherwise, choose **ANY**.

When end users choose a search view that contains this qualifier, the field **Parent Project** accepts the number or name of the desired project, and:

- Search results will be filtered to include only projects that match the name or number that the end user enters in that field.
- If the user leaves the field empty, search results will not be filtered by project name or number.
- If you, the developer, chose a specific custom object type when you specified the **Parent Project** qualifier in the search view, search results will be filtered to include only that custom object type, no matter what value the end user puts in **Parent Project**.

Search results will only show records to which the user has at least "read" authority.

For example, assume that you create a search view that includes the **Parent Project** qualifier, and you specify custom object type **Matter** while defining that qualifier. When an end user runs that search view and enters "Turnstone Incorporation" as a value for that qualifier, then only Matter records that match that name will be searched, even if there are other types of custom object records that have the name "Turnstone Incorporation". If, on the other hand, the end user runs the search view but does not enter any value for **Parent Project**, the search will still be filtered to include only Matter records, regardless of what other qualifiers might be used in the search.

## Searching by Workflow Status and Workflow Action

Possible values for the **Workflow Status** qualifier include the value **Any** (Workflow status will not be considered when filtering search results) or one of the several possible end states of a workflow, such as Rejected, Error, Pending, Approved, or Canceled.

Field **Workflow Status** is available for the results display for a search view to show the status of specific records. **Workflow Status** lists specific values, such as Pending Approval, Approved, Rejected, Error, and Canceled. This field may be blank if a record has not been submitted to workflow.

## Workflow Action

The **Workflow Action** qualifier presents a list of values that vary by object type. Any workflow action that can be defined in an Approval Rule for an object will also appear in the **Workflow Action** qualifier drop-down for that object. For example, the Invoice object allows actions of Post, Void, and Delete. In addition to the actions that are specific to an object type, the qualifier drop-down includes the value **Any**, which means that the qualifier is ignored in the current search.

For example, using this qualifier means that you could limit an Invoice search to searching only for records that are pending approval for posting, not pending approval for void or deletion.

Field **Workflow Action** is available for the results display for a search view to show more detailed status of specific records.

For example, without the **Workflow Action** field the results of an invoice search might show the workflow status of **Pending**. With the inclusion of the **Workflow Action** field the results could say **Pending** and **Void**.

This field may be blank if a record has not been submitted to workflow.

*Note: In cases where there may be multiple workflows open against a single record, only the workflow that was initiated most recently will be evaluated against these qualifiers.*

### 3.1.7.2 Troubleshooting Search Views

This table explains solutions for possible problems with search views.

#### Search View Troubleshooting Tips

Possible problems	Possible solutions
The search view states that the search results display is not configurable, and I cannot define the necessary columns to display the matching search results for the end users.	You are likely to encounter this issue in either contact or appointment search views. If on the <b>General</b> tab of the search view in the contact or appointment search view, <b>Card</b> or <b>Calendar</b> display type is selected, you are unable to change the display of search results.  To customize the display of appointment of contact search results, select <b>Tabular</b> in the <b>Results Display Type</b> drop-down list on the <b>General</b> tab of the search view.
The <b>Results</b> tab or the displayed records section of the object's search screen in the end-user interface fails to have hyperlinks to the records.	Make sure that on the <b>Results Display</b> tab of the corresponding search view, the <b>Object Link</b> system display key is added.
The column width specified on the <b>Results Display</b> tab of the search view fails to take effect on the corresponding <b>Results</b> tab or the displayed records section of the object's search screen in the end-user interface.	If the total percentage of all columns does not equal 100%, the system may automatically resize the column to fit the contents, ignoring your settings.

### 3.1.8 Creating Home Pages and Portal Panes

A *home page* is the first page that users see after logging in to TeamConnect. It is the starting point for all users and can be customized to fit either individual or group user needs. The core functionality of a home page within TeamConnect is to display portal panes, which contain the actual content for the page.

You can create as many different home pages as necessary for your users. Each user can have access to multiple home pages based on which ones you give them access to as members of a group. Each home page that a user has access to is listed on their **Home Pages** menu.

*Portal panes* are the building blocks of a home page and contain specific content (information) that you define. This content is, in one way or another, beneficial to a user's job performance. Examples

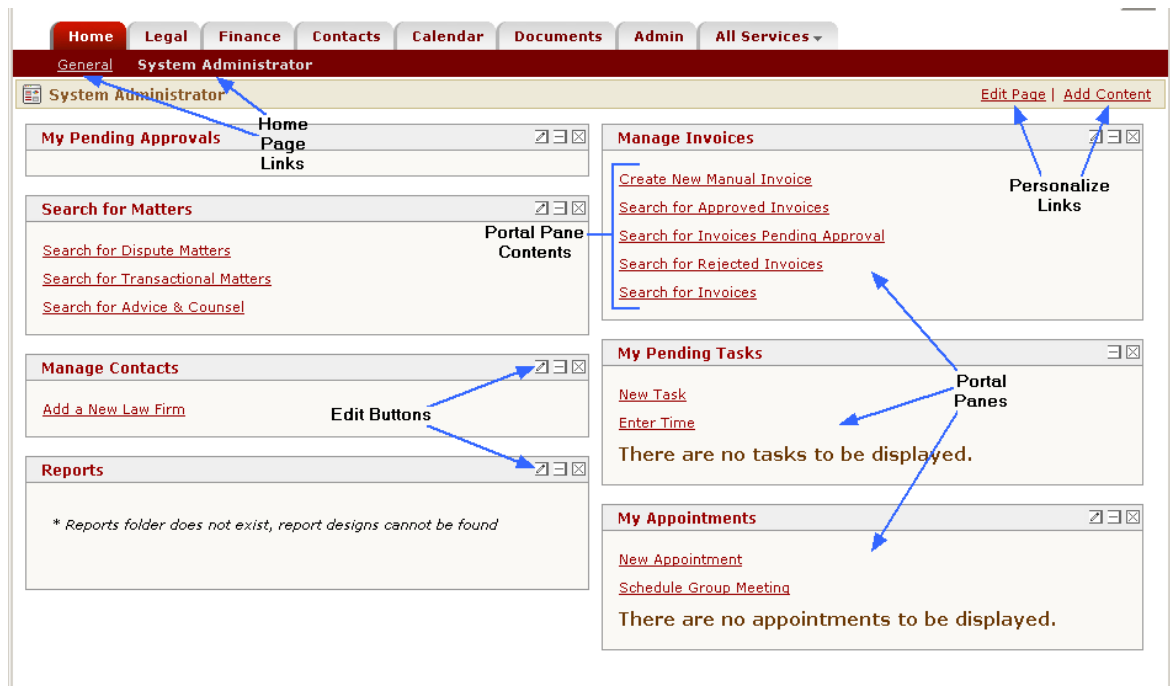


of portal pane content are links to upcoming appointments or helpful web sites, shortcuts to create new contacts, claims, tasks.

### 3.1.8.1 Home Pages and Portal Panes Basics

To specify home pages and portal panes, you may do the following:

- Create and define multiple portal panes.
- Include a particular portal pane on multiple home pages, as necessary.
- Create portal panes that include one or more types of content (see [Portal Panes with Multiple Contents](#)).
- Select the necessary types of content in a portal pane based on its purpose and function as well as the needs of your users.
- Create a common, organization-specific home page that displays corporate logos and messages for users.
- Restrict the availability of a home page to a particular group of users.
- Create a user-customizable home page that displays items such as (but, not limited to):
  - Upcoming appointments and tasks
  - Links to job-related Web sites
  - Stock quotes



Administrator End-User Home Page Components

The following table describes the components of a home page.

#### User Home Page Components

Field	Description
Home Pages Links	<p>Displays the titles of the home pages available to the user.</p> <p>The user selects the desired home page to view by clicking the desired title. The home page currently being viewed appears in a title bar just below the <b>Home Pages</b> links.</p>
Personalize Links	<p>The user can personalize and modify a home page by clicking <b>Add Content</b> and <b>Edit Settings</b>. Once users modify a home page, they must click <b>Refresh</b> to see their changes.</p> <p>You can specify whether to show this menu. For details, see <b>Allow end users to edit layout only</b> in <a href="#">the Home Page Settings Tab table</a>.</p>
Portal Panes	<p>Users work with the information the different portal panes.</p> <p>You can add as many portal panes as necessary to a home page. For details, see <a href="#">About Modifying Home Pages</a>.</p>
Portal Pane Content	<p>Users can perform various tasks by clicking links or viewing content results and information within the different portal panes.</p> <p>You can add and create new content for a portal pane. For details, see <a href="#">Portal Pane Contents</a>.</p>
Portal Pane Title Bar	<p>Displays the name of the portal pane. You can display buttons so that users can edit, move, resize, or delete the portal pane.</p> <p>For details about options that you can set for users, see <a href="#">Portal Pane Settings</a>.</p>

## Design Checklist

Use the following checklist when creating home pages and portal panes:

- Define the various types of users to be seeing home pages.
- Determine how many home pages are necessary.
- Determine how many portal panes are necessary, with the capability of reusing the same portal pane in multiple home pages.
- Define the content of each portal pane.
- Define home pages and add the appropriate portal panes.
- To make a home page visible to users, select the **Visible** check-box on the **Settings** tab of the home page (shown in [the Home Page Settings Tab image](#)).
- Determine whether users are allowed to edit each home page and portal pane or control other options. See [Modifying Home Pages and Portal Panes](#) for more details.

**Important:** Do not turn on editing options for users until after you are finished designing your new home pages and portal panes.

### 3.1.8.2 Portal Panes

As a TeamConnect solution developer, you may define the content and settings of portal panes for your users. By defining the portal panes, you create links to information and control what users can do from their home pages.

Depending on the needs of your users and their job responsibilities, you may select which portal panes are viewable from different home pages. You may either create new portal panes or select the portal panes included with TeamConnect by default.

The explanation of Portal Panes includes the following topics:

- [Default Portal Panes](#)
- [Opening Portal Panes](#)
- [Portal Pane Settings](#)
- [Portal Pane Contents](#)
- [Creating Portal Panes](#)

#### 3.1.8.2.1 Default Portal Panes

The following portal panes are included with TeamConnect and available for use in home pages:

- **My Appointments**—Displays appointments scheduled for the user.
- **My Documents**—Displays links to files contained in the logged-in user's document folder.
- **My Reports**—Displays a specified report on the user's homepage. From the **My Reports** section, click **Edit Parameters** and select a **Report** from the drop-down list. You can also specify the **Chart Width** and **Chart Height**.
- **My Pending Approvals**—Displays the user's pending approvals.
- **My Approvals**—Displays the user's pending approvals.
- **My Requests**—Displays the user's requests in progress.
- **My Workflows**—Displays the processes that the user is responsible for monitoring. This portal pane only appears to Process Managers.
- **RSS Feeds**—Displays the selected RSS Feed content (for example news feeds).
- **My Stocks**—Displays stock information and links to various companies on the Yahoo Finance page.
- **My Tasks**—Displays tasks assigned to the user.

## 3.1.8.2.2 Opening Portal Panes

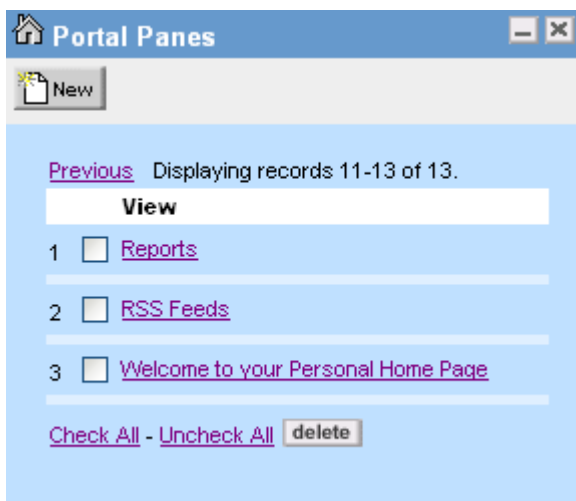
**To open a portal pane**

1. In the **Go to** drop-down list, select **Portal Panes**.

The **Portal Panes** screen displays a list of portal panes, which include the default portal panes and all previously created portal panes.



**Portal Panes Screen**



**Portal Panes Screen (continued)**

- From the portal pane list, click the hyperlink of the desired portal pane.

The **General** tab of the portal pane you selected appears.

**Portal Pane Settings**

\* Title:

Background Color:

Show Title bar

Editable by User

**Portal Pane Contents**

Number of entries you would like to add:

	Content Type	Content	Order	Alignment
1	(Select)	Label: <input type="text"/>	<input type="text"/>	(Select)
2	(Select)	Label: <input type="text"/>	<input type="text"/>	(Select)

[+ add more](#)

	Content Type	Content	Order	Alignment
1	<input type="checkbox"/> Action Link	Label: <input type="text" value="Time Entry Sheet"/> Links To: <input type="text" value="Task Batch Entry"/> Icon Image File: (No Item Selected) Wizard: (No Item Selected)	<input type="text" value="10"/>	<input type="text" value="LEFT"/>
2	<input type="checkbox"/> Search View	Label: <input type="text" value="My Upcoming Tasks"/> Object Type: <input type="text" value="Task"/> Search View: <a href="#">My Pending Tasks</a> View Type: <input type="text" value="Tabular"/> Max Search Results: <input type="text" value="10"/>	<input type="text" value="40"/>	<input type="text" value="LEFT"/>

[Check All](#) - [Uncheck All](#)

General Tab on Portal Pane Screens

The **General** tab of the **Portal Pane** screen has the following sections:

- Portal Pane Settings**—From this section, you may modify general information such as the title and background color of a portal pane. See [Portal Pane Settings](#).
- Portal Pane Contents**—From this section, you can modify the content that defines how a portal pane functions from a home page. See [Portal Pane Contents](#).


### 3.1.8.2.3 Portal Pane Settings

The **Portal Pane Settings** section of the **General** tab allows you to:

- Identify the portal pane by giving it a title and a unique identifying key.
- Set the background color of the portal pane.
- Give users specified controls to the selected portal pane.

The following table describes the items in the **Portal Pane Settings** section.

#### Portal Pane Settings

Field	Description	Comments
<b>Title</b>	<p>Enter a name (maximum 250 characters) that uniquely identifies the portal pane. This name appears in the following locations:</p> <ul style="list-style-type: none"> <li>• The solution developer's <b>Portal Panes</b> screen.</li> <li>• User's <b>Select Content to Add</b> pop-up window displayed by clicking the <b>Add Content</b> link on their home page.</li> <li>• The title bar of the portal pane.</li> </ul>	<p>As long the <b>Editable by user</b> check-box is selected, the title bar of the portal pane appears for the user--even if the <b>Show title bar</b> check-box is not selected.</p>
<b>Unique Key</b>	<p>Enter an alphanumeric combination to uniquely identify the portal pane content. After you save the settings, you can edit this field only if it is blank.</p> <p>The unique key cannot contain spaces, underscores, punctuation, or any special characters and cannot exceed 50 characters in length.</p>	
<b>Background Color</b>	<p>(Optional) Click the <b>Color Picker</b> icon next to the field and select the desired color from the pop-up window that appears.</p> <p>If no background color is specified for the portal pane, the background color will be the normal background color defined in System Settings.</p>	<p>Search View content items always use the background color defined in System Settings. Custom Content items always use the background colors defined in their Java class files.</p>
<b>Show title bar</b>	<p>Select this check-box to display the title bar of the portal pane. For example:</p>  <p>If the appropriate options are turned on, it also displays buttons so that users can edit, move, resize, or delete the portal pane.</p> <p>You can choose which options to turn on.</p>	<p>Changes made by one user from their own home page, do NOT affect the same home page of another user.</p>
<b>Editable by user</b>	<p>Select this check-box to allow users to edit the contents of the portal pane within their home page.</p>	

	The <b>edit</b> button is available from the portal pane title bar.	
--	---	--

### Portal Pane Setting Dependencies

How the settings of your portal pane take effect depends upon the settings of the home page on which that portal pane appears. The following table shows all the possible combinations of portal pane and home page settings, along with the resulting portal pane functionality:

**Home Page and Portal Page Setting Combinations**

Home page settings	Portal pane settings	Results				
		For the portal pane, user can				
		See title bar	Edit	Move	Resize	Remove
<input checked="" type="checkbox"/> Visible <input type="radio"/> Do not allow end users to edit layout or content <input type="radio"/> Allow end users to edit layout only <input type="radio"/> Allow end users to edit layout & content	<input type="checkbox"/> Show Title bar <input type="checkbox"/> Editable by User					
	<input checked="" type="checkbox"/> Show Title bar <input type="checkbox"/> Editable by User	X				
	<input type="checkbox"/> Show Title bar <input checked="" type="checkbox"/> Editable by User	X	X			
	<input checked="" type="checkbox"/> Show Title bar <input checked="" type="checkbox"/> Editable by User	X	X			
<input checked="" type="checkbox"/> Visible <input type="radio"/> Do not allow end users to edit layout or content <input checked="" type="radio"/> Allow end users to edit layout only <input type="radio"/> Allow end users to edit layout & content	<input type="checkbox"/> Show Title bar <input type="checkbox"/> Editable by User					
	<input checked="" type="checkbox"/> Show Title bar <input type="checkbox"/> Editable by User	X		X	X	
	<input type="checkbox"/> Show Title bar <input checked="" type="checkbox"/> Editable by User	X	X	X	X	
	<input checked="" type="checkbox"/> Show Title bar <input checked="" type="checkbox"/> Editable by User	X	X	X	X	

<input checked="" type="checkbox"/> Visible <input type="checkbox"/> Do not allow end users to edit layout or content <input type="checkbox"/> Allow end users to edit layout only <input checked="" type="checkbox"/> Allow end users to edit layout & content	<input type="checkbox"/> Show Title bar <input type="checkbox"/> Editable by User					
	<input checked="" type="checkbox"/> Show Title bar <input type="checkbox"/> Editable by User	X		X	X	X
	<input type="checkbox"/> Show Title bar <input checked="" type="checkbox"/> Editable by User	X	X	X	X	X

### 3.1.8.2.4 Portal Pane Contents

To specify what a portal pane displays, you select a content type. The content type determines the information that the user will access (such as tasks) or the action that they will perform (such as scheduling new appointments).

Portal pane content provides users with quick access to important company announcements, various tasks, search results, reports, Web sites, and so on.

The kinds of information that are added to portal panes are referred to as content types. The following table lists the various content types that you can include in a portal pane, along with a description and usage of each.

#### Portal Pane Content Types and Their Usage

Content type	Description	Use to
<b>Action Link</b>	<p>Allows users to create new records from their home page or to access a TeamConnect user tool from their home page. When a user clicks an action link from a portal pane, the result is the same as selecting the corresponding options from either the <b>Create a new</b> or <b>Tools</b> drop-down lists in the user interface.</p> <p>The portal pane titled "Home Owner Claims" in <a href="#">the Administrator End-User Home Page Components image</a> is an example of an action link.</p> <p>For more details about action links, see <a href="#">Adding Action Links</a>.</p>	<p>Create the following records with or without a wizard:</p> <ul style="list-style-type: none"> <li>Accounts</li> <li>Appointments</li> <li>Contacts (can specify a New Company link or a New Person link)</li> <li>Contact Groups (Address Books)</li> <li>Custom Objects</li> <li>Expenses</li> <li>Invoices</li> <li>Tasks</li> </ul> <p>Access the following tools:</p> <ul style="list-style-type: none"> <li>Document Generator</li> </ul>



		<ul style="list-style-type: none"> <li>• Expense Batch Entry</li> <li>• Task Batch Entry</li> <li>• Scheduler</li> </ul>
<p><b>Object Link</b></p>	<p>Allows users to open specific records or to search for specific records from their home page. When a user clicks an object link from a portal pane to search for a specific record, the result is the same as selecting the corresponding options from the <b>Go to</b> drop-down list in the user interface.</p> <p>The portal pane titled "Important Contacts" in <a href="#">the Administrator End-User Home Page Components image</a> is an example of an Object Link portal pane.</p> <p>Also, allow users to open specific search screens from their home page. For details, see <a href="#">Adding Search View Object Links</a>.</p> <p><i><b>Important:</b> Search View object links function differently from the Search View content type. Search View object links are essentially links to a Filter tab of a search screen, whereas content items of the type Search View display the results of a search within the portal pane.</i></p> <p>For more details, see <a href="#">Adding Object Links</a>.</p>	<p>Create links to access the following:</p> <ul style="list-style-type: none"> <li>• Specific contacts</li> <li>• Specific projects (custom object)</li> <li>• Specific accounts</li> <li>• Specific users</li> <li>• Specific groups</li> <li>• Search views</li> <li>• Specific global forums</li> </ul>
<p><b>Search View</b></p>	<p>Allows users to view specific search results that appear within their portal pane.</p> <p>Search views in portal panes are the same search views defined for each object within TeamConnect. You can include any existing search view in a portal pane—provided that the <b>Usable in Portal Panes</b> option is selected when you create the search view.</p>	<p>Display search results for a specific search view within the portal pane in one of the following ways:</p> <ul style="list-style-type: none"> <li>• <b>Tabular view</b>—Search results appear in a table format, like the <b>Results</b> tab of most search screens.</li> </ul>

	<p><b>Tip:</b> Create separate search views for portal panes.</p> <p>To learn more about Search View qualifiers, see <a href="#">Search View Filter Display</a>.</p> <p>The portal pane titled "My Upcoming Appointments" in <a href="#">the Administrator End-User Home Page Components image</a> is an example of a search view.</p> <p>For more details, see <a href="#">Adding Search Views</a>.</p>	<ul style="list-style-type: none"> <li>• <b>Drop-down list</b>—Search results appear in a drop-down list.</li> </ul>
<p><b>Report</b></p>	<p>Allows users to see a report's results, or to link to a report.</p> <p>The reports must already have been <a href="#">designed and tested</a>.</p> <p>A report is implemented in a portal pane through the use of the <b>tc:report</b> tag or the <b>tc:reportLink</b> tag. Details about these tags are found in <a href="#">Using Tags to Embed Reports in Blocks</a>.</p>	<p>Display report results in one of the following ways:</p> <ul style="list-style-type: none"> <li>• Report results appear directly in the portal pane.</li> <li>• A link in the portal pane allows the user to execute a report, or jump to a specific folder of reports.</li> </ul>
<p><b>Text</b></p>	<p>Serves both of the following purposes:</p> <ul style="list-style-type: none"> <li>• Links to a network location. For example: <pre>&lt;a href="networkLocation" target="_blank"&gt;networkLocationLabel&lt;/a&gt;</pre> <p><b>Important:</b> You must specify a target attribute. If you do not, the network location opens in the same window as TeamConnect.</p> </li> <li>• Displays text to users upon viewing a particular home page. Messages can be displayed to the entire organization, or to a specific group of users, depending on who has access to the particular home page on which the portal pane appears.</li> </ul>	<ul style="list-style-type: none"> <li>• Access network locations through hyperlinks.</li> <li>• Display messages in plain text or HTML format. HTML content can also include references to files hosted on outside servers.</li> </ul>

	<p>The portal pane titled "ClientGuard Messages" in <a href="#">the Administrator End-User Home Page Components image</a> on page 306 is an example of a portal pane displaying text.</p> <p>For more details, see <a href="#">Adding Text</a>.</p>	
<b>Web URL</b>	<p>Links to Web sites outside of TeamConnect. When a user clicks a Web URL link in a portal pane, the specified URL is passed directly to the <b>Address</b> field of a new browser window. Web URL links can be set based on what is necessary for the users to access and view.</p> <p>The portal pane titled "Important Web Links" in <a href="#">the Administrator End-User Home Page Components image</a> is an example of a Web URL portal pane.</p> <p>For more details, see <a href="#">Adding Web URLs</a>.</p>	Access Web sites through hyperlinks.
<b>Custom Content</b>	<p>Displays customized content and formatting through use of Java class files.</p> <p>The portal pane titled "My Stocks" in <a href="#">the Administrator End-User Home Page Components image</a> is an example of a Custom Content portal pane.</p> <p><b>Note:</b> <i>To create a Custom Content portal pane, you need extensive experience with the Java programming language.</i></p> <p>For more details, see <a href="#">Adding Custom Content</a>.</p>	Access specialized content that is created with Java class files, such as TeamConnect publications, weather, stocks, and so on.
<b>WebIntelligence URLs</b>	<p>(WebIntelligence only) Link to WebIntelligence reports.</p> <p><b>Important:</b> <i>You must have <a href="#">System Preferences</a> set appropriately.</i></p>	Access WebIntelligence software through hyperlinks.

## 3.1.8.2.4.1 Adding Action Links


**To add an action link to portal pane content**

1. Open the appropriate portal pane.
2. In the **Portal Pane Contents** section, select **Action Link** from the **Content Type** drop-down list.
3. Fill in the appropriate fields as described in [the Portal Pane Content of Type Action Link table](#).
4. Click **add more** to add more content. If you are finished adding content, click **Save**.

The action link is added to the portal pane. You will see your changes when you add the portal pane to a home page.

**Portal Pane Content of Type Action Link**

Field	Description
<b>Label</b>	<p>Enter a label (maximum 250 characters) that uniquely identifies the action link. The label you enter appears as a link within the user's portal pane.</p> <p><i>Tip: Start your label with a verb that identifies the action that the user can perform. For example, "Create a contact", "Schedule an appointment", or "Generate a document".</i></p>
<b>Unique Key</b>	<p>Enter an alphanumeric combination to uniquely identify the portal pane content. After you save the settings, you can edit this field only if it is blank.</p> <p>The unique key cannot contain spaces, underscores, punctuation, or any special characters and cannot exceed 50 characters in length.</p> <p>This is an optional field.</p>
<b>Links to</b>	<p>Select the action that should occur when the user clicks the link from the portal pane. The action will either be to create a new record or to open a tool.</p> <p><i>Caution: If you link to a user tool to which a user is not granted rights, the hyperlink still appears in the user's portal pane. However, the user will not be able to access the tool.</i></p>
<b>Custom Object Type</b>	<p>Select the specific custom object definition whose records you would like the user to create from the portal pane.</p>

	This field appears when <b>New Custom Object</b> is selected from the <b>Links to</b> drop-down list.
<b>Icon Image File</b>	(Optional) Select an appropriate icon to display next to the action link within the portal pane. Icon image files are in the following location: <b>//Top Level/System/Icons</b> . 
<b>Wizard</b>	(Optional) Select the wizard that the user will access when clicking the action link in the portal pane.  The wizards defined for the object selected in the <b>Links to</b> field are available for you to select.
<b>Order</b>	Enter an integer to indicate the display order of the action link within the portal pane. If a portal pane contains more than one content item, they will be displayed in the order specified in this field, lowest number first.  <b>Note:</b> <i>Items with the same display order are sorted and displayed alphabetically.</i>
<b>Alignment</b>	Select either <b>Left</b> , <b>Center</b> , or <b>Right</b> , to position the action link accordingly in the portal pane.

#### 3.1.8.2.4.2 Adding Object Links

##### To add a record object link to portal pane content

1. Select **Portal Panes** from the **Go to** list.  
The Portal Panes screen appears.
2. Open the appropriate portal pane.
3. In the **Portal Pane Contents** section, select **Object Link** from the **Content Type** drop-down list.

The **Object Type** and **Object** fields appear in the **Portal Pane Contents** section.



4. Fill in the appropriate fields as described in [the Portal Pane Content of Type Object Link table](#).
5. Click **add more** to add more content. If you are finished adding content, click **Save**.

The object link is added to the portal pane. You will see your changes when you add the portal pane to a home page.

The following table describes the object link items in the **Portal Pane Contents** section.

**Portal Pane Content of Type Object Link**

Field	Description
<b>Label</b>	Enter a label (250 characters max) that uniquely identifies the object link. The label appears as a link within the portal pane.
<b>Unique Key</b>	<p>Enter an alphanumeric combination to uniquely identify the portal pane content. After you save the settings, the unique key appears as read-only text and cannot be deleted or modified.</p> <p>The unique key cannot contain spaces, underscores, punctuation, or any special characters and cannot exceed 50 characters in length.</p> <p>This is an optional field.</p>
<b>Links to</b>	Select the desired object. The appropriate records are filtered in the <b>Object</b> field.
<b>Object</b>	<p>Select an existing record from either a search module or a drop-down list. Users will link to the record from the portal pane.</p> <p>The options in the <b>Object</b> field vary depending on the item you selected from the Links to drop-down list.</p> <ul style="list-style-type: none"> <li>• When selecting <b>Contact</b>, <b>Project</b>, or <b>Account</b>, the <b>Object</b> field appears as a search module.</li> <li>• When selecting <b>Group</b> or <b>Forum</b>, the <b>Object</b> field appears as a drop-down list.</li> </ul> <p><i>Note: Only Global Forums are accessible through portal panes.</i></p> <ul style="list-style-type: none"> <li>• When selecting <b>User</b>, the <b>Object</b> field can be displayed as either a search module or a drop-down list. This function for <b>User</b> is dependent on your <a href="#">System Properties</a>.</li> </ul> <p>Selecting <b>Search View</b> from the <b>Links to</b> drop-down list requires further explanation. For more information, see <a href="#">Adding Search View Object Links</a>.</p>
<b>Order</b>	<p>Enter an integer to indicate the display order of the object link within the portal pane. If a portal pane contains more than one content item, they will be displayed in the order specified in this field, lowest number first.</p> <p><i>Note: Items with the same display order are sorted and displayed alphabetically.</i></p>

**Alignment**

Select either **Left**, **Center**, or **Right**, to position the object link accordingly in the portal pane.

## Adding Search View Object Links

A **Search View** object link provides the user with a link to a search view in the portal pane, which is different from displaying the search view itself in the portal pane, as described in [Adding Search Views](#).

Whether the **Filter** or **Results** tab appears when the user clicks a **Search View** object link depends on the search view's settings:

- If the **Auto Search** check-box is selected in the search view, then the **Results** tab appears. This is useful when the search view has search conditions predefined in the **Filter Display** tab.
- If the **Auto Search** check-box is not selected in the search view, then the **Filter** tab appears. The user can then enter the appropriate criteria.

For more details about search view settings, see [Creating Search Views](#).

### To add a search view object link to portal pane content

1. Open the appropriate portal pane.
2. In the **Portal Pane Contents** section, select **Object Link** from the **Content Type** drop-down list.
3. Fill in the appropriate fields as described in [the Portal Pane Object Link of Type Search View](#).
4. Click **add more** to add more content. If you are finished adding content, click **Save**.

The search view object link is added to the portal pane. You will see your changes when you add the portal pane to a home page.

The screenshot shows the 'Portal Pane Contents' configuration window. At the top, there is a dropdown for 'Number of entries you would like to add:' set to '1'. Below this is a table with columns for 'Content Type', 'Content', 'Order', and 'Alignment'. The first row shows 'Object Link' as the content type. The 'Content' field is expanded to show several sub-fields: 'Label' (Search for Disputes), 'Links to:' (Search View), 'Search Object Type:' (Project), 'Custom Object:' (Dispute), and 'Search View:' (Default). The 'Order' field is set to '5' and the 'Alignment' field is set to 'LEFT'. There is an '+ add more' button at the bottom right of the configuration area.

Portal Pane Object Link of Type Search View

The following table describes the object link items of type Search View in the **Portal Pane Contents** section.

#### Portal Pane Object Link of Type Search View

Field	Description
-------	-------------

<b>Label</b>	Enter a label (maximum 250 characters) that uniquely identifies the object link to the search view. The label you enter appears as a link within the user's portal pane.
<b>Unique Key</b>	<p>Enter an alphanumeric combination to uniquely identify the portal pane content. After you save the settings, you can edit this field only if it is blank.</p> <p>The unique key cannot contain spaces, underscores, punctuation, or any special characters and cannot exceed 50 characters in length.</p> <p>This is an optional field.</p>
<b>Links to</b>	Select <b>Search View</b> from the list.
<b>Search Object Type</b>	<p>Select the object in which the user will search for records from the portal pane.</p> <p>When selecting <b>Project</b>, the <b>Custom Object</b> drop-down list appears.</p>
<b>Custom Object</b>	<p>Select the specific custom object for which you would like the user to search.</p> <p>This field appears when <b>Project</b> is selected from the <b>Search Object Type</b> drop-down list.</p>
<b>Search View</b>	<p>Select the search view that the user will see in the portal pane.</p> <p>The options in this drop-down list display the search views of the object selected in the <b>Search Object Type</b> drop-down list, which can be used in the Portal Pane. For more details about search views, see <a href="#">Creating Search Views</a>.</p>
<b>Order</b>	<p>Enter an integer to indicate the display order of the Search View object link within the portal pane. If a portal pane contains more than one content item, they will be displayed in the order specified in this field, lowest number first.</p> <p><b>Note:</b> <i>Items with the same display order are sorted and displayed alphabetically.</i></p>
<b>Alignment</b>	Select either <b>Left</b> , <b>Center</b> , or <b>Right</b> , to display the Search View object link accordingly in the portal pane.



## 3.1.8.2.4.3 Adding Search Views

**To add a search view to portal pane content**

1. Open the appropriate portal pane view.
2. In the **Portal Pane Contents** section, select **Search View** from the **Content Type** drop-down list.
3. Fill in the appropriate fields as described in [the Portal Pane Content of Type Search View table](#).
4. Click **add more** to add more content. If you are finished adding content, click **Save**.

The search view is added to the portal pane. You will see your changes when you add the portal pane to a home page.

**Portal Pane Content of Type Search View**

Field	Description
<b>Label</b>	Enter a label (maximum 250 characters) for the search view results that appear within the user's portal pane.  <i>Tip: Use the search view name.</i>
<b>Unique Key</b>	Enter an alphanumeric combination to uniquely identify the portal pane content. After you save the settings, you can edit this field only if it is blank.  The unique key cannot contain spaces, underscores, punctuation, or any special characters and cannot exceed 50 characters in length.  This is an optional field.
<b>Object Type</b>	Select the type of records for which you would like the user to search from their portal pane.  If you select <b>Project</b> , the <b>Custom Object</b> drop-down list appears.
<b>Custom Object</b>	Select the specific custom object definition for whose records you would like the user to search.
<b>Search View</b>	Select the search view to display in the user's portal pane.  This drop-down list includes only search views for the selected object which are specifically set to be available in home pages. For more details, see <a href="#">Creating Search Views for Home Pages</a> .

<b>View Type</b>	<p>Specify how the results of the search view appear in the user's portal pane by selecting either:</p> <ul style="list-style-type: none"> <li>• <b>Tabular view</b>—Search results appear in a table format, like the <b>Results</b> tab of a search screen.</li> <li>• <b>Drop-down list</b>—Search results appear in a drop-down list.</li> </ul> <p><i>Note: In the present version of TeamConnect, regardless of which choice you specify, the tabular view will be used.</i></p>
<b>Max Search Results</b>	<p>Enter the maximum number of search results that you want returned to the portal pane. If the search finds more results than the maximum number that you allot, a <b>More</b> link appears in the user's portal pane on which they can click to view the additional results.</p> <p><i>Caution: It is recommended that the <b>Maximum Search Results</b> not exceed 1000. Specifying a larger number may affect performance.</i></p>
<b>Order</b>	<p>Enter an integer to indicate the display order of the search view within the portal pane. If a portal pane contains more than one content item, they will be displayed in the order specified in this field, lowest number first.</p> <p><i>Note: Items with the same display order are sorted and displayed alphabetically.</i></p>
<b>Alignment</b>	<p>Select either <b>Left</b>, <b>Center</b>, or <b>Right</b>, to position the search view accordingly within the portal pane.</p>

#### 3.1.8.2.4.4 Adding Text

##### To add text to portal pane content

1. Open the appropriate portal pane view.
2. In the **Portal Pane Contents** section, select **Text** from the **Content Type** drop-down list.
3. Fill in the appropriate fields as described in [the Portal Pane Content of Type Text image](#).
4. Click **add more** to add more content. If you are finished adding content, click **Save**.

The text is added to the portal pane. You will see your changes when you add the portal pane to a home page.



## Portal Pane Content of Type Text

Field	Description
<b>Label</b>	(Optional) Enter a label (maximum 250 characters) for the text that appears within the user's portal pane.
<b>Unique Key</b>	<p>Enter an alphanumeric combination to uniquely identify the portal pane content. After you save the settings, you can edit this field only if it is blank.</p> <p>The unique key cannot contain spaces, underscores, punctuation, or any special characters and cannot exceed 50 characters in length.</p> <p>This is an optional field.</p>
<b>Text</b>	<p>Enter plain text or HTML content that you want to display as a message within the user's portal pane (up to 2000 characters).</p> <p>Or, enter the following code to create a link to a network location:</p> <pre>&lt;a href="Network Location" target="_blank"&gt;networkLocationLabel&lt;/a&gt;</pre> <p><b>Note:</b> <i>HTML code in text content can be fairly rich in functionality. For example, the code snippet below establishes a frame, within the portal pane, that acts similarly to a browser, displaying dynamic content:</i></p> <pre>&lt;iframe&gt; src ="http://www.google.com" width="100%" height="300" frameborder="0" align="middle" scrolling="yes"&gt; &lt;/iframe&gt;</pre>
<b>Order</b>	<p>Enter an integer to indicate the display order of the text. If a portal pane contains more than one content item, they will be displayed in the order specified in this field, lowest number first.</p> <p><b>Note:</b> <i>Items with the same display order are sorted and displayed alphabetically.</i></p>
<b>Alignment</b>	Select either <b>Left</b> , <b>Center</b> , or <b>Right</b> , to position the text accordingly within the portal pane.

## 3.1.8.2.4.5 Adding Web URLs

**To add a web URL to portal pane content**

1. Open the appropriate portal pane view.
2. In the **Portal Pane Contents** section, select Web URL from the **Content Type** drop-down list.
3. Fill in the appropriate fields as described in [the Portal Pane Content of Type Web URL table](#).
4. Click **add more** to add more content. If you are finished adding content, click **Save**.
5. The web URL is added to the portal pane. You will see your changes when you add the portal pane to a home page.

An alternate way of embedding a URL in a portal pane is to place HTML code into text content, as described in [Adding Text](#).

**Portal Pane Content of Type Web URL**

Field	Description
<b>Label</b>	Enter a label (maximum 250 characters) for the Web URL. The label you enter appears as a link within the portal pane.
<b>Unique Key</b>	Enter an alphanumeric combination to uniquely identify the portal pane content. After you save the settings, you can edit this field only if it is blank.  The unique key cannot contain spaces, underscores, punctuation, or any special characters and cannot exceed 50 characters in length.  This is an optional field.
<b>URL</b>	Enter the address of the Web site to which your users need access.  If the URL that you enter does not begin with either http:// or https://, TeamConnect appends http:// to the URL.
<b>Order</b>	Enter an integer to indicate the display order of the Web URL within TeamConnect. If a portal pane contains more than one content item, they will be displayed in the order specified in this field, lowest number first.  <i><b>Note:</b> Items with the same display order are sorted and displayed alphabetically.</i>
<b>Alignment</b>	Select either <b>Left</b> , <b>Center</b> , or <b>Right</b> , to position the text accordingly within the portal pane.

## 3.1.8.2.4.6 Adding Custom Content

**To add custom content to portal pane content**

1. Open the appropriate portal pane view.
2. In the **Portal Pane Contents** section, select **Custom Content** from the **Content Type** drop-down list.
3. Fill in the appropriate fields as described in [the Custom Portal Pane Content table](#), and then click **add more**.

The newly added custom content appears in a list at the bottom of the **Portal Pane Contents** section.

4. Click **Save**.

The custom content is added to the portal pane. You see your changes when you add the portal pane to a home page.

**Custom Portal Pane Content**

Field	Description
<b>Label</b>	Enter a label (maximum 250 characters) for the custom content that appears in a user's portal pane.
<b>Unique Key</b>	<p>Enter an alphanumeric combination to uniquely identify the portal pane content. After you save the settings, the unique key appears as read-only text and cannot be deleted or modified.</p> <p>The unique key cannot contain spaces, underscores, punctuation, or any special characters and cannot exceed 50 characters in length.</p> <p>This is an optional field.</p>
<b>Class Name</b>	<p>Enter the name of the Java class file that you would like to generate the HTML within the portal pane.</p> <p>The Java class file you specify must be uploaded to the <b>Top Level/System/Portal</b> folder in the <b>Documents</b> area.</p> <p>In the end-user interface, portal panes with custom contents may have the <b>Edit Parameters</b> hyperlink, where the users may change the desired parameters through the interface.</p>
<b>Order</b>	Enter an integer to indicate the display order of the custom content. If a portal pane contains more than one content item, they will be displayed in the order specified in this field, lowest number first.

	<b>Note:</b> Items with the same display order are sorted and displayed alphabetically.
<b>Alignment</b>	Select either <b>Left</b> , <b>Center</b> , or <b>Right</b> , to display the custom content accordingly within the portal pane.

#### 3.1.8.2.4.7 Adding WebIntelligence URLs

##### To add a WebIntelligence URL to portal pane content

1. Open the appropriate portal pane view.
2. In the **Portal Pane Contents** section, select WebIntelligence URL from the **Content Type** drop-down list.
3. Fill in the appropriate fields as described in [the Portal Pane of Type WebIntelligence URL table](#).
4. Click **add more** to add more content. If you are finished adding content, click **Save**.

The WebIntelligence URL is added to the portal pane. You will see your changes when you add the portal pane to a home page.



##### Portal Pane Content of Type WebIntelligence URL

Field	Description
<b>Label</b>	Enter a label (maximum 250 characters) for the WebIntelligence URL. The label you enter appears as a link within the portal pane.
<b>Unique Key</b>	Enter an alphanumeric combination to uniquely identify the portal pane content. After you save the settings, you can edit this field only if it is blank.  The unique key cannot contain spaces, underscores, punctuation, or any special characters and cannot exceed 50 characters in length.  This is an optional field.
<b>URL</b>	Enter the address of the WebIntelligence page to which your users need access.  <b>Note:</b> The method of obtaining the link for a WebIntelligence report depends on the version of WebIntelligence you are using. Please see the <i>Business Objects documentation</i> for more information.

<b>Order</b>	Enter an integer to indicate the display order of the WebIntelligence URL within TeamConnect. If a portal pane contains more than one content item, they will be displayed in the order specified in this field, lowest number first.  <i>Note: Items with the same display order are sorted and displayed alphabetically.</i>
<b>Alignment</b>	Select either <b>Left</b> , <b>Center</b> , or <b>Right</b> , to position the Text accordingly within the portal pane.

#### 3.1.8.2.4.8 Portal Panes with Multiple Contents

Depending on the tasks your users perform, you can create portal panes that are multi-functional by adding more than one type of content to the portal pane. By adding multiple content items to a single portal pane, you create a portal pane that functions as a specialized menu of your user's most commonly accessed tasks.

For instance, if you have a group of users that creates contacts, accesses projects, and views a particular Web site on a daily basis, you may want to create a portal pane that accomplishes all of these tasks.

The following image is an example of the General tab of a multi-functional portal pane titled "ClientGuard Menu."

Number of entries you would like to add:

Content Type	Content	Order	Alignment
1 (Select)	Label: <input type="text"/>	<input type="text"/>	(Select)
<a href="#">+ add more</a>			
Content Type	Content	Order	Alignment
1 <input checked="" type="checkbox"/> Action Link	Label: Create a Contact Links To: New Contact Icon Image File: (Select) Wizard: (Select)	1	LEFT
2 <input type="checkbox"/> Web URL	Label: Bar Association Web Sit URL: http://www.abanet.org	2	LEFT
3 <input type="checkbox"/> Object Link	Label: Brandal vs. Daniels Case Object Type: Project Object: <a href="#">WilliamBrandal'sClaim</a>	3	LEFT

[Check All - Uncheck All](#)

**Multi-functional Portal Pane Settings in the Designer Interface**

The following image shows how the "ClientGuard Menu" portal pane displays on a user's home page. Notice how the labels of the content items in [the Multi-functional Portal Pane Settings in the Designer Interface image](#) appear as links in the portal pane in the following image.



#### Multi-functional Portal Pane in the End User Interface

Keep in mind the usefulness and functionality of a portal pane. Creating a portal pane with a large number of content items could be confusing to your users. In most situations, it is best to create a few portal panes with a few content items rather than creating one portal pane that does everything.

#### 3.1.8.2.5 Creating Portal Panes

There are two locations from which you can create portal panes:

- **Portal Panes** screen—When you create a portal pane from the Portal Panes screen, it is available to all users and you can add it to any home page.
- **Home Page** screen—When you create a portal pane from the Home Pages screen, it is only available on the specific home page from which you created it. It never appears in the list displayed on the Portal Panes screen. For details, see [Creating Portal Panes from a Master Home Page](#).

#### To create a portal pane

1. In the **Create a new** drop-down list, select **Portal Pane**.
2. In the **Portal Pane Settings** section, fill in the appropriate portal pane settings as described in [the Portal Pane Settings table](#).
3. Using [the Portal Pane Content Types and Their Usage table](#), identify the content you would like to add.

*Tip: Do not create a portal pane with too much content. Instead, split the content into multiple portal panes.*

4. In the **Portal Pane Contents** section, in the **Number of entries you would like to add** drop-down list, select the number of entries based on how many additional content items you want to add in this portal pane.
5. Select the **Content Type** from the drop-down list.

Once you select a content type, an associated group of fields appears.

6. Complete all fields associated with the selected content type as follows:
  - For action link content types, see [the Portal Pane Content of Type Action Link table](#).
  - For object link content types, see [the Portal Pane Content of Type Object Link table](#).
  - For search view content types, see [the Portal Pane Content of Type Search View table](#).
  - For text content types, see [the Portal Pane Content of Type Text table](#).



- For Web URL content types, [the Portal Pane Content of Type Web URL](#).
  - For custom content types, [the Custom Portal Pane Content table](#).
  - For WebIntelligence content types, [the Portal Pane Content of Type WebIntelligence URL table](#).
  - Type the **Order**.
  - Select the **Alignment** from the drop-down list.
7. When you are done, click **add more**.
  8. If you are finished adding content click **Save**.

The newly created portal pane is saved and added to the list of existing portal panes in the **Portal Panes** screen.



Portal Panes Screen

To add the new portal pane to a home page, see [Creating Home Pages](#).

### 3.1.8.3 Home Pages

Designing home pages consists of the following:

- [Home Page Settings](#)
- [Home Page Content](#)

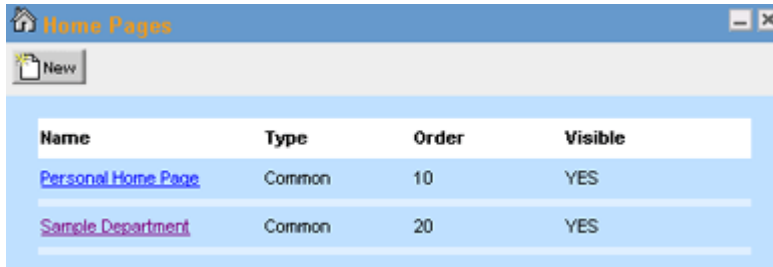
- [Creating Home Pages](#)

For a detailed checklist of tasks to complete, see the [design checklist](#).

### To access existing home page properties

1. In the **Go to** drop-down list, select **Home Pages**.

The **Home Pages** screen appears.



Home Page List Screen

2. Click the hyperlink of the home page you want to edit or to view.

The corresponding home page screen with the following tabs appears:

**Settings** | Content |

- a. **Settings**—Allows you to define general home page settings. For details, see [Home Page Settings](#).
- b. **Content**—Allows you to preview and edit the portal panes of the selected home page or add new portal panes to the selected home page. For details, see [Modifying Portal Panes](#).

#### 3.1.8.3.1 Home Page Settings

You can define settings such as the name and title of a home page on the **Settings** tab of a particular home page. On that tab, you can also define the type of users who are able to view and edit the home page.



#### Home Page Settings Tab

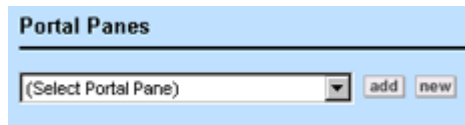
Field	Description
<b>Name</b>	Enter the name (maximum 50 characters) that uniquely identifies the home page. This name appears as a link in the <b>Home Pages</b> screen (see <a href="#">the Home Page List Screen image</a> ).
<b>Title</b>	Enter the title (maximum 250 characters) that identifies the home page for the user. This title appears in the <b>Home Pages Menu</b> (see <a href="#">the Administrator End-User Home Page Components image</a> ).

<b>Home Page Type</b>	<p>Select either <b>Common</b> or <b>Group</b> from the drop-down list to specify which group of users has access to the home page:</p> <ul style="list-style-type: none"> <li>• <b>Common</b> home pages are available to all users.</li> <li>• <b>Group</b> home pages are available to a specific group of users. If <b>Group</b> is selected, another drop-down list appears in which you must select a specific group of users.</li> </ul>
<b>Visible</b>	<p>Select this check-box to make the home page available to the users for which it is defined. If you do not select it, the home page remains unavailable.</p>
<b>Do not allow end users to edit layout or content</b>	<p>Select this check-box to prevent users from changing both the layout and the content.</p> <p><i><b>Important:</b> While you are designing and creating a home page, this is the only option that should be selected.</i></p>
<b>Allow end users to edit layout only</b>	<p>Select this check-box to enable users to change the layout of the home page.</p> <p>Selecting this check-box activates the <b>Edit Settings</b> link on the <b>Personalize Menu</b> on the home page (see <a href="#">the Administrator End-User Home Page Components image</a>).</p> <p>This function allows users to modify the layout of their own home pages only. Changes made by one user do not affect the home pages of other users.</p>
<b>Allow end users to edit layout &amp; content</b>	<p>Select this check-box to enable users to add content and change the page settings of the home page.</p> <p>Selecting this check-box activates both the <b>Edit Settings</b> and <b>Add Content</b> links on the <b>Personalize</b> menu on the home page (see <a href="#">the Administrator End-User Home Page Components image</a>).</p> <p>This function allows users to modify the layout and content (portal panes) of their own home pages only. Changes made by one user do not affect the home pages of other users.</p> <p><i><b>Tip:</b> Do NOT select this check-box while you are designing home pages and portal panes for users.</i></p>

### 3.1.8.3.2 Home Page Content

Defining home page content consists of adding portal panes to a home page. You can add existing portal panes, or you can create and add new portal panes directly from the **Content** tab of the **Home**

**Pages** screen, as shown in the following image. For details, see [Adding Portal Panes to Home Pages](#).



**Home Page Content Tab with No Portal Panes Added**

The following table explains the initial contents of the **Home Page Content** tab:

**Home Page Content Tab with No Portal Panes Added**

Field or button	Description
<b>Select Portal Pane</b>	Select the appropriate portal pane to add to the home page from this drop-down list.
<b>add</b>	Click to add the selected portal pane to the displayed home page.
<b>new</b>	Displays the title bar of a new portal pane. You must save the home page before you can edit the new portal pane.

### 3.1.8.3.3 Creating Home Pages

The following table lists a few tips to keep in mind when creating different home pages for your users:

**Tips for Creating Home Pages**

Tips	Home page settings
<p>If you would like all users to have a home page in which they can add, delete and rearrange content as they please, create a <b>Common</b> home page with the options in the <b>Home page settings</b> column selected.</p> <p>Do not add portal panes. This way, users will have a blank home page to work with that they can modify to meet their individual needs.</p> <p><i>Tip: Make sure users are trained on how to add content that is useful to them.</i></p>	<ul style="list-style-type: none"> <li>• Visible</li> <li>• Allow end users to edit layout and content</li> </ul>
<p>If you would like users to have a home page in which they can only rearrange the content layout, create a Common or Group home page with the options in the <b>Home page settings</b> column selected.</p>	<ul style="list-style-type: none"> <li>• Visible</li> <li>• Allow end users to edit layout only</li> </ul>

If you would like to create a static home page to which only you can make changes but users cannot, create a home page with the options in the <b>Home page settings</b> column selected.	<ul style="list-style-type: none"> <li>• Visible</li> <li>• Do not allow end users to edit layout or content</li> </ul>
Create a Common home page that is useful to the entire organization. Add portal panes that are useful for all users.	As deemed necessary.
Create various Group home pages that are useful to specific groups of users. Add only portal panes that accomplish the needs of the specified group members.	As deemed necessary.

### To create a home page

1. In the **Create a new** drop-down list, select **Home Page**.  
The **Settings** tab of a new home page appears.
2. Fill in the fields to specify the home page settings, as described in [the Home Page Settings Tab table](#).
3. Click the **Content** tab and add the desired portal panes to the home page as described in [the Home Page Content Tab with Portal Panes Added table](#).  
Copies of the portal panes as they are currently defined are added to the home page. For details, see [About Master Home Pages and Portal Pane Templates](#).
4. Click **Save**.
5. If you added the RSS Feeds Portal Pane, you should set the following content definitions:
  - In the **Go To** drop-down list, select **Portal Panes**.
  - Navigate to the **RSS Feeds** link and select it.
  - From the **General** tab, click the **Edit Parameters** link.
  - From the resulting popup, type the target RSS Feed URL in the URL box.
  - From the **Maximum number of feeds** drop-down list, select the number of feed items to display in the Portal Pane.
  - To sort feed items with most recent items on top, check the **Most recent updates** box.
  - Click **OK**. These changes will apply to the RSS Portal Pane on Home Pages that you assign to multiple users.
  - From the RSS Feeds Portal Pane page, click **Save and Close**.
6. If you are ready for your users to start using the new home page, click the following hyperlinks to activate your new home page:
  - Synchronize Settings

- Synchronize Content

**Important:** Do not synchronize settings or content until you are ready for your users to view and start using your new home page.

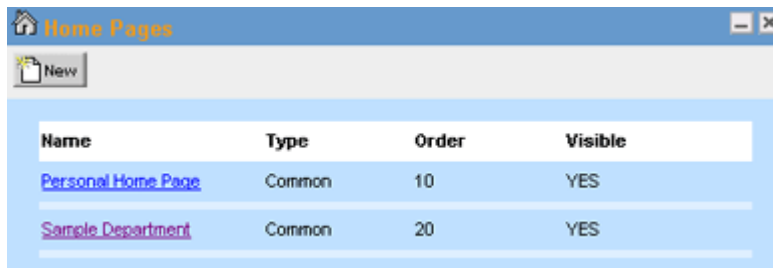
For details about synchronizing settings and content, see [Synchronizing Master Pages, Templates, and Copies](#).

Newly created home pages do not appear in the user's **Home Pages** menu until the user logs off and then logs into TeamConnect again.

### To delete a home page

1. In the **Go to** drop-down list, select **Home Pages**.

The **Home Pages** screen appears.



Name	Type	Order	Visible
<a href="#">Personal Home Page</a>	Common	10	YES
<a href="#">Sample Department</a>	Common	20	YES

Home Page List Screen

2. Select the check-box of the home page you would like to delete, and then click **delete**.

If the home page you selected is visible to a user, it remains visible until the user logs off and logs in to TeamConnect again.

#### 3.1.8.4 Customization Guidelines

Users can customize editable home pages and portal panes to which they have access. When you create new home pages and portal panes from the **Designer** interface, you can set various edit options so that users can customize particular home pages and portal panes to meet their own needs. For details about all home page and portal pane edit options, see [Modifying Home Pages and Portal Panes](#).

New home pages, which you create by selecting **Home Pages** from the **Go to** drop-down list, are master home pages. When users customize their home page, they are actually modifying a copy of the master home page.

The same concept holds true for portal panes. When you create a new portal pane by selecting **Portal Panes** from the **Go to** drop-down list, you create a template. Users who are granted the ability, can modify copies of that portal pane template.

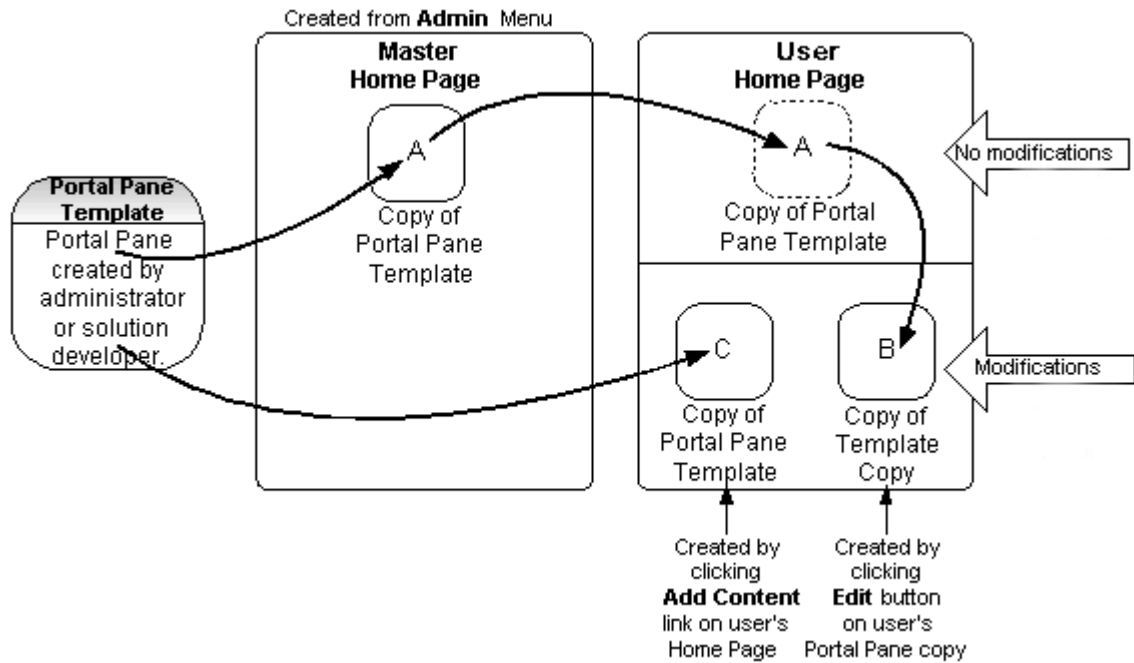
For details about creating and adding home pages and portal panes, see the following sections:

- [About Master Home Pages and Portal Pane Templates](#)
- [Modifying Home Pages and Portal Panes](#)

- [Creating Home Pages](#)
- [Creating Portal Panes](#)
- [Adding Portal Panes to Home Pages](#)

3.1.8.4.1 About Master Home Pages and Portal Pane Templates

It is important to understand the concepts of master home pages, portal pane templates, and copies. The concepts are best conveyed by the following diagram. It shows what happens to the portal pane template that is the original portal pane, created on the **Portal Panes** screen.



**Relations between Master Home Pages, Portal Pane Templates, and Copies**

The following table explains details about each of the portal pane copies shown in the diagram.

**Explanation of the Diagram**

Home page	Portal pane copy	Explanation
Master	<b>A</b>	When you add a portal pane template to master home page, you are actually adding a copy of that portal pane template.
User	<b>A</b>	When users first view their own home page, they see the same master home page that you created—including the same copy of that portal pane template.
User	<b>B</b>	As soon as a user clicks <b>edit</b> on a portal pane on his or her own home page, the template copy is copied and replaced with a revised version. In other words, the user has a copy of a copy

		posted on their home page, and portal pane A is no longer visible on their home page. The user only sees his or her customized version of the portal pane.
User	<b>C</b>	If a user has the ability to add portal panes, they can click the <b>Add Content</b> link on their home page to add a copy of the original portal pane template to their own home page.

In regard to home pages, users initially see an exact replica of the master home page. As soon as users view it, their home page becomes a copy. Therefore, any changes you make as the solution developer, your users cannot see if they have already viewed the home page.

However, there is an exception. If you have deleted a portal pane, then it is deleted from your users' home pages—unless a user has previously modified it. If that is the case, the user's copy of the portal pane that you deleted remains on their home page with their modifications.

### Points To Remember

You can synchronize both copies of home pages and portal panes with their master. For details, see [Adding Portal Panes to Home Pages](#).

What users do see is the sum of the master home page plus their own changes to either home pages or portal panes. For this reason, as stated previously, it is important NOT to set the following home page options until after you are finished designing and creating all of your home pages and portal panes:

- **Visible**
- **Allow end users to edit layout only**
- **Allow end users to edit layout & content**

In other words, the only home page setting that should be turned on when you are designing and creating home pages and portal panes is **Do not allow end users to edit layout or content**.

If you do choose one of the options listed in the table above before you are finished creating a home page or portal pane, you can synchronize your master home page and portal pane template changes with the home page and portal pane copies that have already been viewed by other users. For more details, see [Adding Portal Panes to Home Pages](#).

#### 3.1.8.4.2 Modifying Home Pages and Portal Panes

Users can modify their home page and portal panes if given the ability to do so. Solution developer changes and user changes both affect home pages and portal panes. If users modify their home pages and portal panes, the masters, templates, and users' copies will get out-of-sync with each other.

The good news is that as the solution developer, you have total control. If it is necessary at any time to synchronize the content or settings of a home page or portal pane—or both—you can click the **Synchronize Content** and **Synchronize Settings** hyperlinks to do so.



When you synchronize content or settings of home pages or portal panes, you overwrite all modifications that users made to their home pages. The users see the content and settings as they appear in the master home pages and portal panes.

#### 3.1.8.4.2.1 Synchronizing Master Pages, Templates, and Copies

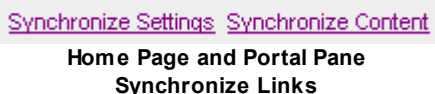
Part of the process of modifying home pages and portal panes is to synchronize your changes from the master home page and template portal panes with the copies that the users have of each.

Synchronizing also plays an important role in creating home pages (not portal panes). To activate a new home page, you must synchronize it. If you are in the process of designing a new home page, it is good practice to NOT synchronize settings or content until your design is final. That way, there is no concern about overwriting user's modifications, in case they decide to customize their home pages or portal panes.

The steps for synchronizing are included within the appropriate instructions in this documentation. It is a matter of clicking links (shown in the following figure), which are in the following locations on the **Designer** interface, at the top of the following screens:

- **Home Pages** screen
- **Portal Panes** screen

**Important:** You must save all your changes to the master home page **BEFORE** you synchronize settings or content.



The screenshot shows a user interface with two purple hyperlinks: "Synchronize Settings" and "Synchronize Content". Below these links, the text "Home Page and Portal Pane Synchronize Links" is displayed.

#### 3.1.8.4.2.2 About Modifying Home Pages

Let's say that you give your users the ability to modify home pages by selecting either the **Allow end users to edit layout only** or **Allow end users to edit layout and content** check-box when you create it. When you do NOT click the following synchronize hyperlinks:

- Any changes the user makes to the page settings take precedence over solution developer changes.
- If you, as a solution developer, add portal panes to a home page, the user will see that particular instance of that portal pane. Even if a user clicks the **Add Content** link from the **Personalize** menu to add a portal pane to a home page, and the solution developer adds the same portal pane to the same home page. As a result, the user sees both of those portal panes on their home page. The user can delete the duplicated portal pane, if desired.
- Modifications to home pages do not take effect immediately. Whether adding, deleting or modifying a home page, users must log off, and then log back on to see the changes.

### What Happens to Modified Home Pages?

As a TeamConnect solution developer, it is important to be aware of the effects that both user and solution developer changes have on home pages.

You may modify home pages according to the following table:

## Effects of Home Page Modifications

Type of user performing task	Task	Changes apply to	Effects of synchronizing
End	Directly from a user's own home page, he or she can click <b>Add Content</b> or <b>Edit Settings</b> links, and click arrows located on portal panes' title bars to rearrange the layout (if the corresponding options are set appropriately).	<ul style="list-style-type: none"> <li>User's home page only.</li> </ul>	User's modifications are overwritten with master home page.
Solution developer	From <b>Go to &gt; Home Pages</b> , open the appropriate home page. Click the <b>Portal Panes</b> tab. Click the arrows located on portal panes' title bars to rearrange the layout.	<ul style="list-style-type: none"> <li>Master home page.</li> <li>Any users who have not yet viewed the master home page.</li> </ul>	All copies of this master home page are overwritten.

## 3.1.8.4.2.3 About Modifying Portal Panes

The same theory that applies to modifying home pages holds true for modifying portal panes. If you give your users the ability to edit or move a portal pane by selecting the **Editable by user** or the **Show title bar** check-box when you create it, any changes the user makes take precedence over solution developer changes.

For example, if a user clicks **edit** on a portal pane and customizes that portal pane, the changes appear only on their home page. Any changes that you made to that same portal pane do not appear on the user's home page. Even if you delete that portal pane from the **Designer** interface, the user still sees their own edited version.

In short, the moment a user customizes a portal pane, those changes take precedence over any solution developer changes to that same portal pane. However, if a user does not edit a portal pane, then solution developer changes take precedence—including the deletion of a portal pane.

As a TeamConnect solution developer, it is important to be aware of the effects that both user and solution developer changes have on portal panes.

You may modify portal panes according to the following table:

## Effects of Portal Pane Modifications

Type of user performing task	Task	Changes apply to	Effects of synchronizing
------------------------------	------	------------------	--------------------------

End	Directly from a user's own home page, he or she can click edit in the appropriate portal pane.	User's portal pane copy only.	User's modifications are overwritten with portal pane template.
Solution developer	From <b>Go to &gt; Portal Panes</b> , open the appropriate portal pane.	Portal pane template only.	All copies of this portal pane template are overwritten.
Solution developer	From <b>Go to &gt; Home Pages</b> , open the appropriate home page, click the <b>Portal Pane</b> tab, and then click <b>edit</b> on the appropriate portal pane.	Solution developer's copy of portal pane on master home page.	All copies of this portal pane template are overwritten.

The [Relations between Master Home Pages, Portal Pane Templates, and Copies diagram](#) shows the different ways in which users can modify editable portal panes, and the results of those modifications (without synchronizing).

To take the task of a user editing a portal pane further—if a user clicks **edit** on a portal pane of their own home page, modifies it, and saves the changes, your actions, as a solution developer, can affect the results. The following table shows the various actions that you might take and the results of those actions. Keep in mind that the results listed are what happens before you click **Synchronize Contents** or **Synchronize Settings** hyperlinks. After you synchronize, portal pane templates overwrite any of its copies.

#### What Happens when User and Solution Developer Edit Portal Panes

Solution developer and user actions	Results
None.	<ul style="list-style-type: none"> <li>The modified portal pane displays only on the user's home page as a separate copy of the portal pane template.</li> </ul>
<p>From <b>Home Pages &gt; Portal Panes</b>, you edit a copy of a portal pane template.</p> <p>User modifies their own copy of that template from their own home page.</p>	<ul style="list-style-type: none"> <li>Your changes are reflected on your copy of the portal pane template. (You do not see user's changes).</li> <li>User's changes are reflected on their own copy of the portal pane template. (User does not see your changes).</li> </ul>
<p>From <b>Portal Panes</b>, you edit a portal pane template.</p> <p>User modifies their own copy of that template from their own home page.</p>	<ul style="list-style-type: none"> <li>Your changes are reflected on the portal pane template only.</li> <li>Your changes only take effect on users' portal pane copies that were created after the template was updated.</li> </ul>

	<ul style="list-style-type: none"> <li>User's changes are reflected on their own copy of the portal pane template. (User does not see your template changes).</li> </ul>
<p>From <b>Home Pages &gt; Portal Panes</b>, a copy of a portal pane template.</p> <p>User modifies their own copy of that template from their own home page.</p>	<ul style="list-style-type: none"> <li>Portal pane is deleted if it is not editable or has not been modified by the user, yet.</li> <li>Does not effect portal panes that have already been viewed by a user.</li> </ul>

The following table lists modifications to portal panes (of which the content and settings were synchronized) and when those changes take effect:

#### When Changes Take Effect in Portal Panes

Modification	Effective
Adding portal panes to home pages.	When user clicks <b>Refresh</b> .
Deleting portal panes that exist on a home page without prior edits by user.	
Deleting portal panes from the <b>Home Page</b> screen with prior edits by user.	When user logs off, and then logs back in.
Modifying portal pane from <b>Home Page</b> screen.	Immediately.
Modifying portal pane from the <b>Portal Pane</b> screen.	When user logs off, and then logs back in.

For each user who has access to an editable portal pane, a separate instance (or copy) of the portal pane template exists. Users with access to an uneditable portal pane, see the solution developer's copy of that portal pane on their home page. Changes to one portal pane copy do not affect another portal pane copy. Nor do the changes affect the portal pane template.

You can modify the following in a portal pane by changing the following settings in the **Portal Pane Settings** section or **Portal Panes Contents** section on the **General** tab of the **Portal Pane** screen:

- Title
- Color
- User options
- Contents

#### To modify a portal pane

1. Access the portal pane you would like to modify. For more information see, [Opening Portal Panes](#).
2. Depending on your needs, do any of the following:
  - Change the desired portal pane settings, such as whether it is editable by users.
  - Add additional content to the portal pane using the data entry rows. For details see [Portal Pane Contents](#).
  - Edit or delete existing content items in the existing content list, as desired.
3. Click **Save**.

**Important:** You must save all your changes *BEFORE* you synchronize settings or content.

4. Click the **Synchronize Settings** and **Synchronize Content** hyperlinks.  
A message to confirm appears.
5. Click **OK**.

All user copies of this portal pane are deleted and then replaced with the solution developer's updated copy, not the template.

You can delete portal panes in the following two places:

- **Portal Panes** screen—Not only deletes portal pane from the list of portal panes, but also from master home pages and users' home page to which it was added.
- An open master home page—Deletes portal pane from the open master home page and from all copies of that master home page (after clicking the **Synchronize Layout** and **Synchronize Content** hyperlinks).

#### To delete a portal pane from the portal pane list

1. In the **Go to** drop-down list, select **Portal Panes**.  
The **Portal Panes** screen appears with all existing portal panes.



Portal Panes Screen


- In the **Portal Panes** list, select the check-box of the appropriate portal pane, and then click **delete**.

A message to confirm the deletion of this portal pane appears.

- Click **OK**.

The portal pane is deleted from the system and you can no longer add it to a home page. For logged in users, this portal pane is deleted from their home page upon clicking **Refresh**, or upon logging out and then logging back in to TeamConnect—even if they have at one time or another modified it.

#### To delete a portal pane from a master home page

- In the **Go to** drop-down list, select **Home Pages**.
- Open the appropriate home page.
- Select the **Portal Panes** tab.
- On the title bar of the portal pane you want to delete, click the **Remove Pane**  button.

A message to confirm the deletion of this portal pane appears.

- Click **OK**.

The portal pane is deleted.

- Click **Save**.

**Important:** You must save all your changes **BEFORE** you synchronize settings or content.

- Click the **Synchronize Content** hyperlink.

A message to confirm that you want to synchronize the master home page with user's copies of this home page appears.

- Click **OK**.

All home pages are in synchronization with the master home page.

#### 3.1.8.4.3 Modifying Master Home Pages

You can preview the layout of a master home page while you are modifying it.

You can view and edit existing portal panes that are already displayed on a particular home page. You can also add other existing portal panes from the **Content** tab of an existing home page. Further, you can create new portal panes directly from the master home page. For details, see [Creating Portal Panes from a Master Home Page](#).

After clicking the **Content** tab, a layout preview of the selected home page appears, along with the **Select Portal Pane** drop-down list. All of the portal panes that appear on the **Content** tab function exactly as they would on the user's home page. You can modify the function, position, and size of these portal panes by clicking the buttons that appear in the **Title Bar** of each individual portal pane. These buttons will always be visible to you on the **Content** tab, even if they are not visible to the user on the user home page.



The screenshot shows a 'Portal Panes' interface with a blue header. Below the header, there is a dropdown menu for 'Yahoo News Headlines' with 'add' and 'new' buttons. The main area contains several panes, each with a title bar and control buttons. The panes include:

- Employee Message:** 'To All ClientGuard Employees: Status reports are due today.'
- My Tasks:** A table with columns 'Due', 'Task', 'Project', 'Category', and 'Priority'. It shows a task due on 07/27/3903 with the task 'Schedule follow-up meeting'.
- My Appointments:** A table with columns 'From', 'To', 'Subject', and 'Project'. It shows an appointment from 07/23/2003 to 07/23/2003 with the subject 'Judgement Hearing (County Clerk's Office)'.
- Yahoo News Headlines:** A list of news items with links, such as 'General Confirms Saddam's Two Sons Killed in U.S. Raid' and 'Fire Near Top of Eiffel Tower Extinguished'.

Home Page Content Tab with Portal Panes Added


Home Page Content Tab with Portal Panes Added

Item	Description
------	-------------

<b>Select Portal Pane drop-down list</b>	Select the desired portal pane you would like to add to the selected home page. All portal panes that were created from <b>Portal Panes</b> are listed.
<b>add</b>	Click to add the selected portal pane to the home page.  If you are adding portal panes to a home page that is already available to a user, the additions are not displayed for the user until he or she clicks the <b>Refresh</b> hyperlink.
<b>new</b>	Click to create a new portal pane directly from the displayed master home page.  <i><b>Important:</b> When you create a new portal pane by clicking this button, the portal pane that you create is not listed on the <b>Portal Panes</b> screen. It is created for this particular home page only.</i>  For details, see <a href="#">Creating Portal Panes from a Master Home Page</a> .
<b>Selected Portal Panes</b>	These are the portal panes that have already been added to the selected home page. The way that they are arranged is the way the user will see them.
<b>Title Bars</b>	Click the appropriate buttons that appear on the title bar to modify the individual portal pane accordingly.
<b>edit</b>	Click to change the selected portal pane.  Clicking this button opens the <b>General</b> tab of the selected portal pane where you can modify the settings of that portal pane. To learn more about modifying portal panes, see <a href="#">About Modifying Portal Panes</a> .
<b>Move Pane</b>	Click the appropriate arrow button to move the selected portal pane to a different position on the home page.  Depending on where the portal pane is positioned in the home page, you may have multiple arrow buttons in the title bar that point up, down, left, or right.
<b>Maximize Minimize</b>	Click to maximize the selected portal pane. After you click the <b>Maximize</b>  button, the portal pane occupies the entire width of the screen, but the height of the portal pane remains the same.  Conversely, if the portal pane is already maximized, the <b>Minimize</b>  button is available. Click this button to resize the portal pane to its original size.



**Remove Pane**

Click the **Remove Pane**  button to remove the selected portal pane from the home page.

The portal pane is removed after the user clicks the **Refresh** hyperlink in the **Personalize** menu.

#### 3.1.8.4.3.1 Adding Portal Panes to Home Pages

Whether you are adding portal panes to a user's home page or editing portal panes that already exist in a user's home page remember the following:

- The user is unable to view the portal panes that you add until after they refresh their screen.
- Portal panes that you remove are still viewable by the user until after they refresh their screen.
- Changes made to a portal pane from the **Portal Panes** tab of the **Home Page** screen will take effect only for that specific portal pane within that specific home page. Other portal panes are not affected.
- When making changes to a portal pane template that already exists as a copy on a user's home page, the changes you make do not take effect until you synchronize the content and settings. For details, see [When Changes Take Effect](#).
- If you change a portal pane that a user has previously customized, the user's changes take precedence. To learn more about solution developer and user customization, see [Modifying Home Pages and Portal Panes](#).

After you save this new portal pane, you cannot reuse it. The portal pane that you created is only intended for this master home page and is not listed on the **Portal Panes** screen.

#### To add portal panes to a home page

1. From the **Content** tab of the desired home page, select the appropriate portal pane from the **Select Portal Pane** drop-down list, and then click **add**.
2. Repeat the previous step to add additional portal panes to the desired home page, if necessary.
3. Click **Save**.

**Important:** You must save all your changes *BEFORE* you synchronize settings or content.

4. Click the **Synchronize Content** hyperlink.

A message to confirm that you want to synchronize the master home page with user's copies of this home page appears.

5. Click **OK**.
6. Click the **Synchronize Settings** hyperlink.

A message to confirm that you want to synchronize the master home page with user's copies of this home page appears.

7. Click **OK**.

The desired portal panes are added to the master home page and all copies are in sync with it.

#### 3.1.8.4.3.2 Creating Portal Panes from a Master Home Page

##### To create a portal pane from a master home page

1. In the **Go to** drop-down list, select **Home Pages**.
2. On the **Home Pages** screen, select the home page to which you want to add a portal pane.
3. Click the **Content** tab, and then click **new**.

A title bar of the new portal pane appears in the open home page.

4. Click **Save** at the top of the **Home Page** screen.

If you attempt to edit your new portal pane before you save your modified home page, a message appears, which prompts you to do so.

5. Click **edit** on the title bar of the new portal pane.

The **Portal Pane** screen appears.

6. Enter appropriate portal pane settings as described in [the Portal Pane Settings table](#).

***Tip:** Do not select the **Editable by user** check-box until your new portal pane is finalized and exactly was you want it.*

7. Select the **Number of entries you would like to add** from the drop-down list, based on how many additional content items you want to add to the portal panes.

8. For each data entry row, do the following actions:

- a. Select the **Content Type** from the drop-down list.

- b. Fill in the fields that display beneath the **Content** column.

For details about each field, see the appropriate content type in the **Portal Pane Contents** section:

- For action link content types, see [the Portal Pane Content of Type Action Link table](#).
- For object link content types, see [the Portal Pane Content of Type Object Link table](#).
- For search view content types, see [the Portal Pane Content of Type Search View table](#).
- For text content types, see [the Portal Pane Content of Type Text table](#).
- For web URL content types, see [the Portal Pane Content of Type Web URL table](#).
- For custom content types, see [the Custom Portal Pane Content table](#).

- For WebIntelligence content types, [the Portal Pane Content of Type WebIntelligence URL table](#).

*Tip: Do not create a portal pane with too much content. Instead, split the content into multiple portal panes.*

- c. Type the **Order**.
  - d. Select the **Alignment** from the drop-down list.
9. Click **add more** to add more content, or click **Save** if you are finished.

*Important: You must save all your changes to the master home page BEFORE you synchronize settings or content.*

10. Click the **Synchronize Content** hyperlink.

A message to confirm that you want to synchronize the master home page with user's copies of this home page appears.

11. Click **OK**.

12. Click the **Synchronize Settings** hyperlink.

A message to confirm that you want to synchronize the master home page with user's copies of this home page appears.

13. Click **OK**.

The newly created portal pane is saved and added to the displayed home page.

### 3.1.9 Using Global Navigation

Global Navigation determines which tabs appear in the TeamConnect tab bar and which navigation items appear in the sub-tabs when you click on a tab. By making changes in the **Global Navigation** screen, you may reorganize existing navigation items. Navigation items link to these kinds of pages:

- System objects
- Custom objects
- Custom tools
- Special collections related to workflow
- Home pages

When you use Global Navigation to alter the appearance or organization of the tab bar, your changes take effect for each user upon logging in. For you to see your own changes in the end-user interface, you must log out and log back in.

*Note: You cannot alter the appearance and contents of some tabs (**Admin, All Services**). TeamConnect controls the contents and positions of these tabs automatically.*

You can reach the Global Navigation portion of the Designer by clicking the **Setup** link in the main user interface, then choosing **Global Navigation** from the **Go To** dropdown list in Designer. Your user group must have Setup rights in order to see the Setup link.

### 3.1.9.1 Working with Tab Bar Appearance

Each tab represents a high-level grouping of main pages. You may wish to alter the appearance of the tool bar to show more or fewer tabs, or change the positions of tabs. The description here explains how to do those tasks.

To learn how to change the navigation items associated with a specific tab, see [Working with Tab Bar Contents](#).

#### To create a new tab

1. Type the name of the new tab in the **New tab name** field.
2. Click **Add**.
3. Your new tab appears to the right of all existing tabs. In the end-user interface, it will be to the right of all tabs except Admin and All Services.
4. Click **Save** or **Save and Close**.



Creating a New Tab

#### To reposition tabs

1. Click on the tab that you wish to reposition.
2. Click on the left-arrow button to move the tab more to the left, or the right-arrow button to move the tab more to the right.
3. If more than one tab needs repositioning, repeat steps 1 and 2 for each tab.
4. Click **Save** or **Save and Close**.



Controls for Repositioning, Deleting, and Renaming Tabs

#### To delete a tab

1. Click on the tab that you wish to delete.

2. Click on the delete ("X") button to delete the tab.
3. In the confirmation message pop-up window, click **Yes**. The remaining tabs adjust to the fill in the gap.
4. Click **Save** or **Save and Close**.

**Important:** *If you delete a tab, end users do not have any way of accessing the navigation items that were associated with that tab, except by finding them in the All Services tab or if you associate those navigation items with another tab.*

#### To rename a tab

1. Click on the tab that you wish to rename.
2. Click on **Rename**.
3. A pop-up window asks you to for the new name of the tab. Type the name and press **OK** to close the pop-up window.
4. Click **Save** or **Save and Close**.

#### 3.1.9.2 Working with Tab Bar Contents

Each tab represents a high-level grouping of navigation items. Some tabs, such as Contacts, are associated with only a single navigation item. TeamConnect recognizes when there is only a single item within a tab. In this situation, when an end user clicks that tab, she goes directly to the linked page.

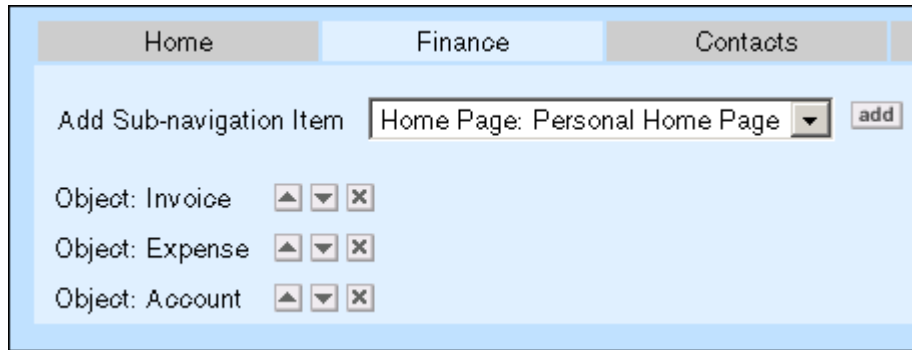
Other tabs, such as Finance, are associated with multiple navigation items. TeamConnect cannot know in advance which item the end user is interested in, so when he clicks such a tab, he sees the available navigation items displayed as sub-tab links in the lower portion of the tab bar. Clicking on one of these sub-tab links then takes him to the linked page.

Only these two forms of navigation are available. You cannot add a navigation item, then add another navigation item as a child of the first item. All items are children of the tab itself.

You can change the navigation items that are associated with a tab in several ways.

#### To add a new navigation item to a tab

1. Click on the tab that you wish to alter.
2. Click on the **Add Sub-navigation** item drop-down list and choose a navigation item. The types of items available are system objects, custom objects, home pages, and custom tools.
3. Click **Add**.
4. Click **Save** or **Save and Close**.



Controls for Repositioning, Deleting, and Creating Navigation Items

#### To reposition navigation items

1. Click on the tab that you wish to alter. The existing navigation items are arranged vertically. The topmost item corresponds to the leftmost link in appearance in the lower part of the tab bar; the bottom item in the list is the rightmost link in the tab bar.
2. Click on the up-arrow button to move the item higher, or the down-arrow button to move the item lower.
3. If more than one item needs repositioning, repeat steps 1 and 2 for each item.
4. Click **Save** or **Save and Close**.

#### To delete navigation items

1. Click on the tab that you wish to alter. The existing navigation items are arranged vertically. The topmost item corresponds to the leftmost link in appearance in the bottom of the tab bar; the bottom item is the rightmost link in the tab bar.
2. Click the delete ("X") button next to an item name.
3. In the confirmation message pop-up window, click **Yes**.
4. Click **Save** or **Save and Close**.

### 3.1.10 Using Templates

Templates define what information is automatically filled in for each type of record and which sub-object or related records are automatically created. You may use templates with wizards, pre-population rules, and custom action rules.

All of these elements support the following functionality:

- Creating a new record and populating its fields
- Creating related and sub-object records for the current object and populating their fields
- Populating assignees and attendees and specifying the main assignee by role
- Specifying the default category with a static value

- Populating categories
- Specifying a phase for custom objects

Pre-population and custom action rules support the following additional template functionality:

- Populating fields of existing records
- Performing phase changes

For information about using templates with rules, see [Using Rules](#). For information about using templates with wizards, see [Whether to Use Templates](#).

### 3.1.10.1 Planning Templates

Your template design depends on your organization's business requirements, as well as the object definition in question. To create useful and effective templates, you must do the following actions:

- Complete the object definitions for which you want to create a template. For more information, see [Creating and Defining Objects](#).
- Familiarize yourself with the TeamConnect object model, whose attributes are described in the [Object Model: Read This First](#) plus the additional related reference tables.
- Learn how to use Object Navigator, which you may use to define fields in templates. See [Using Object Navigator](#).
- If you need to use templates with wizards, plan any necessary wizards before creating templates. See [Creating Wizards](#).

## General Considerations

The following checklist helps you develop a general outline of the templates you may need to create and what kind of information has to be collected:

- Determine the possible scenarios that may take place when a record is created in your organization, such whether users create most records using wizards or also with new, blank records.
- Determine how many wizards and templates you need to create based on the various scenarios, such as one complex wizard with multiple pages and page transition rules creating multiple related records, or multiple small wizards, each taking care of a specific object record.
- Identify the information that must be automatically populated in every new record.
- Determine whether any related records need to be automatically created together with every new record. For example, tasks to be completed, an audit history record to capture the details of a new record, and so on.
- Determine whether any sub-objects need to be added to a record. For example, default categories to add certain blocks, assignees, and so on.
- Identify any fields that may be automatically prepopulated with static or dynamic values in every new record based on naming conventions, company procedures, and other known patterns.

Whether you are defining template fields, sub-objects, or related objects, make sure you understand the following subjects:

- [Field Value Types](#)
- Values, attributes, and using Object Navigator (see [Using Object Navigator](#))
- TeamConnect attributes (see the [Object Model: Read This First](#) plus the additional reference tables related to this reference).
- [Operator Options for Different Attribute Types](#)
- [Defining Values in Templates](#)
- [Sub-Objects](#)
- [Related Objects](#)

## Points To Remember

Keep the following points in mind when planning templates:

- Creating templates is optional, except for pre-population rules.
- Some fields, such as descriptions, dates, number fields, and so on, may be prepopulated using simple actions in a wizard--without the use of a template.
- Whether you create templates for use with wizards, for use with rules, or both, the methods of defining which fields to populate and records to create are the same.
- You may create several templates for the same object definition.
- You may use only one template per wizard or rule, yet you may link related object templates to the main object template.
- Once you create a template, you may select it from the **Action** tab of your pre-population or custom action rule screen.

If you are creating custom action rules for the **User** or the **Group** object definition, you cannot select a template from the **Action** tab.

- You may select existing templates from the **General** tab of wizard screens.
- You may access or create templates on the **Templates** tab of the object definition. However, the following objects do not have a **Templates** tab:
  - Contact Group (Address Book)
  - Document
  - Embedded objects
  - Group
  - Line Items
  - User
- You may create templates for related objects and select them from the main object template. That way, you may reuse the related object template for multiple main objects and scenarios.



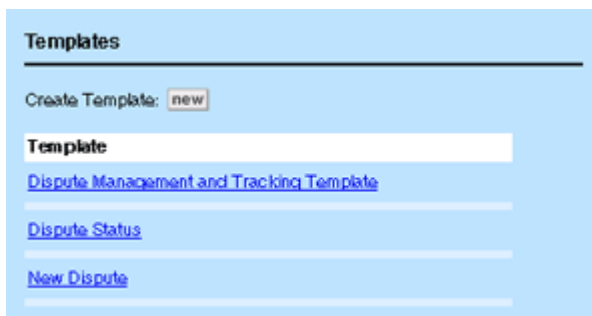
### 3.1.10.2 Using Template Features

Templates are object-specific, so you may access them from the **Templates** tab of the corresponding object definition.

#### To open a template screen

1. In the Designer window, from the **Go to** drop-down list select **Object Definitions**.
2. Select the appropriate object definition.
3. Select the **Templates** tab.

The corresponding **Templates** tab appears with a list of any existing templates defined for the selected object, as shown in the following image.



Templates Tab

4. On the **Templates** tab, do one of the following actions:
  - To open an existing template, click its link.
  - To create a new template, click new.

The **General** tab of the corresponding template screen appears.

5. Click the **Records** tab to view or modify most template definitions.

The template screen has the following tabs:

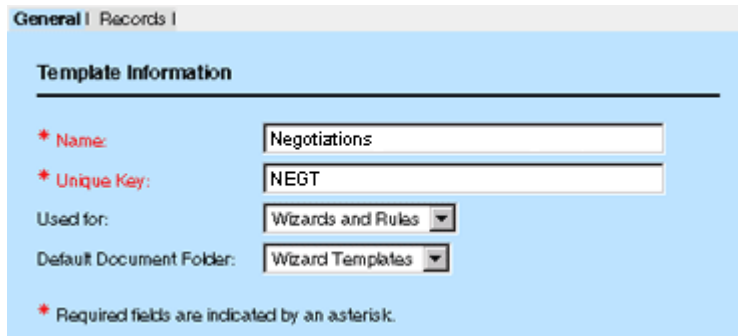
General | **Records**

- **General**—Displays the name of the template and the object it is defined for, and allows you to select a default Documents folder to be associated with each object record. See [General Template Information](#).
- **Records**—Allows you to select fields, sub-objects, and related objects from a tree structure in order to define template fields, their values, and order.

#### 3.1.10.2.1 General Template Information

The **General** tab in the **Templates** screen displays the template's name, its unique key, the name of the object for which it is defined, and the document folder associated with records created by the template.

You may access this tab by following the instructions in [Using Template Features](#).



General | Records |

**Template Information**

\* Name:

\* Unique Key:

Used for:

Default Document Folder:

\* Required fields are indicated by an asterisk.

**General Tab on Template Screens**

The following table explains the contents of the **General** tab on the **Template** screen:

**General Tab on Template Screens**

Field	Description
<b>Name</b>	Enter a name that uniquely identifies the template.
<b>Unique Key</b>	<p>Enter an alphanumeric combination to uniquely identify the template within the object definition.</p> <p><i><b>Note:</b> Templates belonging to different object definitions may have the same key.</i></p> <p>The unique key must be 50 or fewer characters in length and may not contain spaces, underscores, commas, punctuation marks, or any special characters.</p> <p>After you save the template, the unique code appears as read-only text and you may not change it.</p>
<b>Default Document Folder</b>	<p>Select a template folder to have its contents automatically added to the <b>Documents</b> tab of each record that is created based on the template.</p> <p>If you do not create any template folders, this field displays <b>(No Item Selected)</b> in read-only text. For more information about adding folders to records, see <a href="#">Defining Document Folders for Templates</a>.</p>

#### 3.1.10.2.2 Defining Document Folders for Templates

You may select a template folder to have its contents automatically added to the **Documents** tab of each record that is created based on the template. The folder itself is not copied, but any of the following content types may be:

- Subfolders and their contents
- Hyperlinks
- Files, such as Word documents

- Shortcuts to other folders and documents

You may create the appropriate folder structure for the object's templates in the following folder in [TeamConnect's Documents area](#):

#### **Top Level/System/Object Definitions/objectName/Template Folders/**


Where:

objectName is the name of your template's object definition.

For example, if you create the Negotiations/Offers folders within **Template Folders**, the **Negotiations** folder becomes available on the **General** tab of your template.

Each dispute record that is created based on the template has an **Offers** folder on its **Documents** tab that contains any files in the **Offers** folder, such as the **OfferLetter.doc** file. The **Negotiations** folder is not copied to the record.

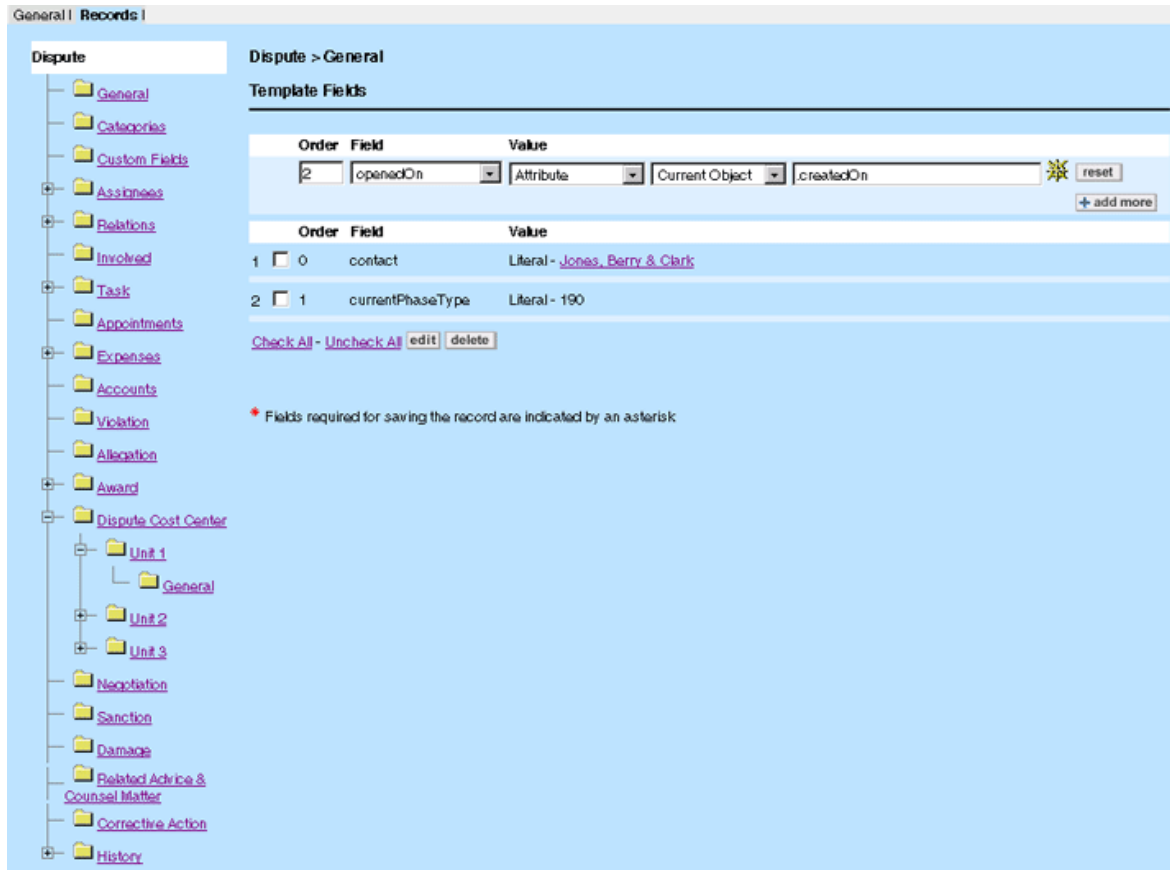
#### **To add folders and their contents to templates**

1. From the end-user interface, select the **Documents** tab.  
Your personal user folder opens.
2. On the Location Indicator, click **Top Level** and navigate to the following folder:  
**Top Level/System/Object Definitions/objectName/Template Folders/**  
Where objectName is the name of your template's object definition.
3. Click **Create New Folder**  and enter a folder name in the **Name** field.
4. Add any subfolders, files, hyperlinks or other contents to the folder.
5. On the **General** tab of the appropriate template screen, select the newly created folder from the **Default Document Folder** drop-down list.  
If no template folders were created for the template's object definition, the **Default Document Folder** field displays **(No Item Selected)** in read only text.
6. Click **Save**.  
The contents of the selected folder are automatically added to the **Documents** tab of each record that is created based on the template.

#### 3.1.10.2.3 Records Tab Structure

The **Records** tab provides a left pane with a directory-tree structure of custom fields, sub-objects, and related objects and a right pane where you may define items in the selected folder.

Depending on the selected folder in the left pane, you may create subfolders for multiple definitions. For example, templates may create multiple involved party records with different categories, if you define multiple subfolders.



Custom Object Records Tab Example

For custom objects, the left pane of the **Records** tab resembles the tree structure of the object definition list that you access in Designer. Every embedded or related child object displayed in the object definition list is available in the **Records** tab. In addition, any applicable sub-objects have folders, such as **Assignees**, **Involved**, **Relations**, **Attendee**, and so on.

When you open an object's folder, you must enter a name in its **Friendly Name** field before you may define any of its values. TeamConnect creates a subfolder with the name you enter, in which it creates a **General** subfolder.

### 3.1.10.3 Creating Templates

You may access or create templates on the **Templates** tab of the corresponding object definition.

To create a template, you must have the following information:

- Object for which you want to create a template.
- System and custom fields you want the template to automatically populate with default values, such as **Name**, **Opened On**. For more details on fields, see [Adding Fields to Templates](#).
- Sub-objects you want the template to automatically add and populate with default values, such as **Categories**, **Assignees**, and **Relations**. For more details on sub-objects, see [Adding Sub-objects to Templates](#).

- Related objects you want the template to automatically create, such as **Tasks**, **Accounts**, **History**, and **Appointments**. For more details on related objects, see [Adding Related Objects to Templates](#).

When you have all this information available you are ready to create your template.

### To create a template

1. Select **Object Definitions** in the Designer window, from the **Go to** drop-down list.
2. Select the appropriate object definition.
3. Select the **Templates** tab.
4. Click **new**.

The **General** tab of a new, blank template screen appears.

5. In the **Name** field, type a name to identify the template.
6. In the **Unique Key** field, type a name or value that uniquely identifies the template within the object definition.  
  
A template's unique key may not contain spaces, underscores, punctuation marks, or any special characters.
7. In the **Default Document Folder** drop-down list, select the folder with contents that you want to automatically populate the **Documents** tab of each record created from the template.
8. Click **Save**.

9. Click the **Records** tab.

The objectName > General folder of the Records tab appears by default.

10. Depending on your template design, add and define the appropriate elements, depending on the following conditions:
  - If you need to define system fields in the template, click the **General** folder (if it is not already displayed).  
  
See [Adding Fields to Templates](#).
  - If you need to add categories to the template, click the Categories folder displayed on the **Records** tab.  
  
See [Adding Categories to Templates](#).
  - If you need to add sub-objects to the template, click the corresponding folder displayed on the **Records** tab.  
  
See [Adding Sub-objects to Templates](#).
  - If you need to add related objects to the template, click the corresponding folder displayed on the **Records** tab.  
  
See [Adding Related Objects to Templates](#).

11. Click **Save**.

12. Link the template to the appropriate wizard, and when the wizard design is complete, test them both.

#### 3.1.10.3.1 Defining Values in Templates

The process of defining template fields, sub-objects, and related objects essentially consists of assigning certain default values to their respective attributes. These attributes usually represent the actual fields in the end-user interface. The values you specify in your template automatically appear in the corresponding fields in the user interface when the user creates a new record based on your template.

Templates use three basic types of field values: Literal (or "static"), Attribute, and Formula, which may be a combination of Literal, Attribute, and operator values. For more information on field values, see [Field Value Types](#). For the explanation of different operators, see [the Operator Options for Different Attribute Types table](#).

#### To define fields in a template

1. Make a list of all fields that you want to be automatically prepopulated with the help of the template you are creating.
2. Decide which default values to assign to each field.
3. Make sure no field definitions in the template conflict with the settings in the selected object definition, for example, Names for custom objects.

For more details, see [Points To Remember](#).

4. Decide which fields also require that their corresponding sub-objects must be defined and prepopulated first, for example, Categories or Assignees.
5. Open the Records tab of the appropriate template.
6. Depending on the value type you want to assign to each field, proceed to the following instructions:
  - [Defining Fields with Literal Values](#)
  - [Defining Fields with Attribute Values](#)
  - [Defining Fields with Formula Values](#)

#### 3.1.10.3.2 Adding Fields to Templates

Templates allow you to define certain fields in an object record to be automatically prepopulated with specified values in each new record created through any wizards or rules to which the template is linked.

Prepopulated fields may include record names, descriptions, amounts, dates, default categories, and so on. For example, every time a user creates a task based on a 30-Day Review Task template, you may want the following fields to be automatically populated with specified values such as:

- Subject
- Due on
- Priority
- Default Category

**Task Fields in End-user Interface**

To accomplish this kind of action using a template, define the following default values to be populated in each record:

**Sample Template Values**

Field label in the user interface	Field	Value	Default value inserted
<b>Subject</b>	shortDescription	Literal > 30-Day Review	<b>30-Day Review</b>
<b>Due on</b>	dueOn	Formula > Attribute > Current Date + Literal > 30 > Days	Depends on the date of creation.
<b>Priority</b>	priorityIID	Literal > High	<b>High</b>
<b>Default Category</b>	defaultCategory	Literal > File Review	<b>File Review</b>

The following image illustrates the values in [the Sample Template Values table](#).

General | Records |

Task > General

Task

- General
- Custom Fields
- History

Template Fields

Order	Field	Value
<input type="checkbox"/>	(Select)	
1	<input type="checkbox"/> 0 currentAssignee	Literal - User Account -
2	<input type="checkbox"/> 0 shortDescription	Literal - 30-Day Review
3	<input type="checkbox"/> 0 priorityIID	Literal - High
4	<input type="checkbox"/> 1 defaultCategory	Literal - File Review
5	<input type="checkbox"/> 2 dueOn	Formula - Attribute - Current Date + Literal - 30 Days

Check All - Uncheck All edit delete

Sample Task Template Values

You may define fields on the **Records** tab of the template screen in the object definition. For instructions on this operation, see [Adding Fields to Templates](#).

For details on attributes, field value types, and paths, see [Field Value Types](#). For details on operators, see [the Operator Options for Different Attribute Types table](#).

## Points To Remember

Keep the following points in mind when adding fields to templates:

- There are certain fields, such as **Default Category** or **Main Assignee**, that you populate only if the corresponding category or assignee is added to the record. This means that in your template, you have to define the corresponding sub-objects first (in this case, a category and an assignee), before setting the corresponding default fields. See [Adding Sub-objects to Templates](#).
- When defining fields for custom objects, keep in mind that the **name** and **number** attributes may or may not be available for selection, depending on the properties defined in the object definition, under the following conditions:
  - If the project name or number are set to be manually entered by the user in the **Name** and **Unique ID** tabs of the corresponding object definition, the corresponding attributes are available in templates.
  - If the project name or number are set to be automatically generated based on the selected pattern in the **Name** and **Unique ID** tabs of the corresponding object definition, the corresponding attributes are not available in templates.
- Depending on your wizard or rule design, all values assigned to the fields of the main record and any of its sub-objects and related objects may be overwritten by your specified wizard or rule actions.



## 3.1.10.3.2.1 Field Value Types

Regardless of whether you are defining template fields, sub-objects, or related objects, you must specify a field value that is automatically populated when the user creates a new record. As shown in the following table, these field values may be:

- Literal: static
- Attribute: dynamic
- Formula: static-dynamic

**Field Value Types**

Literal	Attribute	Formula
<p>Values that users manually enter or select, also known as "static" values.</p> <p>For example, the value for the <b>Default Category</b> in a claim record is manually set to <b>Category 1</b>, which is set to <b>Auto</b>.</p>	<p>Values that are based on a value entered or selected in another field and defined using Object Navigator, also known as "dynamic" values.</p> <p>For example, the value for the sub-object <b>Assignee 2</b> is set to the user who created the record.</p>	<p>Values that are derived from a combination of static and dynamic values, also known as "static-dynamic" values.</p> <p>For example, the value for the <b>Due On</b> date of the related object <b>Task 1</b> is set to be 7 days (static value) after the date when the record is created (dynamic value).</p>

You may use pre-population rules in combination with a template to automatically populate text and memo text fields with static values, dynamic values, or string concatenation.

Usually, fields must be populated by a value of the same type. For example, a number field must be populated by a number. You may prepopulate text and memo-text fields with the following types of values:

- **Boolean**—Populate the field with **Yes** or **No** according to the boolean value (Boolean values are generally shown in the end user interface as check-boxes)
- **Date**—Populate the field with the system date and time formats
- **Number**—Populate the field with a number as text
- **Record**—Populate the field with the display string of the record (its name or number)

You may use pre-population rules in combination with a template to automatically populate number fields using the following calculations:

- Addition
- Subtraction
- Multiplication
- Division

- Percentage of a number
- Ratio between two numbers
- Total values from multiple child records, such as the sum of the **Amount** field from all line items of an invoice
- Number of items in a list, such as the total number of assignees for a matter record

### 3.1.10.3.2.2 Defining Fields with Literal Values

#### To define a field with a static value in a template

1. In the Designer window, from the **Go to** drop-down list select **Object Definitions**.
2. Select the appropriate object definition.
3. Select the **Templates** tab, and then select a template.
4. Open the **Records** tab.
5. Select the **General** folder.
6. Enter the **Order** of the field.  
The order is important only when the value of one field depends on the value from another field you are defining.
7. From the **Field** drop-down list, select the field you want to define.
8. From the **Value** drop-down list, select **Literal**.
9. Enter or select the appropriate value in the **Literal** field.

For specific instructions and examples of how to enter a static value for different field types, see the following table.

10. Click **add more**.

#### Static Value Examples for X Fields

Attribute type	Instructions
<b>Date</b>	Enter the appropriate date manually or select it from the <b>Pop-up Calendar</b> . <i>Note: The format depends on the selected preferences.</i>
<b>Text</b>	Enter the appropriate text string.
<b>Number</b>	Enter the appropriate integer or real number.
<b>Custom Object</b>	Search for the appropriate record and select it using the search module.

<b>Lookup Tables</b>	Select the necessary item from the drop-down list.
<b>Users</b>	Select the necessary user from the drop-down list.  <i>Note: Depending on the selected preferences, this field may be displayed as a user search module.</i>
<b>Check-Boxes</b>	Select or clear the check-box.
<b>IID</b>	Select the item from the drop-down list.

### 3.1.10.3.2.3 Defining Fields with Attribute Values

Attribute values are based on a value entered or selected in another field. They are also known as "dynamic" values. You may define dynamic values using Object Navigator.


#### To define a field with a dynamic value in a template

1. In the Designer window, from the **Go to** drop-down list select **Object Definitions**.
2. Select the appropriate object definition.
3. Select the **Templates** tab, and then select a template.
4. Open the **Records** tab.
5. Select the **General** folder.
6. Enter the **Order** of the field.

The order is important only when the value of one field depends on the value from another field you are defining.

7. From the **Field** drop-down list, select the field you want to define.
8. From the **Value** drop-down list, select **Attribute**.

Another drop-down list appears with applicable values, such as **Current Object**.

9. Click the **Object Navigator**  icon to create the appropriate attribute path.

For details, see [Using Object Navigator](#).

10. Click **add more**.

### 3.1.10.3.2.4 Defining Fields with Formula Values

Formula values are derived from a combination of static and dynamic values. They are also known as "static-dynamic" values. You may define formula values using Object Navigator, Literal values, and operators such as & or +.

**To define a field with static-dynamic values in a template**

1. In the Designer window, from the **Go to** drop-down list select **Object Definitions**.
2. Select the appropriate object definition.
3. Select the **Templates** tab, and then select a template.
4. Open the **Records** tab.
5. Select the **General** folder.
6. Enter the **Order** of the attribute.

The order is important only when the value of one field depends on the value from another field you are defining.


7. From the **Field** drop-down list, select the field you want to define.
8. From the **Value** drop-down list, select **Formula**.

Formulas are only available for fields where a formula makes sense. For example, the `activityItem` field does not support a formula.

A set of drop-down lists appears including Literal, Attribute, and available operators to create the appropriate static-dynamic value.

9. Select the appropriate values from the drop-down lists to define the appropriate field.

Additional fields appear to allow you to select or enter the appropriate Literal or Attribute values.

10. Click the **Object Navigator**  icon to create the appropriate attribute path.

For details, see [Using Object Navigator](#).

11. Click **add more**.

### 3.1.10.3.3 Adding Sub-objects to Templates

To automatically populate sub-objects using of wizards or rules, you may add the corresponding sub-objects to your template, such as categories, assignees, contact addresses, and so on.

Each sub-object consists of a set of fields that allow specific values to be assigned to them to create a valid record.

**To add a sub-object**

1. In the Designer window, from the **Go to** drop-down list select **Object Definitions**.
2. Select the appropriate object definition.
3. Select the **Templates** tab, and then select the appropriate template.

The **General** tab of the template screen appears.

4. Select the **Records** tab.

If no sub-objects have been defined in the selected template, only their main folder is available. That is, no subfolders have yet been defined.
5. Select the appropriate sub-object folder in the left pane.

If the sub-object supports a default value, such as a default **Fax Number** in a Contact template, you may define it in the **Set Default** section using Literal and Attribute values.
6. Depending on the value type you want to assign to the default value, perform one of the following instructions:
  - [Defining Fields with Literal Values](#)
  - [Defining Fields with Attribute Values](#)
  - [Defining Fields with Formula Values](#)
7. If you want to add additional, non-default values, enter a text value in the **Friendly Name** field and click **add more** to create a subfolder.

The value of the **Friendly Name** field is only used within the template. It does not appear in the resulting record.
8. If you have defined a **Friendly Name**, select its new subfolder.
9. Depending on the value type you want to assign to each field, perform one of the following instructions listed in step 6.
10. Click **Save**.

#### 3.1.10.3.3.1 Adding Categories to Templates

##### To specify a category in a template

1. On the **Records** tab, click the **Categories** folder.
2. Select the appropriate default category from the **Category** list.
3. Click **add more**.

The category has been added to the template and are added to records created by the wizard or rule that uses the template.

You may now add custom fields of that category to the template.

To prepopulate the default category of object records, specify the default category with a static value in the template.

##### To specify the default category in a template

1. On the **Records** tab, click **Categories**.
2. Select the appropriate default category from the **Category** list.

3. Select the check-box next to the appropriate category in the list.
4. Click **set as default**.

The default category has been specified in the template and is added as the default category in records created by the wizard or rule that uses the template.

#### 3.1.10.3.3.2 Adding Assignees and Attendees to Templates

You may use pre-population rules in combination with a template to automatically populate the following elements:

- Task assignees
- Project assignees

To populate these elements, you must specify a user, group, contact, or contact group (Address Book). When specifying assignees or attendees, it is best to select a user instead of contacts or contact groups, because only contacts that are active users are assigned or included as an attendee. Adding contacts that are not active users does not cause an error to be displayed, but no assignee or attendee is added.

When users manually enter assignees in a task or project record, the system sets some assignee values automatically, such as **Status** and **Assigned On**, so you do not need to specify them in your template.

**Assignees**

1 - 4 of 4

<input type="checkbox"/>	Line	Primary	Status	Assignee	Role	Action
<input type="checkbox"/>	1	<input checked="" type="radio"/>	Active	<a href="#">Granite, Stacy</a>	Attorney	- +
<input type="checkbox"/>	2	<input type="radio"/>	Active	<a href="#">Smith, Cain</a>	Paralegal	- +
<input type="checkbox"/>	3	<input type="radio"/>	Active	<a href="#">Norman, Kim</a>	Attorney	- +
<input type="checkbox"/>	4	<input type="radio"/>	Active	(Select) <input type="button" value="v"/>	(Select) <input type="button" value="v"/>	- +

**Assignees Batch Screen**

You must define any required values in your template, which users "manually" enter, if you want to create a valid sub-object definition. For example, you must include the user's name and role.

**Important:** In order for a sub-object to be valid you must define its required values.

You may define assignees in a template and then set one as the default assignee.

#### To define an assignee in a template

1. Open the **Records** tab of the appropriate template.
  2. Select the **Assignees** folder.

Although Assignee is a sub-object of the Task object, task templates do not display a separate Assignee folder. You may define the Assignee field in the main General folder of task templates.
  3. Enter a text value in the **Friendly Name** field and click **add more** to create a subfolder.

The value of the **Friendly Name** field is only used within the template to name the subfolders. It does not appear in the resulting record.
  4. Select the new subfolder.
  5. Enter a value in the **Order** field.

The order is important only when the value of one field depends on the value from another field you are defining.
  6. From the **Field** drop-down list, select appropriate field name, such as **user**.
  7. From the **Value** drop-down list, select **Literal**.

Another drop-down list appears with applicable values.
  8. Select **User Account**.

Another drop-down list appears with available usernames. If there are more than 100 users, a search module appears.
  9. Select the appropriate username.
  10. Click **add more**.
  11. To assign a role to the assignee, select **type** from the **Field** drop-down list.
  12. From the **Value** field, select **Literal**.

Another drop-down list appears with applicable values.
  13. Select the appropriate role from the list.
  14. Click **add more**.
- Once you have defined at least one Assignee value in a template, you may select one as the default assignee.

#### To define the default assignee in a template

1. Open the **Records** tab of the appropriate template.
2. Select the **Assignees** folder.
3. In the **Set Default Assignee** section, select **Literal**.
4. From the drop-down list to the right of the **Literal** value, select the friendly name you specified for the folder with the appropriate assignee.

5. Click **add more**.

#### 3.1.10.3.3.3 Deleting Sub-objects from Templates

Do not delete sub-objects that were added for the following reasons:

- To make custom fields created for a category (other than Root) accessible in a wizard.
- To prepopulate certain types of fields in the template that require a sub-object, for example, Main Assignee or Default Category.

#### To delete a sub-object from a template

1. In the Designer window, from the **Go to** drop-down list select **Object Definitions**.
2. Select the appropriate object definition.
3. Select the **Templates** tab, and then select a template.
4. Select the **Records** tab.
5. Select the sub-object's folder in the left pane.
6. Select the check-box next to the sub-object's **Friendly Name** field.
7. Click **delete**.

The sub-object definition is removed from the template.

#### 3.1.10.3.4 Adding Related Objects to Templates

Related objects are those objects whose records may be associated or linked to another object record and may be accessed directly from that object record. For details on the TeamConnect objects and their default relationships with one another, see [About Objects](#).

Related objects in templates are associated with the main object in the following hierarchical ways:

- Child-parent
- Involved object
- Embedded object

Non-hierarchical related objects are not available on the **Records** tab of templates.

The **Records** tab of template screens displays a folder for each related object in the object definition. For example, a Dispute (custom object) template includes folders for the following related objects:

- Involved
- Task
- Appointments
- Expenses
- Accounts



- Embedded projects
- Child projects
- History

You may use templates to specify which related object records to create and which of their fields to populate with values.

**Note:** Although Document is a related object to projects, you may not define documents on the **Records** tab of the template screen. For more information, see **Default Document Folder** in [the General Tab on Template Screens table](#).

## Points To Remember

Consider the following points when defining related objects in a template:

- You have to add a separate related object for each associated record you want to be automatically created every time a new record is created. For example, if you need one account, one history record, five involved records, and five task records to be created, you must add twelve related objects in your template.
- You have the option of either creating a separate template for each related object or defining it within the main template.
  - If you need to define only a few default values for certain fields in a related object, you may want to do this within the main template, such as the **Due Date** and **Description** for a task.
  - If you need to define a more complex related object, with sub-objects, you have to create a separate template for it and the link it to the main template, for example, an account with its default category, posting criteria and a set of child accounts.
- You may also define a default document folder for each new record that is created based on the template. See **Default Document Folder** in [the General Tab on Template Screens table](#).

You may define sub-objects on the **Records** tab of the template screen in the appropriate object definition.

### To add a related object to a template

1. Make a list of all associated records you want to be created and linked to each record created based on the object template you are creating.
2. Decide which related objects you need to define for each associated record.
3. Decide what default values you want to be assigned to each attribute.
4. Decide for which related objects you might need to create separate templates.

See [Points To Remember](#).

5. Create the necessary separate templates to be linked to the selected related objects in the template you are creating.
6. Open the **Records** tab of the appropriate object template. See [Using Template Features](#).

7. Select the appropriate related object folder in the left pane.
8. Enter a text value in the **Friendly Name** field and click **add more** to create a subfolder.

The value of the **Friendly Name** field is only used within the template to name the subfolders. It does not appear in the resulting record.
9. Select the new subfolder.
10. Do one of the following actions:
  - Select **Use an existing template** if you want to use the fields and sub-objects that are already defined for the selected object within an existing template.
  - Select the **General** subfolder of the friendly named folder if you want to define fields and sub-objects that are already defined for the selected object within an existing template.

The process of defining values for this option is the same as defining individual fields on the **Fields** tab of a template. The only difference is that you may not define fields that require sub-objects (for example, assignees). In this case, you must create a separate template for the related object and select the **Use an existing template** option to link it to the current template.
11. If you select the **General** subfolder option, perform to the following instructions, depending on the value type you want to assign to each attribute:
  - [Defining Fields with Literal Values](#)
  - [Defining Fields with Attribute Values](#)
  - [Defining Fields with Formula Values](#)

## Adding Involved Objects

You may define Involved object values in a template to automatically add an involved party record and populate its **Contact** field. To do this, you must specify one of the following values:

- An active group or user as a dynamic value
- A group as a static value (the group must include at least one active user)
- A user's contact as a static value
- A contact group (Address Book) as a static value (the contact group must include at least one active user's contact record)
- A dynamic value ending with the following object attributes:
  - assigneeList(role).user
  - assigneeList(Any).user

**Note:** Only active assignees may be used.

### 3.1.11 Creating Wizards

A wizard is a user-interface utility that helps users create valid records by guiding them through the data-entry process, executing rules, populating defined values, and saving the record on completion.

When creating a new record, users have one or both of the following options:

- Creating the record manually by using the record screens to enter data in various fields. The record screens may not be organized in the same order in which users collect information when they create records.
- Using a wizard designed to guide users through the process of entering required information in predefined pages.

You may define rules for wizards to make sure users enter valid data. You may also create wizards that use templates, which automatically insert specific values into every record created from the wizard.

For more information about templates, see [Using Templates](#).

### Accessing Wizards

Once you define wizards, users can access them in the following locations in TeamConnect:

- The **New** button in the list view of a record—When you click this button, a drop-down list one or more wizards that are defined for the selected record type.
- The **New** button in a related object block—When you click this button, a drop-down list displays all of the wizards that are available for the selected related object.

#### 3.1.11.1 Designing Wizards

When designing a wizard, you must consider various scenarios that users may face when creating records. For example, you need to determine the minimum information that must be captured by the wizard to quickly create a valid record, which fields may be automatically prepopulated, and how object definition rules might conflict with the wizard.

You must also design the layout of individual pages, page components, and page order. You may be able to reuse existing blocks of fields used in custom object record screens or create new blocks for the wizard.

Consider the following when designing wizards:

- [General Considerations](#)
- [Writing a Wizard Specification Document](#)
- [Whether to Use Templates](#)
- [Designing Wizard Layout](#)
- [Identifying Wizard Flow and Page Transitions](#)

### Prerequisites

To create a set of useful and effective templates and wizards, you must meet the following requirements:

- Define your organization's business requirements and accepted practices.
- Understand TeamConnect's object model, whose attributes are described in [Object Model: Read This First](#) and the additional related reference tables.
- Understand template and wizard basics and the major TeamConnect components, such as sub-objects and related objects.

For details, see:

- [Planning Templates](#)
- [Adding Pages to Wizards](#)
- [Introduction to Customization](#)
- Write a wizard design specification based on your system design. It must specify custom fields, blocks, and so on, which need to be included in the wizard. For an example of this document, see [the Wizard Design Specification Document Example table](#).

## General Considerations

Use the following checklist to develop a general outline of the wizards you may need to create and what kind of information has to be collected:

- Determine the possible scenarios that may take place when a user must create a record.  

For example, in a legal department, a representative receives a call reporting a transaction. The representative attempts to create a Transaction record in TeamConnect. The appropriate wizard checks whether the user is a member of the Transaction Attorneys or Paralegal groups and preselects the wizard pages according to the type of information the current user is allowed to enter.
- Identify the information that must be collected when creating a record through each wizard and whether child records must be created and prepopulated with dynamic or static values.  

For example, certain wizards may assign a default category for the record, then create related records such as task, account, or audit-history records.
- Decide how many wizards are necessary to account for different scenarios, for example, one complex wizard with multiple pages and page transition rules or several smaller wizards, with each taking care of a specific scenario.  

For example, you may want to create separate wizards to collect transaction details depending on the specified categories, and separate wizards for different involved party roles or child records.

Once you have a general outline for the wizard, you might want to write a specification document detailing all the wizard requirements. If you decide to create more than one wizard, the best practice is to create a separate document for each wizard. See [Writing a Wizard Specification Document](#) for more details.

## 3.1.11.1.1 Writing a Wizard Specification Document

A wizard specification document details all wizard requirements by determining the following information for each step (page) of the wizard:

- Object for which the wizard and any associated templates must be created.
- System and custom fields, or blocks of fields, which capture the appropriate information that needs to be gathered.
- Required fields as specified in the object definition, such as:
  - The parent project field.
  - Fields whose values are used in the unique ID or name patterns.
  - Fields that are required by the rules created for the selected object on the Create trigger--create rules are not triggered until the user reaches the end of the wizard, which is when the record is created.
  - Required custom fields to be included in the object view, and so on.

For more details on required fields, see [Required Fields](#).

- Additional fields necessary only in the wizard. For example, parameter fields might require the user to enter a **Yes**, **No**, or **Unknown** value to determine the next wizard page. Such fields are not saved or displayed in the resulting record. For details, see [Defining Parameters](#).
- Default values for certain fields.
- Additional rules to ensure the needed values are entered through the wizard.

For example, if a user selects **Yes** in response to a wizard question, the user must make at least one selection from a set of later-displayed check-boxes. This requirement might be accomplished using a custom block or by creating a page transition rule that directs the user to an end page.

- Error messages displayed for the user if a page transition rule is triggered.
- Prompts to help the user enter information for the fields in the record.
- The field order on each page.
- Page transitions, including dependencies on the input from the user.

The following table provides an example of a wizard requirements specification document for an insurance claim.

**Wizard Design Specification Document Example**

#	Wizard question	Field type	Field label	Display in record	Wizard page component	Comments
1	What is the caller's first name and last name?	Text	None	No	Parameter	If the caller is a third party, an action must be

						written to add the names to the involved party record.
2	What is the caller's phone number?	Text	None	No	Parameter	If the caller is a third party, an action must be written to add the number to the new contact record.
3	Is the caller insured or third party?	Radio Button	None	No	Parameter	Insured or third-party values <ul style="list-style-type: none"> <li>• If Insured, go to #4</li> <li>• If third party, go to #16</li> </ul>
4	What is the insured's policy number?	Search Module	Policy Number	Yes	System Field	This is the name or number of the policy.
5	When did the accident occur?	Date & Time	Date of Loss	Yes	Custom Field	This is the date of the accident.
6	Where did the accident occur? 1 Location 2 State	3 Text field 4 List	5 Accident Location 6 Location State	7 Yes 8 Yes	9 Custom Field 10 Custom Field	This is the physical location (for example, intersection of 5th and Main St.)  Use State List lookup table
7	Did the accident involve an intersection?	Radio Button	None	No	Parameter	Yes or No Value <ul style="list-style-type: none"> <li>• If Yes, go to #8</li> <li>• If No, skip to #9</li> </ul>
8	What Type of Intersection was involved?	List	Intersection Type	Yes	Custom Field	Default to No.

9	Did the accident involve traffic controls or signage?	Radio Button	None	No	Parameter	Yes or No Value <ul style="list-style-type: none"> <li>If Yes, go to #10</li> <li>If No, skip to #11</li> </ul>
10	What Type of Traffic Control was involved?	List	Traffic Control	Yes	Custom Field	Default to No.
11	Please describe the accident.	Memo Text	Description of Loss	Yes	Custom Field	For example, the insured vehicle failed to yield right of way and made a left turn in front of claimant vehicle.
12	Are there any Area Risk Factors that apply to this accident?	Radio Button	None	No	Parameter	Yes or No Value <ul style="list-style-type: none"> <li>If Yes, go to #13</li> <li>If No, skip to #14</li> </ul>
13	Indicate the Area Risk Factors that apply to this accident.	Check-Boxes	Multiple - Area Risk block	Yes	Block	A rule must be added to require at least one selection.
14	Additional required information					

#### 3.1.11.1.2 Whether to Use Templates

Templates define what values are automatically populated for each type of record and what related records are automatically created. You may create wizards in association with a template or independently of a template.

**Note:** The information listed in your wizard specification document helps you decide whether using a template is worth the additional effort (see [Writing a Wizard Specification Document](#)).

If any of the following conditions exist, you must create a template for your wizard:

- Sub-objects that need to be added to the record. For example, you need to add sub-objects if you are:
  - Using custom fields of a category other than root in your wizard

- Setting a certain non-root category, such as **Lawsuit**, as a default category for all your records created from the wizard
- Adding several assignees to the record and setting one of them as the main one by default
- Fields that may be automatically prepopulated in every new record, such as the **Default Category** or the **Main Assignee**.
- Related records that need to be automatically created together with every new record, such as tasks or accounts.

You may create separate templates for associated records, such as accounts or invoices.

If you only need to prepopulate individual fields that do not require sub-objects, such as descriptions, dates, number fields, and so on, they may be prepopulated with the help of simple actions in the wizard itself without the need for a template.

### 3.1.11.1.3 Designing Wizard Layout

Once you specify the information that you want to capture, you may design the layout of your wizard.

Organize the fields and any questions into separate groups as you would expect them to appear in each page of the wizard, such as the following example:

<p><b>Page 1</b></p> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;"><b>Caller Information</b></p> <p>Caller's First Name <input style="width: 80%;" type="text"/></p> <p>Caller's Last Name <input style="width: 80%;" type="text"/></p> <p>Insured?      <input type="radio"/> Yes   <input type="radio"/> No</p> </div>	<p><b>Page 2</b></p> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;"><b>Insured's Policy Information</b></p> <p>Policy Number <input style="width: 80%;" type="text"/></p> <p>Policy Name <input style="width: 80%;" type="text"/></p> </div>
<p><b>Page 3</b></p> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;"><b>Claimant's Information</b></p> <p>Address <input style="width: 80%;" type="text"/></p> <p>City <input style="width: 80%;" type="text"/></p> <p>Zip Code <input style="width: 80%;" type="text"/></p> <p>Country <input style="width: 80%;" type="text"/></p> <p>Home Phone <input style="width: 80%;" type="text"/></p> <p>Business Phone <input style="width: 80%;" type="text"/></p> </div>	<p><b>Page 4</b></p> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;"><b>Accident Details - General</b></p> <p>Date &amp; Time of Loss <input style="width: 60%;" type="text"/> <input style="width: 20%;" type="text"/></p> <p>Loss Location <input style="width: 80%;" type="text"/></p> <p>State <input style="width: 80%;" type="text"/></p> <p>Did the accident involve an intersection?  <input type="radio"/> Yes   <input type="radio"/> No</p> </div>

**And so on**

**Wizard Page Layout Example**

**Tip:** If necessary, you may add wizard prompts to wizard pages as a component of the type Text.

Through the process of laying out page components, you may gradually determine the contents of each page, such as individual custom and system fields, blocks, text, or parameters. For more details, see [Defining Page Components in Wizards](#).

Through this process, you may decide to create new custom blocks for certain wizard pages.



**Tip:** You may reuse the blocks created for object views of the object for which you are creating the wizard, or create new custom blocks for wizard pages, for example, to include parameters in them.

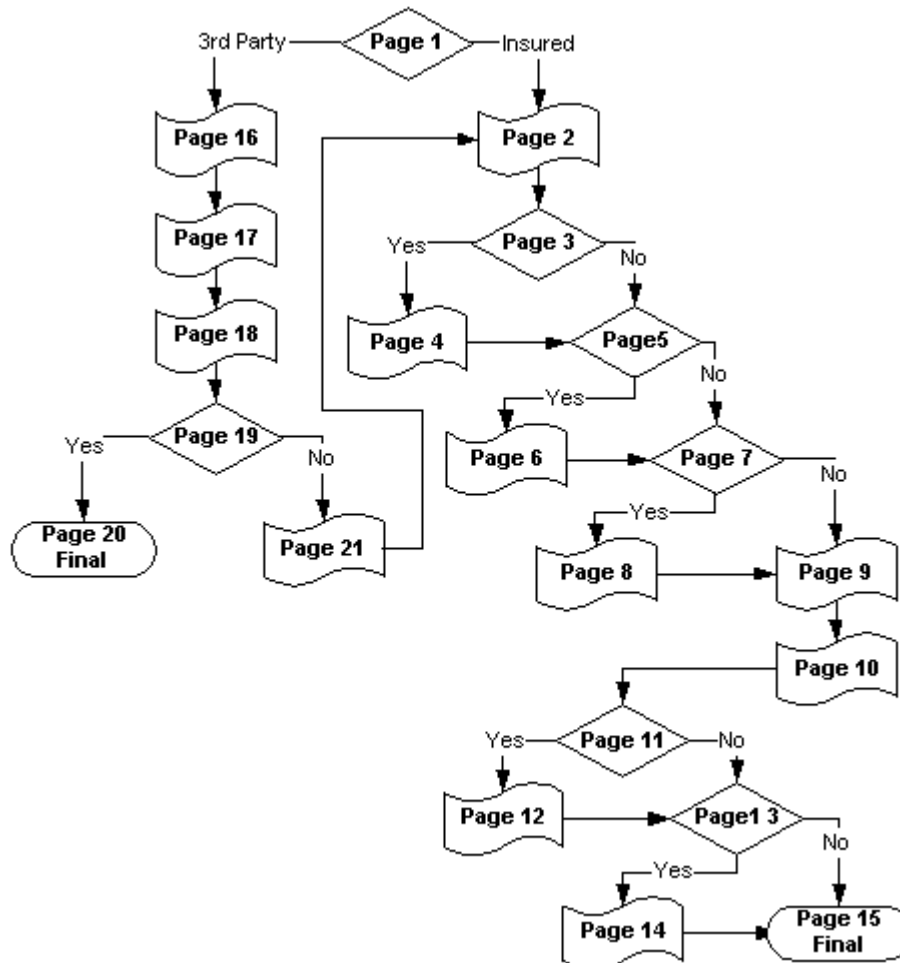
#### 3.1.11.1.4 Identifying Wizard Flow and Page Transitions

After you have decided on the page layout, you need to make sure the pages appear in the appropriate order to the user creating a record. Plan your page order as follows:

- Finalize the general flow of the wizard. Namely, define the sequence in which the wizard pages follow each other through page transitions.

For example, in the following diagram, an alternative sequence may result from Page 1 if the involved is a third party.

- Identify the pages that require page transition rules and the qualifiers. These pages are marked by the diamond shapes in the following diagram.
- Determine how many final pages may be required. These requirements include pages that allow the user to exit the wizard and pages that allow them to save the record, and dependencies such as missing information, limited rights, and so on. These pages are marked by the oval shapes in the following diagram.



Wizard Page Transition Diagram Example

- Make a list of the page transition rules you have to create for page transitions based on the input data. You must:
  - Identify the parameters used in qualifier items (conditions) for each rule.
  - Identify the pages to which you want the user to be directed in each of the possible alternative flows.

The order you indicate for the wizard pages, as you add them to the wizard, is always set as default order.

### 3.1.11.2 Creating Wizards

Once you plan your wizard, determine its alternative flows, its wording, and the necessary fields and blocks, you are ready to create wizard pages. For each page, you must define individual components, define actions, and specify the necessary page transitions.

The following general procedure summarizes the process that is necessary for creating a wizard and refers to other sections in this guide for details:

1. Create a wizard specification document and complete all the prerequisites outlined in [Designing Wizards](#).

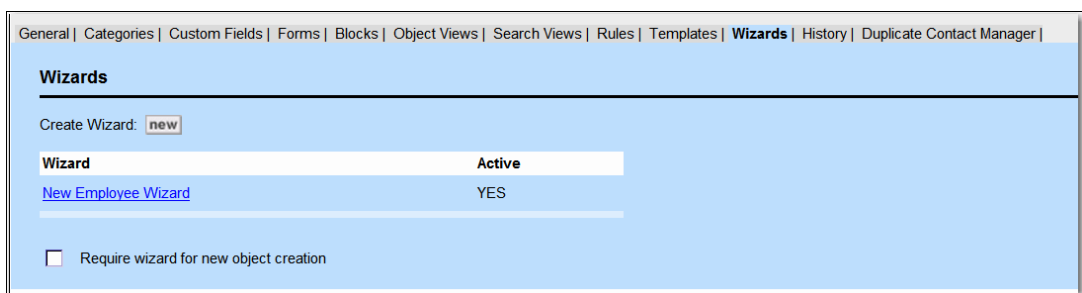
- Determine whether you are going to use a template with your wizard. See [Whether to Use Templates](#).
- If necessary, create the appropriate templates. See [Defining General Wizard Information](#).
- In the Designer window, from the **Go to** drop-down list select **Object Definitions**.
- Open the appropriate object definition.
- Click the **Wizards** tab, and then click **New**.
- Define the general wizard information as described in [the General Tab on Wizard Screens table](#).
- Click the **Pages** tab and add all appropriate pages as described in [Adding Pages to Wizards](#).
- Click the **Page Components** tab. Define and add all wizard page components to each page as described in [Defining Page Components in Wizards](#).
- Click the **Actions** tab and define any actions for the appropriate wizard pages as described in [Using Actions in Wizards](#).
- Click the **Page Transitions** tab and modify the default page transition order by adding page transition rules to the appropriate wizard pages as described in [Defining Page Transition Logic](#).
- Click the **General** tab and select the **This wizard is active** check-box and/or the **Require wizard for a new object definition** check-box if necessary.
- Save the wizard and thoroughly test it. See [Testing and Troubleshooting Wizards](#).

#### 3.1.11.2.1 Opening the Wizard Screen

##### To open the Wizard screen from an object definition

- Select **Object Definitions**, in the Designer window, from the **Go to** drop-down list.
- Select the appropriate object definition.
- Open the **Wizards** tab.

The **Wizards** tab opens with a list of existing wizards defined for the selected object (if any).



Wizards Tab

- Do one of the following actions:
  - To open an existing wizard, click the respective link.
  - To create a new wizard, click **new**.

The corresponding **Wizard** screen opens with its **General** tab displayed by default.

The Wizard screen consists of the following tabs:

- **General**—Displays the name of the wizard, the object it is defined for, and its template (if any). You can also activate the wizard and specify that users may only create objects of this type using a wizard. See [Defining General Wizard Information](#).
- **Pages**—Allows you to view and create all the pages for the wizard. See [Adding Pages to Wizards](#).
- **Page Components**—Allows you to view and define the contents of each page in the wizard. See [Defining Page Components in Wizards](#).
- **Actions**—Allows you to define the necessary actions you may want for the selected wizard pages. See [Using Actions in Wizards](#).
- **Page Transitions**—Allows you to check what page follows the currently selected page and to define the appropriate page transition rules. See [Defining Page Transition Logic](#).

#### 3.1.11.2.2 Defining General Wizard Information

The **General** tab of the **Wizard** screen displays:

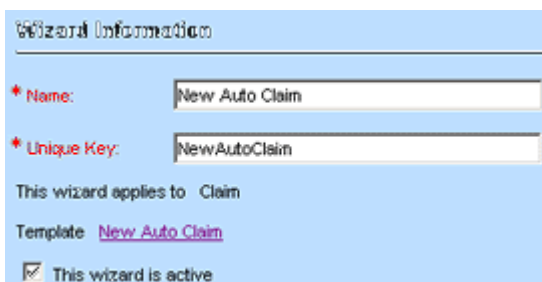
- The wizard's name
- The unique key of the wizard within the object
- The name of the object it is defined for
- The name of the template it is associated with (if any)
- Whether the wizard is active
- Whether to provide only a wizard to create objects of this type

#### To define general wizard information

1. Access the **General** tab of the wizard screen by following the instructions in [Opening the Wizard Screen](#).

The **General** tab of the wizard screen opens by default.

2. Define the information on the **General** tab according to the information in [the General Tab on Wizard Screens table](#).



Wizard Information

\* Name:

\* Unique Key:

This wizard applies to

Template [New Auto Claim](#)

This wizard is active

General Tab on Wizard Screens

The following table describes the items on the **General** tab of the **Wizard** screen.

**Note:** The fields in the table below are the default fields that appear for all object definitions. You may see additional fields that only apply to the selected object definition.

#### General Tab on Wizard Screens

Item	Description
<b>Name</b>	Enter a name that uniquely identifies the wizard and describes its purpose. This name appears to the users when they start to create a new record.
<b>Unique Key</b>	<p>Enter an alphanumeric combination to uniquely identify the wizard within the object definition. It must be 50 or fewer characters in length and must not contain special characters such as spaces, underscores, or punctuation marks.</p> <p>After you save the wizard, the unique code appears as read-only text and you cannot change it.</p> <p>Wizards within the same object definition cannot have the same unique key. However, wizards belonging to different object definitions may have the same key.</p>
<b>This wizard applies to</b>	Automatically displays the name of the object that the wizard is created for.
<b>Template</b>	<p>Select one of the available templates you want to use in the wizard (if applicable).</p> <p><b>Note:</b> If you intend to use a template with the wizard you are creating, make sure you create and define the template first. See <a href="#">General Template Information</a>.</p>
<b>This wizard is active</b>	<p>Select this check-box to activate the wizard. Clear the check-box to deactivate the wizard.</p> <p>This field is available only after at least one page is added to the wizard. Until then, instead of this field, the following message appears: "The wizard is currently inactive. To be able to activate it, please add a page."</p> <p><b>Caution:</b> Even when the check-box appears, do not activate the wizard until it is completely finished and ready for testing.</p>
<b>Require wizard for new object</b>	Select this check-box if you want users to only have the option to create a record using a wizard. When users click the <b>New</b> button, the

<b>definition</b>	<p>wizard opens automatically.</p> <p>Clear the check-box if you want users to have the options to create a record using a wizard and create a record manually using a blank record.</p> <p>This field is only visible if at least one active wizard is defined for this object.</p>
-------------------	--

### 3.1.11.2.3 Adding Pages to Wizards

Wizard pages are represented by separate web page screens in the end-user interface. To create wizards, you add pages, define their individual contents, and specify page transitions. Each page may have a number of fields and may include a **Next** button, a **Back** button, a **Cancel** button, and a **Finish** button.

All of the necessary wizard pages must be added on the **Pages** tab of the corresponding wizard screen in Designer. You must define the order in which the pages must follow each other by default.

Depending on the number of alternative flows in the wizard and their page transitions, you may need to define more than one final page, as described in [Identifying Wizard Flow and Page Transitions](#).

The screenshot displays the 'Create a Subpoena' wizard in the TeamConnect application. The interface includes a top navigation bar with tabs for Home, Legal, Finance, Contacts, Calendar, Documents, Admin, and All Services. Below the navigation bar, there is a search bar and a 'GO' button. The main content area is titled 'Create a Subpoena' and shows 'Step 1 of 5' of the wizard. The 'General Subpoena' section contains several input fields: 'Subpoena Matter Name', 'Date Due', 'Date Rec'd', 'Type of Subpoena' (a dropdown menu set to 'Order to appear'), '\*Issuing Court', and '\*Docket No.'. The 'Assignees' section features a table with one row, showing '1 - 1 of 1' items. The table has columns for 'Line', 'Primary', 'Status', 'Assignee', 'Role', and 'Action'. Below the table are buttons for 'Remove', 'Add New Item', 'Unassign', and 'Reassign'. The wizard also includes 'Cancel' and 'Next' buttons at the top right of the step indicator.

Example: Wizard Page

### To add a page to a wizard

1. Open the appropriate wizard screen and select the **Pages** tab.

The **Pages** tab opens.

**Wizard Pages**

Number of entries you would like to add:

	Title	Order	Hide Page Title
1	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
2	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
<a href="#">+ add more</a>			
	Title	Order	Hide Page Title
1	<input type="checkbox"/> Accident Details - General	9	NO
2	<input type="checkbox"/> Accident Details - Injuries	12	NO
3	<input type="checkbox"/> Accident Details - Injuries	15	NO
4	<input type="checkbox"/> <input type="text" value="Accident Details - Injuries"/>	<input type="text" value="18"/>	<input type="checkbox"/>
5	<input type="checkbox"/> Accident Details - Injuries	21	NO
6	<input type="checkbox"/> Accident Details - Police	30	NO
7	<input type="checkbox"/> Accident Details - Witnesses	24	NO
8	<input type="checkbox"/> Accident Details - Witnesses	27	NO
9	<input type="checkbox"/> Caller Information	3	NO
10	<input type="checkbox"/> Policy Information	6	NO

[Check All](#) - [Uncheck All](#)

**Wizard Pages Tab**

2. Select the **Number of entries you would like to add** from the drop-down list.
3. For each data entry row, define the page title, its order, and whether the page title must be hidden in the wizard as described in [the Wizard Pages Tab table](#).
4. Click **add more** and repeat step 3 until you've added all of the wizard pages.

For each page that you define, you must add the fields and blocks that you want to appear on each page. Click the **Page Components** tab to define the appropriate components for each page of the wizard. For instructions, see [Defining Page Components in Wizards](#).

The following table describes the items on the **Pages** tab of the wizard screen.

**Wizard Pages Tab**

Item	Description
<b>Title</b>	Enter a name for the page, This name appears at the top of the wizard page as its heading. See <a href="#">the Wizard Page image</a> for an example.

<b>Order</b>	<p>Enter a number to specify the default order in which the page is to appear in the wizard.</p> <p><i>Tip: When defining the page order, you may want to skip a few numbers before each page (for example, use 3, 6, 9, 13, 18, and 21, ) in case you need to insert a few additional pages later if your wizard design changes.</i></p> <p>This is also the order in which the names of the pages you are creating appear in <b>Current Page</b> drop-down list of the <b>Page Components</b>, <b>Actions</b>, and <b>Page Transitions</b> tabs and in the left pane when users are using the wizard.</p> <p>The default order that you create for the wizard pages may be modified through the appropriate page transition rules (<a href="#">Defining Page Transition Logic</a>).</p>
<b>Hide Page Title</b>	<p>Select this check-box if you want the page title to be hidden. By default, this option is not selected.</p> <p>If you applied a block template to custom blocks, and your custom blocks have their block titles specified in the XML, this option prevents the title from being displayed twice.</p>
<b>add more</b>	<p>Click to add the defined page to the page list.</p>
<b>Page List</b>	<p>Displays all pages added to the selected wizard page. Select the check-boxes next to the components that you want to delete or edit and click the appropriate command button at the bottom.</p>

#### 3.1.11.2.4 Defining Page Components in Wizards

You may define the contents of each wizard page on the [Page Components tab of the wizard screen](#), which provides the following types of components:

- Text
- Block
- System Field
- Custom Field
- Parameter

To use the **Page Components** tab, you must first create wizard pages.

**Note:** *If duplicate contact checking is enabled and you have created a wizard that creates contacts using the Contact search module (Create New Contacts pop-up window), under certain conditions certain contacts may not be created. See the [Contacts](#) for more information.*



## To define a page component in a wizard

1. Open the appropriate wizard and select the **Page Components** tab.

Wizard Page Components

Current Page:

Component	Order	Value
1 <input type="text" value="Block"/>	<input type="text"/>	<input type="text" value="Access Info"/>
<a href="#">+ add more</a>		
1 <input type="checkbox"/> Text	1	Where did the accident occur?
2 <input type="checkbox"/> Custom Field	2	Label: Location Category: Auto Custom Field: AccidentLocation Default Value:
3 <input type="checkbox"/> Custom Field	<input type="text" value="3"/>	Label: <input type="text" value="State"/> Category: Auto Custom Field: AccidentState Default Value: <input type="text"/>
4 <input type="checkbox"/> Text	4	What are the parties involved in this accident?
5 <input type="checkbox"/> Block	5	WZ Involved Parties
6 <input type="checkbox"/> Text	6	Did the accident involve an intersection?
7 <input type="checkbox"/> Parameter	7	Label: Name: Intersec Type: Radiobutton List List: Yes,No Default Value: No

[Check All](#) - [Uncheck All](#)

Wizard Page Components Tab

2. Select the wizard page you want to add components to from the **Current Page** drop-down list.
  3. From the **Component** list, select the component you want to add to the selected page.  
The available components and their values are described in [the Wizard Page Component Value Fields table](#).
  4. In the **Order** field, enter an integer to indicate the order in which the component must appear in the wizard page.
  5. In the **Value** area, enter or select the appropriate values for the selected component.  
Different fields appear based on the **Component** that you selected.
  6. Click **add more**.
  7. Repeat steps 3 through 6 to add other components to the selected page and click **Save**.
- Repeat steps 2 through 7 for each page in the wizard.

For an example of how the components shown in [the Wizard Page Components Tab image](#) appear in the resulting wizard page, see [the Wizard Page Component Value Fields table](#).

The following image describes the fields in the **Page Components** tab.

#### Wizard Page Components Tab

Field or button	Description
<b>Current Page</b>	Select the wizard page you want to define components for.  All pages in this field are listed in the order assigned to them on the <b>Pages</b> tab.  If no pages have been created for the wizard, <b>(No Item Selected)</b> appears in read-only text.
<b>Component</b>	Select the component you want to add to the selected wizard page.
<b>Order</b>	Enter an integer to indicate the order in which the components must follow each other when displayed in the wizard page.  <i>Tip: When defining the order, you may want to skip a few numbers before each component (for example, 3, 6, 9, 13, 18, 22, and so on.) in case you may need to insert a few additional components later, for example, if your page design changes.</i>
<b>Value</b>	Enter the appropriate values in the corresponding fields displayed in the <b>Value</b> column.  The number and type of fields varies depending on the component selected and the type of field it is represented by on the wizard page.  The <b>Default Value</b> is not a required field in most of the cases, because it depends on the user input.  For details on various <b>Value</b> fields for different components, see <a href="#">the Wizard Page Component Value Fields table</a> .
<b>add more</b>	Click to add the new item to the list of page components.
<b>Page Component List</b>	Displays all components added to the selected wizard page. Select the check-boxes next to the components you want to delete or edit then click the appropriate command button at the bottom.

The following table describes the component values available on the **Page Components** tab.

#### Wizard Page Component Value Fields

Component	Description
-----------	-------------

<b>Text</b>	<p>Type the <b>Value</b> text that you want to appear in the wizard page, for example, text prompts for the user.</p> <p>(optional) For <a href="#">localization purposes</a>, enter a <b>Unique Key</b> that you want to assign to this text field.</p> <p><i>Tip: You may also want to use this component instead of field labels.</i></p>
<b>Block</b>	<p>Select the system or custom block that you want to appear in the selected wizard page.</p> <p>The contents of this drop-down list vary depending on the object you are creating the wizard for.</p> <p>For more details on blocks in TeamConnect, see <a href="#">Blocks</a>.</p>
<b>System Field</b>	<p>Complete the following steps to add a system field from the object for which you are creating the wizard:</p> <ol style="list-style-type: none"> <li>1. Type a <b>Label</b> for the system field, as you want it to appear in the wizard page. <p>The label does not have to be the same as in the system object views.</p> </li> <li>2. Select the appropriate attribute in the <b>Name</b> drop-down list.</li> <li>3. Enter a <b>Default Value</b> for the select attribute, if any. <p>The <b>Default Value</b> field varies depending on the attribute selected.</p> </li> </ol> <p>Labels for fields are optional because they may be self-explanatory or explained by the wizard prompts and questions.</p> <p>You may add only the system fields of the main object whose records are created through the wizard—not those of related objects.</p> <p>If you associate a template with the wizard and add a sub-object to it, you may select its main assignee or default category.</p>
<b>Custom Field</b>	<p>Complete the following steps to add a custom field from the object for which you are creating the wizard:</p> <ol style="list-style-type: none"> <li>1. Type the appropriate <b>Label</b> for the custom field as you want it to appear in the wizard page (optional). <p>The label does not have to be the same as in the custom object views.</p> </li> </ol>

	<ol style="list-style-type: none"> <li>2. Select the <b>Category</b> for which the custom field you want to add to the page was created.</li> <li>3. In the <b>Custom Field</b> drop-down select the name of the field you want to add to the selected wizard page.</li> <li>4. (Optional) Enter the appropriate <b>Default Value</b> for the selected custom field.  The <b>Default Value</b> field varies depending on the custom field selected.</li> </ol> <p>You may add the custom fields created for the object's root category only. However, if you associate a template with the wizard and add categories other than Root, then their respective custom fields are also available for you to add to the wizard. For details on adding a category to a template, see <a href="#">Adding Sub-objects to Templates</a>.</p>
<b>Parameter</b>	<p>Parameters allow you to add items that you may need to use as qualifiers for page transitions or to perform certain actions on the object attributes. Parameter data is not stored in the database and is only used within the wizard.</p> <p>Complete the following steps to add a parameter to the wizard page:</p> <ol style="list-style-type: none"> <li>1. Type the appropriate <b>Label</b> for the Parameter as you want it to appear in the wizard page.</li> <li>2. Select the <b>Type</b> of the Parameter you want to add to the page. For details on parameters, see <a href="#">Defining Parameters</a>.</li> <li>3. Type a unique name for the parameter you are creating.  This is the name that are required when you use the parameter in page transition rules or page actions.</li> <li>4. Check <b>Display in blocks only</b> if you intend to use parameters in a custom block. Otherwise, the parameter appears twice on the page, in the block and as an individual field.</li> <li>5. Select the Custom Lookup Table that will contain the list for this parameter. This applies to check-boxes, radio buttons, and drop-down lists that are used in custom blocks.</li> <li>6. (Optional) Enter the appropriate <b>Default Value</b> for the selected Parameter.  The <b>Default Value</b> field varies depending on the type of parameter selected.</li> </ol>

<b>System Field</b>  <b>Custom Field</b>  <b>Parameter</b>	of type <b>String</b> <b>(Text Fields)</b>	For system fields, custom fields, and parameters represented by a Text Field in the wizard page, there are two additional fields in the <b>Value</b> column: <ul style="list-style-type: none"> <li>• <b>Field Size</b></li> <li>• <b>Maximum Length</b></li> </ul> These two values allow you to specify how many characters the text field may display ( <b>Field Size</b> ) and how many characters it may contain ( <b>Maximum Length</b> ).
	of type <b>Long String</b> <b>(Memo Text Fields)</b>	For system fields, custom fields, and parameters that are represented by a Memo Text Field in the wizard page there are two additional fields in the <b>Value</b> column: <ul style="list-style-type: none"> <li>• <b>Columns</b></li> <li>• <b>Rows</b></li> </ul> These two values allow you to specify the size of the memo text box vertically ( <b>Rows</b> ) and horizontally ( <b>Columns</b> ).

#### 3.1.11.2.4.1 Defining Wizard Parameters

Parameters allow you to add items to be used as qualifiers for page transitions or to perform operations on the object attributes. Parameter data is not stored in the database and is only used within the wizard. Parameters may be simple, such as Number, String, or List, or use functionality that is stored in the database.

For each parameter that you add on the **Page Components** tab of the wizard definition, you may enter:

- A **Label** (required)
- The **Type** (required)
- A unique **Name** (required)
- **Custom Lookup Table** (required for check-boxes, radio buttons, and drop-down lists)
- The **Default Value** (optional)

The unique name is used in the following locations in the wizard screen:

- On the **Qualifier** tab of the page transition rules in the drop-down list for the **Left Argument**
- In the **Right Argument** if you select **Attribute**
- On the **Actions** tab in the **Parameter** drop-down list
- In the name attribute of the `<tc:wizardParameter name>` and `<tc:wizardParameterLabel name>` tags, when the parameter is included in a custom block. For details, see [Block Tags](#).

**Caution:** When you are including a parameter in a custom block that also contains custom fields, the parameter names and field names must be unique. Otherwise, errors may occur in page transition rules.

If you plan to include the parameter in a custom block, make sure to select the **Display in blocks only** check-box, or it appears twice on the wizard page.

## Database Parameters

Complex parameter values are stored in TeamConnect's database. The following parameter types used in wizard page components are mostly system lookup tables that appear in drop-down lists in the wizard:

- Activity Item
- Address Type
- Area Item
- Category
- Check-Box List
- Contact Relation Type
- Country Item
- Drop Down List
- Email Type
- Fax Type
- Group
- Internet Address Type
- Invoice Rejection Reason
- Integration Message Queue
- Phone Type
- Project Relation Type
- Radio Button List
- Resource Type
- Skill Type
- String
- Territory Type
- User

Depending on system and user preferences, these items may also appear in an applet. The **User** parameter type may appear in a search module.

If you select any of these parameter types, the **Default Value** field in the **Value** column on the **Page Components** tab appears a drop-down list with the corresponding lookup table items so you may select a default value. In the resulting wizard page, the selected parameter is represented by the same drop-down list, applet, or search module with its respective items.

The following parameter types used in wizard page components are used to create a corresponding search module field, where the user may select the necessary record:

- Account
- Contact
- Custom Object

## Transient Parameters

The following parameter types used in wizard page components have transient values. That is, they are not stored in the database and are created within the wizard itself:

- **Simple types**—Depending on the type of user input, select the appropriate parameter to create the corresponding type of field in the wizard page.
  - Boolean
  - Date
  - Date&Time
  - Decimal
  - Integer
  - Memo Text
  - (Text) String
- **List types**—Add list parameters to present the user with simple options on the wizard page, such as **Yes**, **No**, or **Unknown**.
  - Check-Box List
  - Drop Down List
  - Radio Button List

### To define simple list type parameters

On the **Page Components** tab of the wizard definition, select an option in the **Custom Lookup Table** drop-down list.

*Tip: Automated actions may populate a list parameter with values retrieved from records in the database. For example, you might display a list of outside counsel with certain rates and skills next to corresponding radio buttons.*

## Points To Remember

The following list summarizes wizard parameters:

- All parameters have transient values that exist only within the wizard in which they are defined.
- Parameter data is not stored in the TeamConnect database.
- Use its value to perform certain actions or as a qualifier for page transitions.
- Parameter values may be used in the wizard page for which you defined it and any of the subsequent pages in the wizard for actions or page transitions.

For example, if you add a parameter of type Integer to page 3 to obtain the number of days required to complete a task, on page 6, you may use its value to create an action to set the **Due On** date for the task.

- Parameters types are listed in the **Type** drop-down list in the **Value** column of the **Page Components** tab.
- Like custom fields, you may include wizard parameters in custom blocks to control their layout on the page.

You may do this by using the `<tc:wizardParameter>` and `<tc:wizardParameterLabel>` tags. For details, [Block Tags](#).

**Important:** If you intend to use a parameter in a custom block, make sure the **Display in blocks only** check-box is selected. Otherwise, the parameter appears twice on the page, in the block and as an individual field.

#### 3.1.11.2.5 Using Actions in Wizards

Wizard actions are operations that allow you to create appropriate behavior in a wizard, such as creating additional records, inserting values, accessing other records, or performing calculations to assign values to object attributes. Depending on your wizard design, applying actions to wizard pages is optional.

All wizard actions may be classified into the following groups based on:

- **When they are executed**
  - **Initial Actions**—Executed when a wizard is started by the user but before the first page opens and may not be defined as a simple action.
  - **Page Actions**—Executed when the user clicks **next** on wizard pages for which they are defined and may be defined as a simple or automated action.
- **How they are defined**
  - **Simple actions**—Defined through the user interface and are limited to a predefined set of operators
  - **Automated actions**—Defined in Java class or JavaScript files and may perform any operations on record values without limitation

The following table summarizes the relationships between the four types of wizard actions:

Wizard Action	Simple	Automated
---------------	--------	-----------



<b>Initial</b>		<b>X</b>
<b>Page</b>	<b>X</b>	<b>X</b>

**Important:** You may only use one initial action file per wizard and only one automated page action file per page. However, you may define several simple actions for a page.

To define an initial action for your wizard or to add an automated action to a page, you must first create each page, and upload the Java class or JavaScript file to the appropriate document folder for the object.

To define a simple page action, see [Defining Simple Actions](#).

#### 3.1.11.2.5.1 Applying Automated Actions

Automated actions may perform any operations on record values without limitation. In wizards, these actions include initial actions and page actions.

You must define automated actions in their respective Java class or JavaScript files and upload them to the appropriate directory in the Documents area. Otherwise, they are not available on the **Actions** tab of the wizard screen.

Although automated actions may perform simple operations, you may use them to create more complex logic that may not be defined through the user interface, such as:

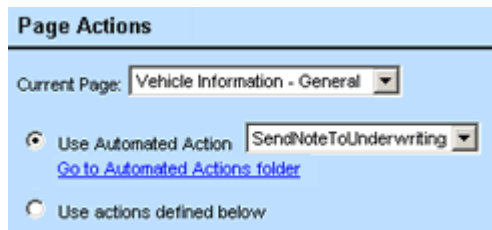
- Validating or verifying information entered by the user
- Opening existing records at the end of a wizard
- Inserting values in parameters of the type List.

**Note:** When you insert values in these parameters, you insert data gathered through the wizard into the appropriate screens of the object record. By default, duplicate contact checking is not performed when custom Java code creates or updates contacts. See the `ContactService.readDuplicateContacts()` method in the API Javadocs to use duplicate contact checking functionality.

#### To apply an automated action to a page in a wizard

1. Upload the Java class or JavaScript files to the **Automated Actions** folder.
2. Open the **Actions** tab in the appropriate wizard.
3. In the **Page Actions** section, select the wizard page you want to apply the action to in the **Current Page** drop-down list.
4. Select the **Use Automated Action** radio button.  
The **Use Automated Action** drop-down list opens.
5. From the **Use Automated Action** drop-down list, select the file with the action you want to be applied to the selected wizard page.

6. Click **Save**.



Automated Action Option on Wizard Actions Tab

## Points To Remember

- You may use only one automated action file per page.
- Automated actions are optional.
- The **Automated Actions** folder location in the Documents area is:

**Top Level/System/Object Definitions/objectName/Rules/Automated Actions/**

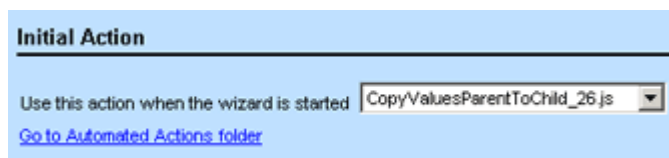
For information on how to create automated actions, see [Writing Automated Actions](#).

### 3.1.11.2.5.2 Specifying Initial Actions

Initial actions are used to run an initialization routine when a wizard is started by the user but before the first page of a wizard opens. For example, these actions may compute the values inserted from the template associated with the wizard.

#### To specify the initial action for a wizard

1. Make sure the necessary files are uploaded to the **Automated Actions** folder of the object definition.
2. Open the **Actions** tab in the appropriate wizard.
3. In the **Initial Action** section, from the **Use this action when the wizard is started** drop-down list, select the file with the action that you want to be executed.
4. Click **Save**.



Initial Action Section on Wizard Actions Tab

## Points To Remember

- Initial actions are optional and depends on your wizard design.
- Initial actions must be defined as automated actions.

- You may only define one initial action per wizard.

See also [Execution and Population Order](#).

#### 3.1.11.2.5.3 Defining Simple Actions

Simple actions are operations, such as calculations, string concatenation, and so on, that allow you to assign specific values to object attributes. These actions are always defined through the user interface and are limited to the number of the predefined operators available for each value type.

You may assign values to object attributes belonging to the object's related or child objects—provided you have defined the objects in a template and associated the template with the wizard. Once defined in the template, the sub-objects are listed in the **Object** drop-down list of the **Actions** tab.

You may assign the following types of values:

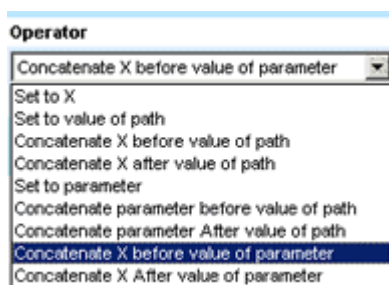
- Static values
- Values of wizard parameters
- Values of other attributes using a path in the object model

Use simple actions in a wizard when:

- You want to populate certain fields in the main record or its associated records with the values of parameters.
- When you want to overwrite the values assigned in a template with the ones specified on the **Actions** tab.
- You are not planning to use templates with your wizard, yet need to have certain fields of the main record prepopulated with default values.

**Important:** *Without a template you may not assign values to related objects or sub-objects.*

Simple actions are performed with the help of operators preselected for each value type. Depending on the attribute type, these operators may include the values shown in the following image and listed in [the Operator Options for Different Attribute Types table](#).



**Operator Options**

#### To define a simple action for a page in a wizard

1. Open the **Actions** tab in the appropriate wizard screen.

**Page Actions**

Current Page:

Use Automated Action  
 Use actions defined below

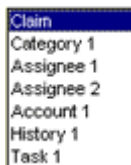
Order	Object	Attribute	Operator	X	Parameter Path
1	<input type="text" value="Task"/>	<input type="text" value="defaultCategory"/>	<input type="text" value="Set to X"/>	X: <input type="text" value="File Review"/>	
<a href="#">+ add more</a>					
Order	Object	Attribute	Operator	X	Parameter Path
1	<input type="checkbox"/> 1	Task	shortDescription	Set to X	30-Day Review
2	<input type="checkbox"/> 1	Task	dueOn	Add X days to value of path	30 .createdOn
3	<input type="checkbox"/> 1	Task	priorityID	Set to X	High
4	<input type="checkbox"/> <input type="text" value="0"/>	Task	defaultCategory	Set to X	<input type="text" value="File Review"/>

[Check All](#) - [Uncheck All](#)

#### Simple Actions in the Page Actions Section of the Actions Tab

- In the **Page Actions** section of the **Actions** tab, select the appropriate page from the **Current Page** drop-down list.
- Select the **Use actions defined below** radio button.  
The screen refreshes to display data entry fields necessary for defining a simple action.
- In the **Order** field, enter a number to indicate the order in which the action must be executed.
- From the **Object** drop-down list, select the appropriate object for which this action is being defined.

If you have associated a template with the wizard that defines child objects, these objects are listed in the **Object** drop-down list. They are numbered the same way as they are in their respective template screens, for example, Task 1, Task 2.



**Object  
Drop-down  
List**

- From the **Attribute** and **Operator** drop-down lists, select the appropriate attribute and the operator.

Depending on the attribute type, operators may include a set of operators listed in [the Operator Options for Different Attribute Types table](#), in addition to another set of the same operators using the value of a parameter. For example, instead of **Path** or **X**, the operators could be **Set to Parameter**, **Set to Value of Parameter**, **Subtract X from Value of Parameter**.

- Depending on the field value type of the selected attribute, enter the appropriate values for **X**, **Parameter**, or **Path**.


To learn more about setting a path, see [Paths](#).

- Click **add more**.
- Repeat steps 4 through 7 to add other actions to the selected page, if necessary, and click **Save**.

The following table describes the Action options on the **Actions** tab.

Simple Actions Options on Wizard Actions Tab

Field or button	Description
<b>Current Page</b>	Select the wizard page from which you want to define actions.  <i>Note: All pages in this field are listed in the order assigned to them on the <b>Pages</b> tab.</i>
<b>Use Automated Action</b>	Select this option to apply an automated action to the selected wizard page.  The <b>Page Actions</b> section refreshes to display a list of available automated actions created for the selected object (for example, <a href="#">the Automated Action Option on Wizard Actions Tab image</a> ).  For more details, see <a href="#">Applying Automated Actions</a> .
<b>Use actions defined below</b>	This is the default option. Select it to define your own simple actions.  The <b>Page Actions</b> section refreshes to display data entry fields where you may assign appropriate values to the selected fields through their respective attributes. For details, see <a href="#">Defining Simple Actions</a> .
<b>Order</b>	Enter an integer to indicate the order in which the actions must take effect in the wizard page.  <i>Tip: When defining the order, you may want to skip a few numbers before each action (for example, 3, 6, 9, 13, 18, 22, and so on) in case you may need to insert a few additional actions later, for example, if your design changes.</i>
<b>Object</b>	Select the object for which you want to define a simple action.  The objects displayed in this drop-down list include:

	<ul style="list-style-type: none"> <li>The object for which wizard is being created, for example, Claim.</li> <li>All sub-objects or related objects defined in the template, if you are using one with the wizard.</li> </ul> <p>These objects are numbered the same way as they are in their respective template screens, for example, Assignee 1, Assignee 2, Category 1, Task 1, Task 2, and so on.</p>
<b>Attribute</b>	Select the attribute of the field in the user interface to which you want to assign a value.
<b>Operator</b>	<p>Select the appropriate operator, depending on the operation you want to perform on the attribute value. For examples, see <a href="#">the Operator Options for Different Attribute Types table</a>.</p> <p><b>Note:</b> Along with the operators in <a href="#">the Operator Options for Different Attribute Types table</a>, there is a set of the same operators available. These operators use the value of a <b>Parameter</b> instead of <b>Path</b> or <b>X</b>, for example, <b>Set to Parameter</b>, <b>Set to Value of Parameter</b>, <b>Subtract X from Value of Parameter</b>, and so on.</p>
<b>Path</b>	<p>This field and the <b>reset</b> button appear when you select a dynamic or static-dynamic option from the <b>Operator</b> drop-down list. For examples of the operators, see <a href="#">the Operator Options for Different Attribute Types table</a>.</p> <p>Click the <b>Object Navigator</b>  icon next to the field and select the appropriate path in the displayed Object Navigator window. For details on how to use Object Navigator, see <a href="#">Using Object Navigator</a>.</p>
<b>reset</b>	Click to clear the <b>Path</b> field and start adding the path again.
<b>add more</b>	Click to add the new item to the list of page actions.
<b>Page Action List</b>	Displays all simple actions added to the selected wizard page. Select the check-boxes next to the actions that you want to delete or edit and click the appropriate command button at the bottom.

The following table lists possible options for different attribute types and examples of how they might be used.

**Operator Options for Different Attribute Types**

Attribute type	Operator	Value type	Description	Example
----------------	----------	------------	-------------	---------

<b>Lookup Table</b> <b>Check-Box</b> <b>Object</b> <b>IID</b>	Set to X	Static	Assigns the value entered or selected in the <b>X</b> field.	The <b>Main Assignee</b> for the claim is the assignee with the role <b>Adjuster</b> .
	Set to value of path	Dynamic	Assigns the value of the path specified in the <b>Path</b> field.  <i><b>Note:</b> For check-boxes, the value of path may be another check-box. For lookup tables, the value of path must be the same lookup table in another record.</i>	None
<b>Date</b>	Set to X	Static	Assigns the value entered in the <b>X</b> field.	The <b>Due On</b> date of a task is set to the date entered in the <b>X</b> field, for example, '07/01/02'.
	Set to value of path	Dynamic	Assigns the value of the path specified in the <b>Path</b> field.	A task is due the same day as the date the claim is created.  If the claim is created on 07/01/02, the <b>Due On</b> date for the task is set to 07/01/02.
	Add X days to value of path	Static-Dynamic	Adds the number of days entered in <b>X</b> to the value of the path.	The task '30-Day Review' is due 30 days after the date the claim is created.  If the claim is created on 07/01/02, the <b>Due On</b> date for the, '30-Day Review' task is set to 07/31/02.
	Subtract X days from value of path	Static-Dynamic	Subtracts the number of days entered in <b>X</b> from the value of the path.	The Progress Report for a claim must be printed 2 days before the, '30-Day Review' meeting.

				If the, '30-Day Review' meeting is scheduled for 07/31/02, the due date for the Progress Report is 01/29/01.
<b>Text (Strings)</b>	Set to X	Static	Assigns the value entered in the <b>X</b> field.	The value of the <b>Short Description</b> field for a task is set to the string entered in the <b>X</b> field, for example, '30-Day Review'.
	Set to value of path	Dynamic	Assigns the value of the path specified in the <b>Path</b> field.	The value of the <b>Name</b> field of the account is set to the name of the claim. For example, 'Allison McDonald Auto Accident'.
	Concatenate X before value of path	Static-Dynamic	Include the value entered in <b>X</b> before the value from the <b>Path</b> .	The value of the <b>Short Description</b> field for a task is set to the string '30-day review for' entered in the X field + the name of the claim. For example, '30- day review for Allison McDonald Auto Accident'.
	Concatenate X after value of path	Static-Dynamic	Include the value entered in <b>X</b> after the value from the <b>Path</b> .	The value of the <b>Short Description</b> field for a task is set to the name of the claim + the string '- 30-day review' entered in the <b>X</b> field. For example, 'Allison McDonald Auto Accident - 30- day review'.
<b>Number</b>	Set to X	Static	Assigns the value entered in the <b>X</b> field.	The value of the <b>Allocation Limit</b> field of an account is set to the number entered in the <b>X</b> field, for example, '1,000,000.00'.
	Set to value of path	Dynamic	Assigns the value of the path specified in the <b>Path</b> field.	The value of the <b>Allocation Limit</b> field of an account is equal to that of the <b>Allocation Limit</b> of its parent account.



				<p>If the Allocation Limit of the parent account is set to `250,000`, then the Allocation Limit of the account is set to `250,000`.</p>
Add X to value of path	Static-Dynamic	Adds the number entered in <b>X</b> and to the value of the path.	<p>The value of the <b>Allocation Limit</b> field of an account is the sum of `50,000` and the <b>Allocation Limit</b> of the parent account.</p> <p>If the Allocation Limit of the parent account is set to `250,000`, then the Allocation Limit of the account is set to `300,000`.</p>	
Subtract X from value of path	Static-Dynamic	Subtracts the number entered in <b>X</b> from the value of the path.	<p>The value of the <b>Allocation Limit</b> field of an account is the difference of `50,000` and the <b>Allocation Limit</b> of the parent account.</p> <p>If the Allocation Limit of the parent account is set to `250,000`, then the Allocation Limit of the account is set to `200,000`.</p>	
Multiply the value of path by X	Static-Dynamic	Multiplies the value of the path by the number entered in <b>X</b> .	<p>The value of the <b>Allocation Limit</b> field of an account is 25% of the <b>Allocation Limit</b> of the parent account.</p> <p>If the Allocation Limit of the parent account is set to `250,000`, then the Allocation Limit of the account is set to `62,500`.</p>	
Divide X by the value of path	Static-Dynamic	Divides the number entered in <b>X</b> by the value of the path.	None	

<b>Number</b>	Divide the value of path by X	Static-Dynamic	Divides the value of the path by the number entered in <b>X</b> .	<p>The value of the <b>Allocation Limit</b> field of an account is half of the Allocation Limit of the parent account.</p> <p>If the <b>Allocation Limit</b> of the parent account is set to `250,000`, then the Allocation Limit of the account is set to `125,000`.</p>
<b>User</b>	Set to X	Static	Assigns the value selected in the <b>X</b> field.	The value of the <b>Assignee</b> in a claim is set to the user selected in the <b>X</b> field, for example, Jane Smeeth.
	Set to value of path	Dynamic	Assigns the value of the path specified in the <b>Path</b> field.	<p>The value of the <b>Assignee</b> field in a claim is set to the user who created the claim.</p> <p>If Jane Smeeth is a claim associate who created the new claim, she is automatically set as the assignee for the claim.</p>
	Set to user with role X in project of path	Dynamic	Assigns the user with the specified role in another project.	<p>The value of the <b>Assignee</b> field in a claim is set to the user who is specified as the <b>Underwriter</b> for its parent policy.</p> <p>If Mathew Anderson is the Underwriter for the parent policy, he is automatically set as the assignee for the claim.</p>

#### 3.1.11.2.6 Defining Page Transition Logic

Page transition logic is the sequence in which the wizard pages follow each other. Depending on user input, this sequence may be linear (all wizard pages following each other), may skip pages, or may branch into alternative flows.

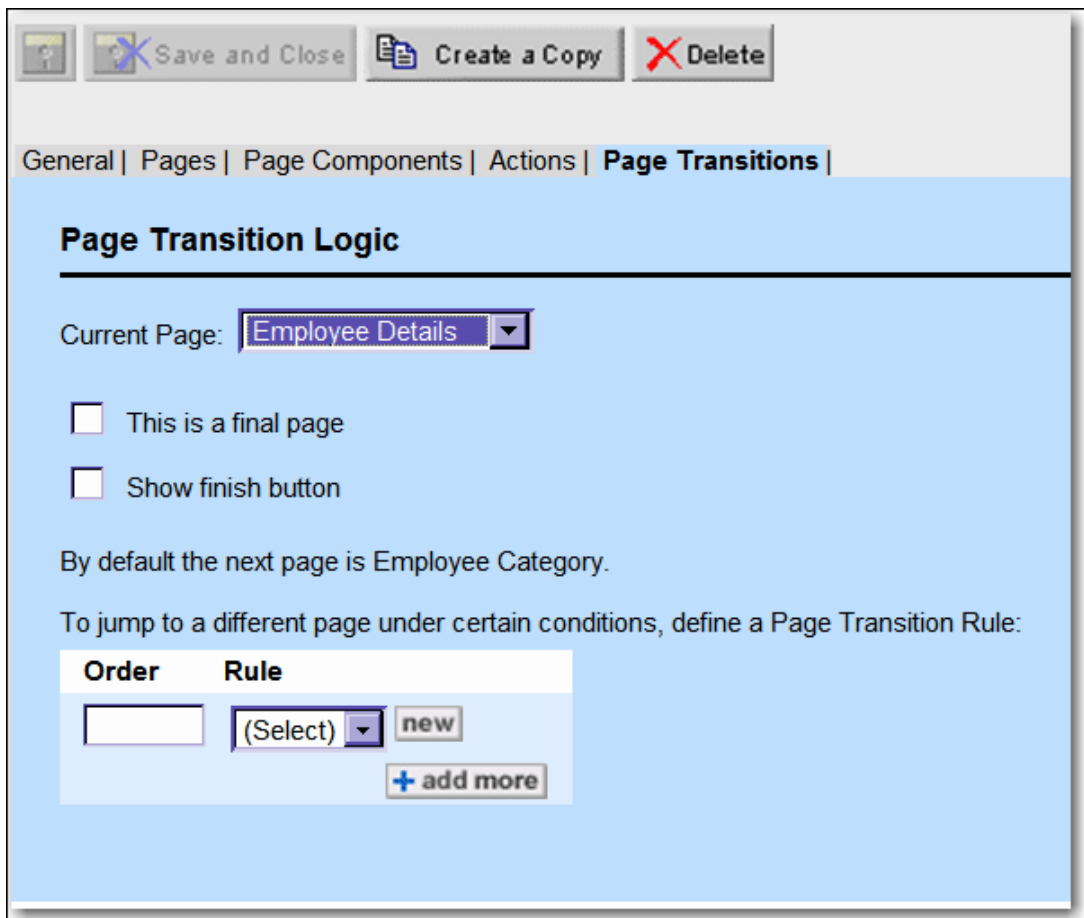
You can also allow users to end a wizard on any page by adding a **Finish** button. This allows users to choose where they want to stop the wizard, and then use the record's additional pages to add more information at a later time.

You must plan your page transition logic and create the wizard pages before specifying the necessary page transition rules. By default, the **Order** numbers you assign to each page defines the sequence of pages in a wizard.

### To define page transition logic

**Note:** This procedure assumes that the wizard is open to its tabbed pages.

1. Click the **Page Transitions** tab for the appropriate wizard.



The screenshot shows a software interface for configuring page transitions. At the top, there are four buttons: a help icon, 'Save and Close', 'Create a Copy', and 'Delete'. Below these is a tabbed menu with 'Page Transitions' selected. The main area is titled 'Page Transition Logic' and contains the following elements:

- 'Current Page:' followed by a dropdown menu showing 'Employee Details'.
- Two unchecked checkboxes: 'This is a final page' and 'Show finish button'.
- A text label: 'By default the next page is Employee Category.'
- A text label: 'To jump to a different page under certain conditions, define a Page Transition Rule:'
- A table with two columns: 'Order' and 'Rule'. The 'Order' column has an empty input field. The 'Rule' column has a dropdown menu showing '(Select)', a 'new' button, and an '+ add more' button.

Page Transitions Tab

2. From the **Current Page** drop-down list, select the page for which you want to define the transition logic.
3. Complete the following information:
  - If the selected page is the last page of the wizard, click the **This is a final page** check-box.

**Caution:** If you select the **This is a final page** check-box, the **Create the record on finish** check-box appears. If you do not check the **Create the record on finish** check-box, when you reach the end of the wizard, all of the information may be discarded.

- If you want to add a **Finish** button to the selected wizard page, click the **Show finish button** check-box.
4. In the **Order** field, enter a number to indicate the order in which the rule must take effect in the wizard page.
  5. Select one of the following options:
    - To select an existing rule, click the appropriate rule in the **Rule** drop-down list.  
You may view the details of a rule by selecting it and clicking **Open**, which displays the Rule screen where the qualifier and action are defined.
    - To create a new rule, see [Creating Page Transition Rules](#).  
The newly created rule is now displayed in the **Rule** drop-down list.
  6. Click **add more**.  
The selected rule is added to the list of rules defined for the page.
  7. Repeat steps 2 through 6 to add other rules to the page, if necessary, and then click **Save**.

#### ☐ To display another wizard after a user completes the first wizard

**Note:** You can only add another wizard to wizards for custom objects.

1. From the **Page Transition** tab of a wizard, place a checkmark in the **This is a final page** checkbox.  
The **Pick Wizard to launch on finish** options appear.
2. Select the **Object Definition** with the second wizard.
3. Select the wizard name from the **Wizard** drop-down list.

The following table describes the items on the **Page Transitions** tab.

**Page Transitions Tab**

Field or button	Description
<b>Current Page</b>	Select the wizard page for which you want to view or define page transitions.  <b>Note:</b> All pages in this field are listed in the order assigned to them on the <b>Pages</b> tab.

<b>This is a final page</b>	Select this check-box if the selected page is the last page that users will see, regardless of the flow that they select. The screen refreshes to display the <b>Create the record on finish</b> check-box and the <b>Post Wizard to launch on finish</b> options.
<b>Show finish button</b>	Select this check-box if you want to add a <b>Finish</b> button to the selected page of the wizard. You can add a <b>Finish</b> button to any page of the wizard.
<b>Create the record on finish</b>	Select this check-box if you want to create the record when the user clicks the <b>Finish</b> button. If you clear the check-box, all the information collected throughout the wizard may be discarded.
<b>Order</b>	Enter an integer to indicate the order in which the page transition rule takes effect in the selected wizard page.
<b>Rule</b>	Select the page transition rule you want to apply to the selected wizard page.
<b>new</b>	Click to create a new page transition rule for the selected page.
<b>add more</b>	Click to add the selected rule to the page transition rule list at the bottom of the screen.
<b>Pick Wizard to launch on finish</b>	<p>If you want another wizard to display after the user completes this wizard, select the following options:</p> <ul style="list-style-type: none"> <li>• <b>Object Definition</b>—The object with the wizard that you want to open.</li> <li>• <b>Wizard</b>—The wizard that you want to open after the user completes the first wizard.</li> </ul> <p>These options only appear when you select the <b>This is a final page</b> check-box.</p>

#### 3.1.11.2.6.1 Creating Page Transition Rules

Page transition rules allow you to specify alternative flows in response to user input and other qualifiers, so the wizard displays the appropriate pages to the user. Otherwise, all pages follow each other in the order specified on the wizard's **Pages** tab.

Page transition rules are similar to TeamConnect rules located on the **Rules** tab of an object definition screen. However, page transition rules are wizard-specific, so you may only create and access them from the **Page Transition** tab of the corresponding wizard screen.

Page transition rules have the following tabs:

- **General**—General information about the rule, including its name, whether it is active, and its trigger. The rule's **Type** field has the value **Wizard Page Transition**, which cannot be changed.
- **Qualifier**—Allows you to define the qualifier that must be met in order for the rule to take effect. Wizard parameters may serve as qualifier items.
- **Action**—Allows you define which page comes next, if a specified condition exists.

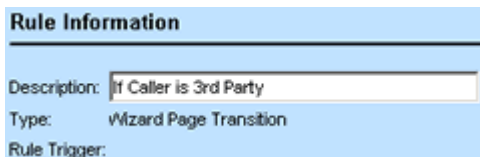
When the user clicks **next**, page transition rules display the appropriate page based on the specified conditions. Like validation rules, they may check the values entered in certain fields, check who the user is, and so on.

Wizard parameters that you add on the **Page Components** tab of the wizard screen become available on the **Qualifier** tab of the page transition rule screen. That way, user entries in the wizard may be used as qualifiers in the rule.

### To create a page transition rule

1. Open the **Page Transitions** tab of the appropriate wizard.
2. In the **Current Page** drop-down list select the wizard page for which you want to define the rule.
3. In the **rule** column click **New**.

The **General** tab of the wizard page transition rule screen opens.



Rule Information	
Description:	If Caller is 3rd Party
Type:	Wizard Page Transition
Rule Trigger:	

#### General Tab in Wizard Page Transition Rules

4. In the **Description** field, enter a self-explanatory name for the rule.
5. Click **Save** to allow you to add qualifiers.
6. Select the **Qualifier** tab.

Qualifier Tab in Wizard Page Transition Rules

7. Select the appropriate **Left Argument**, **Operator**, and **Right Argument** for the rule you are creating.

The **Left Argument** list includes all parameters that have been added to the wizard.

8. Click **add more**.
9. Select the **Action** tab.

Action Tab in Wizard Page Transition Rules

10. From the **Next Page When Condition Verified** field, select the page that is next if the qualifiers are met.
11. Click **Save**.

The rule is now available in the **Rule** list on the **Page Transitions** tab.

## Points To Remember

Keep in mind the following important information about page transition rules:

- Page transition rules are executed on a per-page basis—that is, when the user clicks **next** in the wizard, not on Create when the user clicks **finish**.
- The **Action** tab of a page transition rule allows you to specify the page to which the wizard must direct the user, if the conditions on the **Qualifier** tab is verified.
- When a user clicks **next** in a wizard page for which a rule is defined, page transition rules are executed on a per-page basis according to the **Order** numbers you assign to each rule.
- Page transition rules are wizard-specific and may be created and accessed only from the **Wizard** screen of the object for which the wizard is created, not from the object's **Rules** tab.

- To direct a user to a specific page depending on the rights of their user group, check the current user in the page transition rule. The **Current User** option is available from the **Left Argument** drop-down list on the **Qualifiers** tab.
- You may add wizard parameters as page transition qualifiers on the **Qualifier** tab. Select them from the **Left Argument** drop-down list. Parameters are also available when you select **Attribute** from the **Right Argument** list.



**Example: Left  
Argument  
Drop-down List**

## Page Transition Rule Example

Assume there are five pages in a wizard regarding view obstructions that a claimant and insured might have had. The user must provide the following information in each subsequent page depending on the answers:

**Page Transition Rule Example**

Page #	Wizard question	Field type	Wizard page component	Page transition
1	Did the Insured have any view obstructions?	Yes or No Radio buttons	Parameter1	If yes, go to Page 2. If no, go to Page 3.
2	Indicate the view Obstructions that the Insured had.	Check-Boxes	Block	Go to Page 3.
3	Did the Claimant have any view obstructions?	Yes or No Radio buttons	Parameter2	If yes, go to Page 4. If no and yes on Page 1, go to Page 5. If no and no on Page 1, go to Page 6.
4	Indicate the view Obstructions that the Claimant had.	Check-Boxes	Block	Go to Page 5.



5	Provide a summary of the view obstructions that the Insured and Claimant had.	Memo Text	Custom Field	Go to Page 6.
6	Did the Insured receive any citations?	Yes or No Radio buttons	Parameter3	

For example, to create a rule to skip to Page 6 from Page 3 when neither Insured nor Claimant had any view obstructions, you have to do the following actions:

1. On the **Qualifier** tab of the rule, you must specify the following conditions:
  - a. Select Parameter1 in the **Left Argument** drop-down list and set it equal to **No**.
  - b. Select Parameter2 in the **Left Argument** drop-down list and set it equal to **No**.
2. On the **Action** tab of the rule, you must select Page 6.

#### 3.1.11.2.6.2 Checking Page Order in Wizards

To determine the page order, alternate flows, and final pages in a wizard, refer to the **Page Transitions** tab of the corresponding wizard screen.

The **Page Transitions** tab does not list all page transitions at once. Instead, you select each wizard page in the **Current Page** drop-down list to determine which page follows it, based on the default page transition order or the selected page transition rules.

#### To check the page order in a wizard

1. Open the **Page Transitions** tab in the appropriate wizard.
2. From the **Current Page** drop-down list, select the wizard page for which you need to check the page order.

The screen either displays the page that follows the selected page, or informs you that the selected page is a final one, as shown in the following images.

**Page Transition Logic**

Current Page:

This is a final page

By default the next page is: 25 - Policy Information

To jump to a different page under certain conditions, define a Page Transition Rule.

Order	Rule	
1	<input type="text" value="(Select)"/>	<input type="button" value="open"/> <input type="button" value="new"/>
<a href="#">+ add more</a>		
Order	Rule	
1	<input type="checkbox"/> 1 <a href="#">If User is not Licensed Adjuster</a>	<input type="button" value="Next Page: End of Wizard"/>
2	<input type="checkbox"/> 2 <input type="text" value=""/> <a href="#">If Caller is 3rd Party</a>	<input type="button" value="Next Page: Accident Details - General"/> <input type="button" value="ok"/> <input type="button" value="cancel"/>

[Check All](#) - [Uncheck All](#)

---

Current Page:

**Page Transition Logic**

This is a final page

Create the record on finish

#### Page Transitions Tab Screens

- Repeat the previous step for all wizard pages you need to know the transition order for and write down the page transition sequence for each.
- Make any necessary changes and close the wizard screen when you are done.

You may use the list of page transitions to help you avoid breaking wizard flows when making changes to the page transitions.

### 3.1.11.3 Testing and Troubleshooting Wizards

After you have designed a wizard, you must thoroughly test it and resolve any problems that may arise.

#### To test a new wizard

- Make sure you have the rights to the object for which you have created the wizard.
- From the end-user interface, click **New**.



Select a Wizard Pop-up Window

3. Click the name of the wizard you are testing, in the pull-down menu.
4. Go from page to page selecting different options and checking where each option leads you.
5. Click **Finish**.
6. Repeat steps 4 and 5 for each of the alternative flows in your wizard, if applicable.
7. Make sure the wizard works as designed by checking:
  - The page order
  - Page transition rules' operation
  - The record state when you click **Finish** (record saved correctly)
  - Whether values are populated as assigned
  - Related objects specified in the linked template, making sure they are created and populated appropriately
  - Whether the wizard has correct layout, typographical errors, and field-and-block alignment
8. Make the necessary changes to fix any problems and test the wizard again.

The following table lists common problems and possible solutions to them.

#### Wizard Troubleshooting Tips

Possible Problems	Possible Solutions
The <b>New</b> button does not appear at all on the TeamConnect user interface.	<p>The <b>Require wizard for new object definition</b> check-box is selected, but there are no active wizards.</p> <p>Either clear this check-box or activate a wizard.</p> <p><b>Note:</b> <i>This does not apply to Contacts—a <b>New</b> button is always visible because it is also used to create Address Books.</i></p>
The wizard is not displayed in the New drop-down list.	Make sure the wizard is activated by selecting the <b>This wizard is active</b> check-box on the General

		tab of the wizard.
	I want users to only be able to create records by using a wizard, but when I click the <b>New</b> drop-down list, I still see the wizard option and the blank record option.	Select the Require wizard for new object definition check-box on the <b>General</b> tab for the wizard. See <a href="#">the General Tab on Wizard Screens table</a> for more information.
	The selected wizard cannot be started.	<ul style="list-style-type: none"> <li>• Make sure the correct automated action file is selected as the initial action for the wizard.</li> <li>• Ask a developer to check the code in the initial action file.</li> </ul>
	The wizard never ends.	Make sure the page transition rules are added properly, and no loops are created by switching the same pages over and over again.
Page order is not correct.	Pages do not follow each other in the order that was assigned to them on the <b>Pages</b> tab.	Make sure there are no page transition rules added to your wizard pages on the <b>Page Transitions</b> tab, because this action modifies your default order.
		Make sure there are no Final pages added in the middle of your default page order.
	Certain pages never display.	Make sure they have not been accidentally deleted from the wizard.
		Make sure the page transition rules are added correctly. All possible options for the user input are accounted for. No loops are created.
		Make sure there are no final pages added in the middle of your default page order.
	Make sure no pages with the added page transition rules have been deleted, thus breaking the sequence.	
Pages are not skipped as appropriate or appear when they are supposed to be skipped.	Make sure the qualifiers are set correctly in your page transition rules. All possible options for the user input are accounted for.	
Transition rules use the wrong values.	Make sure the parameter and custom field names in custom blocks are unique.	

	<p>The wizard goes to the final page without having gathered all the necessary information.</p>	<p>Make sure you use the appropriate user account to test your wizard.</p> <p>For example, if you are an Administrator, and the wizard is set to allow creating records only to the users with the role Claim Associate or Claim Adjuster, you are not able to proceed.</p>
<p>Records are not created.</p>	<p>The wizard does not allow you to save the record when clicking <b>Finish</b>.</p>	<p>Required fields may not have been populated or were not added to the wizard. For example, a parent record, or a custom field that are included in the object views.</p>
	<p>The wizard exited but the record was not created.</p>	<p>Make sure the <b>Create the record on finish</b> check-box is selected on the <b>Page Transitions</b> tab of the Final pages appropriately.</p>
	<p>The wizard was supposed to end without creating the record, but the record was created.</p>	
	<p>No related records were created when the wizard finished.</p>	<p>Make sure the appropriate template is linked to the wizard in its <b>General</b> tab.</p> <p>Make sure the appropriate related objects are added to the template linked to the wizard.</p>
<p>The assigned default values are not populated correctly in the saved record.</p>		<p>Make sure the correct attributes or paths were selected for the fields in the template or on the <b>Actions</b> tab of the wizard.</p>
		<p>Make sure the values assigned in the template are not overwritten on the Actions tab of the wizard, by the initial action, or manually when working with the wizard.</p>
<p>Layout and Formatting</p>	<p>The fields and their labels are not aligned.</p>	<p>Consider putting individual fields into a separate custom block.</p>
	<p>The fields and their values are not formatted appropriately, for example, to display time, dollar amounts, and so on.</p>	<p>Create custom blocks for the custom fields and add the appropriate formatting tag attributes.</p>

		<p><b>Tip:</b> You may add basic HTML formatting to the text components, such as <b>bold</b>, <i>italics</i>, <u>underline</u>.</p>
	The last page in the wizard does not have the finish button.	Make sure it is set as a final page on the <b>Page Transitions</b> tab.
	Duplicate parameter fields appear on the same page.	If you have included parameters in a custom block, make sure the <b>Display in blocks only</b> check-box (see <a href="#">the Select a Wizard Pop-up Window image</a> ) is selected on the page where the parameter is defined.
	Parameters do not appear on the page.	<p>If the parameters are not included in custom blocks, make sure the <b>Display in blocks only</b> check-box (see <a href="#">the Select a Wizard Pop-up Window image</a>) is not selected.</p> <p>If the parameters are included in custom blocks:</p> <ul style="list-style-type: none"> <li>• Make sure the block is added to the page where the parameters are defined or on any other page that follows.</li> <li>• Check whether the parameters are within a <code>&lt;tc:section&gt;</code> tag that contains custom fields in the custom block.</li> </ul> <p>If they are, either make sure the category to which the custom fields belong is added to the record that is being created through the wizard, or move the parameters outside a <code>&lt;tc:section&gt;</code> tag (see <a href="#">Sample XML File</a>).</p>
Custom Blocks	Blocks take long to load.	The first time you are using your wizard, it may take a little longer to load the custom blocks, because they need to be cached locally.
	Block does not appear on the wizard page. There is an error instead.	<p>The XML file may have problems. Make sure the custom block was created properly. For details, see:</p> <ul style="list-style-type: none"> <li>• Screen Designer Help</li> <li>• <a href="#">Troubleshooting Custom Blocks</a></li> </ul>
	Miscellaneous error messages may appear during the testing phase.	Make sure the fields used in actions or page transition rules are populated.

### 3.1.11.3.1 Execution and Population Order

To better understand how templates and wizards work and to be able to troubleshoot them, it is important to understand the order in which values are populated and actions are executed.

## Starting a Wizard

When a user starts a wizard the following actions happen, in this order:

1. All values from the associated template are automatically populated.  
The sub-objects and related objects specified in the template are created and the default values are assigned to the fields of the main record.
2. The automated action selected as the initial action for the wizard is executed.  
If there is no initial action specified, this step is skipped.
3. The parameters defined on the first page of the wizard are created.
4. The first page of the wizard opens with the latest values of all objects and parameters.
5. When the user clicks **next**, the available page actions and page transition rules associated with the page are executed.
6. The next wizard page opens with the updated values.

## 3.1.12 Using Rules

*Rules* allow you to automate your organization's business requirements within TeamConnect. Rules prevent records from being processed incorrectly or without proper approval by performing validations, security checks, and approval processes that control the flow of data. Rules eliminate the need for users to perform such checks and create a more reliable workflow.

TeamConnect contains many rules already, to coordinate workflow among and between its objects. As you add more types of objects and implement more capabilities, you may need to create additional rules.

Creating rules is one of the last steps in the customization process (See [Customization Sequence](#)). Before starting to create rules, make sure you have finished the following operations:

- Defining custom objects
- Adding necessary items to system lookup tables
- Defining object categories
- Creating custom fields and custom lookup tables
- Organizing custom fields by creating blocks
- Setting object views and search views
- Creating group accounts with the appropriate rights

### 3.1.12.1 Rule Components

Rule definitions include the following components:

- [Rule Types](#)—Select the appropriate rule type and save the rule before defining other components.
- [Rule Triggers](#)—The end-user action that triggers the rule, such as creating a record.
- [Rule Qualifiers](#)—The information that the rule checks before executing its action. A rule qualifier consists of a list of qualifier items and the logic used to evaluate those items.
- [Rule Actions](#)—The action that the rule performs when its qualifiers are met, such as denying the operation attempted by the user or sending it for approval. More complex actions, such as generating files when the rule's qualifiers are met, are considered custom actions and may be accomplished by writing rules using class files or JavaScript. The pre-population of fields may be defined through the use of templates.

### 3.1.12.2 Rule Types

You may create several different types of rules in TeamConnect:

- [Security Rules](#)—Protect records when they are in certain states. For example, when users open a matter record that is in a certain phase, no one may be able to modify it except for a certain group of users.
- [Pre-population Rules](#)—Populate fields with predefined values. For example, a filing date field may be populated when the **Lawsuit** category is added to a dispute record. You may define pre-population rules in conjunction with templates, which you may create through Designer without the need for class files or JavaScript.
- [Validation Rules](#)—Confirm the conditions that exist in a record at the time an action is attempted. For example, when a user clicks save, determine whether the appropriate custom fields are filled out for each category. If not, prompt the user with a message and do not save the changes.
- [Approval Rules](#)—Send requests for approval to specified users before an action is completed, such as posting an invoice. The requests then appear in each of those users' **My Approvals** screen.
- [Custom Action Rules](#)—Perform custom actions according to the needs of your organization. These actions are defined in class files or JavaScript or through the use of a template.
- [Scheduled Action Rules](#)—Custom action rules with parameters set through Designer to schedule when and how often an action is executed. Unlike custom action rules, scheduled actions do not support templates.
- [Audit Rules](#)—Capture and log events by creating history records that contain information about user activities in TeamConnect.
- [Post Commit Rules](#)—Perform a task after an action has completed successfully. For example, send an email notification to associated users after a record has been updated.

These types of rules may be categorized into the following groups based on the actions that they may perform:

- **Predefined actions**—Defined through Designer, security, pre-population, validation, approval, and audit rules have predefined actions such as the following:



- Deny an operation.
- Send the operation on a route for approval.
- Populate fields based on values in a template.
- Record user activities in history records.
- **Custom and scheduled actions**—Defined by creating class files or JavaScript and the TeamConnect API, custom action rules may perform any action written in Java or JavaScript.

Scheduled action rules use custom Java or JavaScript actions that you may schedule using certain parameters in Designer.

Custom action rules also support templates, whereas scheduled action rules do not. However, the predefined actions defined by a template are not custom actions.

By default, TeamConnect does not perform duplicate contact checking if custom code creates or updates contacts. See the `ContactService.readDuplicateContacts()` method in the API Javadocs to use duplicate contact checking functionality.

- **Page transitions**—Defined on wizard screens, you may create rules to govern how users enter information in a wizard. For more details about page transition rules, see [Creating Page Transition Rules](#).

#### 3.1.12.2.1 Security Rules

*Security rules* prevent an operation (such as creating, updating, or deleting) from being completed under specified circumstances and based on the original values of the record when it was opened by the user. When the user tries to alter a record, if the original values match the qualifier specified in the rule, the rule prevents the operation and displays a message to the user.

Keep the following points in mind when working with security rules:

- When a user attempts an operation, TeamConnect executes any applicable security rules before executing other types of rules.
- Security rules compare the rule qualifier to the conditions of the record when the user first opened it, so they prevent a user from changing values in the record to get around the rule. Qualifiers such as **isChanged**, **Item Deleted**, **Item Modified**, or **Item Added** will not have any effect on the rule's actions, since the rule is checked before any changes have occurred.
- Security rules do not provide field-level security. For example, you cannot specify that a certain field's value may not be changed. Security rules may only prevent the user from updating or deleting the entire record based on certain specific, pre-existing conditions in the record.
- You may use security rules to control the Delete and Update operations for existing records belonging to any object, or for the Phase Change operation in a custom object. For a complete list of operations that may trigger security rules, see [the Rule Triggers table](#).
- You may define security rules for specific user accounts and user groups. For example, if the current user is not a member of the Adjusters group, that user cannot update the record while it is in a specified phase.

#### 3.1.12.2.2 Pre-population Rules

Pre-population rules may populate fields, sub-objects, or related objects with values predefined in a template, which you may access or create on the **Templates** tab of the object definition. Once you create a template, you may select it from the **Action** tab of your pre-population rule screen.

**Note:** *In addition to pre-population rules, you may use templates with custom action rules and wizards.*

With templates, you may create rules that:

- Populate fields of existing records
- Create a new record and populate its fields
- Create a related record for the current object, and populate its fields
- Populate assignees and attendees and specify the main assignee by role
- Perform phase changes
- Populate the default category of a record
- Specify the default category with a static value

The available triggers and qualifiers for pre-population rules are the same as validation rules. For a complete list of triggering operations, see [the Rule Triggers table](#).

For information about defining templates for object definitions, see [Using Templates](#).

#### 3.1.12.2.3 Validation Rules

Based on the values in the record, *validation rules* prevent users from creating or updating the records, or changing project phases. If the qualifier in the rule is met, the rule prevents the operation and displays a message to the user.

### Points To Remember

- When a user attempts an operation, validation rules execute after all applicable security rules and pre-population rules, as described in the execution order listed in [Points to Remember](#).
- Validation rules compare their qualifiers to the values in the record at the time when the user attempts an operation, not when the record was opened.
- Validation rules may control the Create or Update operations for any object, or for the Phase Change operation in a custom object. See [Rule Triggers](#) for a complete set of triggers according to each object.
- Page Transition rules in wizards function like validation rules. They "validate" the information that the user enters in each wizard page to ensure that the data is entered properly. They also use this information to determine which wizard page to display next. For more information, see [Creating Wizards](#).

### Validation Rule Example

For example, you might write a validation rule for involved party records of the Dispute object, which checks:

- Whether the current user who is attempting to update a dispute record is a member of the Dispute Attorney group
- Whether the involved party record has the **Outside Counsel Firm** category
- Whether the **Involvement Date** field of the involved party record is empty (has a null value)

If all of these conditions are present, the rule prevents the record from being updated and displays a message to the user such as **Involvement date of the involved party cannot be empty**.

This is a validation rule rather than a security rule because the value in the **Involvement Date** field when the user opened the record is irrelevant. The purpose of the rule is to make sure that the **Involvement Date** field has a value when it is saved.

## Complementary Validation Rules

When writing a validation rule for the Create operation, make sure that you also write a validation rule for the Update operation that complements the Create rule, if necessary. This action ensures that the conditions you specify are met at all times.

For example, if you want to ensure that dispute records are always saved with at least one assignee, you must create two identical validation rules: one for Create and one for Update. If only the Create rule is used, a user may open an existing record and save (update) it after that user has deleted the assignee. To prevent this action, write a rule that requires an assignee when users update the record.

### 3.1.12.2.4 Approval Rules

Approval rules designate certain users who must approve an operation before it may be completed. For example, you may define approval rules to require permission to post invoices, deactivate accounts, change the phase of a matter. The [Rule Triggers table](#) lists the available operations for approval rules.

Use approval rules in conjunction with routes to define approval processes. A Route specifies a logical sequence of approvers used in an approval process. These approvers are included in routes through their user group memberships, since routes are based on groups, not individual users. For more details about routes, see [Creating Routes](#).

TeamConnect performs the following operations when an approval rule is triggered:

- When a user attempts a triggering operation, TeamConnect executes applicable approval rules after all security and validation rules for the operation are executed, as described in the execution order listed in [Points to Remember](#).

For example, if there is a security rule preventing users from deleting a record when it is in a certain phase, this rule is executed before the approval rule sends the deletion for approval.

- If the qualifier defined in the approval rule is met, the operation (also known as a request) is put into a pending status.
- The specified message appears at the top of the record, such as **This object has an action that is pending approval and it cannot be modified**.

- The approval request goes through the specified route of approvers. As each approver receives the request, it appears on his or her **My Approvals** screen.
- If the required users approve the operation, TeamConnect completes the requested operation.
- If the required users reject the operation, TeamConnect does not do the requested operation.
- During the approval process, decisions are listed on the **Workflow** tab of the record. If the needed approval is for deleting the record, the decisions cannot be accessed after the record is deleted.

You may create qualifiers for approval rules so that the user's attempt to perform an operation only goes for approval under certain conditions. For example, you may create an approval rule that requires approval for deleting a closed dispute record only if the deletion is attempted by a user in the Dispute Attorney group.

You may also specify an approval period for approval rules. You must use a value other than zero when specifying the length of an approval process. Otherwise, the approval period expires almost immediately.

If you do not set any qualifier information for an approval rule, the operation requires approval under all circumstances.

If an approval rule is triggered by the only user who is included in the route, the rule still requires that user's approval. The workflow status message immediately appears at the top of the record and the user must approve the operation before it is completed.

Many other actions and events may take place during [an approval process](#).

#### 3.1.12.2.5 Custom Action Rules

You may write custom action rules in Java or JavaScript and use Designer to upload the files to TeamConnect and select them for use in the rule. The qualifier for a custom action rule may be defined in Designer, or it may be a custom qualifier created in Java or JavaScript (an automated qualifier). For details, see [Writing Automated Actions](#).

You may define custom actions using a template, which you may access or create on the **Templates** tab of the object definition. You may select the template from the **Action** tab of your custom action rule.

**Note:** *By default, duplicate contact checking is not done when custom Java code creates or updates contacts. See the `ContactService.readDuplicateContacts()` method in the API Javadocs to use duplicate contact checking functionality.*

### Custom Action Rule Example

Custom action rules define actions that cannot be specified in Designer. For example, you might write a custom action rule in Java to automatically create budgets when a user adds an outside counsel firm to a dispute record. You upload the class file to the appropriate folder in TeamConnect's Documents area and then select it from the list of class files on the rule's **Action** tab.

### 3.1.12.2.6 Scheduled Action Rules

A *scheduled action rule* is similar to a custom action rule. You upload custom Java or JavaScript files to use for the rule action. The difference is that when you define these rules in the user interface, you have the option to specify when and how often the action is executed.

For example, your system might need a custom rule requiring, when the qualifiers are met, a task is created every 15 days to remind the user to do something, for a period of 90 days. To accomplish this requirement, a custom Java class might be uploaded to TeamConnect, and you then select this class file in the scheduled action rule. However, the actual number of days between tasks and the number of times that a task must be created are specified on the **Action** tab of the scheduled action rule.

When a scheduled action rule is triggered, you may view information about actions pending in the Scheduled Actions Monitor, provided that you have the rights to this tool. If necessary, you may also delete (or cancel) pending actions using the Monitor screen.

If the qualifying conditions for a scheduled action rule are true, and the rule is triggered, it will continue to run as scheduled even if the qualifying conditions subsequently become false during the schedule. The qualifiers are only checked the first time the scheduled action rule is invoked.

For more details about the Scheduled Actions Monitor and scheduled action rules, see:

- [Scheduled Actions Monitor](#)
- [Creating Automated Qualifiers and Actions](#)
- [Defining Settings for Scheduled Action Rules](#)

By default, duplicate contact checking is not done when custom Java code has been written to create or update contacts. See the `ContactService.readDuplicateContacts()` method in the API Javadocs to use duplicate contact checking functionality.

### 3.1.12.2.7 User Invoked Actions

You may define custom actions to be user invoked. Instead of the rule being triggered by an action like the creation or deletion of a record, the user intentionally invokes the action by selecting it from a drop-down list. The qualifier items are then checked and the action is executed if the qualifiers are met.

For example, you could define a user invoked custom action to trigger an email message to be sent containing information from the current record.

You may develop user invoked custom actions to create child records. To do so, you must specify whether the record (and the child record to be created) may be created when the user invoked rule is performed or when the user clicks **Save**. Either alternative is possible.

The drop-down list for user invoked actions appears to users on the toolbar, as a **More Actions** button.



Example: User Invoked Actions Drop-Down List

### To create a user invoked custom action rule

1. Define the custom action as a Java class or JavaScript file.
2. Define a custom action rule for the appropriate object definition and set the rule trigger to **User Invoke**.

When you define a user invoked action, the user invoked action becomes available on the **Rights** tab of user and group accounts for the corresponding object definition.

3. Give rights to all groups to whom you want to give access to the custom action.

The next time they log in, members of those groups have the action available to them on the record toolbar.

#### 3.1.12.2.8 Audit Rules

*Audit rules* may keep track of activity in TeamConnect by logging information in history records. You may apply these rules to user actions that are relevant for record keeping or monitoring changes, such as when a user updates one or more system or custom fields in a record. Audit rules provide you with a way to define business rules through Designer rather than developing and testing custom code to record user activity.

Audit rules may create history records that include the name of the user performing an action and how the record was affected. For example, you may create an audit rule to track when a contact record's **Social Security Number** field is updated. When the rule is triggered, it records the old value, new value, and the user who made the change.

Recording information in history records allows you to control what is recorded and how it appears. For example, you may create a custom description for the event that triggers the audit rule, record information in custom fields, and export data for reporting purposes.

### Points To Remember

Keep the following points in mind when creating audit rules:

- You may use audit rules to track modifications to data, and the creation or deletion of records.
- When an audit rule is triggered and its qualifiers are met, it creates one history record. To define audit rules that are triggered on Update, define qualifiers that only capture information when appropriate. Otherwise, a history record will be created whenever a user updates any part of a record.

- Rules with multiple triggers have the potential to execute more than once for a single event. If this happens with an audit rule, then multiple history records will be created.
- TeamConnect executes audit rules after all other business rules are applied. That way, operations that are prevented because of another rule will not cause the creation of an audit history record.
- The current user's access rights do not affect the creation of an audit history record. For example, even if the current user does not have rights to create related records in matters, the audit rule may still create a history record for the current matter record.
- If an audit rule is triggered but a history record cannot be created, for whatever reason, no additional information is logged, and no error is given to the user.

For information about the specific settings available in audit rules, see [Defining Actions for Audit Rules](#).

#### 3.1.12.2.8.1 Updates, Triggers, and Audit Rules

If you choose "use default description" in the Actions tab of an Audit rule, TeamConnect autopopulates the "Text" field in the audit history record. The way in which "Text" is autopopulated differs depending on the nature of the trigger.

For example, the text differs depending on whether an object is being created, updated, or deleted. Here is an example of the "Text" field in an audit history record that is triggered by an Update of a Project object:

**[When the action was completed the following information was changed in Matter 'test']**

- **Amount was '10,000' and was changed to '15,000'**
- **Transaction Date was '01/01/05' and was changed to '01/02/06'**
- **Quantity was '25' and was changed to '50'**

The default description for an update contains an entry for each field that changed during the update (three fields in this example). However, if you choose a trigger other than "Update", you may not get as much information.

For example, the Account object allows you to define an audit rule that uses the "Withdraw" trigger. The resulting audit history record tells you that a withdrawal took place, but does not specify the actual amount of the withdrawal, whereas an audit history record based on the "Update" trigger would show the change in amount.

You must decide how much detail is required and choose whether one trigger is sufficient or whether you need to associate both triggers with the rule. Here is a more complete syntax for how TeamConnect constructs the default description for an Update trigger:

```
[When the action was completed the following information was changed in <Record>
- <Field> was <Old Value> and was changed to <New Value> [For non memo text fields]
- <Field> was changed [For memo text fields]

[List of the changed, added or removed sub-objects and their associated events and
fields in separate lines. Note that if there is no change in sub-objects, nothing
goes here.]

[A new <Sub-object> was added]

-----
```

[An existing <Sub-object> was deleted]

-----  
 [An existing <Sub-object> was updated]

[List of all modified sub-object fields]

### 3.1.12.2.8.2 Audit Default Description

The default description (the value that TeamConnect puts into the "Text" field of an audit history record) is fairly detailed. In most cases, you do not need to override the default description with your own description.

In the case of Update triggers, the description includes changes for not only the fields of the main object, but also changes for some sub-objects. In addition to the sub-objects specifically named in the table below, every category assigned to an object is also considered a sub-object, for purposes of audit history logging.

#### Objects and their Sub-objects

Main Object	Sub-objects
Account	(none)
Appointment	Attendee Resource
Contact	Address Email Fax Internet Address Phone Rate Skill Territory
Document	(none)
Expense	(none)
History	(none)
Invoice	(none)
Task	Assignee



Custom Object	Assignee Embedded Custom Object
Involved	(none)
Embedded Custom Object	(none)

### 3.1.12.2.8.3 Configuring History Object Definition for Audit Tracking

If you want the information captured by audit rules to be in custom fields, you must define these fields under a particular category of the History object. Capturing details in separate fields makes it easier to export that information using Business Objects® reports.

**Note:** *If you are creating audit rules for the **User** or the **Group** object definition, you cannot define fields for History categories.*

To distinguish audit history records from other history records, create descriptive categories in the History object definition, such as the **Information Change - Lawsuit Key Dates** category, and then select the appropriate category while configuring audit rules. This technique is also useful if you decide to set security by the category rather than preventing user groups from accessing all history records.

The following table shows examples of History object categories and their custom fields that you may configure to be populated by audit rules.

**History Object Definition Category and Custom Field Examples**

Category name	Custom fields
Information Change - Lawsuit Key Dates	<ul style="list-style-type: none"> <li>• Old Trial Date</li> <li>• New Trial Date</li> </ul>
Information Change - Contact	<ul style="list-style-type: none"> <li>• Address</li> <li>• Address (old)</li> <li>• Contact Rate</li> <li>• Contact Rate (old)</li> <li>• Email Address</li> <li>• Email Address (old)</li> <li>• Fax Number</li> <li>• Fax Number (old)</li> <li>• Phone Number</li> </ul>

- Phone Number (old)

For details about defining categories, see [Using Categories](#). For details on defining custom fields, see [Custom Fields](#).

#### 3.1.12.2.9 Post Commit Rules

Unlike the other rule types, which are all executed before the user's actions are completed and committed to the database, post commit rules are only executed when it is confirmed that the triggering event has completed successfully. This prevents cases where, for example, a notification email might be sent out even if a transaction failed to commit for any reason.

The qualifier tab and triggers available to post commit rules are the same as those available for custom action rules.

Keep the following points in mind when creating post commit rules:

- Actions in post commit rules are executed outside of a database transaction. No updates to the current record are made within a post commit rule.
- The **Use template** option is not available for post commit rules. The available options are **Use automated Action** and **Notification** (when implemented).

#### 3.1.12.3 Rule Triggers

*Triggers* are operations that users perform in TeamConnect that affect specific records, such as creating tasks, activating accounts, posting invoices, voiding invoices, deleting contact records, and so on. Each type of rule may place restrictions on certain operations. You select the operation that triggers the rule on the **General** tab of the rule.

##### 3.1.12.3.1 Triggering Events

For a full list of which triggering events are available for each object type and rule type, see [the Rule Triggers table](#).

- **Activate**—The rule is triggered when an account is activated.
- **Assign**—The rule is triggered when a task is assigned.
- **Check In**—The rule is triggered when a document is checked in.
- **Check Out**—The rule is triggered when a document is checked out.
- **Create**—The rule is triggered when an object is created.
  - Rules for the Contact object that use the Create trigger apply when users are creating contacts through a search module. For example, if you make the **Date of Birth** field required, users will be unable to create a contact record through the search module because there is no **Date of Birth** field in the search module.
  - Rules that use the Create trigger for objects that have wizards are executed when the user reaches the end of the wizard and the system attempts to save the new record. If the rule requires that a field be populated, make sure that the user is able to enter a value and knows that the field is required.

- **Deactivate**—The rule is triggered when an account is deactivated.
- **Delete**—The rule is triggered when an object is deleted.
- **Deposit**—The rule is triggered when a budget amount is deposited into an account.
- **Phase Change**—The rule is triggered when a project's phase is changed. When the Phase Change trigger is selected, checkboxes and drop-down lists with further options appear.
  - **From**—The rule is triggered when a project's phase is changed from the selected phase in the drop-down list.
  - **To**—The rule is triggered when a project's phase is changed to the selected phase in the drop-down list.
  - **Transition**—The rule is triggered when the selected phase transition in the drop-down list occurs. This is the default option when the Phase Change trigger is selected.
- **Post**—The rule is triggered when an expense, invoice or task is posted.
- **Transfer From**—The rule is triggered when a budget amount is transferred from an account.
- **Transfer To**—The rule is triggered when a budget amount is transferred to an account.
- **Update**—The rule is triggered when an object is updated.
- **User Invoke**—The rule is triggered when a user with the appropriate rights selects the rule from the More Actions button in an object definition.
- **Void**—The rule is triggered when an expense, invoice or task is voided.
- **Withdraw**—The rule is triggered when a budget amount is withdrawn from an account.

## Points To Remember

Keep the following points in mind when creating rule triggers:

- Each rule type may be used with almost all triggers (see [the Rule Triggers table](#)), so base your choice of rule type on your business requirements, not the trigger.
- A rule can use more than one trigger. If more than one trigger is associated with a rule, it is possible that the rule will fire more than once for the same event. For example, the **Account** object custom action rule type has an available trigger named **Withdraw**, and it also has the **Update** trigger that is found in all object definitions. If you define a custom action rule that uses both of these triggers, and a Withdraw event occurs, the monetary amounts in the Account record will change, which causes an Update event. So the rule will execute twice, once as a result of the Withdraw trigger and once as a result of the Update trigger.
- Execution order—If an operation applies to multiple rules, the rules types are executed according to type. The following execution order is indicated on the **General** tab of a new rule that has not yet been saved:
  - a. Security
  - b. Pre-population
  - c. Validation
  - d. Approval

- e. Custom Action
  - f. Scheduled Action
  - g. Audit
  - h. Post Commit
- Each rule of the same type is executed according to the value of their **Order** fields. For example, after all security rules are executed, validation rules are executed, starting with the rule that has the lowest order. Rules of the same type that have the same values in their **Order** fields are executed in alphabetical order.

The following table lists the triggers that are available for each type of rule. A rule can use multiple triggers.

#### Rule Triggers

Object	Trigger	Security	Pre-population/ Validation	Approval	Custom/ Scheduled Action	Audit	Post Commit
All Objects	<b>Create</b> (Save, Save & Close, Save & Preview, or Save & New on a new record)		x		x	x	x
	<b>Update</b> (Save, Save & Close, Save & Preview, or Save & New on a new record)	x	x		x	x	x
	<b>Delete</b>	x		x	x	x	x
	<b>User Invoked</b>				x		x
Account	<b>Activate</b>	x	x	x	x	x	x
	<b>Deactivate</b>	x	x	x	x	x	x
	<b>Deposit</b>	x	x	x	x	x	x
	<b>Transfer From</b>	x	x	x	x	x	x

	Transfer To	x	x	x	x	x	x
	Withdraw	x	x	x	x	x	x
Docum ent	Check In	x	x		x		x
	Check Out	x	x	x	x		x
Expen se	Post	x	x	x	x	x	x
	Void	x	x	x	x	x	x
Group	Create		x		x	x	x
	Delete	x			x	x	x
	Update	x	x		x	x	x
	User Invoked				x		x
Invoic e	Post	x	x	x	x	x	x
	Void	x	x	x	x	x	x
Project	Phase Change	x	x	x	x	x	x
Task	Assign	x	x	x	x	x	x
	Post	x	x	x	x	x	x
	Void	x	x	x	x	x	x
User	Create		x		x	x	x
	Delete	x			x	x	x
	Update	x	x		x	x	x
	User Invoked				x		x

### 3.1.12.4 Creating Rules

This topic provides information about rule types, components, and settings to help you create and administer rules.

To create rules through the user interface, determine the following for each rule:

- The object records that the rule affects, such as accounts records.
- The actions the rule takes, such as require approval before a user may perform an operation.
- The conditions that must be met to trigger the rule.
- The specific fields within records that the rule uses.
- The users or groups the rule affects.

*Tip:* Consider whether [group rights](#) may accomplish your objectives without creating a rule.

#### 3.1.12.4.1 Defining General Rule Information

General rule information includes several settings, such as the name of the rule, the rule type, and what actions will trigger the rule. This information may be viewed, added, or modified on the **General** tab of the **Rules** screen.

Each rule has its own screen where it is defined. This screen contains the following tabs:

**General** | Qualifier | Action |

- **General**—[General information about the rule](#), including its name, whether it is active, its rule type, and its trigger(s).
- **Qualifier**—Allows you to define the qualifier that must be met in order for the rule to take effect. For details, see [Creating Rule Qualifiers](#).
- **Action**—For security and validation rules, this tab allows you to enter the message that appears to the user when an operation is denied. For details, see [Defining Messages for Security and Validation Rules](#).

For pre-population rules, the **Action** tab allows you to select the template used to add or populate records. For details, see [Using Templates](#).

For approval rules, the **Action** tab allows you to select the route and specify details about what happens when the route is in effect. For details, see [Defining Actions for Approval Rules](#) and [Creating Routes](#).

For scheduled action rules, the **Action** tab allows you to schedule the rule. For details, see [Defining Settings for Scheduled Action Rules](#).

#### 3.1.12.4.2 Accessing the Rules Screen

All rules that are defined within TeamConnect may be created, viewed, and modified on the **Rules** tab of the corresponding object definition.

**To access the screen for a new or existing rule**

1. Click the **Rules** tab of the object definition for which you want to create or access rules.  
A list of existing rules for the selected object appears, as shown in the following image.

The screenshot shows the 'Object Definition: Account' page in the TeamConnect Setup application. The page is titled 'Rules' and contains a table of existing rules. The table has four columns: Order, Type, Rule, Trigger(s), and Active?. Below the table, there are controls for creating a new rule and filtering the list.

Object Definition: Account

General | Categories | Custom Fields | Forms | Blocks | Object Views | Search Views | Custom Messages | **Rules** | Templates | Wizards | History |

### Rules

Create Rule: [new](#)

[Go to Rules folder](#)

Show  Rules on  action.

Order	Type	Rule	Trigger(s)	Active?
0	Security	<a href="#">Defined Update</a>	Update	No
0	Custom Action	<a href="#">Secondary Update</a>	Update	No
0	Custom Action	<a href="#">Activation actions</a>	Activate	Yes
0	Custom Action	<a href="#">Matter check</a>	Activate, Deactivate, Update	Yes
1	Validation	<a href="#">Balance check</a>	Deposit, Update, Withdraw	No

Object Definition Li... | Object Definition: A...

Rules Tab



2. If appropriate, sort the list of rules you want to view by either or both of the following criteria:
  - **By rule type**—Select the type from the **Show \_\_ Rules** drop-down list.
  - **By rule trigger**—Select the operation from the **on \_\_ action** drop-down list.

The rule list automatically refreshes to show you a list of matching results according to your selections.

3. Click the name of a rule to access it, or click **new** to create a rule.

The following table describes the items on the **Rules** tab.

**Rules Tab**

Field or column	Description
<b>new</b>	Click to create a new rule.
<b>Go to Rules folder</b>	Click to open the <b>Rules</b> folder for the selected object in the Documents area, where all files for custom action and scheduled action rules are stored.
<b>Show __ Rules</b>	Select a rule type by which to filter the list.
<b>on __ action.</b>	Select an action (trigger) by which to filter the list.  <i>Note: The contents of this drop-down list varies according to the rule type you have selected. For a list of rule triggers, see <a href="#">the Rule Triggers table</a>.</i>
<b>Rule list</b>	Displays all rules that have been created for the selected object that meet your view criteria.
<b>Order</b>	Displays the order in which rules are executed when a user attempts the corresponding action. Rules are executed in order from lowest to highest.  The order only pertains to rules of the same type. For example, all security rules are executed first according to order. After all security rules are executed, validation rules are executed according to order, and so on, according to the rule type.  For details on execution order, see <a href="#">Points To Remember</a> .
<b>Rule</b>	Displays the name of the rule as a hyperlink, which you may click to view the rule definition.

<b>Trigger(s)</b>	Displays the names of the triggers that are currently associated with the rule. For a list of rule triggers, see <a href="#">the Rule Triggers table</a> .
<b>Active?</b>	Displays whether the rule is currently active in TeamConnect.

#### 3.1.12.4.2.1 Setting General Rule Information

Before you may define a rule's actions and qualifiers, you must complete the fields on the **General** tab and save the rule.

##### To set the general rule information

1. Click the **Rules** tab of the appropriate object definition.
2. Click **new** on the **Rules** screen to open a new rule.

The **General** tab of a new **Rule** screen appears.

The screenshot shows a software interface for configuring an account rule. At the top, there is a navigation bar with buttons for 'Create a new...', 'Go to...', 'Tools...', and 'System Se...'. Below this is a window title bar for 'Account Rule: Custom ActionCA' with standard window controls. The main area has tabs for 'General', 'Qualifier', and 'Action', with 'General' selected. The 'Rule Information' section contains the following fields and options:

- This Rule is Active
- \* Description: Custom ActionCA
- \* Unique Key: CustomActionCA
- \* Order: 0
- Type: Custom Action
- \* Triggering Event:
  - Activate
  - Create
  - Deactivate
  - Delete
  - Deposit
  - Transfer From
  - Transfer To
  - Update
  - User Invoke
  - Withdraw

\* Required fields are indicated by an asterisk.

At the bottom, a taskbar shows three open windows: 'Object Definition Li...', 'Object Definition: A...', and 'Account Rule: Custom...'.

General Tab on Rule Screens

3. Enter the necessary information as described in [the General Tab on Rule Screen table](#).

4. Make sure that **This rule is Active** is not selected until you are ready to test the rule.  
Before activating the rule, you must set the necessary information on the **Qualifiers** and **Action** tabs.
5. Click **Save**.

The following table describes the items on the **General** tab of the **Rule** screen.

**General Tab on Rule Screen**

Field	Description
<b>This Rule is Active</b>	<p>Select to activate the rule after you have finished developing it and you are ready to test it or make it available for users.</p> <p><i><b>Important:</b> When designing rules, you must always deactivate rules you have not completed or are not using. Having active rules may hinder your ability to design and test other rules.</i></p>
<b>Name</b>	<p>Enter a name that clearly and uniquely identifies the rule. This field cannot exceed 250 characters.</p> <p><i><b>Tip:</b> Avoid similar descriptions for different rules. Emphasize important information, such as the rule trigger, in the description such as the following example:</i></p> <p style="text-align: center;">When a dispute is closed it cannot be deleted</p> <p>If this rule is an approval rule, this description appears on the <b>Workflow</b> tab of a record that is pending approval for an operation. This helps users to understand why the operation is being sent for approval.</p>
<b>Unique Key</b>	<p>Enter an alphanumeric combination to uniquely identify the rule within the object definition. After you save the rule, the unique code appears as read-only text and you cannot change it.</p> <p>The unique key cannot contain spaces, underscores, commas, punctuation marks, or any special characters and must not exceed 50 characters in length.</p> <p>Although rules within the same object definition cannot have the same unique key, rules belonging to different object definitions may have the same key.</p>
<b>Order</b>	<p>Enter an integer to indicate the order in which the rule must take effect in relation to other rules that are created for the selected object.</p> <p>Organize your rules in the order of their importance in your system, with the most important rule being executed first.</p>

<b>Description</b>	Enter a detailed description of the rule. This field cannot exceed 2000 characters.
<b>Type</b>	Select the radio button next to the type of rule you want to create. For an explanation of each type of rule, see <a href="#">Rule Types</a> .  <b>Important:</b> After the rule is saved, you cannot change this selection.
<b>Triggering Event</b>	Select the events, or operations attempted by the user, that you want to trigger the rule. Multiple events can be selected.  <b>Note:</b> The checkboxes available vary according to the rule type you select. For details, see <a href="#">Rule Triggers</a> .

#### 3.1.12.4.3 Rule Qualifiers

A *qualifier* for a rule may be defined either in a file that is uploaded to TeamConnect, or directly through the user interface. This section describes qualifiers defined in the user interface. For details about files that are uploaded (custom code), see [Writing Automated Qualifiers for Rules](#).

A qualifier defined through the user interface consists of two parts:

- A list of qualifier items, which identify data values from fields in TeamConnect, that are to be checked by the rule.
- Logic for determining whether the entire qualifier is met: all of the items must be met, any of them must be met, or a more complex logic that you define.

When the rule is triggered, if the qualifier as a whole is met, then the rule executes its action. If the qualifier is not met as you have defined it, the action is not executed.

It is helpful to use this sentence as a model for creating a validation or security rule and defining its qualifier items through the user interface:

**"If the entire qualifier is met, do not allow the user to [update, create, delete] the record."**

Following the construction of this sentence when creating qualifiers will help you write them accurately.

**Important:** If an active rule has no qualifier information defined, the rule action is always executed when the rule's triggering events occur, whether by a user, wizard, or another rule.

## 3.1.12.4.3.1 Qualifier Tab Information

You may view, create, modify, or delete the qualifier information for a rule on the **Qualifier** tab of the respective **Rules** screen of an object definition ().

**Rule Qualifier**

Use Automated Qualifier file  
 Use Qualifier defined below

Left Argument	Operator	Right Argument
Current Object   IsClosed [reset]	Is   Equal To	Literal   No
1 <input type="checkbox"/> Current Object .detailList(Motions:Motion to Amend)	Exists	
2 <input type="checkbox"/> Current Object .detailList(Motions:Motion for Sanctions)	Exists	
3 <input type="checkbox"/> Current Object .detailList(Motions:Request for Admissions)	Exists	
4 <input type="checkbox"/> Current Object .detailList(Motions).detailObjValueList(InherentlyDangerousBusiness).detailValue	Is Equal To	(Yes)
5 <input type="checkbox"/> Current Object .detailList(Motions).detailObjValueList(PunitiveDamages).detailValue	Is Equal To	(Yes)

[Check All](#) - [Uncheck All](#) [edit](#) [delete](#)

**Run the rule action when:**

All of the above conditions are met (AND logic)  
 Any of the above conditions are met (OR logic)  
 The following combination of the above conditions is met (1 or 2 or 3) and (4 or 5) [View Qualifier](#)


Use condition numbers in [formula](#), for example (1 and 2) or (3 and 4)

**Qualifier Tab on Rule Screens**

The following table describes the items in the **Qualifier** tab in **Rule** screens.

**Qualifier Tab on Rule Screens**

Field	Description
<b>Use Automated Qualifier file</b>	Select this option to use a Java class or JavaScript file as the rule's qualifier. When this option is selected, the screen displays a drop-down list labeled <b>Use this Automated Qualifier</b> , where you may <a href="#">select the appropriate Java or JavaScript rule qualifier</a> . For details about automated qualifiers, see <a href="#">Writing Automated Qualifiers for Rules</a> .
<b>Use Qualifier defined below</b>	Select this option if you want to use the user interface to define the qualifier conditions and logic.
<b>Left Argument</b>	Allows you to create a path to the attribute that you want to use in the qualifier condition.

	<p>Select whether you want to create a path to an attribute that is associated with the <b>Current Object</b> or the <b>Current User</b>:</p> <ul style="list-style-type: none"> <li>• Select <b>Current Object</b> if you want to use a field in TeamConnect as a qualifier (for example, the Current Phase of the record that the user is attempting to update).</li> <li>• Select <b>Current User</b> if you want the rule to check the identity of the current user working with a record (for example, to verify that the user who is attempting the operation is a member of a specified group).</li> </ul> <p>Click the <b>Navigator</b>  icon to open Object Navigator, the tool for creating paths in TeamConnect, and edit the path displayed in the <b>Attribute</b> field, if appropriate. To create a path, you must understand the object model as described in <a href="#">Object Model: Read This First</a> and in the additional tables it points to.</p>
<b>reset</b>	Clears the path that appears in the Object Navigator field. This button appears after an option has been selected from the drop-down list in the Left Argument.
<b>Operator</b>	Select the appropriate operator settings for the qualifier condition. For details, see <a href="#">Available Operators in Qualifier Items</a> .
<b>Right Argument</b>	Select the attribute, literal value, or other setting that you want to compare to the attribute that you selected for the left argument. For details, see <a href="#">Right Argument Options in Qualifier Items</a> .
<b>Qualifier Item List</b>	<p>Displays a list of qualifier items that have been added to the rule. These are the conditions used by the rule, in combination with the logic that you specify, to determine whether to prevent the operation that is being attempted, or whether the rule's action must be run.</p> <p>Select or clear the corresponding check-boxes to <b>edit</b> or <b>delete</b> qualifier items.</p>
<b>Run the rule action when</b>	Select how you want the defined qualifier items to be evaluated by the rule. For details, see <a href="#">Qualifier Logic</a> .

#### 3.1.12.4.3.2 Qualifier Items

To define a qualifier item in Designer, you specify all of the following:

- The left argument
- The operator
- The right argument

These three pieces together form a statement, such as these:

## Sample Qualifier Statements

Left argument	Operator	Right argument
The value selected in the Loss Type list field	is	Fire Damage
The current user	is not	a member of the Outside Counsel group
The Trial Date	is changed	from any date to any date
The Expiration Date field	is populated	[none]

If you add these statements as qualifier items, or conditions, the rule will evaluate the statements to determine whether the rule's action must be executed when a user triggers the rule.

The type of data being compared in the left and right arguments must match. For example, if your left argument identifies a list field, then your operator and right argument options allow you to compare the selection with a selection made in another list field or to a literal list value.

The **Designer** user interface is designed to deny illogical combinations of left arguments, operators, and right arguments, such as: "The Trial Date is the Outside Counsel group."

**Tip:** *If you find a rule executing in a situation where you did not expect it to, carefully check how you have constructed your qualifiers. In some cases, when the left qualifier evaluates to an unknown value and the right qualifier also evaluates to an unknown value, the rule engine will consider this a match. For example, consider an Audit rule triggered by Create that affects Involved records, and has a qualifier with these characteristics:*

```
Left qualifier = inactiveDate
Operation = X Days from Today
Right qualifer = (no value)
```

*In this case, when you create a new Involved record that is active, the rule will execute. That's because the value of inactiveDate is null, since the record has never been inactivated. And because the right qualifier was specified incorrectly (an integer should have been entered, but wasn't), the right qualifier's value is also null. Thus the rule engine considers the two nulls a match, and the rule executes. You should fix the incorrect specification of the right qualifier. Also, to guard against such inadvertent errors, you could add a second qualifier, testing the isPopulated condition for the inactiveDate field, so that the rule would only execute when that field was actually populated.*

For details about the options available when constructing qualifier items for each type of data, see:

- [Available Operators in Qualifier Items](#)
- [Right Argument Options in Qualifier Items](#)



The operator in a rule qualifier statement determines how the left and right arguments are compared. It may have several components, depending on the type of data value you are working with in the qualifier (text, lookup table value, date, and so on).

The following table describes TeamConnect operators and the corresponding types of data that may be compared.

#### Rule Qualifier Operator Options

Selection	Description	Data types where operator selection is available					
		Text	Number	Date	List	Check-box	Related object
<b>Is</b>	Determines whether the qualifier statement is a positive or negative statement.	x	x	x	x	x	x
<b>Not</b>		x	x	x	x	x	x
<b>Equal To</b>	Evaluates whether the left argument and right argument are the same value.	x	x	x	x	x	x
<b>Populated</b>	Evaluates whether the field identified in the left argument has a value.  Do not use this selection to check whether a field that was previously null now has a value. Instead, use Changed from No Value.	x	x	x	x		x
<b>Changed</b>	Evaluates whether the value for the field identified in the left argument has changed. It compares the old value in the database to the value in the record when rule is triggered.  <i>Tip: This option is useful for triggering audit rules.</i>	x	x	x	x	x	x
<b>Begins With</b>	Evaluates whether the text value in the field identified in the left argument begins with, ends with, or contains	x					

<b>Ends With</b>	the text that is identified in the right argument.	x					
<b>Contains</b>		x					
<b>Less Than</b>	Evaluates the number value identified in the left argument compared to the number value in the right argument.		x				
<b>Less Than Or Equal To</b>	For example, you may specify that if a dollar amount <b>Is - Greater Than Or Equal To</b> 5000, send the action for approval.		x				
<b>Greater Than Or Equal To</b>			x				
<b>Greater Than</b>			x				
<b>Exists</b>	These options appear when you select a particular category by creating a path such as <code>.detailList.(Motions)</code> . Evaluates whether the category has been added to the record.						
<b>Does Not Exist</b>							
<b>Item Added</b>	Evaluates whether a related object (such as involved) or sub-object (such as assignee) was added, deleted, or modified.				x		
<b>Item Deleted</b>					x		
<b>Item Modified</b>					x		
<b>After</b>	Compares the date in the left argument to the date in the right argument.  An additional field appears when you select one of the options that requires you to specify a number of days. Enter the value for <b>X</b> in this			x			
<b>Before</b>				x			
<b>After The Next X Days</b>				x			

<b>Within The Next X Days</b>				<b>x</b>			
<b>Within The Last X Days</b>				<b>x</b>			
<b>Before The Last X Days</b>				<b>x</b>			

Depending on the data type of the qualifier item that you are adding to your rule qualifier, the selections that are possible for the right argument are automatically provided to you.

The following table describes the possible selections for the right argument of a rule qualifier.

#### Rule Qualifier Right Argument Options

Selection	Description	Data types for which selection is available					
		Text	Number	Date	List	Check-box	Custom Object/Involved
<b>Literal</b>	<p>Select this option when you want to specify a value to compare to the field identified by your left argument.</p> <p>For example, select <b>Literal</b> for a qualifier item that checks whether the invoice total is greater than 5000.</p> <p>You must also select <b>Literal</b> when you want to check whether a certain item is selected in a List field.</p>	<b>x</b>	<b>x</b>		<b>x</b>	<b>x</b>	<b>x</b>
<b>Attribute</b>	<p>Select this option when you need to compare the values of two fields in TeamConnect.</p> <p>After you select <b>Attribute</b>, you must first select</p>	<b>x</b>	<b>x</b>	<b>x</b>	<b>x</b>	<b>x</b>	<b>x</b>

	<p><b>Current Object</b> or <b>Current User</b>, and then use Object Navigator to identify the field to which you want to compare the field identified by your left argument.</p> <p>For example, if you want to compare the Contact of a contact-centric project to the Contact identified in a custom field of type Involved, select <b>Attribute</b> to identify the custom field.</p>						
<b>Current Object</b>	Select one of these options after you have selected <b>Attribute</b> and you need to identify a field in TeamConnect to which you are comparing the field in your left argument.	x	x	x	x	x	x
<b>Current User</b>	For details about using Object Navigator, see <a href="#">Using Object Navigator</a> .	x	x	x	x	x	x
<b>Any Value</b>	These options work in conjunction with the <b>Changed</b> option to form a statement, such as <b>Is changed from any value to this value (California)</b> .	x	x	x	x	x	x
<b>No Value</b>		x	x	x	x		x
<b>This Value</b>		x	x		x		x
<b>Today</b>	The date that the rule is being triggered.			x			
<b>This Date</b>	Allows you to compare the date in the left argument to a date that is identified in another field.			x			
<b>From Now</b>	Allows you to specify a number of days from the date that the rule is being triggered.			x			

<b>From This Date</b>	Allows you to specify a number of days from a date that is identified in another field.			<b>x</b>			
<b>Yes</b>	These options correspond to a check-box being selected or not selected.					<b>x</b>	
<b>No</b>						<b>x</b>	

### 3.1.12.4.3.3 Qualifier Logic

The **Designer** user interface is designed for you to complete the qualifier logic in stages. After listing one or more qualifier items on the **Qualifier** tab, the following options appear, allowing you to specify when the rule action runs:

- All of the conditions are met.
- Any of the conditions are met.
- The conditions are met according to the logic that you specify.

The "logic" option allows you to specify a formula for how the rule evaluates the conditions. You refer to a qualifier condition using the number that is automatically generated on the batch screen for each item (see ). For example, once you create a list of five qualifier conditions, you might create the following formula:

```
(1 or 2 or 3) and (4 or 5)
```

### Points To Remember

- Keep the following points in mind while working with qualifier logic:
- Define all of the qualifier conditions that you need in your rule qualifier before writing the formula.
- The only words that you may use in your formula are "and" and "or."
- Use parentheses ( ) to logically group conditions. You may also nest parentheses when needed. Make sure to use complete pairs of parentheses.
- If you do not group the conditions where grouping is necessary, grouping is automatically applied by TeamConnect, and you will have to verify that the appropriate logic is achieved. For example:

```
1 or 2 and 3
```

is automatically converted to:

```
1 or (2 and 3)
```

It is best to apply the grouping yourself to ensure that it is correct.

- If necessary, your formula may repeat the same qualifier condition. For example, the following formula is valid:

```
(1 and 2 and 3) or (1 and 2 and 4)
```

## Viewing Completed Qualifier

Once you have added all of the necessary qualifier conditions and have written the formula that references those conditions, you may view the complete qualifier expression by clicking **View Qualifier**.

If you modify your qualifier conditions or formula, you may refresh the display of your qualifier by clicking **Save**, or by clicking **View Qualifier** again. The following figure shows an example of qualifier logic:

**Run the rule action when:**

All of the above conditions are met (AND logic)  
 Any of the above conditions are met (OR logic)  
 The following combination of the above conditions is met (1 or 2 or 3) and ( 4 or 5 ) Hide Qualifier

Use condition numbers in [formula](#), for example (1 and 2) or (3 and 4)

**Qualifier**

```

AND
├── OR
│   ├── -1- Current Object .detailList(MATR_MOTI_MOAM) Exists
│   ├── -2- Current Object .detailList(MATR_MOTI_MOSA) Exists
│   └── -3- Current Object .detailList(MATR_MOTI_READ) Exists
└── OR
    ├── -4- Current Object .detailList(MATR_MOTI).detailObjValueList(InherentlyDangerousBusiness).detailValue Equal To Yes
    └── -5- Current Object .detailList(MATR_MOTI).detailObjValueList(PunitiveDamages).detailValue Equal To Yes
  
```

Qualifier Logic Options

The following table describes the items in the qualifier logic section.

Qualifier Logic Options

Field	Description
<b>Run the rule action when</b>	Specify when the rule action must be run based on the qualifiers. You may specify AND or OR logic, or a formula using the qualifiers. See <a href="#">Qualifier Logic</a> .
<b>View Qualifier/Hide Qualifier</b>	Click to view or hide the qualifier logic display.
<b>formula</b>	Click this hyperlink to view additional help on how to write the formula for a rule qualifier.
<b>Qualifier</b>	Displays a visual representation of your entire rule qualifier according to the formula that you have provided.  You cannot edit the qualifier in this area. It is provided to help you determine whether the formula you have written is correct.

#### 3.1.12.4.3.4 Creating Rule Qualifiers

Creating rule qualifiers requires an understand of the following items:

- **Rule components**—Creating appropriate qualifiers is only possible when you understand how rules function, including all of the following:
  - [Rule Types](#)
  - [Rule Triggers](#)
  - [Qualifier Items](#)
  - [Qualifier Logic](#)
- **TeamConnect object model**—The object model is described in detail in [Object Model: Read This First](#) and the additional reference tables it points to.
- **Creating paths**—You need to understand how to create paths to attributes using Object Navigator. For details, see [Using Object Navigator](#).

#### To create a qualifier for a rule

1. Plan your rule so that you know which attributes and conditions you need to identify in TeamConnect.
2. Open the **Qualifier** tab of the appropriate rule.

***Important:** Before you may create the qualifier, you must save the rule with the correct information on the **General** tab.*

3. Create the qualifier conditions as described in [Qualifier Items](#). You may refer to [Qualifier Item Examples](#) for examples that may be used as models for your rules.
4. Select whether you want to run the rule when:
  - **All of the above qualifiers are met (AND logic).** (This is the default.)
  - **Any of the above qualifiers are met (OR logic).**
  - **The following combination of the above conditions is met.**
5. If you selected one of the first two options, this action completes your rule's qualifier. Click **Save**.
6. If you selected the third option, use the text field that appears to write a formula that uses the qualifier conditions that you have defined. For details, see [Qualifier Logic](#).
7. Click **View Qualifier**.

Your entire qualifier expression appears, showing the logical grouping that you have defined in your formula. For an example, see [Viewing Completed Qualifier](#).

8. Click **Save**.

To complete a rule, you must define its action. For details about the rule action for various rule types, see [Rule Actions](#).

## 3.1.12.4.3.5 Qualifier Item Examples

The following are examples of qualifier items for rules. Use these examples as models for constructing your own rule qualifiers.

Each example demonstrates exactly what you would click, select, and type to define the qualifier item, using numbered steps that start with the left argument selections and finish with the right argument selections.

These examples use hypothetical values for categories, custom fields, user group names, and so on.

## Rule Qualifier Item Examples

Can be used for:	What this qualifier item is checking	Left argument selections	Operator selections	Right argument selections
Any object	Checks whether the current user who is attempting the operation is a member of the <b>Paralegals</b> group.	1 Current User 2 Click the Object Navigator icon 3 groupMemberList-> (traverse) 4 group-> (ok)	5 Is 6 Equal To	7 Literal 8 Paralegals
Any object	Checks whether the current user is Bob Johnson.	1 Current User	2 Is 3 Equal To	4 Literal 5 Bob Johnson
Any object	Checks whether the current user is the same user who created the record.	1 Current User	2 Is 3 Equal To	4 Attribute 5 Current Object 6 Click the Object Navigator icon 7 createdBy-> (ok)
Contact-centric custom object	Checks whether the contact selected in the contact-centric field of the project (such as the <b>Insured</b> ) is the contact selected in a custom field of type Involved (such as <b>Requestor</b> ).	1 Current Object 2 Click the Object Navigator icon 3 contact-> (ok)	4 Is 5 Equal To	6 Attribute 7 Current Object 8 Click the Object Navigator icon 9 detailList-> (traverse) 10 RequestedBy-> (traverse) 11 contact-> (ok)



Any object	Checks whether the value of a custom field <b>Trial Date</b> has changed (for example, to trigger an audit rule).	1 Current Object 2 Click the Object Navigator icon 3 detailList-> (traverse) 4 Select the category (traverse) 5 TrialDate (ok)	6 Is 7 Changed	8 Any Value 9 Any Value
Custom object (such as a matter)	Checks whether an assignee with the role <b>Outside Counsel</b> has been added to the record.	1 Current Object 2 Click the Object Navigator icon 3 assigneeList-> (traverse) 4 Outside Counsel (ok)	5 Not 6 Item Added	None
Custom object (such as a matter)	Checks whether the phase is changed from some other phase to <b>Closed-Pending</b> .	1 Current Object 2 Click the Object Navigator icon 3 currentPhaseType -> (ok)	4 Is 5 Changed	6 From: Any Value 7 To: This Value 8 Closed- Pending
Any object	Checks whether the custom field <b>Disposition Date</b> has no value (is null).	1 Current Object 2 Click the Object Navigator icon 3 detailList-> (traverse) 4 Select the category (traverse) 5 Disposition Date (ok)	6 Not 7 Populated	None
Any object	Checks whether the <b>Suit Served</b> date is after the <b>Suit Filed</b> date (both custom fields).	1 Current Object 2 Click the Object Navigator icon	6 Is 7 After	8 This Date 9 Current Object

		<p>3 detailList-&gt; (traverse)</p> <p>4 Select the category (traverse)</p> <p>5 Suit Served (ok)</p>		<p>10 Click the Object Navigator icon</p> <p>11 detailList-&gt; (traverse)</p> <p>12 Select the category (traverse)</p> <p>13 Suit Filed (ok)</p>
Any object	Checks whether the <b>Incident Date</b> was during the last 30 days. Incident Date is a custom field.	<p>1 Current Object</p> <p>2 Click the Object Navigator icon</p> <p>3 detailList-&gt; (traverse)</p> <p>4 Select the category (traverse)</p> <p>5 Incident Date (ok)</p>	<p>6 Is</p> <p>7 Within The Last X Days</p> <p>8 30</p>	9 From Now
Any object	Checks whether the <b>Loss Estimate</b> is greater than or equal to 1000 dollars. Loss Estimate is a custom field.	<p>1 Current Object</p> <p>2 Click the Object Navigator icon</p> <p>3 detailList-&gt; (traverse)</p> <p>4 Select the category (traverse)</p> <p>5 Loss Estimate (ok)</p>	<p>6 Is</p> <p>7 Greater Than Or Equal To</p>	<p>8 Literal</p> <p>9 1000</p>
Task or Expense	<p>Checks whether the project associated with the record is in <b>Closed</b> status. (This is the record status—not the phase.)</p> <p><b>Tip:</b> You may use this qualifier with the Post rule trigger to</p>	<p>1 Current Object</p> <p>2 Click the Object Navigator icon</p> <p>3 project-&gt; (traverse)</p> <p>4 isClosed (ok)</p>	<p>5 Is</p> <p>6 Equal To</p>	<p>7 Literal</p> <p>8 Yes</p>

	<i>prevent posting to a closed project.</i>			
Invoice	Checks the type of invoice, since some types may be subject to more stringent verification and approval than other types are. You may want to restrict your rule to a specific invoice type, or you may want to exclude a specific type (for example, "Not Shadow".)	1 Current Object 2 Click the Object Navigator icon 3 typeID (ok))	4 Is 5 Not	6 Standard 7 Accrual 8 Credit Note 9 Shadow
Task or Expense	Checks whether the Current Phase of the project associated with the record is <b>Pending</b> .  <i>Tip: You may use this qualifier with the Post rule trigger to prevent posting to a project when it is in an inappropriate phase for posting.</i>	1 Current Object 2 Click the Object Navigator icon 3 project-> (traverse) 4 currentPhaseType -> (ok)	5 Is 6 Equal To	7 Literal 8 Pending
Contact	Checks whether there are any phone numbers added to the contact.	1 Current Object 2 Click the Object Navigator icon 3 phoneList-> (ok)	4 Not 5 Populated	None

#### 3.1.12.4.3.6 Limitations for User Interface Qualifiers

If your rule requirements call for qualifiers that cannot be created using Designer, you must write automated qualifiers in Java or JavaScript. For details about creating automated qualifiers, see [Writing Automated Qualifiers for Rules](#).

For example, the following qualifiers cannot be created using Designer and therefore must be written as automated qualifiers:

- Check the values for more than one attribute of a single sub-object.

For example:

- A qualifier that checks whether there is an active assignee of the type Underwriter.
- A qualifier that checks whether an invoice line item has a certain task category and the amount of the same line item.
- A qualifier that checks the state and the city of the same address of a particular type from a contact's list of addresses. This is only possible if you are checking the state and city of a contact's default address.

- Check the values for more than one attribute of a single related object from a list of related objects.

For example:

- A qualifier that checks whether there is an involved (through involvedList->) with a certain role, and in the same involved's contact record, whether it has a certain firm as its company in the Company field.
- A qualifier that checks whether there is a task related to the project with a certain category and that same task is not completed.

- Check the value in a record that is not somehow related to the current object for which the rule is written. This requires searching, which is possible in rule qualifiers only through custom code.
- Perform complex calculations.

#### 3.1.12.4.4 Rule Actions

A rule action is what a rule performs when the entire qualifier is met. The fields on the **Action** tab depend on the rule type:

- The action for security and validation rules is to deny the operation that is attempted by the user if the qualifier conditions in the rule are met. If an operation is denied by a security or validation rule, a message appears.

You may view, create, or modify a rule's rejection message on the **Action** tab. For details, see [Defining Messages for Security and Validation Rules](#).

- The action for approval rules is to follow a Route of approvers who decide whether to deny or allow the operation.

You must set various parameters for an approval rule, including which route to use, on the **Action** tab of the rule. For details, see [Defining Actions for Approval Rules](#).

- The action for custom action or scheduled action rules is defined using Java or JavaScript. You select the file that defines the action on the **Action** tab of the rule.

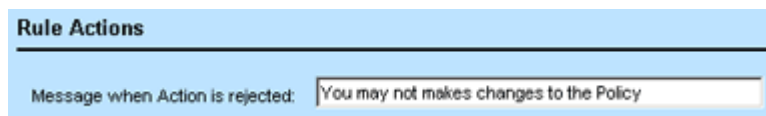
You may also need to make additional selections on the **Action** tab, depending on the rule type and whether additional fields appear that are specific to the custom rule. For details about the fields available for scheduled action rules, see [Defining Settings for Scheduled Action Rules](#).

#### 3.1.12.4.4.1 Defining Messages for Security and Validation Rules

When the qualifier for a security or validation rule is met, the operation attempted by the user is automatically rejected. You are only required to enter a message that appears for users when the operation is rejected. If the operation is approved, the rule does not display a message to the user.

Keep the following points in mind when writing messages for rules:

- The message may not only help the user understand what to do, but it may also help you identify which rule was triggered when users are having difficulties. For example, you may include a reference number that identifies the rule.
- If you do not specify a message, a generic message box is shown to users. Make sure to include a message that is descriptive and includes a possible solution for the user.
- Messages are limited to 2000 characters. If you type more characters than that, the extra characters will be truncated without warning.



Action Tab on Security or Validation Rule Screens

#### To specify a denial message for a security or validation rule

1. Open the **Action** tab of the rule whose message you want to specify.
2. Enter a message in the **Message when Action** is rejected field.

For example:

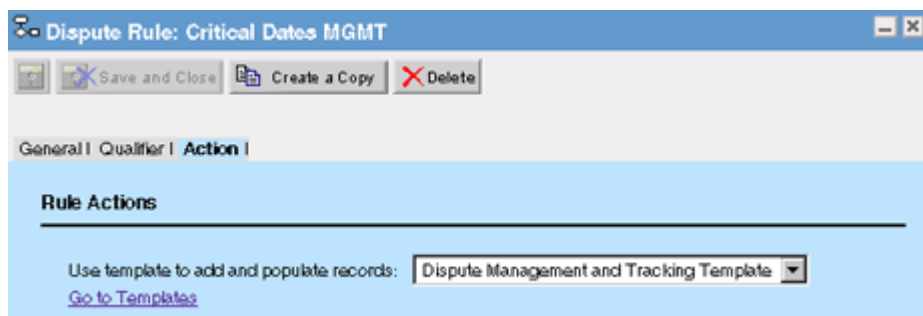
```
You must include an Attorney and a Main Assignee to save this Litigation record.
```

3. Click **Save** to save the message as part of the rule.

#### 3.1.12.4.4.2 Defining Actions for Pre-population Rules

When a pre-population rule is triggered, the rule uses its template to prepopulate specific fields or child records. For more information about creating templates for an object definition, see [Using Templates](#).

To be able to specify a template, you must create at least one template defined for use with rules. If you do not specify a template, no fields are prepopulated.

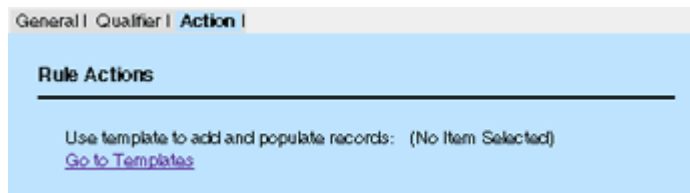


Action Tab on a Pre-population Rule

### To specify the template used by a pre-population rule

1. Open the **Action** tab of the pre-population rule whose template you want to specify.
2. From the **Use template to add and populate records** drop-down list, select the appropriate template.  
  
If no list appears, no templates have been defined to be used with rules for the object definition.
3. Click **Save** to associate the template with the rule.

The following image shows the **Action** tab when no templates are available for the object definition. **No Item Selected** appears as read-only text and the **Go to Templates** link allows you to open the **Templates** tab of the object definition.



**Action Tab of Rule Screen with No Available Templates**

For more information about templates, see [Using Templates](#).

#### 3.1.12.4.4.3 Defining Actions for Approval Rules

When defining the action of an approval rule, Use the Action tab to perform the following actions:

- Select the route that the approval rule follows when the qualifier conditions are met.
- Specify a Process Manager. The user group that you specify as Process Manager must contain at least one active user.
- Specify settings, either at the route level or stop level, for the rule as follows:
  - Whether or not to allow the record to be sent for review.
  - The expiration action at the route level or for each individual stop, such as automatically approve, automatically reject, or send to Process Manager.
  - How to handle approvers who appear at multiple stops.
  - Who may update the record while it is pending approval.

The settings available for the approval rule action depend on whether you want to set route-level parameters or stop-level parameters.

### About Process Managers

**Note:** Although this documentation discusses Process Manager as if it were a single person, a Process Manager is always defined via a user group. You cannot assign an individual user to the Process Manager role. All members of the assigned group receive email related to

*workflow requests, and all have equal authority to perform Process Manager tasks. A task requires only one group member to complete it; whoever handles it first handles it for the entire group.*

An approval rule always requires a Process Manager. The default value for Process Manager is the group "Workflow Process Manager." You may wish to change the default to some other user group when editing the rule. The Process Manager becomes responsible for [monitoring approval processes](#) triggered by that rule on the **My Workflows** screen. The Process Manager may perform the following actions:

- Reject the requested operation by returning it to the user who triggered it.
- Restart the approval process at the first stop of the route.
- Reassign any current approval requests to any TeamConnect user.
- Reassign the approval request to approvers in the current stop who rejected the requested operation.

## About Route-level versus Stop-level Parameters

The main difference between Stop-level versus Route-level parameters is that for Stop-level parameters, you may set specific options, such as allowing approvers to create an impromptu reviewer list, allowing stop members to update records (pending approval), and setting the approval time limit and expiration action on a per stop basis. If you choose Route-level parameters, those options are available, but applied universally to all stops for that route.

### To define the action settings for an approval rule

1. Open the **Action** tab of the approval rule.
2. Select the route of approvers that you want this rule to follow from the drop-down list. For details about routes, see [Creating Routes](#).
3. Type or select the rest of the settings as described in the Action Tab on Approval Rule Screens tables for [Stop-level Parameters Selected](#) or [Route-level Parameters Selected](#).
4. Click **Save** to save the settings as part of the approval rule.

For details about how approval rules function, see [Approval Rules](#). For more details about routes, see [Creating Routes](#).



The following table describes the fields in the **Action** tab of approval rule screens when using stop-level parameters.

**Action Tab on Approval Rule Screens (Stop-level parameters selected)**

Field	Description
-------	-------------

<b>Approval route</b>	Select an existing route from the drop-down list to specify the users who must authorize the request. If this area displays (No Items to Select), then you must first create a route. For details, see <a href="#">Creating Routes</a> .
<b>Process Manager</b>	<p>Select a user group from the drop-down list. (You may first need to define a new group and/or add users to a group, as explained in "Account Administration" chapter of the TeamConnect Administration Guide.) The group must contain at least one active user. When you add a Process Manager, any user in that group may reassign processes, restart processes, or return requests to requestors for the following statuses:</p> <p><b>Action on reject - Send to Process Manager</b> (see below in this table)</p> <p><b>Expiration action - Send to Process Manager</b> (see below in this table)</p>
<b>Action on reject</b>	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Return to requestor</b>—This stops the approval process. The requestor will still have the option to resubmit the approval request.</li> <li>• <b>Send to first stop</b>—If an approver at the first stop rejects the operation, the approval process will end. If an approver at the second stop or later rejects the operation, the approval will be pending for the first stop's approver. No matter where in the route the rejection took place, the request is rejected. The Workflow block of the record indicates who rejected the operation and the date and time.</li> <li>• <b>Send to Process Manager</b>—Select to allow a Process Manager to reassign the request to a new user (non-approver), restart the approval process, or return the request to the requestor.</li> </ul>
<b>Action on return to requestor</b>	<p>This field is used when the previous field, Action on Reject, has the option "Return to Requestor" selected. If Action on Reject has a different option, the setting in this field is ignored.</p> <p>Select one of the following:</p> <ul style="list-style-type: none"> <li>• <b>No Action</b>—The approval is marked as rejected and it is returned to the requestor with no other action taken.</li> <li>• <b>Use template</b>—The approval is marked as rejected and the template that is selected in the drop-down list is executed. The link <b>Go to Templates</b> allows you to view and manage templates, then return to this page and choose a template to be executed with this action.</li> <li>• <b>Use automated action</b>—The approval is marked as rejected and the automated action that is selected in the drop-down list is</li> </ul>



	<p>executed. The link <b>Go to Automated Action Folder</b> allows you to view and manage automated actions, then return to this page and choose an automated action to be executed with this rule action.</p>
<p><b>If the same Approver is at multiple stops</b></p>	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Approver approves at each assigned stop</b>—The assigned approver must approve at all assigned stops. This is the default option.</li> <li>• <b>Approver approves once and is skipped in subsequent stops</b>—Once the approver approves at the first stop, all subsequent stops where this user is an approver are marked "Approved" with comments populated. Workflow functions as if the user manually approved each of these stops.</li> </ul> <p><i>Notes: This option only applies to actual approver action and does not apply if stop members are auto-approved via an expiration date.</i></p> <p>Also, if a stop member approves under this option, and later, a process manager assigns the same user to a stop, the stop member must manually approve the new assignment and approval automation will not overwrite the previously-automated approval.</p>
<p><b>Use Stop-level parameters</b></p>	<p>If you select the Use Stop-level parameters option, you may define workflow approval behavior per stop.</p>
<p><b>Options available prior to approval</b></p>	<p>You may select one or more of the following for each route stop:</p> <ul style="list-style-type: none"> <li>• <b>Send for review</b>—(optional) Select this option to allow the approver the option to assign reviewers to the operation before he or she approves.</li> <li>• <b>Send for additional approval</b>—Select this option to allow the approver, before approving, to send the request to another user. If the new approver approves the request, the request continues to the original approver. If the new approver rejects the request, the specified <b>Action on Reject</b> occurs.</li> <li>• <b>Update records pending approval</b>—If approval is pending for one or more approvers at a stop level, then any of the approvers at that level have edit rights for the tabs and fields in the record. This field is mutually exclusive with the "<b>All users that are approvers can update</b>" option present in the <b>Options available upon approval</b> section.</li> <li>• <b>All users that are approvers can update</b>—All users on the route will have edit rights for the tabs and fields in the record. This field is mutually exclusive with the "<b>Update records pending approval</b>" option present in the <b>Options available</b></li> </ul>

	<p><b>prior to approval</b> section. <b>Note:</b> <i>This selection is only available if all users are required on the stop.</i></p>
<b>Options available upon approval</b>	<p>You may select the following for each route stop:</p> <ul style="list-style-type: none"> <li>• <b>Send for additional approval</b>—Select this option to allow the approver, after approving, to send the request to another user.</li> </ul> <p>If the new approver approves the request, the request continues to the next stop. If the new approver rejects the request, the specified <b>Action on Reject</b> occurs.</p> <p>If the rule uses a route with conditions, this settings does not appear on your rule.</p>
<b>Approval time limit</b>	<p>You may set the number of days approvers in each stop have to respond.</p> <p>This number is converted into exact hours. For example, entering 5 gives the approvers five days, or 120 hours, to make a decision concerning the action. So if the rule is invoked on October 15, 2008 at 3:00 pm, the approvers have until October 20, 2008 3:00 pm to approve the action.</p> <p>The countdown starts when the operation is first attempted by the user.</p> <p>You must use a value other than zero when specifying the length of an approval process. Otherwise, the approval period expires almost immediately.</p>
<b>Expiration action</b>	<p>You may select one of the following for each route stop:</p> <ul style="list-style-type: none"> <li>• <b>Send to Process Manager</b>—Select this radio button to allow the Process Manager to reassign or reject a request after the approval time limit has passed.</li> <li>• <b>Auto-approve</b>—Select this radio button to allow the operation to be performed after the approval time limit has passed.</li> <li>• <b>Auto-reject</b>—Select this radio button to prevent the operation from being performed after the approval time limit has passed.</li> </ul>
<b>Email Actions</b>	<p>The <b>Allow approval/rejection via email</b> field is a check-box you select if you want users who receive approval requests during the stop to receive an email from which they can approve or reject the request. This setting also allows them to enter internal comments and comments to the requestor after approving or rejecting through email.</p>
<b>Email Template</b>	<p>The drop-down contains all email templates you can select as the approval request email users receive during this stop. If you do not</p>

	select a template, TeamConnect uses a default email template. You can design an approval request email from the <b>Notifications</b> page of <b>Admin Settings</b> . See <a href="#">Notification Settings</a> for more information.
<b>Hold Conditions</b>	Hold conditions allow you to ensure approvals on a certain stop are contingent on meetin predetermined stipulations. Use the Hold Condition message entry field to provide users with detail on why the approval is blocked. For more details, see <a href="#">Hold Conditions</a> . <b>Note:</b> <i>Feature only available in TeamConnect 4.0.8+.</i>



The following table describes the fields in approval rule screens when using route-level parameters.

**Action Tab on Approval Rule Screens (Route-level parameters selected)**

Field	Description
<b>Approval route</b>	Select an existing route from the drop-down list to specify the users who must authorize the request. If this area displays (No Items to Select), then you must first create a route. For details, see <a href="#">Creating Routes</a> .
<b>Process Manager</b>	Select a user group from the drop-down list. When you add a Process Manager, any user in that group may reassign processes, restart processes, or return requests to requestors for the following statuses:  <b>Action on reject - Send to Process Manager</b> (see below in this table)  <b>Expiration action - Send to Process Manager</b> (see below in this table)
<b>Action on reject</b>	Select one of the following: <ul style="list-style-type: none"> <li>• <b>Return to requestor</b>—Stops the approval process. The requestor will still have the option to resubmit the approval request.</li> <li>• <b>Send to first stop</b>—If an approver at the first stop rejects the operation, the approval process ends. If an approver at the second stop or later rejects the operation, the approval will be pending for the first stop's approver(s).</li> <li>• <b>Send to Process Manager</b>—Select to allow a Process Manager to reassign the request to a new user (non-approver), restart the approval process, or return the request to the requestor.</li> </ul>

<p><b>If the same Approver is at multiple stops</b></p>	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Approver approves at each assigned stop</b>—The assigned approver must approve at all assigned stops. This is the default option.</li> <li>• <b>Approver approves once and is skipped in subsequent stops</b>—Once the approver approves at the first stop, all future stops where this user is an approver are marked "Approved" with comments populated. Workflow will function as if the user has manually approved each of these stops.</li> </ul> <p><i>Notes: This option only applies to actual approver action and does not apply if stop members are auto-approved via an expiration date.</i></p> <p>Also, if a stop member approves under this option, and later, a process manager assigns the same user to a stop, the stop member must manually approve the new assignment and approval automation will not overwrite the previously-automated approval.</p>
<p><b>Use Route-level parameters</b></p>	<p>This option applies a general set of workflow approval settings across all stops for the selected route. See more details below.</p>
<p><b>Options available prior to approval</b></p>	<p>You may select the following for the route:</p> <ul style="list-style-type: none"> <li>• <b>Send for review</b>—(Optional) If you select <b>Use Route-level parameters</b>, you may select this option to allow the approver the option to assign reviewers to the operation before the approver approves.</li> <li>• <b>Send for additional approval</b>—Select this option to allow the approver, before approving, to send the request to another user.</li> </ul> <p>If the new approver approves the request, the request continues to the original approver. If the new approver rejects the request, the specified <b>Action on Reject</b> occurs.</p>
<p><b>Options available upon approval</b></p>	<p>You may select the following for the route:</p> <ul style="list-style-type: none"> <li>• <b>Send for additional approval</b>—Select this option to allow the approver, after approving, to send the request to another user.</li> </ul> <p>If the new approver approves the request, the request continues to the next stop. If the new approver rejects the request, the specified <b>Action on Reject</b> occurs.</p> <p>If the rule uses a route with conditions, this settings does not appear on your rule.</p>

<p><b>Updates to records pending approval</b></p>	<p>If you select Use Route-level parameters, you may select one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Not allowed.</b> While the request is pending approval, no contacts have edit rights for the tabs and fields related to that object.</li> <li>• <b>Allowed to approvers of current stop.</b> The approval request is sent to all members of the group. While the request is pending approval, any approver from the current stop has update rights for the tabs and fields in the record.</li> </ul> <p><i>Note: If the Route Stop has been set to require all members to approve, and this option is selected, the stop members will not be able to update the record.</i></p> <ul style="list-style-type: none"> <li>• <b>Allowed to anyone.</b> While the request is pending approval, any approvers have update rights for the tabs and fields in the record.</li> </ul>
<p><b>Approval time limit</b></p>	<p>(Optional) Enter the number of days approvers have to respond.</p> <p>This number is converted into exact hours. For example, entering 5 gives the approvers five days, or 120 hours, to make a decision concerning the action. So if the rule is invoked on October 15, 2001 at 3:00 pm, the approvers have until October 20, 2001 3:00 pm to approve the action.</p> <p>The countdown starts when the operation is first attempted by the user. It does not reset each time a route stop is completed.</p>
<p><b>Expiration action</b></p>	<p>You may select one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Send to Process Manager</b>—Select to allow the Process Manager to reassign or reject a request after the approval time limit has passed. The Process Manager will have option to either restart the workflow approval process or return the request to the requestor</li> <li>• <b>Auto-approve</b>—Allow the operation to be performed after the approval time limit has passed.</li> <li>• <b>Auto-reject</b>—Prevent the operation from being performed after the approval time limit has passed.</li> </ul>
<p><b>Internal auto-reject comment</b></p>	<p>(Optional) If the <b>Auto-reject</b> option was selected above, then you may type a comment that will be displayed to the requestor, such as the reason for rejection. The maximum length is 250 characters.</p>

## 3.1.12.4.4.4 Specifying Actions for Custom Action Rules

When a custom action rule is triggered, its action is specified by one of the following:

- JavaScript file
- Java class file
- Template defined in the object definition for use by rules

For information about defining custom actions using JavaScript or Java classes, see [Writing Automated Actions](#). For details about creating templates for an object definition, see [Using Templates](#).

## Specifying a Class or JavaScript File

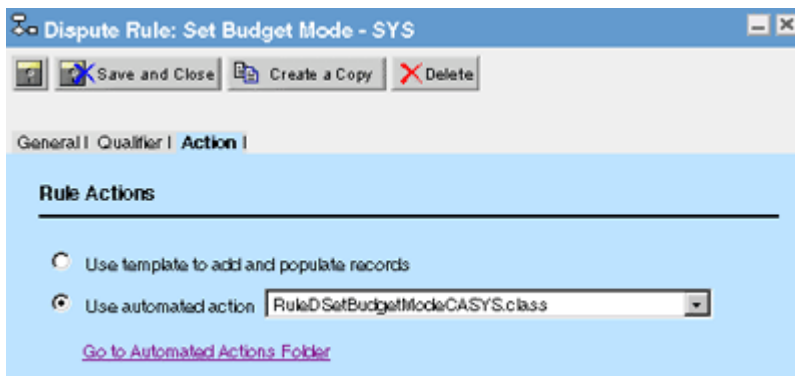
To specify an automated action, you must upload at least one class or JavaScript file to the following folder in the Documents area:

**/System/Object Definitions/objectName/Rules/Automated Actions/**

**Note:** The **Action** tab of the custom action rule screen provides a link to the **Automated Actions** folder for the object definition (see the [Action Tab Displaying an Error with a Custom Action image](#)).

### To specify the Java class or JavaScript file used by a custom action rule

1. Open the **Action** tab of the custom action rule.
2. Select the **Use automated action** radio button.  
The **Use automated action** drop-down list appears.
3. From the **Use automated action** drop-down list, select the appropriate file.



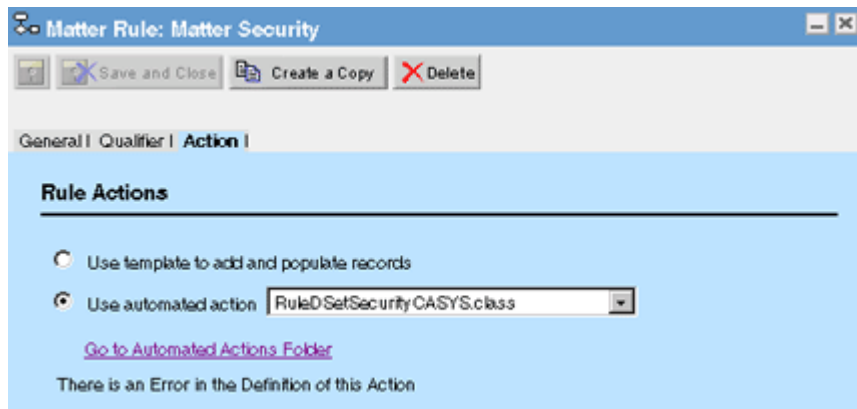
**Action Tab Specifying a Custom Action**

If **No Item Selected** appears as read-only text, that means no Java class or JavaScript files have been uploaded to the Automated Actions folder for the object definition.

4. Click **Save** to associate the action with the rule.

If TeamConnect detects errors while instantiating the selected class file (in relation to the object definition and its other classes), it displays the following error on the **Action** tab:

### There is an Error in the Definition of this Action



Action Tab Displaying an Error with a Custom Action, Click Link to Open Automated Actions Folder

## Specifying a Template

Templates allow you to define specific fields or child records to populate when your qualifiers are met. You may access or create templates from the **Templates** tab of the object definition.

When used with templates, custom action rules have the same functionality as pre-population rules. However, the execution order of custom action rules is after pre-population, validation, and approval rules. This may be useful for some business cases, such as running approval workflow before populating a record.

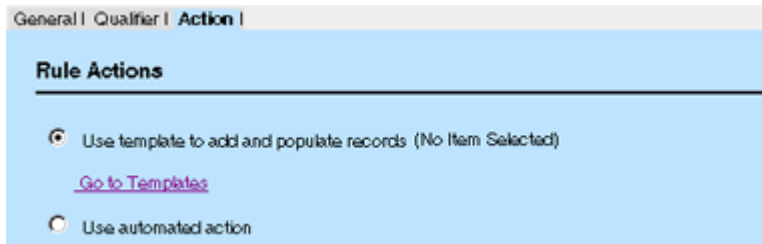
To be able to specify a template for a rule, you must create at least one template for use with rules.

**Note:** You cannot specify a template for a custom action rule for the **User** or the **Group** object definition.

### To specify the template used by a custom action rule

1. Open the **Action** tab of the custom action rule whose template you want to specify.
2. Select the **Use template to add and populate records** radio button.  
The **Use template to add and populate records** drop-down list appears.
3. From the **Use template to add and populate records** drop-down list, select the appropriate template.  
If **No Item Selected** appears as read-only text, no templates have been defined to be used with rules for the object definition.
4. Click **Save** to associate the template with the rule.


The following image shows the **Action** tab when no templates are available for the object definition. **No Item Selected** appears as read-only text and the **Go to Templates** link allows you to open the **Templates** tab of the object definition.



Action Tab of Rule Screen with No Available Templates, Click Link to Open Templates Folder

For more information about templates, see [Using Templates](#).

### To specify the template used to send notification

1. Open the **Action** tab of the custom action rule whose template you want to specify.
2. Select the **Use template to send notification** radio button.
3. From the **Use template to send notification** drop-down list, select the [appropriate template](#).
4. From the **To/CC/BCC** drop-down list, select the type of email to be sent to the recipient: A regular email, a carbon copy email, or a blind carbon copy email.
5. From the **Recipient** drop-down list, select the location of the recipient. Notification emails will be sent to the default email address of a user or contact.
  - Select **Path** and click the **Navigator**  icon to open Object Navigator, and then select an attribute. For details, see [Using Object Navigator](#).
  - Select **Group** to send the notification to all members of a defined group.
  - Select **Address Book** to send the notification to users in an address book.
  - Select **Contact Object Collection** to send the notification to users in a defined object collection, such as All Contacts, Recently Created, Recently Modified, etc.
  - Select **Email String** to send the notification to an email address.
6. (Optional) Select the **Track in History** check-box if you want a history record to be created every time a notification is sent out. A drop-down list appears.

From the **Location** drop-down list, select which level you want the history record to be saved in: **Record**, **Object**, or **System**. **Record** will not be available if the rule triggering event is **Delete**.

7. Click **add more** to continue adding notification recipients if necessary, or click **Save** to save the custom action rule.

**Important:** Notification templates that contain an object link or object link display string will not display these items correctly in the notification email for Custom Action rules that use the Create triggering event. This is because the object link has not yet been created when the Create triggering event occurs. To avoid this issue, use the Post Commit rule type instead of the Custom Action rule type.



## 3.1.12.4.4.5 Defining Settings for Scheduled Action Rules

Scheduled action rules have a set of fields on the **Action** tab where you must do the following actions:

- Select the file that is executed or the template that is used according to the schedule that you specify.
- Specify when, and how often, the action is executed after the rule has been triggered.

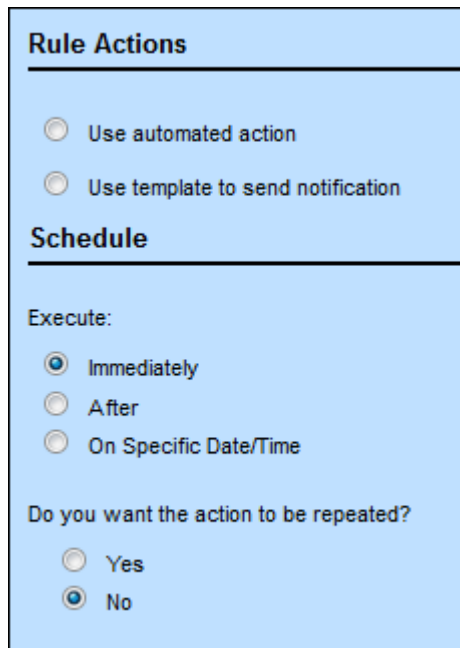
To specify an automated action, you must upload at least one class or JavaScript file to the following folder in the Documents area:

**/System/Object Definitions/objectName/Rules/Automated Actions/**

**To define the action settings for a scheduled automated action rule**

1. Open the **Action** tab of the scheduled action rule.
2. Select the **Use this Automated action** radio button, then select the custom action Java or JavaScript file that you want the rule to execute from the **Use this Automated action** drop-down list.  
  
If **No Item Selected** appears as read-only text, that means no Java class or JavaScript files have been uploaded to the Automated Actions folder for the object definition.
3. Type or select the rest of the settings as described in [the Action Tab on Scheduled Action Rule Screens table](#).
4. Click **Save** to save the scheduled action settings as part of the rule.

For more details about scheduled action rules, see [Scheduled Action Rules](#).



**Rule Actions**

Use automated action

Use template to send notification

**Schedule**

Execute:

Immediately

After

On Specific Date/Time


Do you want the action to be repeated?

Yes

No

**Action Tab on Scheduled Action Rule Screens**

### To define the action settings for a scheduled notification action rule

1. Open the **Action** tab of the scheduled action rule.
2. Select the **Use template to send notification** radio button.
3. From the **Use template to send notification** drop-down list, select the [appropriate template](#).
4. From the **To/CC/BCC** drop-down list, select the type of email to be sent to the recipient: A regular email, a carbon copy email, or a blind carbon copy email.
5. From the **Recipient** drop-down list, select the location of the recipient. Notification emails will be sent to the default email address of a user or contact.
  - Select **Path** and click the **Navigator**  icon to open Object Navigator, and then select an attribute. For details, see [Using Object Navigator](#).
  - Select **Group** to send the notification to all members of a defined group.
  - Select **Address Book** to send the notification to users in an address book.
  - Select **Contact Object Collection** to send the notification to users in a defined object collection, such as All Contacts, Recently Created, Recently Modified, etc.
  - Select **Email String** to send the notification to an email address.
6. Click **add more** to continue adding notification recipients if necessary.
7. Type or select the rest of the settings as described in [the Action Tab on Scheduled Action Rule Screens table](#).
8. Click **Save** to save the scheduled action settings as part of the rule.

The following table describes the fields in **Scheduled Action Rule** screens.

**Action Tab on Scheduled Action Rule Screens**

Field	Description
<b>Use this Automated action</b>	<p>Select the Java or JavaScript file that defines the action that is executed according to the schedule. This same action is executed for each repetition that you specify in these parameters.</p> <p>This drop-down list contains documents that have been uploaded to <b>Top Level/System/ObjectDefinitions/ObjectName/Rules/Automated Actions</b> and have the document type <code>JavaClass</code> or <code>JavaScriptCode</code>.</p> <p>To open the folder where Automated Action files for this object definition are stored, click <b>Automated Actions Folder</b> hyperlink.</p>
<b>Use template to send notification</b>	<p>Select the email notification template that is to be sent according to the schedule. This same notification template is sent for each repetition that you specify in these parameters.</p>

<b>Track in History</b>	<p>(Optional) Select whether a history record will be created every time a notification is sent out. A drop-down list appears when the <b>Track in History</b> check-box is selected.</p> <p>From the <b>Location</b> drop-down list, select which level you want the history record to be saved in: <b>Record</b>, <b>Object</b>, or <b>System</b>. <b>Record</b> will not be available if the rule triggering event is <b>Delete</b>.</p>
<b>Execute</b>	<p>Specify when the automated action or notification is to be executed.</p> <ul style="list-style-type: none"> <li>• <b>Immediately</b>—The action is executed immediately after the rule is triggered. This is the default option.</li> <li>• <b>After</b>—The action is executed a specified period of time after the rule is triggered. Select from <b>Minutes</b>, <b>Hours</b>, <b>Days</b>, <b>Weeks</b>, <b>Months</b>, or <b>Years</b> in the drop-down list, then enter a value in the text box to specify how long after the trigger the rule will be executed.</li> <li>• <b>On Specific Date/Time</b>—Use the fields that appear to specify the calendar date and time for the scheduled action to be executed.</li> </ul>
<b>Do you want the action to be repeated?</b>	<p>Select whether the action is repeated. If you select <b>Yes</b>, then additional options appear, where you specify how often and how many times the action is repeated. The default option is <b>No</b>.</p>
<b>Repeat</b>	<p>Specify the period of time between rule action repetitions after the initial execution after the rule has been triggered. Additional options will appear depending on the selection in the <b>Repeat</b> drop-down list.</p> <ul style="list-style-type: none"> <li>• <b>Hours and Minutes</b>—The scheduled action will repeat after the specified number of <b>Hours</b> and <b>Minutes</b> have passed.</li> <li>• <b>Daily</b>—The scheduled action will repeat every day. Select the <b>Weekdays only</b> check-box to set the rule action to be repeated only from Monday through Friday.</li> <li>• <b>Weekly</b>—The scheduled action will repeat every week. Use the <b>Day</b> drop-down list to select on which day of the week the action will be repeated.</li> <li>• <b>Monthly</b>—The scheduled action will repeat every month. Use the <b>Day of the Month</b> drop-down list to select on which day of the month the action will be repeated, from <b>1</b> to <b>31</b>, or <b>Last</b> for the last day of the month.</li> <li>• <b>Yearly</b>—The scheduled action will repeat every year. Use the <b>Day and Month</b> drop-down lists to select on which month and day of the year the action will be repeated.</li> </ul>
<b>End</b>	<p>Specify when the rule action repetitions will stop.</p> <ul style="list-style-type: none"> <li>• <b>No End Date</b>—The scheduled action will repeat indefinitely until the rule is deleted or inactivated.</li> </ul>

- **After**—The scheduled action will execute for the specified number of repetitions. This is the default option, with a value of 1.
- **Until**—The scheduled action will repeat until the specified date and time.

**Important:** Actions that are scheduled to repeat monthly on the 29th, 30th, or 31st day of the month will not occur in months that do not have those dates. To avoid this issue, use the **Last** option for the last day of the month.

#### 3.1.12.4.4.6 Defining Actions for Audit Rules

When the qualifier for an audit rule is met, details about the operation that triggered the rule are automatically logged in a history record. Audit rules are designed to capture information about events, but you may define what is captured and how it is displayed.

The **Action** tab of an audit rule allows you to specify:

- A location for the history records.
- The description of the event.
- A category and custom fields to separate information about the event.

The first thing to consider when you want to audit an event is the target location to create the history record. For example, you may select a contact record as the target location to record a contact name every time it is updated. When the rule is triggered, an audit history record is created, and afterwards, you may view the contact record's **History** tab to track the events.

Next, you need to define the description for the event to be recorded. This description populates the **Description** field of the history record that is created when the rule is triggered. There is already a default description, depending on the triggering event; however, you may customize your own using a batch screen, if you need more specific information.

Finally, you may separate details about the event into specific custom fields for reporting purposes. To accomplish this, you must select a History category and populate its custom fields. When the rule is triggered, in addition to the **Description** field's contents, the custom fields that you defined are populated for the appropriate category on the **Details** block of history records.

#### To define the action settings for an audit rule

1. Open the **Action** tab of the audit rule.

For a description of the fields on an audit rule's **Action** tab, see [the Action Tab on Audit Rule Screens table](#).

2. From the **Location** drop-down list, select a target where the history record will be created when the audit rule is triggered.
3. Define a description for the triggering event to populate the **Description** field of the history record.

4. (Optional) From the **Category** drop-down list, select a History category and define its custom fields to capture event details.
5. Click **Save** to save the settings as part of the audit rule.

The following images show the **Action** tab of an audit rule screen. The action settings for the rule in this image creates a history record in the affected matter record with a custom description that states, for example, "The main assignee was changed from Jake Ashby to Larry Finch." It will also populate a custom field called "Previous Assignee" in the Audit category on the **Details** block of the history record.

General | Qualifier | **Action**

Create History record with the following parameters:

Location:

### History Description

Use default description (i.e. "Matter record has been updated")  
 Use description defined below:

Item Type	Item Value	Order
<input type="text" value="Object Attribute"/>	<input type="text" value="Current Object .mainAssignee"/> <input type="text" value="New Value"/> <input type="text" value="30"/>	
<a href="#">+ add more</a>		
Item Type	Item Value	Order
1 <input type="checkbox"/> Literal	The main assignee was changed from	0
2 <input type="checkbox"/> Character Space		5
3 <input type="checkbox"/> Object Attribute	.mainAssignee - Old Value	10
4 <input type="checkbox"/> Character Space		15
5 <input type="checkbox"/> Literal	to	20
6 <input type="checkbox"/> Character Space		25

[Check All](#) - [Uncheck All](#)

### History Category and Custom Fields

Category:

Field Label	Field Value
<input type="text" value="Previous Assignee"/>	<input type="text" value="(Select)"/>
<a href="#">+ add more</a>	
Field Label	Field Value
1 <input type="checkbox"/> Previous Assignee	Current Object .mainAssignee.user - Old Value

[Check All](#) - [Uncheck All](#)

General | Qualifier | **Action**

Create History record with the following parameters:

Location:

### History Description

Use default description (i.e. "Matter record has been updated")

Use description defined below:

Item Type	Item Value	Order
<input type="text" value="Object Attribute"/>	<input type="text" value="Current Object"/> <input type="text" value=".mainAssignee"/> <input type="text" value="New Value"/> <input type="text" value="30"/>	
<a href="#">+ add more</a>		
Item Type	Item Value	Order
1 <input type="checkbox"/> Literal	The main assignee was changed from	0
2 <input type="checkbox"/> Character Space		5
3 <input type="checkbox"/> Object Attribute	.mainAssignee - Old Value	10
4 <input type="checkbox"/> Character Space		15
5 <input type="checkbox"/> Literal	to	20
6 <input type="checkbox"/> Character Space		25

[Check All](#) - [Uncheck All](#) [edit](#) [delete](#)

### History Category and Custom Fields

Category:

Field Label	Field Value
<input type="text" value="Previous Assignee"/>	<input type="text" value="(Select)"/>
<a href="#">+ add more</a>	
Field Label	Field Value
1 <input type="checkbox"/> Previous Assignee	Current Object .mainAssignee.user - Old Value

[Check All](#) - [Uncheck All](#) [edit](#) [delete](#)


Action Tab on Audit Rule Screens

The following table describes the items on the **Action** tab.


Action Tab on Audit Rule Screens

Field/Button	Description
<b>Location</b>	<p>Select the target location for storing the history record that is created when the audit rule is triggered.</p> <p>Depending on the object that you are defining, this drop-down list may include some or all of the following options:</p>

	<ul style="list-style-type: none"> <li>• <b>Current record</b>—For example, if you select Matter, a related history record will be created in the affected Matter record.</li> <li>• <b>Object definition</b>—For example, if you select Matter Object Definition, a history record will be created in the Matter custom object definition.</li> <li>• <b>Parent project or projects</b>—For example, if you select a Claim's parent project, Policy, a related history record will be created in the policy record.</li> <li>• <b>System</b>—If you select this item, the history record will be created in an area reserved for the TeamConnect System user. You may access history records assigned to this location on the <b>History</b> tab of the <b>System Settings</b> screen.</li> </ul>
<p><b>Use default description</b> (i.e. "Matter record has been updated")</p>	<p>Select this option if you want TeamConnect to insert a default description of the event that triggers the audit rule in the history record's <b>Description</b> field.</p> <p>The default description combines the object type, record name, and triggering event to form a description such as "Matter Simmons vs. Acme Auto Rental has been updated."</p>
<p><b>Use description defined below</b></p>	<p>Select this option if you want TeamConnect to insert a custom description of the event that triggers the audit rule in the history record's <b>Description</b> field.</p> <p>This is the default option, and you must use the associated batch screen to define a custom description.</p> <p>For more information on whether you need a default description or a custom description, see <a href="#">Audit Default Description</a>.</p>
<p><b>Item Type</b></p>	<p>Select an item from this list to build the description for the history record.</p> <p>Depending on the option that you select in this field, different settings appear for the Item Value:</p> <ul style="list-style-type: none"> <li>• <b>Object Attribute</b>—Select this option to define an object attribute.</li> <li>• <b>Literal</b>—Select this option to manually enter a text value.</li> <li>• <b>Line Break</b>—Select this option to insert a carriage return in the text description.</li> <li>• <b>Character Space</b>—Select this option to insert a space between items in the text description.</li> </ul>
<p><b>Item Value</b></p>	<p>Define a value to build the description for the history record.</p>

	<ul style="list-style-type: none"> <li>For object attributes, select <b>Current Object</b> or <b>Current User</b>, click the <b>Navigator</b>  icon to open Object Navigator, and then select an attribute. For details, see <a href="#">Using Object Navigator</a>.</li> <li>For literals, enter a custom message in the text field.</li> </ul>
<b>Old Value or New Value</b>	<p>Select <b>Old Value</b> or <b>New Value</b> for the attribute.</p> <p>This drop-down list only appears if the <b>Item Type</b> is an object attribute. The old value refers to the attribute's value in the TeamConnect database before the record is changed and updated by an action. The new value refers to the attribute's value after the record has been updated and saved.</p> <p><i>Tip: This may be useful in building the history record's description by capturing information that might not exist in the record after it is modified and saved.</i></p>
<b>Order</b>	Enter an integer to indicate the Order of the item you defined to place it correctly within the entire description statement.
<b>Item List</b>	<p>Displays the items that are already added to be used in the custom description.</p> <p>Select the check-box of an item and click <b>edit</b> or <b>delete</b> to make changes.</p>
<b>Category</b>	<p>Select a category from this list to be automatically added to the history record.</p> <p>This list shows all the categories available in the History object definition.</p> <p>Make sure that you have created descriptive categories in the History object definition and defined custom fields within these categories that you plan to populate with the audit rule. For details, see <a href="#">Configuring History Object Definition for Audit Tracking</a>.</p> <p>If you do not select a different category, the <b>Root</b> category is automatically used.</p> <p>After you select a category, you may use the <b>Field Label</b> and <b>Field Value</b> fields to define the content that will populate the category's custom fields.</p>
<b>Field Label</b>	Select the label of the custom field that you want to use in the history record.



	<p>This drop-down list contains the labels for the custom fields in the History object definition. The values available in this list depend on the History category selected in the <b>Category</b> field.</p> <p>These field labels appear on the <b>Details</b> block of the history record that is created when the rule is triggered.</p> <p>This field is not available for the <b>User</b> and <b>Group</b> object definitions.</p>
<b>Field Value</b>	<p>Select <b>Current Object</b> or <i>Current User</i>, click the <b>Navigator</b>  icon to open Object Navigator, and then select an attribute to populate the field that is identified in <b>Field Label</b>. For details, see <a href="#">Using Object Navigator</a>.</p> <p>You must select whether you want the field to be populated with the <b>Old Value</b> or <b>New Value</b>.</p> <p>Make sure you have defined any necessary custom fields for the category you want to use in the History object definition.</p> <p>Make sure to match the field type to the attribute type, such as matching a date field to an object attribute that stores a date value.</p> <p>This field is not available for the <b>User</b> and <b>Group</b> object definitions.</p>

#### 3.1.12.4.5 Rule Administration

When you have finished defining all of the necessary rule specifications for your system, you must perform the actual administration of rules in Designer. This involves tasks such as creating, deactivating, modifying, deleting, and troubleshooting rules. This section describes these procedures in detail so that you may work with your own rules in TeamConnect.

##### 3.1.12.4.5.1 Defining User Interface Rules

The following procedure describes the process of creating rules with predefined actions within TeamConnect. Security, validation, and approval rules may be created using these instructions. To create rules using Java class files or JavaScript for their qualifiers or actions, see [Creating Automated Qualifiers and Actions](#).

Before creating a rule, be sure that you create the correct rule type. For descriptions of security, validation and approval rules, see [Rule Types](#).

#### To create a rule

1. In the Designer window, from the **Go to** drop-down list select **Object Definitions**.
2. Select the appropriate object definition and then click the **Rules** tab.
3. Click **New**.

4. The corresponding **General** tab appears with blank fields.
5. Set the **General Rule Information**. For details, see [Setting General Rule Information](#).
6. Click the **Qualifier** tab and create all of the necessary qualifier conditions. For details, see [Rule Qualifiers](#).
7. Click the **Action** tab and define all of the necessary Rule Action information. For details, see [Rule Actions](#).
8. Click the **General** tab.
9. Select the **This rule is Active** check-box to activate the rule.
10. Click **Save**.

The rule is immediately active and is triggered when a user attempts to perform the operation that is selected as the trigger.

#### 3.1.12.4.5.2 Deactivating Rules

You may easily deactivate rules that are currently active. For example, you may want to temporarily deactivate a rule so that it does not interfere with the behavior of a particular rule that you are testing. You may also want to deactivate a rule because you are replacing it with a different rule.

##### To deactivate a rule

1. Open the rule that you want to deactivate.  
For details about accessing rules, see [Accessing the Rules Screen](#).
2. On the General tab, clear the **This Rule is Active** check-box.
3. Click **Save**.

The rule has been deactivated and will no longer be triggered. However, it still appears in the list of rules for the Object Definition.

##### Points To Remember

When deactivating a rule, consider the following points:

- For all rule types, as soon as the rule is deactivated, it ceases to be triggered on the triggering events. For example, when users click **Save** on a new record, a deactivated rule with a **Create** trigger is not triggered.
- When you deactivate a rule, you may re-activate it later as needed. It remains in the rule list for the object definition.
- When you deactivate an approval rule, approval processes that have already been started are not affected when you deactivate the rule.
- When you deactivate a scheduled action rule, actions that have already been scheduled by the rule are not affected when you deactivate the rule. If you want to cancel actions that have already been scheduled, you must delete them using the [Scheduled Actions Monitor](#).

### 3.1.12.4.5.3 Modifying Rules

You may modify an existing rule when necessary. If the rule is active, the changes immediately take effect and may be observed the next time the rule is triggered. If the rule is inactive, the changes may be observed when the rule has been activated and then triggered.

You may modify existing rules for any of the following reasons:

- To modify existing qualifier items.
- To add or delete qualifier items.
- To change the message when the rule is triggered.
- To change the order in which the rule is executed.
- To change the rule trigger(s).
- To change the route for an approval rule to a different route, or modify other information on the approval rule's **Action** tab.

Approval processes that have already been started are not affected if you modify the approval rule's action settings or select a different route for the rule.

- To change the scheduling of a scheduled action rule.

Actions that have already been scheduled are not affected by modifications to any of the rule's action settings. The changes only affect actions that are scheduled by the rule after you save the changes to the rule.

- To change the file being used as the action for a custom action or scheduled action rule.

Actions that have already been scheduled by a scheduled action rule are not affected if you select a different file for the rule's action. The change only affects actions that are scheduled by the rule after you select the new file for the rule.

- To deactivate the rule (see [Deactivating Rules](#)).

**Note:** Some modifications to rules may require you to click the **Refresh** button to see the modified information reflected in the list of rules.

### 3.1.12.4.5.4 Deleting Rules

You may need to delete a rule during the design phase of your TeamConnect implementation, or because you are sure that the rule is not necessary.

You may delete active and inactive rules. When you delete an active rule, the rule immediately stops taking effect. Users are able to perform the corresponding operation, such as creating, modifying or posting records, without the previous constraints of the rule.

However, rules that have already been triggered are not affected when you delete a rule. For example, if you delete an approval rule that has already been triggered and the approval request has already been routed, then the approval process is not affected by the deletion of the rule.

**Tip:** If you want to keep the rule for future use or future reference, but you do not want the rule to continue to be triggered, you may deactivate the rule. For details, see [Deactivating Rules](#).

### To delete a rule

1. Click the **Rules** tab of the appropriate object definition.
  2. Click the name of the rule you want to delete.  
The **General** tab of the rule opens by default.
  3. Click **Delete** at the top of the screen and confirm that you want to delete the rule.
- The rule has been deleted and will no longer be triggered by user operations.

#### 3.1.12.4.6 Troubleshooting Rules

To assist you with creating rules, this table provides a quick troubleshooting guide of common problems and their solutions.

##### Rule Troubleshooting Tips

Possible problem	Possible solution
An error message appears for a rule that must not be triggered in the situation you are testing.	<p>Check for the following issues:</p> <ul style="list-style-type: none"> <li>• Open the error messages of the rules being triggered.</li> <li>• Do they have the correct messages?</li> <li>• Does the rule have multiple triggers associated with it? Are those the correct triggers?</li> </ul> <p><b>Note:</b> You may have another rule being triggered with the same error message, especially if you used the <b>Create Copy</b> button to create this rule.</p>
A rule is not being triggered.	<p>Once you create any rules, test them to see whether modifications are needed to get a rule to trigger in the correct situations. If your rule is not being triggered, check for the following issues:</p> <ul style="list-style-type: none"> <li>• Is the rule active?</li> <li>• Did you select the correct rule type? Security and validation rules are easy to confuse.</li> <li>• Is the qualifier defined correctly?</li> <li>• Is the entire qualifier met?</li> </ul> <p>If at least one of the qualifier conditions or the logic that you have defined is not met, the rule is not executed.</p>
My Approval rule is not being triggered at the right time.	<p>Check for the following issues:</p> <p>Is the rule active?</p>

	<p>Is a route selected?</p> <p>Does the route have members?</p> <p><b>Note:</b> <i>If a group is the only stop member in the selected route and there are no active users added to the group, there is no indication that the rule was triggered, because the approval request was not sent to any users.</i></p> <ul style="list-style-type: none"><li>• Is the qualifier defined correctly?</li><li>• Is the entire qualifier met?</li></ul> <p>If at least one of the qualifier conditions or the logic that you have defined is not met, the rule is not executed.</p>
--	--

### 3.1.12.5 Creating Routes

A *route* is a sequence of users whose approval is required for users to perform a certain action. Routes are required for approval rules. For details about how approval rules are constructed, see [Approval Rules](#).

Routes may have multiple *stops*, which define the order in which the approval is sent to each user whose approval is required. The members of a route may approve or reject the attempted operation, reassign it to another user, or even edit items in a workflow that are still pending approval. **Note:** *Editing items pending approval in workflow is only available in TC 4.0.8 and later.*

Members of a route may include:

- User groups
- Users who have a certain role in a project record
- Other users who are identified by association

If a route is associated with all objects, only the first choice (user groups) is available when creating stops. The choices for users with roles, or users identified by association, are available for routes that are linked to one specific object type.

If the sequence of approval is important, the stops must be separate. Otherwise, all members may be in the same stop.

For example, to create a route that requires approval from a transaction attorney and a supervisor before an invoice payment over \$10,000 is posted, you would need an approval rule that checks whether an invoice being posted is for more than \$10,000. If only one person must approve the posting, then the route would need only one stop with all members.

If, however, the invoice may only be posted after a transaction attorney and a supervisor both approve the posting, in that order, then the route would contain two stops. The first stop would send the approval to the transaction attorney. If the transaction attorney approves the posting, the approval would then be sent to the supervisor.

You may also define multiple branches (with the same rule) for the same operation when the levels of authority to approve differ depending on the qualifiers that trigger the approval rule. This is because it

is impossible to have an approval finish part of the way through the route. It must finish the route before it is approved.

For example, for invoice posting approval:

- **Up to \$5,000:** the Main Assignee's approval is sufficient.
- **More than \$5,000, but less than \$20,000:** the Main Assignee and the supervisor must approve the posting.
- **More than \$20,000:** the first two individuals and the vice president must all approve the posting.

### 3.1.12.5.1 Accessing Routes Screen

If you have user group rights to Routes, you may access them from the **Setup** link, which starts Designer.

#### To open a new or existing route definition

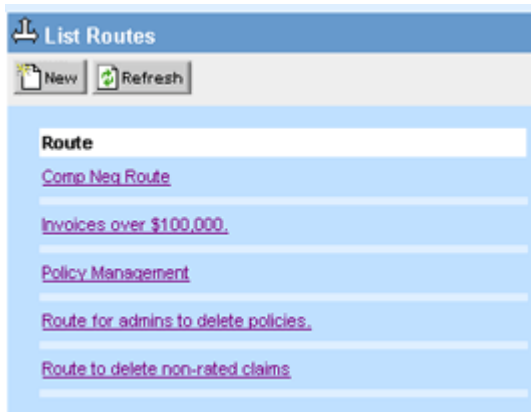
1. In the Designer window bar, in the **Go To** drop-down list, select **Routes**.

The **List Routes** screen appears.

2. Do one of the following actions:
  - In the list of existing routes, click the name of the route you want to open.
  - Click **New** to create a route.

The **General** screen displays by default.

3. Click the **Stops** tab to view the stops information, if necessary.



Routes Screen

You may create, view, and modify a route on its **Route** screen, which has the following tabs:

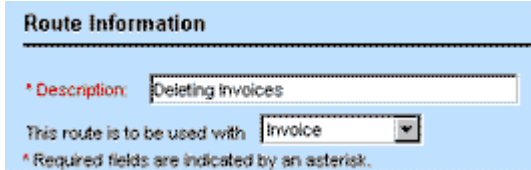
**General** | Stops |

- **General**—Displays the name of the route and object for which it may be used. See [Defining General Route Information](#).

- **Stops**—Displays the stops and members of each stop. See [Route Stops](#).

### 3.1.12.5.2 Defining General Route Information

General route information includes the name of the route and the object to which it applies. These fields are located on the **General** tab of the route.



**General Tab on Route Screens**

The following table describes the items in the **General** tab of **Route** screens.

**General Tab on Route Screens**

Field	Description
<b>Description</b>	Enter a name (maximum 250 characters) that uniquely identifies the route. This description appears on the <b>Action</b> tab of an approval rule to allow you to select it.
<b>This route is to be used with</b>	<p>Select the object for which the route may be used. You may either specify one object, or select <b>(All)</b> to make the route accessible to all objects.</p> <ul style="list-style-type: none"> <li>• If the route you are creating is only used for one object, such as Expense, then select the name of that object in the list. You must select a specific object if you want to add a <b>User with role</b> or <b>User path</b> as a stop member (see <a href="#">Member Types</a>).</li> <li>• If you want to be able to use the same route of approvers for more than one object, you must select <b>(All)</b>. For example, you may need to use the same route for both task and expense approval rules. If the route is available for all objects, then you may only add specific users or groups to the route. You are unable to add a user with role or user path as a stop member (see <a href="#">Member Types</a>).</li> </ul> <p>When the route is saved, you cannot change this selection.</p>

After saving the name and the object selection for the route, you may click the **Stops** tab and define the stops in the route.

### 3.1.12.5.3 Route Stops

Stops represent approval rule check points. The route follows the order in which you organize the stops, starting with Stop 1. A stop consists of members who may approve or reject the operation

attempted by a user. All members of each stop receive the approval notice in their **My Approvals** screen.

### Points To Remember

- If all users in a stop do not exist, the stop is skipped and the route proceeds to the next stop. For example:
  - If a stop includes only the Main Assignee of a Claim and there is no Main Assignee, then the stop is skipped and the route proceeds to the next stop.
  - If a stop includes only a certain assignee in the parent project of a project that is being deleted. If there is no parent project, then the route proceeds to the next stop.
  - If a user with a specified role is assigned to a stop in a route, and there is no such role assigned to any active assignee or group, the assignee is skipped and the request goes to the next approver.
- If an assigned user with a specified role exists, but the user account is inactive, an error message appears if all members are required for approval or if there are no other valid members in the stop.
- Rejection of an approval request always works on a "first come, first serve" basis. Once any designated approver rejects a request, the request record is removed from the **My Approvals** list of all route members, regardless of which stop in the route the rejection occurred.
- Approval may work on a "first come, first serve" basis only if you want it to. For each stop of the route, you may control whether one approval is enough to allow the operation, or all stop members must approve the operation before the route proceeds to the next stop.
- If the sequence of approval does not matter, then you must place all of these members in one stop. Placing multiple members in one stop and requiring approval from all stop members (selecting the check-box) allows you to give more time for all members to approve the operation, rather than having them wait for the previous stop members to approve the operation first.
- If the sequence of approval does matter, then each member whose approval is required must be placed in a separate stop, in the order in which their approval is required. You must also make sure that the expiration time allows for all stops to be reached.

Stops may be viewed, added, deleted, or modified on the **Stops** tab of the route. The **Stops** tab also defines the order of the stops and whether all members of each stop must approve the request before the route proceeds to the next stop.



Create Stop: [new](#)

**Route Stop 1** ▼ delete

All members of this stop are required for approval

Member Type	Member
1 <input type="text" value="User"/>	<input type="text" value="Anderson, Mathew"/> <a href="#">+ add more</a>

Member Type	Member
1 <input type="checkbox"/> Group	<a href="#">Senior Claim Officers</a>

[Check All](#) - [Uncheck All](#) delete

**Route Stop 2** ▲ delete

All members of this stop are required for approval

Member Type	Member
1 <input type="text" value="(Select)"/>	<a href="#">+ add more</a>

Member Type	Member
1 <input type="checkbox"/> User	<a href="#">Anderson, Mathew</a>
2 <input type="checkbox"/> User path	.JinItemList.project.mainAssignee.user
3 <input type="checkbox"/> User with role	User with role Supervisor In JinItemList.project

[Check All](#) - [Uncheck All](#) delete

**Stops Tab**

The following table describes the items on the **Stops** tab.

**Stops Tab**

Field or button	Description
<b>new</b>	Click to create a new stop for the route. The new stop is added to the end of the route.
<b>All members of this stop are required for approval</b>	<p>Select this check-box if all members of the stop are required to approve the request. If the check-box is selected, then the route does not continue to the next stop until all users in the stop have approved the operation.</p> <p>If this check-box is NOT selected, then the route continues to the next stop as soon as one user in the stop has approved the operation.</p> <p>This check-box requires all members no matter how they are added to the stop. This means that a member's approval is required in all of the following situations:</p>

	<ul style="list-style-type: none"> <li>• More than one member (of any type) is added to the stop. Each member's approval is required.</li> <li>• A <b>Group</b> is a member of the stop. This check-box requires an approval from all members of the group.</li> <li>• A <b>User with role</b> or <b>User path</b> stop member points to more than one user with the specified role. For example, there are two General Counsel assignees (user with role). Approval is required from each assignee with the specified role.</li> </ul> <p>Whether or not this option is selected, a single user may reject the attempted operation and thus delete the request from the <b>My Approvals</b> screen of all route members. You cannot set a route to reject an operation only after all members have rejected it.</p>
<b>Member Type</b>	Select the type of stop member you want to add to the stop. For details, see <a href="#">Adding Stop Members</a> .
<b>Member</b>	Select or identify the user or group you want to add to the stop. For details, see the following: <ul style="list-style-type: none"> <li>• <a href="#">Adding Groups as Stop Members</a></li> <li>• <a href="#">Adding User Paths as Stop Members</a></li> <li>• <a href="#">Adding Users With Roles as Stop Members</a></li> </ul>
<b>Member List</b>	Displays the members of the corresponding stop.
<b>Move Up or Down Arrows</b>	Use these arrows to change the order of the stops if needed. Click the respective arrow to move the entire stop up or down one level.  The route follows the order in which you organize the stops, starting with <b>Stop 1</b> .
<b>delete</b>	Click to delete the corresponding stop from the route. Each delete button corresponds to the stop with which it is aligned.  <b>Note:</b> Do not confuse the delete buttons for the stops with the delete button at the bottom of the screen that appears in all batch screens. The delete button for the batch screen applies to stop members rather than an entire stop.
<b>Visit stops in sequence</b>	Select to specify that the route visits each stop in numerical order.  This option does not appear if you can use the route with all record types.

<b>Custom sequence</b>	<p>Select to place a condition on a stop to determine whether the route visits the stop. Refer to <a href="#">the Custom Sequence Entries table</a> for details about this field.</p> <p>This option does not appear if you can use the route with all record types.</p>
------------------------	--

### 3.1.12.5.3.1 Adding Stop Members

You may add stop members on the **Stops** tab of the route. You may add static members and dynamic members to a stop, depending on your needs.

**Important:** Make sure that there is at least one active user in a route that is being referenced by an approval rule. If no users are identified in a route, the route behaves as if everyone in the route approved the operation. The operation will be completed.

Adding dynamic stop members requires an understanding of Object Navigator. For details, see [Using Object Navigator](#).

## Member Types

The Member Type determines how users are included in a route stop as members: as members of a static user group, or as dynamically selected members. The following table explains the three member types.

### Stop Member Types in Routes

Member type	Description	Example
<p><b>Group</b></p>	<ul style="list-style-type: none"> <li>Allows you to select a specific group to add as a stop member. The request is sent to all active users who are part of the group.</li> <li>This type of stop member is static. It identifies one group by name.</li> </ul>	<p>The group Senior Claim Officers must approve posting of invoice payments over \$100,000.</p>
<p><b>User path</b> (Not available in routes that are specified for <b>All</b> objects in the route's <b>General</b> tab.)</p>	<ul style="list-style-type: none"> <li>Allows you to add a stop member based on how they are associated to the object record. For example, a user path could identify the user who created a record, the user who is the main assignee, and so on.</li> <li>This type of stop member is dynamic. It identifies a user (or, in some cases, multiple users) based</li> </ul>	<p>Posting of invoice payments over \$100,000 must be approved by the user who is the main assignee of the claim that is identified in the line item of the invoice.</p> <p>In this example, if more than one project is listed in the line items of the invoice, then the approval goes to the main assignee of each project.</p>

	<p>on the user's relationship to the record.</p> <ul style="list-style-type: none"> <li>• A user who is not active does not receive the approval request.</li> </ul>	
<p><b>User with role</b> (Not available in routes that are specified for <b>All</b> objects in the route's <b>General</b> tab.)</p>	<ul style="list-style-type: none"> <li>• Allows you to add a stop member based on the user's assignee role in a project.</li> <li>• If a project has more than one assignee with the same role, the approval request is sent to each user who has the selected role.</li> <li>• This type of stop member is dynamic. It identifies a user (or, in some cases, multiple users) based on the user's relationship to the record.</li> <li>• An assignee who is not active does not receive the approval request.</li> <li>• An assignee whose underlying user is inactive does not receive the request.</li> </ul>	<p>Posting of invoice payments over \$100,000 must be approved by the user who is the Supervisor of the claim.</p> <p>In this example, if the claim record has more than one supervisor, then the approval goes to each supervisor.</p>

The steps necessary to add members to a stop are different for each member type. The following are step-by-step instructions to help guide you through adding stop members:

- [Adding Groups as Stop Members](#)
- [Adding User Paths as Stop Members](#)
- [Adding Users With Roles as Stop Members](#)

You may add user groups to route stops. If any or all members of a certain group must always approve certain actions, then adding the group as a member of the route is a good idea.

Even though a group is a static stop member type, the route automatically uses only the current, active members of the group in the approval process.

	Member Type	Member
1	Group	Senior Claim Officers
		<a href="#">+ add more</a>

**Groups as Route Stop Members**

#### To add a group as a stop member

1. Click the **Stops** tab of the route to which you want to add a group as a member.
2. From the **Member Type** drop-down list, select **Group**.

3. From the list of groups, select the appropriate group.
4. If appropriate, select the check-box requiring approval for all members.  
This requires approval for all members of the group and all other members of the stop, if there are other members.
5. Click **add more** to continue adding stop members, or click **Save** to save the route.

A **User path** stop member identifies a dynamic user. The user who receives the approval depends on which user is identified by the path. This stop member type is only available if a specific object is identified for the route on its **General** tab.

#### To add a user path as a stop member

1. Click the **Stops** tab of the route to which you want to add a user path as a stop member.
2. From the **Member Type** drop-down list, select **User path**.

	Member Type	Member	
1	User path	null   jinetItemList.project.mainAssignee.us	<input type="button" value="reset"/> <input type="button" value="+ add more"/>

User Paths as Route Stop Members

3. Click the **Navigator** icon and create a path using Object Navigator.
4. If appropriate, select the check-box to require all members of the stop to approve the requested operation.  
This requires approval from all members of the stop, including all users who are identified by this path, as well as other members of the stop.
5. Click **add more** to continue adding stop members, or click **Save** to save the route.

When adding a user path as a member of a stop, keep in mind that if there is no user identified by the path, then the member is not added to the stop. For example, if you want the route to go to the Main Assignee of a project and this is the only member of the stop, then the stop is skipped if there is no Main Assignee, and the route automatically proceeds to the next stop.

When creating a path to a user for a member of a stop, the path must end with an attribute in the object model that identifies a user. This means that your final selection in Object Navigator must be one of the following:

- user->
- createdBy->
- modifiedBy->
- expensedBy->
- enteredBy->
- transitionedBy->
- Another attribute that identifies a user.

**Important:** Only users may be members of a route. Do not select an attribute that identifies a contact as your final selection for a user path.

The following table provides sample user paths created for a stop member.

#### User Path Examples for Route Stops

User identified by this path	What to select in Object Navigator to build this path
The user who created the record.	createdBy->(ok)
The user who is the Main Assignee of the project for which you are creating a route.	mainAssignee->(traverse) user->(ok)
The user who is the Main Assignee of the project to which the current object for which you are creating a route (such as Account, Expense, Task, etc.) is a related object.	project->(traverse) mainAssignee->(traverse) user->(ok)
The user who is the main assignee of the project indicated in the invoice's line items (for a route created for the Invoice object).	lineItemList->(traverse) project->(traverse) mainAssignee->(traverse) user->(ok)
The user who entered or posted the expense (for a route created for the Expense object).	expensedBy->(ok)

A **User with role** stop member identifies a dynamic user who is an assignee of a project. The user who receives the approval depends on which user is identified by the Assignee role indicated in the stop. This stop member type is only available if one specific object is identified for the route in its **General** tab.

#### To add a user with role as a stop member

1. Click the **Stops** tab of the route to which you want to add a user with role as a stop member.
2. From the **Member Type** drop-down list, select **User with role**.

Member Type	Member
1 User with role	: Supervisor


In null lineItemList.project

reset

+ add more

#### Users with Specific Roles as Route Stop Members

3. Select the appropriate role in the second field.
4. Do one of the following actions:

- If the role you selected belongs to the object for which you are creating the route, you do not need to create a path. Instead, continue to the following step.
- If the role you selected belongs to a different object than the object for which you are creating the route, click the **Navigator**  icon and create a path to the necessary project using Object Navigator.

For example, if you want the stop member to be the user with the role GC Attorney in the parent project of the project for which you are creating the route, select parent-> in Object Navigator.

5. Click **add more** to continue adding stop members, or click **Save** to save the route.

When you create a user with role stop member, follow these guidelines:

- If you are creating a user with role stop member and the Assignee role exists within the object for which you are creating the route, you do not need to create a path to identify the role.
- If the role belongs to a different object, you must create a path that indicates which object the role is from, in relation to the object for which you are creating the route, as shown in the examples in [the User with Role Examples for Route Stops table](#).

When creating a path to a user with role, the path must end with an attribute in the object model that identifies a project (custom object). This means that your final selection in Object Navigator must be one of the following:

- project->
- parent->
- applProject->
- leftProject-> (this is not common in routes)
- rightProject-> (this is not common in routes)
- Another attribute that identifies a project.

**Important:** To build a path that identifies the PROJECT in which the user has the specified assignee role, do not select an end-of-path attribute that identifies the user.

The following table provides sample **User with role** paths for stop members.

**User with Role Examples for Route Stops**

Project identified by this path	Example	What to select in Object Navigator to build this path
The parent project of the current project.  <i>Note: This path may only be used for custom objects that</i>	When creating a route for Claim (a custom object):  If Policy is the parent object of Claim, this would send the approval request to assignees of the policy	parent-> (ok)

<i>have parent-child relationships.</i>	record that is the parent of the claim record.	
The project to which the current object (such as Task or Expense) is related.	When creating a route for Task, Expense, or other system object that may be related to a project record:  This would send the approval request to assignees of the project that is selected in the <b>General</b> information for the record.	project->(ok)
The project to which an account belongs.	When creating a route for account:  This would send the approval request to assignees of the project that is selected in the <b>General</b> information for the account.	applProject->(ok)
The project referenced on a line item of an Invoice. If more than one project is listed, then the approval goes to each user who has the selected role in each of the projects.	When creating a route for invoice:  This would send the approval request to assignees of the projects that are referenced by each line item on the invoice. If all line items in the invoice reference the same project record, then only the assignees of that project receive the approval request.	lineItemList->(traverse) project->(ok)
The parent of the parent project of the current project.	When creating a route for litigation (a custom object):  In a litigation record that has a parent claim, which in turn has a parent policy, this would send the approval request to the assignees of the policy record.	parent->(traverse) parent->(ok)

When adding a user with role as a member of a stop, keep in mind that if there is no user identified by the path, or if the user is inactive, then the member is not added to the stop. For example, if you want the route to go to the General Counsel assignee of a project and this is the only member of the stop, then the stop is skipped if there is no such assignee, and the route automatically proceeds to the next stop. If there are no further stops, then the request is approved.

**Tip:** You may add all project assignees as members of a stop by adding a **User with role** member for each role that is possible for the project. TeamConnect automatically routes the approval request to all users who are assignees of the project.



## 3.1.12.5.3.2 Customizing Route Stops

When you add stops to a route, records visit each stop by default. If you want a record to visit stops when certain conditions are present, you can customize the route.

Select **Custom Sequence** on the **Stops** tab of a route to [enter the custom sequence](#). The following table describes the fields for entering a custom sequence.

**Custom Sequence Entries**

Column Name	Description of Fields	How Fields Work
<b>Starting Stop</b>	<p>Specifies where the route starts or the last stop that just occurred. You can select the following fields:</p> <ul style="list-style-type: none"> <li>• <b>Start</b>—To indicate the condition that a route evaluates at the beginning of the workflow process.</li> <li>• <b>Stops in route</b>—To indicate the condition that a route evaluates after the specified stop is complete.</li> </ul>	<p>If more than one entry has the same field in the first column, the route evaluates the first of those entries. The route continues to each entry with the same field under <b>Starting Stop</b> until a condition is true. If no conditions are true, the workflow process ends.</p> <p><i>Note: When a workflow process ends, the action that put the record into workflow occurs.</i></p>
<b>Condition</b>	<p>Specifies the conditions available for the record type of the route. You can select the following fields:</p> <ul style="list-style-type: none"> <li>• <b>Default</b>—To indicate that a route does not have to evaluate a condition before visiting the <b>Destination Stop</b>.</li> <li>• <b>Condition name</b>—To indicate the condition that must be true for the route to visit the <b>Destination Stop</b>.</li> </ul>	<p>When evaluating a condition, the route visits the <b>Destination Stop</b> for the conditions that are true.</p> <p>See <a href="#">Conditional Expressions</a> for more information about conditions.</p>
<b>Destination Stop</b>	<p>Specifies the stop that the route visits if the condition is true. You can select the following fields:</p> <ul style="list-style-type: none"> <li>• <b>Stops in route</b>—To indicate the stop the route visits if the condition is true.</li> <li>• <b>End</b>—To indicate that the workflow process ends if the condition is true.</li> </ul>	<p>For each entry, the <b>Destination Stop</b> number must come after the <b>Starting Stop</b> number.</p>

In the following image, consider an example of how the route evaluates each entry for a record. In this example, the record visits only Stop 2 and Stop 4 before completing. **Invoice Condition3** and **Invoice Condition4** are true for the record and all other conditions are false.

Custom sequence  
[Go to Condition List](#)  
 Number of entries you would like to add:

	Starting Stop	Condition	Destination Stop
1	<input type="text" value="Start"/>	<input type="text" value="Default"/>	<input type="text" value="End"/>
<a href="#">+ add more</a>			
	Starting Stop	Condition	Destination Stop
1	<input type="checkbox"/> Start	<a href="#">Invoice Condition1</a>	Stop 1
2	<input type="checkbox"/> Stop 2	<a href="#">Invoice Condition2</a>	End
3	<input type="checkbox"/> Start	<a href="#">Invoice Condition3</a> <b>(true)</b>	Stop 2
4	<input type="checkbox"/> Start	<a href="#">Invoice Condition4</a> <b>(true)</b>	Stop 3
5	<input type="checkbox"/> Stop 2	Default <b>(true)</b>	Stop 4
6	<input type="checkbox"/> Stop 4	<a href="#">Invoice Condition5</a>	End
<a href="#">Check All</a> - <a href="#">Uncheck All</a> <a href="#">delete</a>			

**Customization Sequence for a Route**

The following actions occur for the example:

- From the first entry with **Start** as the **Starting Stop** (line 1), the route evaluates **Invoice Condition1**. Because **Invoice Condition1** is not true, the route continues to the next **Start** entry (line 3).
- From the second entry with **Start** as the **Starting Stop** (line 3), the route evaluates **Invoice Condition3**. Because **Invoice Condition3** is true, the route visits Stop 2.

*Note: Even though **Invoice Condition4** is true, the route does not evaluate the condition because the previous **Start** entry is true.*

- After Stop 2 is complete, from the first **Stop 2** entry (line 2), the route evaluates **Invoice Condition2**. Because **Invoice Condition2** is not true, the route continues to the next **Stop 2** entry (line 5).
- The second **Stop 2** entry (line 5) lists **Default** as the **Condition**, which means that the route visits Stop 4 without evaluating a condition.
- After Stop 4 is complete, from the only **Stop 4** entry (line 6), the route evaluates **Invoice Condition5**. Even though **Invoice Condition5** is not true, the workflow process still ends and can no longer send requests to approvers because no more entries with **Stop 4** exist in the **Starting Stop** column.

### To customize the sequence conditions

1. From the route, click the **Stops** tab.
2. Add all stops to the page. See [Adding Stop Members](#) for information about adding stops.
3. Select the **Custom Sequence** option, as shown in the [Customization Sequence for a Route](#) image.
4. From the **Starting Stop** drop-down list, select **Start** or a stop. **Start** indicates which entry the route evaluates first before visiting a stop. A stop indicates which entry the route evaluates after that stop is complete.
5. From the **Condition** drop-down list, select the condition that must be true for the record to continue to the **Destination Stop**.
6. From the **Destination Stop** drop-down list, select the stop that the route visits if the condition is true.
7. Click **add more** to save the entry.
8. Add additional conditions if necessary.
9. Save the route.

#### 3.1.12.5.4 Adding Email Notification Recipients

You may configure email notifications to dynamically selected users or groups for the following events related to a pending approval request:

- Final approval
- Final rejection
- Error

#### To add a user or group as an email notification recipient

1. Click the **Email Notifications** tab of the route to which you want to add an email notification recipient.

**Email Notifications**

Number of entries you would like to add:

Recipient Type	Recipient	Event
1 (Select) ▼	(Select) ▼	(Select) ▼

Recipient Type	Recipient	Event
1 <input type="checkbox"/> User	<a href="#">Clouds, Steve</a>	Final Approval

[Check All](#) - [Uncheck All](#)

Email Notifications Tab

2. From the **Number of entries you want to add** drop-down list, select the number of recipients to add.
3. From the **Recipient Type** drop-down list, select **Group**, **User Path**, or **User with Role**.
4. From the resulting **Recipient** drop-down list, select either the user name or group name.
5. From the **Event** drop-down list, select one of the following:
  - Final approval
  - Final rejection
  - Error
6. Repeat steps 3-5 for adding multiple recipients.
7. Click **add more**.
8. Click **Save**.

**Note:** When a user is the only recipient of an email notification, it is sent in the user's preferred language. When there are multiple recipients, the email is sent in the selected system language.

The following table describes the items in the **Email Notifications** tab.

Email Notifications Tab

Field or button	Description
<b>Number of entries you want to add</b>	Select the number of users or groups to add to the email notifications recipients list.

<b>Recipient Type</b>	Select <b>Group</b> , <b>User Path</b> , or <b>User With Role</b> .
<b>Recipient</b>	Select either the group name or a dynamic user. Make this selection using the steps previously described in: <ul style="list-style-type: none"> <li>• <a href="#">Adding Groups as Stop Members</a></li> <li>• <a href="#">Adding User Paths as Stop Members</a></li> <li>• <a href="#">Adding Users With Roles as Stop Members</a></li> </ul>
<b>Event</b>	Select one of the following: <ul style="list-style-type: none"> <li>• Final approval—Sends an email notification when a request has been approved by the final approver/stop members.</li> <li>• Final rejection—Sends an email notification when a request has been rejected at the first stop</li> <li>• Error—Sends an email notification.</li> </ul>
<b>Email notification recipients list</b>	Displays the current route's list of email notification recipients.
<b>delete</b>	Select the current email notification recipient box(es) and click the <b>delete</b> button to remove that recipient from the notifications list.

**Important:** To set [email notifications](#) for users involved with the approval process as well as users with no direct role in the approval process, email notifications must be properly configured.

### 3.1.12.6 Requirements for Rule Specifications

Before writing your custom code, prepare the following information:

- A description of the system design with all object definitions, their custom fields, blocks, object views, and so on.
- A complete list of user groups and their rights.
- A complete list of business rules that need to be implemented.

For each rule in the rule specification documents, a minimum of the following requirements must be specified:

- Object definition for which the rule must be created and defined.
- Rule description/title that will be used when defining the rule.
- Rule type (for a complete list, see [the Rule Types table](#)).

- Trigger (for a complete list, see [the Rule Triggers table](#)).
- Qualifier/condition:
  - Whether it is done through the user interface or API.
  - If through the API, are you choosing Java or JavaScript?
  - Whether it has parameters or not. If, yes, which ones?
- Action:
  - Whether it denies, approves, or performs a custom action.
  - If the action is to deny, the message that appears to the user.
  - If the action is custom, the use of Java or JavaScript.
  - Whether it has parameters or not. If, yes, the parameters needed.
- List of fields used in the rule, with their names (not labels), full tree positions (for custom fields), full lookup item tree positions (for custom fields of type List), full category tree positions under which custom fields are created.

Consider the following when reviewing rule specifications:

- The rule does not duplicate or replace rights set in the user or group accounts (functional level security).

For example, if a rule is to prevent a user from creating, deleting, or updating records of a specific object definition, determine whether this may be done through the rights set at the user or group account level. You may also use record-level security to assign rights to users.
- Whether the indicated rule type and action match.

For example, if the action is to deny and display a message, use a validation rule instead.
- Whether the trigger is identified properly.

This is important because the exact appropriate behavior must be identified. For example, in rules, there is no such trigger as Save. Instead, there are Create and Update triggers that are associated with the **Save** button in the user interface. The list of triggers is provided in [the Rule Triggers table](#).
- Whether the specified trigger is available for the indicated rule type.

This needs to be taken into consideration for the rule types other than custom or scheduled actions. For example, you cannot create approval rules for the Update trigger or the Check-In trigger in documents.
- Whether the rule may be created through the user interface.

For example, just because a rule is of type custom action, it does not necessarily mean that its qualifier must be automated as well. Simple qualifiers, such as checking a value in a field or a user's group membership, may often be created through the user interface.
- Whether any automated actions and qualifiers may be reused. Indicate which ones and for which rules.

For example, several rules may have to perform the same action but under different conditions, or vice versa, perform different actions under the same condition. Always keep in mind the

concept of separating qualifiers and actions in a rule that allows you to mix-and-match files to minimize duplication of effort and to maximize the flexibility of the rule development process.

- Whether there are any complementary rules, rules with the same qualifier and action but a different trigger. Indicate which ones.

For example, two rules may need to be created to enforce the workflow process when the **Save** button is clicked—one rule with the Create trigger and the other with the Update trigger.

- Whether any additional or complementary rules are needed to enforce the workflow process. Indicate which ones.

For example, you may need to break a more complicated rule into several rules, or a rule with the Update trigger may be required to complement the specified rule with the Create trigger.

- Whether there is any missing design information that you would need in the rule.

For example, categories' full tree positions may not be provided for custom fields, or lookup items' full tree positions for List fields or sub-objects, and so on.

## Rule Component Separation

Depending on the rule type and business requirements, either component may be defined through the user interface or written in a separate Java or JavaScript file (for details on which rule types may have automated qualifiers and actions, see [the Rule Types table](#)). In either case, strive to maintain the separation between the two components, especially when writing automated qualifiers and actions.

Writing each rule component in a separate file allows you to mix and match qualifiers with actions in different rules by reusing files. For example, the same component files may be used for two rules, one with the Create trigger and the other with the Update trigger. This division between the qualifier and the action is also useful in rules when one of the components is defined through the user interface and the other is created through the API.

There is nothing to stop you from placing all of your rule code in a condition to enable multi-stage interactions between condition requirements and actions taken, and having your action file do nothing. However, if you want to maintain an object-unspecific rule set, and be able to mix-and-match qualifiers and actions at will, it is advisable to keep your qualifiers and your actions granular and separate.

## Legacy Rules

Rules for TeamConnect versions preceding version 1.6 were written as a single file, with no division between qualifiers and actions. These rules are now called legacy rules, and have a different method of inclusion into TeamConnect from rules for the TeamConnect 2.x architecture. Inclusion methods for legacy rules will be covered in the future versions of this document.

### 3.1.13 Conditional Expressions

Conditions contain one or more expressions that point to records or other parts of TeamConnect. Conditions also contain a setting for the combination of expressions. Depending on this setting, the condition is true or false.

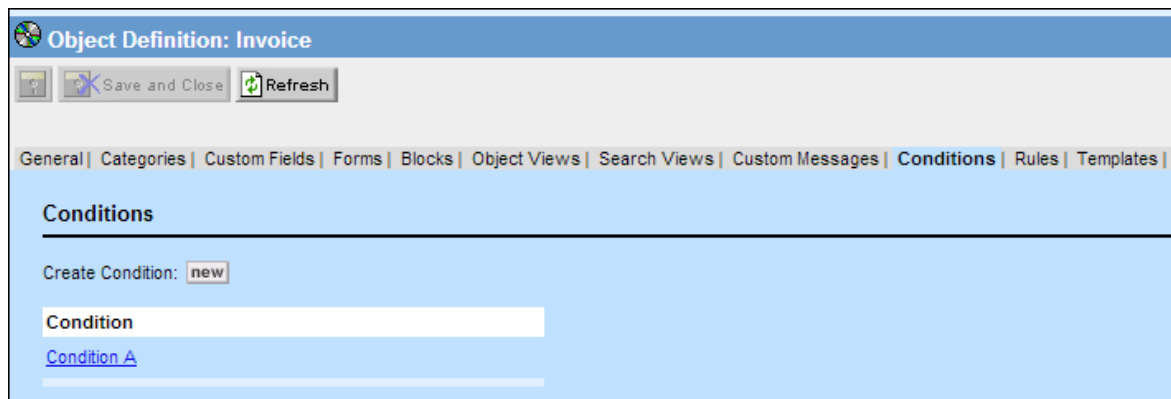
## Use Case for Conditions

Use a condition to base the stops in a route on these expressions.

- You can create a route so that different users can approve invoices, depending on the total amount on the invoice. For example, you can have a different user approve each of the following situations:
  - Invoices amounts less than or equal to \$1,000.
  - Invoices amounts greater than \$1,000 and less than or equal to \$5,000
  - Invoices amounts greater than \$5,000.
- You can create a route so that if users try to delete a record that requires approval for deletion, different groups can approve the deletion based on different fields in the record.

## Finding Conditions

Each object definition has its own set of conditions. To find conditions for an object definition, click the **Conditions** tab from the object definition.



Conditions Tab

### 3.1.13.1 Creating a Condition

You create conditions from the **Conditions** tab of a object definition.

#### To create a condition

1. From the **Conditions** tab for an object definition, click **New**.
2. Enter information about the condition. See [the General Tab of a Condition table](#) for a description of each field.
3. Click **Save**.


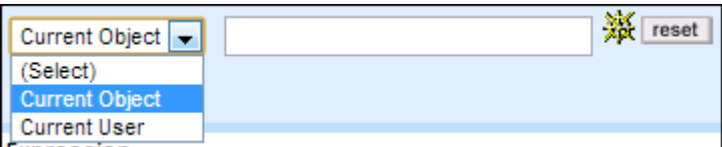
**Note:** Click the **Delete** button of an open condition to delete it.



### To create a copy of a condition

1. From the condition, click **Create a Copy**.
2. Change the **Name** and the **Unique Key** of the condition.
3. Click **Save**.

#### General Tab of a Condition

Field or Column	Description
<b>Name</b>	The name of the condition. The name you enter is also the name that appears in any drop-down list of conditions.
<b>Unique Key</b>	The unique name of the condition.
<b>Description</b>	Additional information about the condition.
<b>Expression Type</b>	<ul style="list-style-type: none"> <li>• <b>Automated Qualifier</b>—Select this option to add a Java class or JavaScript file as an expression for the condition.</li> <li>• <b>User Expression</b>—Select this option to add an expression with TeamConnect fields to the condition.</li> </ul>
<b>Expression</b> (when <b>Automated Qualifier</b> is selected)	<p>A drop-down list with Java class and JavaScript files that you can select for this condition. Click the <b>Automated Qualifiers Folder</b> link to open the folder with Java classes and JavaScript files.</p> <p>For examples of when to create automated qualifier, see <a href="#">When to Create Automated Qualifiers</a>.</p>
<b>Expression</b> (when <b>User Expression</b> is selected)	<p>The following fields appear:</p> <ul style="list-style-type: none"> <li>• <b>Left Argument</b>—Select the path you want to create for the condition. <ul style="list-style-type: none"> <li>○ Select <b>Current Object</b> if you want to use a field in TeamConnect as the qualifier, for example <b>Current Phase</b>.</li> <li>○ Select <b>Current User</b> if you want to compare the identity of each user with a field that identifies a user.</li> </ul> </li> </ul> <p>Click the  icon to <a href="#">open the Object Navigator</a> and select the path to the field.</p>  <p style="text-align: center;"><b>Left Argument</b></p>

	<ul style="list-style-type: none"> <li>• <b>Operator</b>—Select the appropriate way to connect the right and left arguments. See <a href="#">the Operator Fields for Creating Expressions table</a> for more information about operators.           <div data-bbox="614 324 901 392" style="border: 1px solid black; padding: 2px; margin: 5px 0;"> <span style="border: 1px solid black; padding: 2px;">Is</span> <span style="border: 1px solid black; padding: 2px;">Equal To</span> </div> <p style="text-align: center; margin: 0;">Operator</p> </li> <li>• <b>Right Argument</b>—Select the attribute or literal value that you want to compare to the attribute that you selected for the left argument. See <a href="#">the Rule Qualifier Right Argument Options table</a> for more descriptions of each field.           <div data-bbox="614 582 877 795" style="border: 1px solid black; padding: 2px; margin: 5px 0;"> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;"> <span style="border: 1px solid black; padding: 2px;">Literal</span> </div> <div style="border: 1px solid black; padding: 2px;">           User1, User1            User2, User2            User3, User3            User4, User4         </div> </div> <p style="text-align: center; margin: 0;">Right Argument</p> </li> </ul>
<b>add more</b>	Click this button to add the expression to the condition.
Expressions table	This table lists all the expressions that you have added to the condition. Place a check-mark in the check-box of an expression and click <b>edit</b> to edit the expression. Place a check-mark in one or more check-boxes and click <b>delete</b> to delete the expression(s).
<b>Condition is true when</b>	<p>Select one of these options to determine which expression in the condition must be met:</p> <ul style="list-style-type: none"> <li>• <b>All of the above expressions are met (AND logic)</b>—Select this option for all expressions to apply to the condition.</li> <li>• <b>Any of the above expressions are met (OR logic)</b>—Select this option for one of the expressions to apply to the condition.</li> <li>• <b>The following expressions of the above conditions is met</b>—Enter a formula for the combination of expressions. Use the number of the expression to refer to the expression. See <a href="#">Points To Remember</a> for information about how to create this combination. Click <b>View Qualifier</b> to display the qualifier logic visually.</li> </ul>

#### 3.1.13.1.1 Defining Expressions

To define an expression, you specify the following:

- The left argument
- The operator
- The right argument

These three pieces together form a statement, such as these:

#### Sample Expressions

Left Argument	Operator	Right Argument
The value selected in the Loss Type list field	is	Fire Damage
The current user	is not	a member of the Outside Counsel group
The Trial Date	is changed	from any date to any date
The Expiration Date field	is populated	[none]

If you add these statements as expressions, the condition evaluates the statements to determine whether the condition is true.

The type of data being compared in the left and right arguments must match. For example, if your left argument identifies a list field, then your operator and right argument options allow you to compare the selection with a selection made in another list field or to a literal list value.

## Operator Fields

The operator connects the rights and left arguments. Refer to the following table for a description of each operator field.

#### Operator Fields for Creating Expressions

Selecti on	Description	Data types where operator selection is available					
		Text	Num ber	Date	List	Check- box	Related object
<b>Is</b>	Determines whether the qualifier statement is a positive or negative statement.	x	x	x	x	x	x
<b>Not</b>		x	x	x	x	x	x
<b>Equal To</b>	Evaluates whether the left argument and right argument are the same value.	x	x	x	x	x	x
<b>Popula ted</b>	Evaluates whether the field identified in the left argument has a value.	x	x	x	x		x

	Do not use this selection to check whether a field that was previously null now has a value. Instead, use <b>Changed from No Value</b> .						
<b>Changed</b>	Evaluates whether the value for the field identified in the left argument has changed. It compares the old value in the database to the value in the record when rule is triggered.  <i>Tip: This option is useful for triggering audit rules.</i>	x	x	x	x	x	x
<b>Begins With</b>	Evaluates whether the text value in the field identified in the left argument begins with, ends with, or contains the text that is identified in the right argument.	x					
<b>Ends With</b>		x					
<b>Contains</b>		x					
<b>Less Than</b>	Evaluates the number value identified in the left argument compared to the number value in the right argument.  For example, you may specify that if a dollar amount <b>Is—Greater Than Or Equal To</b> 5000, send the action for approval.		x				
<b>Less Than Or Equal To</b>			x				
<b>Greater Than Or Equal To</b>			x				
<b>Greater Than</b>			x				

<b>Exists</b>	These options appear when you select a particular category by creating a path such as <code>.detailList.(Motions)</code> . Evaluates whether the category has been added to the record.						
<b>Does Not Exist</b>							
<b>Item Added</b>	Evaluates whether a related object (such as involved) or sub-object (such as assignee) was added, deleted, or modified.				<b>x</b>		
<b>Item Deleted</b>					<b>x</b>		
<b>Item Modified</b>					<b>x</b>		
<b>After</b>	Compares the date in the left argument to the date in the right argument.  An additional field appears when you select one of the options that requires you to specify a number of days. Enter the value for <b>X</b> in this field.			<b>x</b>			
<b>Before</b>				<b>x</b>			
<b>After The Next X Days</b>				<b>x</b>			
<b>Within The Next X Days</b>				<b>x</b>			
<b>Within The Last X Days</b>				<b>x</b>			
<b>Before The Last X Days</b>				<b>x</b>			

**Right Argument Fields**

The types of right arguments you can add depend on the data type of the left argument.

#### Rule Qualifier Right Argument Options

Select ion	Description	Data types for which selection is available					
		Text	Number	Date	List	Check-box	Custom Object/Involved
<b>Literal</b>	<p>Select this option when you want to specify a value to compare to the field identified by your left argument.</p> <p>For example, select <b>Literal</b> for a qualifier item that checks whether the invoice total is greater than 5000.</p> <p>You must also select <b>Literal</b> when you want to check whether a certain item is selected in a List field.</p>	x	x		x	x	x
<b>Attribute</b>	<p>Select this option when you need to compare the values of two fields in TeamConnect.</p> <p>After you select <b>Attribute</b>, you must first select <b>Current Object</b> or <b>Current User</b>, and then use Object Navigator to identify the field to which you want to compare the field identified by your left argument.</p> <p>For example, if you want to compare the Contact of a contact-centric project to the Contact identified in a custom field of type</p>	x	x	x	x	x	x

	Involved, select <b>Attribute</b> to identify the custom field.						
<b>Current Object</b>	Select one of these options after you have selected <b>Attribute</b> and you need to identify a field in TeamConnect to which you are comparing the field in your left argument.	x	x	x	x	x	x
<b>Current User</b>	For details about using Object Navigator, see <a href="#">Using Object Navigator</a> .	x	x	x	x	x	x
<b>Any Value</b>	These options work in conjunction with the <b>Changed</b> option to form a statement, such as <b>Is changed from any value to this value (California)</b> .	x	x	x	x	x	x
<b>No Value</b>		x	x	x	x		x
<b>This Value</b>		x	x		x		x
<b>Today</b>	The date that the rule is being triggered.			x			
<b>This Date</b>	Allows you to compare the date in the left argument to a date that is identified in another field.			x			
<b>From Now</b>	Allows you to specify a number of days from the date that the rule is being triggered.			x			
<b>From This Date</b>	Allows you to specify a number of days from a date that is identified in another field.			x			
<b>Yes</b>	These options correspond to a check-					x	

No	box being selected or not selected.					x	
----	-------------------------------------	--	--	--	--	---	--

### 3.1.13.1.2 Qualifier Logic

The **Designer** user interface is designed for you to complete the condition logic in stages. After listing one or more expressions on the **Qualifier** tab, the following options appear, allowing you to specify when the rule action runs:

- All of the expressions are met.
- Any of the expressions are met.
- The expressions are met according to the logic that you specify.

The "logic" option allows you to specify a formula for how the condition evaluates the expressions. You refer to an expression using the number that is automatically generated on the batch screen for each item. For example, once you create a list of five expressions, you might create the following formula:

```
(1 or 2 or 3) and (4 or 5)
```

### Points To Remember

Keep the following points in mind while working with qualifier logic:

- Define all of the expressions that you need in your condition before writing the formula.
- The only words that you may use in your formula are "and" and "or."
- Use parentheses () to logically group expressions. You may also nest parentheses when needed. Make sure to use complete pairs of parentheses.
- If you do not group the expressions where grouping is necessary, grouping is automatically applied by TeamConnect, and you will have to verify that the appropriate logic is achieved. For example:

```
1 or 2 and 3
```

is automatically converted to:

```
1 or (2 and 3)
```

It is best to apply the grouping yourself to ensure that it is correct.

- If necessary, your formula may repeat the same expression. For example, the following formula is valid:

```
(1 and 2 and 3) or (1 and 2 and 4)
```

### 3.1.13.2 Finding Routes that Use a Condition

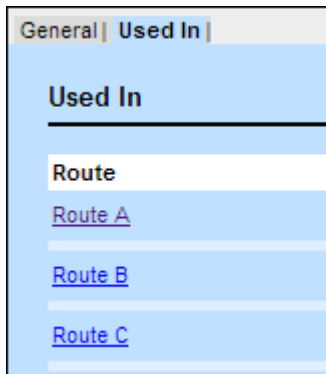
From an existing condition, you can find out whether one or multiple routes already use that condition.

#### To view routes that use a condition



1. From the condition, click the **Used In** tab.

You see a list of routes that use the condition.



Used In Page of a Route

2. Click the link for a route to find out more about how it uses the condition.

### 3.1.14 Adding Custom Code to TeamConnect

Once you have written and compiled your automated qualifier and/or automated action Java classes or finished writing your JavaScript and XML files, you must include them in TeamConnect. To do this, you need to upload the files to [the appropriate folders in your system's Documents area](#), and define the rules.

The following sections describe how to include automated qualifiers and automated actions in TeamConnect:

- [Uploading Rule Component Files](#)
- [Defining Rules that Use Custom Code in the User Interface](#)
- [Defining Wizard Page Transition Rules](#)
- [Defining Wizard Page Actions](#)

For information on defining non-automated qualifiers and actions through the user interface, see [Rule Qualifiers](#) and [Rule Actions](#).

#### 3.1.14.1 Uploading Rule Component Files

To include API rule components in TeamConnect, you must upload the corresponding Java, JavaScript, or XML rule component files to the TeamConnect Documents area. There are three major [folders in the Documents area](#) for such files:

- **Top Level/System/Custom Rules**—Upload only supporting Java files that are shared between more than one object definition. These files are not available in the **Qualifiers** or **Actions** tab of the Rule screen in the object definitions, nor in the **Page Actions** tab in wizard screens. These files may be only referenced by the files uploaded to the object definitions' **Rules** folder.
- **Top Level/System/Object Definitions/objectDefinitionName/Rules**—In each object definition, the **Rules** folder has three subfolders:

- **Automated Actions**—Upload your files for automated actions for custom action rules and wizard page actions here.
- **Automated Qualifiers**—Upload your files automated qualifiers for regular rules and wizard page transition rules here.

The **Automated Actions** and **Automated Qualifiers** folders are the two locations that you will most frequently use to upload your files.

- **Top Level/System/Libraries**—Allows you to deploy JAR files outside of the **TeamConnect.war** file or the **WEB-INF/lib** folder, which has the advantage of not requiring you to restart TeamConnect. This folder may be used for more than rule files.
- **Top Level/System/StartUp**—Allows you to deploy class files that must be executed when TeamConnect starts. This may be useful for installation scripts, to initialize TeamConnect using custom code, and to check consistency in the application for the session. For more details, see [Defining Start Up Classes](#).

***Important:** It is a good idea to limit access to the **StartUp** folder to prevent the addition of malicious code.*

## Ways of Accessing Rule Folders

You may either upload all your files to the respective object definitions before you start defining rules, or you may upload each component file during the definition process of each rule. Aside from your personal preference, this often depends on the way you choose to access the rule folders.

There are two ways of accessing the **Automated Actions** and **Automated Qualifiers** folders where you will typically upload your files:

- From the Documents area by selecting **Documents** in the **Go to** menu in Designer and then navigating to the necessary object definition folder.
- Directly from the **Qualifiers** or **Actions** tabs in the Rule screen by clicking the hyperlink that takes you to the appropriate **Automated Actions** or **Automated Qualifiers** folder of the displayed object definition. For example, see the hyperlink in the following image.

General | Qualifier | **Action**

**Rule Actions**

Use this Automated action: RuleDCreateActualsBudgetCASYS.class in [Automated Actions Folder](#)

How many days do you want between the event and the action? 0E-12

Do you want the action to be repeated?

Yes

No

How often will the action be repeated (in days)? 1.0000

How many times will the action be repeated?

Indefinitely


Number of Times 3

**Rule Screen - Actions Tab**

This ensures that the files are uploaded to the correct folder, as you do not have to manually navigate to it in the Documents screen.

## Uploading Techniques

Depending on the display version of the Documents screen, which may be either HTML or Applet, you may use one of the following ways to upload a file:

- Manually, by using the **Upload File**  button on the toolbar.  
You may do this in either version of the screen.
- By dragging and dropping the file(s) from any location on your machine.  
You may do this only in the Applet version of the screen.

For more specific details on the size you can upload a document in TeamConnect, see [Managing Documents](#).

### 3.1.14.2 Defining Start Up Classes

The **Top Level/System/StartUp** folder allows you to deploy class files that execute when TeamConnect starts. You can use this folder for installation scripts, to initialize TeamConnect using custom code, and to check consistency in the application. Your Java class files can use the TeamConnect's API, other libraries in TeamConnect's **Documents** area, and libraries in other locations.

***Important:** You may want to limit access to the **StartUp** folder to prevent the addition of malicious code.*

TeamConnect finishes its standard initialization before it executes the class files in the **StartUp** folder. Your custom code can affect TeamConnect's initialization after its standard initialization and before TeamConnect becomes available via a web browser.

***Note:** Custom code may not be compatible across TeamConnect releases.*

## Start Up Settings

Use the **web.xml** file to update settings for the **StartUp** folder.

To enable the execution of custom code in the **StartUp** folder, set the **app.runStartUp** application parameter to **YES** in the **web.xml** file.

To stop the execution of all class files in the **StartUp** folder in the event of any runtime errors or exceptions, set the **app.runStartUp.failOnError** parameter to **YES** in the **web.xml** file.

## How TeamConnect Uses Start Up Classes

By default, the [system user](#) executes the start up classes. TeamConnect executes classes in the **StartUp** folder under the following conditions:

- The files must be Java classes. TeamConnect ignores non-Java class files and any sub-folders in the **StartUp** folder, unless you reference them from the Java class. For example, you can include JAR files and reference them from class files.

- The Java classes must include implement the `ScheduledCustomAction` interface and the method of `public void action()` to hold the action to run. If you set the system logging level to **Info** or **Debug** and upload Java classes that do not implement the `ScheduledCustomAction` interface, TeamConnect logs the following message and continues to the next java class:

**ClassName does not implement TCStartupAction or ScheduledCustomAction. Not running ClassName.**

- In the alphabetical order of the file names if multiple class files are present in the **Startup** folder.

### 3.1.14.3 Defining Rules that Use Custom Code in the User Interface

For all rules that use custom code in some aspect, you must incorporate them into the TeamConnect application after the code has been developed. The following is a reminder of the steps for defining rules in TeamConnect:

1. Setting general rule information, as described in [Setting General Rule Information](#).
2. Defining the necessary action. See:
  - [Rule Actions](#)
  - [Setting Automated Actions](#)
3. Adding the necessary qualifiers. See:
  - [Creating Rule Qualifiers](#)
  - [Setting Automated Qualifiers](#)
4. Activating the rule. See [Enabling Rules](#).

For details on defining Page Transition rules in wizards, see [Defining Wizard Page Transition Rules](#).

#### To set an automated action

**Note:** Automated actions may only be set for rules of type Custom Action.

1. In the rule screen, click the **Actions** tab.
2. In the **Use this Automated action** drop-down list, select the appropriate automated action file.

If you have not uploaded your automated action file, click the **Automated Actions Folder** link to proceed directly to the **Automated Actions** folder of the selected object definition. Upload the file, and it will become available for selection in the **Use this Automated Action** drop-down list.
3. If there are any parameters in the action, enter values in the provided fields.
4. Click **Save**.

General | Qualifier | **Action**

**Rule Actions**

Use this Automated action:  in [Automated Actions Folder](#)

How many days do you want between the event and the action?

Do you want the action to be repeated?

Yes

No

How often will the action be repeated (in days)?

How many times will the action be repeated?

Indefinitely

Number of Times

Rule Screen - Actions Tab

### To set an automated qualifier

1. In the rule screen, click the **Qualifiers** tab.
2. Click the Use **Automated Qualifier** radio button.
3. In the Use this **Automated Qualifier** drop-down list, select the appropriate automated qualifier file.

If you have not yet uploaded your automated qualifier file, click the **Automated Qualifiers Folder** link to proceed directly to the **Automated Qualifiers** folder of the selected object definition. Upload the file, and it will become available for selection in the Use this **Automated Qualifier** drop-down list.

4. If there are any parameters in the qualifier, enter values in the provided fields.
5. Click **Save**.

General | **Qualifier** | Action |

**Rule Qualifier**

Use Automated Qualifier file

Use Qualifier defined below

Use this Automated Qualifier:  in [Automated Qualifiers Folder](#)

Rule Screen - Qualifier Tab

### To enable your rule

1. In the **General** tab of the rule screen, select **This Rule is Active**.
2. If you have defined a user invoked rule, make sure to give the right to that rule to the appropriate users, through the appropriate [group accounts](#).

## User Invoked Rules

User invoked rules are, as the name implies, invoked directly by the user. When you define a user invoked rule for an object definition, the rule appears in a drop-down list when a user with the appropriate rights views a record or other user interface associated with that object definition. This operation uses the **More Actions** button in the toolbar, the same button that accesses user invoked actions (see [User Invoked Actions](#)).

### 3.1.14.4 Defining Wizard Page Transition Rules

#### To define a page transition rule in a wizard

1. Click the **Page Transitions** tab in the wizard.
2. Click **new** to open a new rule screen.
3. In the **General** tab, enter a description that clearly identifies the rule.
4. Click the **Action** tab and select the page to which the wizard takes the user if the qualifiers are met and click **Save**.

Wizard Page Transition rules do not have automated actions, and their actions only indicate to which page in the wizard to go next, when the qualifiers are met.

5. Click the **Qualifiers** tab and make the appropriate selections as explained in [Setting Automated Qualifiers](#).
6. Click **Save and Close**.

The rule becomes available in the **Rule** drop-down list in the **Page Transitions** tab of the wizard:

**Page Transition Logic**

Current Page:

This is a final page

By default the next page is: 25 - Policy Information

To jump to a different page under certain conditions, define a Page Transition Rule:

Order	Rule
1	<input type="text" value="If caller is 3rd Party"/> <input type="button" value="open"/> <input type="button" value="new"/>
2	<input type="checkbox"/> 1 <input type="text" value="If User is not Licensed Adjuster"/> Next Page: End of Wizard
2	<input type="checkbox"/> 2 <input type="text" value="If Caller is 3rd Party"/> Next Page: Accident Details - General <input type="button" value="ok"/> <input type="button" value="cancel"/>

[Check All](#) - [Uncheck All](#)

Example of Page Transitions Tab in Wizards

By default, all Wizard Page Transition rules are active. However, they do not take effect until you add them to a wizard page.

7. Select the rule you have defined and enter the order in which it must be triggered when the user clicks next on the wizard page.

8. In the **Current Page** drop-down list, select the page whose transition must trigger the rule and click **add more** or save the wizard.

### 3.1.14.5 Defining Wizard Page Actions

Currently, you cannot upload automated action files directly from the **Page Actions** tab in wizards. Make sure to upload your files to the **Rules\Automated Actions** folder under the appropriate object definition in the Documents area. For details, see [Uploading Rule Component Files](#).

#### To define an automated page action in a wizard

1. Click the **Page Actions** tab in the wizard.
2. In the **Current Page** drop-down list, select the page for which you need to define the automated action, and select the **Use Automated Actions** radio button.
3. In the **Use this Automated Action** drop-down list, select the appropriate automated action file.
4. If there are any parameters in the action, enter values in the provided fields.
5. Click **Save**.

### 3.1.15 Localizing TeamConnect

TeamConnect supports localization - the display of web pages in multiple languages and formatting styles.

The most important part of the localization infrastructure is the *i18n key*. This is a unique value assigned to each element that is subject to localization in the entire application. Examples of such elements are:

- Labels (of tabs, push buttons, fields, etc.)
- Lookup table entries
- Category names
- Messages
- Other static text on the page

**Note:** *User-entered data is not localized. For example, if an Appointment record is named "North Hills Partners Settlement Conference," that is the text that will be rendered to the Name field of every end user's Appointment web page, regardless of their locale. But the static label next to that field might say "Name" or "Nom" or "Nombre," depending upon the end user's locale.*

The first step in localization is to register all the locales that TeamConnect must support. For details, see [Working with Locales](#). After those locales are registered, you can export your application's design element resources.

All elements that have i18n keys can be exported to a spreadsheet. The sheet contains a column for the i18n key (Message Key), a column for Default Value (the static text, in the system default locale), and an additional column for each other locale that has been registered by the system administrator. To localize the application, enter static text in other languages in those additional columns, then import the edited spreadsheet to the application. For details of these procedures, see [Working with Locales](#).

Design elements that are already unique will have their i18n key generated automatically. For example, a custom field name (when combined with its object definition's unique code and its category) is already guaranteed to be unique. But some design element names are not required to be unique - for example, a report name. In cases like these, the solution developer must assign a value to a Unique Key field in that design element. The Unique Key will be used as part of the automatically generated i18n key. Unique Key fields are found in several places in Designer, as well as in report design pages. Design elements that have empty Unique Key fields will not be exported to the localization spreadsheet.

### 3.1.15.1 How are i18n Keys Generated?

For "out of the box" static text that is present in TeamConnect before any customization, all i18n keys are already present. Those keys are permanent and are not editable by you. These elements of static text become rows in the localization spreadsheet. Shown below are examples of spreadsheet entries to show the patterns used in generating such i18n keys.

**i18n key and default text examples**

Design element	Example i18n key and default text	Notes
<b>Static text</b>	account.accountProjectType Posting Project to this Account	
<b>Static text</b>	accountView.title Account - {0}	This message accepts a parameter - the name of the account. When localizing it is important to preserve parameter tokens.
<b>Block</b>	objdef.ACCT.block.Account_Children Child Accounts	
<b>Button text</b>	button.saveAndPreview Save & Preview	
<b>Enumeration</b>	enum.docucontent.26 Rich Text Format	Enumerations can be used to populate fixed drop-down lists, in this case for choosing the file type of a document.



<b>Enumeration</b>	enum.lineitem.adjustmenttarget.neta mount  Net Amount	
<b>Error message</b>	error.screendesign.tagUsage  Could not parse screen file due to improper use of a tag in file. Please check logs.	
<b>Error message</b>	common.typeMismatch  Value "{0}" entered for criterion "{1}" is invalid.	Parameter tokens must be preserved when localizing.
<b>Search view (criterion)</b>	objdef.ACCT.searchview.DefaultTem plate.criterion.Vendorr290c1  Vendor	Tokens like "r290c1" were used by prior versions of TeamConnect to control the positioning of a criterion on the filter page.
<b>Search view (result column)</b>	objdef.ACCT.searchview.DefaultTem plate.resultscolumn.Balance  Balance	
<b>System lookup table entry</b>	table.system.CURR.item.CAD  Canada, Dollars	
<b>Custom field</b>	objdef.CONT.customfield.Eligible  Eligibility	"Eligible" is field name. "Eligibility" is field label.

#### 3.1.15.1.1 Features that Require Your Attention

The preceding sections described TeamConnect features whose i18n keys are generated automatically. The sections below describe features where you can influence the generation of i18n keys for localization.

**Note:** Many of the features below use **Unique Key** fields. Characters entered in those fields must be ASCII letters and numbers.

#### 3.1.15.1.2 Reports

In order to localize report properties such as data series names, column headers, etc., there must exist an i18n key for each property. Since you can design brand-new reports that didn't exist in the out-of-the-box design, you need a way to assign i18n keys to the new reports' properties.

This need is addressed by the **Unique Key** fields that are found in the various report design pages. If you do not plan to localize a report, you may leave these fields empty. But if you need to localize the report, each of these fields in the report must have a value.

For details about localizing reports, see [Localizing Reports](#).

#### 3.1.15.1.3 Custom Messages

Custom messages are created using the **Custom Messages** page within **System Settings** in Designer (for global messages) or using the **Custom Messages** page within an object definition (for object-specific messages). They consist of an i18n key and default text. The i18n key takes two forms:

- **custom.common.someUniqueKey**—for global messages, where "someUniqueKey" is the only portion of the key that is actually entered by the solution developer.
- **custom.INVC.someUniqueKey**—for object-specific messages (object INVC, in this example), where "someUniqueKey" is the only portion of the key that is actually entered by the solution developer.

Custom messages are exported to the localization spreadsheet and you can supply alternate locales' text to the messages in that spreadsheet.

Custom messages can be used in custom actions to provide localized error messages.

Localized text for custom messages is available through the API methods: `localize()` and `localizeNumber()`.

#### 3.1.15.1.4 Custom Blocks

The static text of custom blocks can be localized during design by referencing custom messages rather than typing literal text.

The Screen Designer tool allows you to reference custom messages by key. If you are editing block XML files manually, you can use tags **tc:message** and **tc:messageParam** to achieve the same effect.

Text that is added to a block by entering literal text in the Screen Designer, or by using the **out** tag with literal text, will not be localized. It will always be rendered as it was originally entered.

#### 3.1.15.1.5 Portal Panes

The Designer page for **Portal Pane Settings** provides the opportunity to enter **Unique Key** values for the pane and its contents, which are converted to full i18n keys during export to the localization spreadsheet.

#### 3.1.15.1.6 Rules

Each rule has a **Unique Key** field that is converted to a full i18n key during export to the localization spreadsheet. Unlike some other design elements, the name of a user-invoked rule can be localized so that it appears in the user's locale in the **More Actions** drop-down list at runtime. There are rows in the spreadsheet related to security rules' message text and validation rules' message text, so those can be localized as well.

### 3.1.15.1.7 Wizards

Wizard design pages have **Unique Key** fields in several places, covering the wizard itself plus its text elements. Unique Key values are converted to full i18n keys during export to the localization spreadsheet.

### 3.1.15.1.8 Saved Searches

A saved search is one end user's copy of a search view, edited for the needs of that user and saved under a name that is assigned by the user. Saved searches do not have i18n keys, so there is no action to be taken by the solution developer with regard to localization. However, there is some localization behavior that should be understood. Under one uncommon and specific set of circumstances, a saved search can behave slightly differently from other features, with respect to localization.

The labels of criteria and result fields are always obtained, when possible, from the original search view that underlies the saved search. However, since the end user can remove results and criteria, and add brand-new results and criteria, there can be elements in the saved search that are not found in the original search view. In this case, one of two things will happen:

- If the element is a custom field, its label will be obtained from the object definition.
- If the element is not a custom field, the label used will be the one that was in existence at the time the search was saved. Even if that end user is now using a different locale, the label in the saved search will reflect the locale at the time the search was saved.

## 3.1.15.2 Upgrade Considerations for Localizing

If you upgrade TeamConnect from a version older than version 3.4, your application will be enabled for localization but, at first, only text in the system default locale (English-U.S.) will be available. The installer will take the following actions automatically:

- **System default locale**—Set to English (US).
- The **Region** block of User Preferences previously contained separate fields for country and language. That has been replaced by a **Locale** block containing a single drop-down list in which "English (United States)" is selected by default for all end users.
- Wizard list parameters that use comma delimited strings will still be supported for backward compatibility. But the new model for parameters of type list requires custom lookup tables, and all future editing of such parameters can only be done with custom lookup tables.

For localization to work properly, your database requires additional configuration beyond the basic requirements for TeamConnect. For details, see Database Requirements.

### 3.1.15.2.1 Custom Blocks

Significant work is required to make custom blocks ready for localization. Custom blocks are defined by XML files which, in previous versions of TeamConnect, will contain literal text such as:

```
<br/>  
    * No E-Billing Role Available  
<br/>
```

To localize such blocks, all occurrences of literal text must be converted to custom messages such as:

```
<br/>
  <tc:message
    key="custom.CSM$.screen.xml.eBillingRoles.noEbillingRoleAvailable" />
<br/>
```

...where the referenced message key should have been already added via the Custom Messages facility in the object definition for CSM\$. Such custom messages are visible in file **custommessages.xml**, after a design snapshot export, in the following format:

```
<CustomMessage
messageKey="custom.CSM$.screen.xml.eBillingRoles.noEbillingRoleAvailable"
messageDefaultText="* No E-Billing Role Available"/>
```

#### 3.1.15.2.1.1 Parameters

Text that used dynamic values will need to be rewritten to use the `tc:messageParam` tag. For example, a message that said:

```
Share documents to <tc:out value="{cjb.vendorName}" /> in Collaborati.
```

Becomes:

```
<tc:message
key="custom.VEN$.screen.xml.vendorDocumentSharing.shareDocumentsToVendor">
  <tc:messageParam value="{cjb.vendorName}" />
</tc:message>
```

...and you create a corresponding custom message with key "custom.VEN\$.screen.xml.vendorDocumentSharing.shareDocumentsToVendor" and text "Share documents to {0} in Collaborati."

#### 3.1.15.2.1.2 Date Formatting

The example below shows date formatting and text localization. A custom message is supplied to the `localize()` method.

```
public String getLastPublishedOnString() {
    Date lastPublished = new Date();
    String localizedDate =
        platform.getUtilityService().formatDateByUserSetting(lastPublished);
    return
        platform.getInternationalizationService().localize("customKey.publishedDateMess
age", localizedDate);
}
```

#### 3.1.15.2.1.3 Best Practices

Avoid using custom java code to simply append dynamic values in a message.

If the label of a field was previously hard-coded in a custom screen, use `<tc:message>` instead of `<tc:label>` to keep exactly the same text formatting.

Avoid making new custom messages for field labels if an existing label will do the job. For example, `<tc:label label="Company Tax ID" />` could be replaced by the object definition field label `<tc:label category="VEN$" name="companyTaxID" />`, presuming that the field label from the object definition already says "Company Tax ID". The label from the object definition will have been localized already, so this approach avoids double work.

### 3.1.16 Migrating Custom Designs

The typical life cycle of any new TeamConnect design is:

- Design/Development phase
- Testing phase
- Production phase

The following design tools are provided for migrating custom designs from a development TeamConnect instance to a QA or production TeamConnect instance:

- [Design Snapshot Tool](#)—From your development TeamConnect instance, captures your latest design snapshot, provides a summary of archived design snapshots, generates a design update file (used to update a TeamConnect instance at custom design version x to design version y). You may also revert your current design to a previously saved snapshot using this tool.
- [Design Import Tool](#)—From your QA or production TeamConnect instance, applies a design update file to upgrade the instance from design version X to design version Y.
- [Configuration Transfer Utility](#)—Allows users to transfer custom designs and configurations to a new production version of TeamConnect.

**Caution:** *It is neither recommended nor supported for a user to make separate changes to both the development and QA instance designs, and then use the Design Snapshot Tool and Design Import Tool to merge the respective designs.*

For descriptions on assigning rights to work with the Design Snapshot Tool and Design Import Tool, see [Group Tool Rights](#).

#### 3.1.16.1 Requirements

For the design import process, it is required that both the source (development) and destination (QA or production) TeamConnect instances are at the same version.

#### 3.1.16.2 Working with the Design Snapshot Tool

Use the Design Snapshot Tool from your development TeamConnect instance to:

- Capture design snapshots
- Create design upgrade files
- Revert to previous snapshots

For information about setting rights to use the Design Snapshot Tool, see [Group Tool Rights](#).

**Create Snapshot**

To **create** a snapshot of the current system design, specify a comment and click the button below. Creating a snapshot may take a few minutes.

Notes:

[View Log](#)

✓ The snapshot was successfully created.

**Snapshot Files**

To **generate** a file for import, select two design versions and click the button below. This file will contain the differences between the two versions.

	Created On	Author	Name	File Size (Bytes)	Notes
<input type="checkbox"/>	06/23/2008 12:00 AM		<a href="#">Design0.zip</a>	0	
<input type="checkbox"/>	08/04/2008 03:44 PM	<a href="#">system,</a> <a href="#">system</a>	<a href="#">Design1.zip</a>	1717784	Before creating the Injunction custom object

[View Log](#)

Design Snapshot Tool

### 3.1.16.2.1 Creating a Design Snapshot

After you make your custom design changes on the development TeamConnect instance, you can use the Design Snapshot Tool to capture a snapshot of the current design. It is recommended that you take regular snapshots of your design to back up changes.

#### To create a design snapshot

1. From the Designer, **Tools** drop-down list, select **Design Snapshot Tool**.
2. From the **Create Snapshot** section, type notes about design changes implemented. (Be detailed in your description, because you may need good notes at a later time if you perform a Revert operation as described in [Reverting to a Previous Snapshot](#).)
3. Click **Create Snapshot**.
4. Click **OK** in the confirmation dialog.

Afterward, the new design snapshot is added to the **Snapshot Files** section below.

5. If the export does not complete, click the **View Log** link. For more information, see the Troubleshooting and Warnings section.

### 3.1.16.2.2 Creating a Design Upgrade File

After you've captured one or more design snapshots, you can create a design upgrade file using the Design Snapshot Tool from the development TeamConnect instance.

**Note:** *The first design snapshot file, Design0.zip, represents the out-of-box TeamConnect design and is provided for you.*

#### To create a design upgrade file

1. From the Designer, **Tools** drop-down list, select **Design Snapshot Tool**.
2. From the **Design Versions** section, select the current design version of the target TeamConnect instance (QA or production).
3. Also select the latest design version that you would like to upgrade the target instance to.

For example, if you have captured two snapshots on the development TeamConnect instance, and would like to upgrade your out-of-box QA TeamConnect instance to the second snapshot, select Design0.zip and Design2.zip (where Design0.zip is the default snapshot provided for you).

4. Click **Generate Import File**.
5. From the confirmation pop-up window, click **Save**.

If the file generation does not complete, click the **View Log** link. For more information, see [Troubleshooting](#).

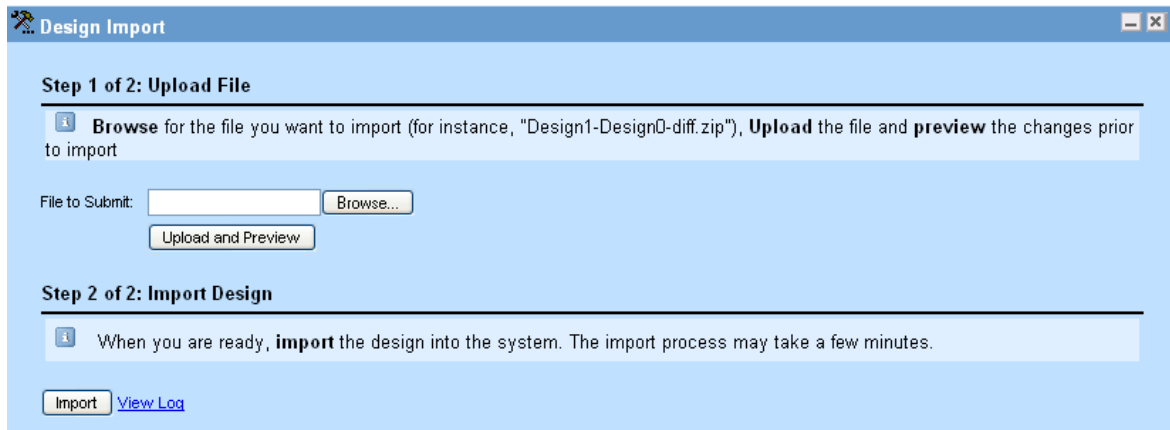
### 3.1.16.2.3 Reverting to a Previous Snapshot

From the Snapshot page, you can revert your design to any existing, previously captured snapshot. To do so, click the button **Revert to this version** next to the desired snapshot. TeamConnect will then capture a snapshot of your current design, generate a diff file between your current design and the desired snapshot, and import that diff file to your design.

### 3.1.16.3 Working with the Design Import Tool

Use the Design Import Tool from your QA or production TeamConnect instance to upgrade the current design to a newer design version that was created on your development TeamConnect instance. It is recommended to perform a design upgrade to a production TeamConnect instance during a pre-scheduled period of application down-time.

For information about setting rights to use the Design Import Tool, see [Group Tool Rights](#).

**Design Import Tool**

### 3.1.16.3.1 Performing a Design Upgrade

After you've generated a design upgrade file, you can use the Design Import Tool from a QA or production TeamConnect instance to upgrade to a target design version.

Before you begin, back up the destination TeamConnect database.

**Important:** *Since both a database backup and a design import could require significant resources, Mitrtech recommends that you perform design import during off-peak hours on production application servers.*

**Note:** *If the QA or production TeamConnect instance that you are upgrading is not at the same base design version supported in the design upgrade file, then the import will not complete. For example, if the QA TeamConnect instance is at design version 3 and the design upgrade file updates the design from version 4 to version 5, then the import will not complete.*

#### To perform a design upgrade

1. From the Designer application, **Tools** drop-down list, select **Design Import Tool**.
2. From the **Step 1 of 2: Upload File** section, click **Browse**. Navigate to the location of the design upgrade file.
3. Click **Upload and Preview**.  
A pop-up window appears displaying a summary of design changes that results from the design import.
4. Click **Done** after you finish reviewing the summary to close the window.
5. From the **Step 2 of 2: Import Design** section, perform the import. If you wish to take a snapshot of the current design just before performing the import, click the relevant checkbox. Click **Import**. Click **OK** in the confirmation pop-up menu.
6. DMT checks the existing design to see if any incompatibilities exist with the design that is about to be imported. If any are found, warning messages are displayed. In some cases, you



are given the option to ignore the warning and continue with the import. For more details about possible warning messages, see [Warnings](#).

Depending on the extent of design changes, the import may take a few minutes or longer. The status icon at the upper right turns while the import is in progress.

- After the import completes, a success message appears. If the import does not complete, and an error displays, click the **View Log** link. For more information, see [General Troubleshooting Steps](#).

Even when your import completes with a success message, you should scan the log file for warning messages. Some items, such as object views for embedded objects, are ignored during import. A warning will be generated when an item is ignored. You should be aware of such situations.

#### 3.1.16.4 Working with Configuration Transfer Utility

Like the older version of the [Design Import Tool](#), the new Configuration Transfer Utility (CTU) allows users to transfer the current design that was created on your development TeamConnect instance to a different design version in your Production instance.

Use the CTU to make small incremental changes (e.g. add a rule for a custom object), medium changes (create a set of workflows), and large changes (provision a new server).

The process for using the new CTU is slightly different and changes have been made on how artifacts and design dependencies are handled.

For instance, in an example scenario a user has added a rule and then later made modifications three separate times. All entries are captured automatically by the tool, and in order to export the change the user needs to include all four entries (the initial rule creation and subsequent changes; i.e. the dependencies) in the export package. Packages can be in one export file or split up, but the import must follow the sequence that creates the object first, then the rules on which they depend. This differs from the old Design Import Tool in that, previously, the package would have only one entry for that rule creation.

**Note:** The previous Design Import Tool could sometimes be used as a design backup, in that if an upgrade failed users could revert back to the old design, removing any partial changes. CTU does not have this capability.

The Old Way: Design Migration Tool	The New Way: Configuration Transfer Utility
1. Make changes in Development environment	1. Make changes in Development environment
2. Export entire configuration as a design snapshot	2. Filter and select only the changes you want
3. Edit XML files in the snapshot to select the changes	3. Export the changes to a .zip package

The Old Way: Design Migration Tool	The New Way: Configuration Transfer Utility
4. Import the edited snapshot	4. Import the package by uploading it
5. If import fails, repeat steps	

Click the links below for instructions on using CTU:

- [Export Design Changes](#)
- [Import Design Changes](#)
- [Troubleshooting and Best Practices](#)
- [Artifacts Captured for Custom Designs](#)

**Important Note:** In order to use the CTU, the user's Group must have Contact, Document, and View Rights for **Admin Settings**. If the user does not have Contact Rights they cannot see the Changed By nor the Create By values; if the user does not have Document Rights they are unable to view Export Packages.

#### 3.1.16.4.1 Export Design Changes

As a TeamConnect administrator, you can select appropriate changes that were made in **TeamConnect Setup** or **Admin Settings** to be imported to your next environment. This provides the flexibility to migrate designs in parts depending on the demands of your project, as well as allow for precise scrutiny over the changes to be made.

##### Accessing the Export Design Changes page:

From your TeamConnect homepage, click or hover your mouse pointer over the **All** tab. Select **Export Design Changes**, located under the **Tools** section.

The data found in the top Design Changes table records changes made in TeamConnect Setup (ex: removing, changing, or adding look-up tables). Changes are listed by the date and time the change was made (starting with the most recent), who made the change, change ID number, description of the change, the category (Type) of change, and any notes appended. Each design change is associated with a design category, shown in the **Change Type** column (e.g. Audit Appender, Custom Tool, Global Navigation, Group Account, etc).

If a change has previously been exported, the file name of that package appears in the **Recent Export Package** column. Identifying previously exported packages streamlines the process of making corrections or additions to a group of changes, and focuses attention to packages that have not yet been exported.

If a change has been previously imported, an [IMPORTED] indicator will appear in the **Description** column.

Changed On	Changed By	ID	Description	Type	Notes	Recent Export Package
03/03/2015 09:08 PM	Shelton, Jonathan	18	Custom Lookup Table qwertqwertwer; added	Audit Appender	mass changes	
03/09/2015 01:42 PM	Marino, Samuel	19	Detail Field Test added	Object Definition: Custom Field	mass changes	
03/09/2015 01:42 PM	Marino, Samuel	20	Custom Lookup Table Categories added	Custom Lookup Table	mass changes	
03/09/2015 01:42 PM	Marino, Samuel	21	Lookup Table Item adfesf added	Custom Lookup Table	mass changes	
03/09/2015 01:55 PM	Marino, Samuel	22	Tool dfhh added	Custom Tool	mass changes	
03/09/2015 02:00 PM	Marino, Samuel	23	Activity Item fdfg added	System Lookup Table	mass changes	
03/09/2015 04:20 PM	Marino, Samuel	24	System View Default Contact removed; System View Default FooBar View added	Settings	mass changes	
03/09/2015 04:20 PM	Marino, Samuel	25	System View Default FooBar View removed; System View Default Contact added	Settings	mass changes	
03/09/2015 04:29 PM	Marino, Samuel	26	Group Functional Access Document, Mass Edit removed; Group Functional Access Document, Marino, David removed; Group Fu...	Group Account	mass changes	

Design Changes Table and Filter

Use the filter tool above the table to show only changes by date range, individual who made the changes, description of changes, note keywords, type, or object (e.g. Appointment, Line Item, Task, etc.). Click **Apply Filters** to sort the list, or **Reset Filters** to list all design changes. The **Changed By** filter layout depends on settings defined in **Admin Settings | User Interface | User Selection**.

If a design change has been made under Admin Settings, the **Change Type** column will display "Admin Settings".

Control the amount of design changes displayed in the table by selecting an option from the Design Changes per Page drop-down menu, located at the bottom right corner of the table. Navigate through the design change pages by clicking the numbers under the table. **Warning: Moving from page to page resets the checkboxes you selected.** Click **Add Selected to Export List** before you move on to the next page.

To include a note or details for an individual export change, click the pencil icon under the **Notes**

column, type into the text box and click the check mark to confirm or the red "x" to cancel. Notes are transferred when [importing](#) these design changes.

To include a note or details for multiple design changes in the table, check the boxes of the items for which you would like to include a note, then click the pencil icon located inside the **Notes** column. A pop-up window appears where you can type in your notes or instructions. Click **Apply** to confirm or **Cancel** to exit without saving. **Note: If none of the checkboxes are marked, clicking the bulk edit icon has no effect.**

Notes

Edit Notes

### To create an export package, use the following steps:

1. Click any number of checkboxes from the Design Changes table.
2. After selecting the desired design changes, click **Add Selected to Export List**, located at the bottom of the table. The page refreshes to reflect your selections in the Export List Summary table.

**Note:** TeamConnect issues a warning if you export changes from multiple versions of TeamConnect, and you can see what versions are included in the **TCE Version** column. You also receive a warning if there are non-consecutive changes added to your export list. A non-consecutive change is considered a "skip" in the numeric sequence of ID numbers (i.e.

exporting 1, 2, 4 would issue a warning because ID number 3 has been omitted). The warnings are an opportunity to correct unintentional changes, and do not prevent the user from dismissing the warning and continuing with the export.

The items in the Export List Summary table are listed by change ID numbers in ascending order and match the ID numbers in the table from which you initially selected the items.

3. Click **Export Changes**. To remove a change from this selection, click the corresponding checkbox and click **Remove Selected from Export List**. You do not need to select the checkboxes in order to export the changes.

When exported, each change includes a list of host names of the servers on which the change was made and any servers on which it was previously imported. It also indicates the user who originally performed the changes. When a change that has been imported is re-exported, the user identified as the one making the change will be displayed as the one who imported it.

Export List Summary

Upon clicking **Export Changes**, a text field appears in a pop-up window. Using the Export List Summary field, add notes about individual changes or a description of the export. Click **Finish Export** to confirm.

Date	Time	Changed By	ID	Description	Type	Notes	Bundle	Version
Nothing found to display.								
<div style="display: flex; justify-content: space-between;"> <span>Remove Selected from Export List</span> <span>Export Changes</span> </div>								
Recently Created Export Packages								
Name	Created By	Export Package Description						
DesignExport_20150326_022909.zip	Taylor, Tim							
DesignExport_20150326_022731.zip	Taylor, Tim							

Export Changes

Package files can be downloaded from the Recently Created Export Packages table (in the bottom table, shown above) by clicking the hyperlink under the **Name** column.

#### 3.1.16.4.1.1 View Older Packages

Because the **Export Design Changes** page only displays the ten (10) most recently created export packages, you can click the **View older packages here** hyperlink at the bottom of the screen to view all export packages created by any user in the TeamConnect application.



Recently Created Export Packages		
Name	Created By	Export Package Description
<a href="#">DesignExport_20150326_022909.zip</a>	Taylor, Tim	
<a href="#">DesignExport_20150326_022731.zip</a>	Taylor, Tim	
<a href="#">DesignExport_20150323_080219.zip</a>	Shelton, Jonathan	
<a href="#">DesignExport_20150323_080210.zip</a>	Shelton, Jonathan	
<a href="#">DesignExport_20150323_075733.zip</a>	Shelton, Jonathan	
<a href="#">DesignExport_20150323_075731.zip</a>	Shelton, Jonathan	
<a href="#">DesignExport_20150323_075631.zip</a>	Shelton, Jonathan	
<a href="#">DesignExport_20150323_075625.zip</a>	Shelton, Jonathan	
<a href="#">DesignExport_20150323_075516.zip</a>	Shelton, Jonathan	
<a href="#">DesignExport_20150323_070917.zip</a>	Shelton, Jonathan	











[View older packages here](#)

[View Older Packages Here Hyperlink](#)

The Design Migration Packages table records the package name, category, file size and type, the name of the individual who has checked out the package (if applicable) and most recent date and time the package was modified.

Additionally, under the **Action** column, users can view package properties and/or check out the package.

- —View package properties
- —Check out package

Location: Top Level » System » Design Versions » Design Migration Packages							
Documents 1 - 10 of 43							
Action	Name	Category	File Size	File Type	Checked Out By	Modified On	Custom Search
	<a href="#">Demo_3-12-15_2-17_PM.zip</a>	Document	654 Bytes	Binary		3/13/15 1:40 PM	
	<a href="#">DesignExport_20150323_070917.zip</a>	Document	1.3 KB	Binary		3/23/15 2:09 PM	
	<a href="#">DesignExport_20150323_075516.zip</a>	Document	701 Bytes	Binary		3/23/15 2:55 PM	
	<a href="#">DesignExport_20150323_075625.zip</a>	Document	701 Bytes	Binary		3/23/15 2:56 PM	
	<a href="#">DesignExport_20150323_075631.zip</a>	Document	701 Bytes	Binary		3/23/15 2:56 PM	
	<a href="#">DesignExport_20150323_075731.zip</a>	Document	701 Bytes	Binary		3/23/15 2:57 PM	
	<a href="#">DesignExport_20150323_075733.zip</a>	Document	701 Bytes	Binary		3/23/15 2:57 PM	
	<a href="#">DesignExport_20150323_080210.zip</a>	Document	700 Bytes	Binary		3/23/15 3:02 PM	
	<a href="#">DesignExport_20150323_080219.zip</a>	Document	700 Bytes	Binary		3/23/15 3:02 PM	
	<a href="#">DesignExport_20150326_022731.zip</a>	Document	1.3 KB	Binary		3/26/15 9:27 AM	

1 2 3 4 5 Documents per page: 10

Design Migration Packages Table


To filter the table layout, use the arrows in each column header (↕) to sort data alphabetically, chronologically, or by size. Use the Documents per page drop-down menu at the bottom right of the table to control how many documents you see per page.

Open a package with Windows Explorer by clicking the hyperlink under the **Name** column. In the new pop-up window, select the option to open or save the file to your computer, then click **OK**.

For additional filter criteria, click [Custom Search](#) at the top right of the table.

By clicking the **Custom Search** hyperlink in the **View Older Packages** page within **Export Design Changes**, you can further filter the results for design migration packages.

The top portion of the **Custom Search** page allows you to specify the criteria on which you wish to search by value of a particular field (e.g. package name, category, file type), or date.

- Using the radio buttons above the Filter Criteria table, determine whether you want to include all of the following values, any of the values, or group the values.
- Select a field and operator from the drop-down lists, then type in the value for which you wish to search. For the dates section, click on the  icon to select a date.
- Under the **Action** column, use the plus or minus buttons to add or erase additional search fields, or click the **Add Criteria** button at the bottom of the table for the same effect as the plus button.
- Click **Clear Values** at the bottom of the page to remove text under the **Value** column, or click **Reset to Default** to remove all values and added search fields.

After filling in all search criteria, click the **Search** button to apply the filters. Click **Cancel** to return to the previous page.

The bottom portion of the **Custom Search** page allows you to select the columns you wish to display in the Design Migration Packages table.

- Under the **Order Field** column, make a selection from the drop-down lists. If you would like to remove one of the columns, leave the drop-down list on the "(Select)" option, or click the minus button under the **Action** column.
- Add a column by clicking the plus button under the **Action** column, or click the **Add Another Column** hyperlink at the bottom of the table.

**Note:** Unless you choose a field from the drop-down list, the added column does not appear in the Design Migration Packages table.

- Click **Clear Values** to remove text under the **Value** column, or click **Reset to Default** to reset fields and remove added columns.

After choosing your display criteria, click the **Search** button to apply the filters. Click **Cancel** to return to the previous page.

## Viewing Your Search Results

Once you have entered your search criteria and selected **Search**, you can view the modified Design Migration Packages table.

In the new screen, you have additional hyperlinks above the table.



- **Custom Search**—Return to the **Custom Search** page with cleared values.
- **Save Search**—Save your current custom search criteria.
- **Modify Search**—Return to the **Custom Search** page with your current values in place.

### 3.1.16.4.1.2 System Settings Excluded From Export

As part of the **Configuration Transfer Utility** tool, certain changes made in **Admin Settings** are captured to be imported into new environments. A number of system settings are excluded from the design migration tool, and these items are listed below:

Admin Settings Tab	Excluded Setting
General	Business Objects
	TeamConnect URL
Region	Default Locale
	Default Currency
Calendar	All Settings
Email	Outgoing Mail Server Settings
	Incoming Mail Server Settings
	TeamConnect IMAP Server Settings
Maintenance	Maintenance Notification
	Shutdown Alert
Connections	Proxy Settings

Admin Settings Tab	Excluded Setting
	Collaboration Portal
History	All Settings
About	Instance Information
	Available Updates
	Support Options

Click **Update** to add changed system settings. Users must update first before exporting, or the changes cannot be imported.

### 3.1.16.4.2 Import Design Changes

As a TeamConnect Administrator, you can create design migration changes by importing a package not only made to the target environment, but also logged with Data Migration Tools. This ability aids in pinpointing changes that have already been imported, and identifying changes that still need to be made. Additionally, you can create an audit trail for changes that are being made to an environment where they potentially should not be. The import operation runs on foreground processing, as opposed to background processing, so the transaction can run reliably and unattended.

#### Accessing the Import Design Changes page:


From your TeamConnect homepage, click the **Tools** button in the upper right corner of TeamConnect. Select **Import Design Changes**.

At the top of the **Import Design Changes** page, select **Browse** to import your package. Once you have selected the appropriate file from your computer, click **Open**, then click **Upload**.

Your imported package appears in the Import Details table. If any notes were made in the **Export Design Changes** page, they appear in the **Notes** column. **Note:** *Uploading an incompatible file does not display results in the table.*

Package Description						Exported Date	Status
Demo Description						3/27/15 8:12 PM	Checking for prior imports took 0.00 seconds.
ID	Date	Time	Changed By	Description	Notes	Status	
58	3/27/15	2:15 PM	Taylor, Tim	Custom Lookup Table Demo Lookup Table added			
59	3/27/15	2:16 PM	Taylor, Tim	Lookup Table Item Demo Item 1 in table Demo Lookup Table added			
60	3/27/15	2:16 PM	Taylor, Tim	Lookup Table Item Demo Item 2 in table Demo Lookup Table added			
61	3/27/15	2:16 PM	Taylor, Tim	Lookup Table Item Demo Item 3 in table Demo Lookup Table added			

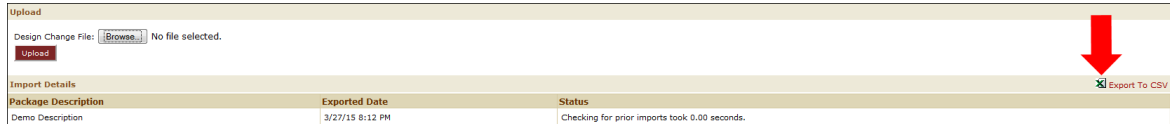
Top of Import Design Screen

Export your list of uploaded changes to Excel by selecting the  icon in the top right section of the Import Details table, or click the **Export to CSV** button below the table.



Choose to open or save the file by selecting the corresponding button in the download bar that appears at the bottom of your screen, or click **Cancel** to exit.

A .CSV file is downloaded through your browser containing all of the data shown in the Import Details table (i.e. design change ID, date, time, changed by, description, notes, and status), and also includes any other fields that are included in the XML of the change record, but are not displayed in the on-screen table.

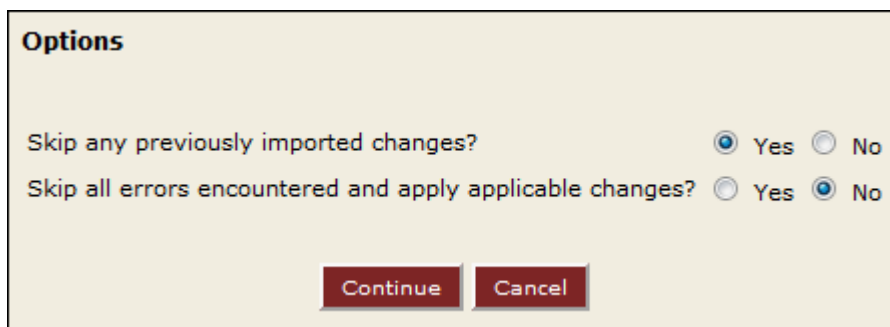


Export to Excel Icon

Once the data is generated into the Import Details table, click the checkboxes of the changes to be applied, then click **Apply**. All applied changes appear in the Imported Change History table at the bottom of the page.

**Important Note:** You cannot stop the import unless an error occurs. If you log out of TeamConnect, the import will continue to change the environment, but will go no further if an error is encountered.

Upon clicking **Apply**, choose to skip previously imported changes or errors encountered by clicking the "Yes" radio button in the **Options** pop-up window. Previously imported packages are already identified in the **Status** column after file upload. If you choose not to skip errors, you still have the option to skip all errors after the first error is generated [during importation](#). The reason for having multiple options to skip errors is helpful if the imported package has a long list of changes and the first error may not be identified for a while.



Import Options

Previously imported packages are skipped without stopping the import process as an error might. In order to eliminate duplicate packages, any package that has already been imported displays the information in the second **Status** column (see example below).

Error notices help you identify a solution for the issue, and TeamConnect provides warnings for the following cases:

- A change has already been imported (duplicate ID and data).
- There is an ID gap between the last imported change and the first change of the uploaded package (this may have occurred when selecting design changes to add to the [Export List](#)).
- Any of the uploaded design changes were done on a TeamConnect version that is later than the current instance—**Note:** TeamConnect allows the import of design changes that were done on an older TeamConnect version. The error serves as notice in case this was done unintentionally.

Import Error						
Change #58, Custom Lookup Table Demo Lookup Table added, encountered an error: Duplicate Unique Code						
<input type="button" value="Stop Import"/> <input type="button" value="Retry Change"/> <input type="button" value="Skip Change"/> <input type="button" value="Skip All Errors"/>						
Import Details						
Package Description	Exported Date			Status		
Demo Description	3/27/15 8:12 PM			Checking for prior imports took 0.00 seconds.		
ID	Date	Time	Changed By	Description	Notes	Status
58	3/27/15	2:15 PM	Taylor, Tim	Custom Lookup Table Demo Lookup Table added		Duplicate Unique Code

#### Import Error Options

At this point, you have several options:

- **Stop Import:** Cancel design import and start over.
- **Retry Change:** Reattempt the import if you have made the necessary changes.
- **Skip Change:** If the Import Details table has more than one item within the package, you can select this option to skip the first error to test the remaining items one-by-one. Selecting **Skip Change** does not import failed changes.
- **Skip All Errors:** If the Import Details table has more than one item within the package, you can select this option to test all items simultaneously. Selecting **Skip All Errors** does not import a failed change.

#### 3.1.16.4.2.1 Imported Change History

At the bottom of the **Import Design Changes** page, you can view all imported packages in the Imported Change History table.

Imported Change History						
Date Range: <input type="text" value="1/20/2015"/> To: <input type="text" value="4/20/2015"/>		Imported By: <input type="text" value="(Select)"/>		Notes: <input type="text"/>		
Description: <input type="text"/>		Package Name: <input type="text"/>				
<input type="button" value="Apply Filters"/>		<input type="button" value="Reset Filters"/>				
ID	Imported On	Imported By	Description	Notes	Package Name	
123	04/14/2015 09:28:37 AM	McNeilxxx, Charles	Rule TestDelete removed		DesignExport_20150414_022757.zip	
105	04/13/2015 03:20:41 PM	McNeilxxx, Charles	Rule CmTCTOOL42 added		DesignExport_20150413_064604.zip	
106	04/13/2015 03:20:41 PM	McNeilxxx, Charles	Rule Condition 1 for Rule CmTCTOOL42 added; Rule CmTCTOOL42 changed; Rule Condition 2 for Rule CmTCTOOL42 added		DesignExport_20150413_064604.zip	
106	04/13/2015 02:57:22 PM	McNeilxxx, Charles	Rule Condition 1 for Rule CmTCTOOL42 added; Rule CmTCTOOL42 changed; Rule Condition 2 for Rule CmTCTOOL42 added		DesignExport_20150413_064604.zip	
105	04/13/2015 02:09:48 PM	McNeilxxx, Charles	Rule CmTCTOOL42 added		DesignExport_20150413_064604.zip	
21	04/10/2015 01:21:49 PM	Marino, Samuel	Object Definition Test added		Test Complete Red.zip	
33	04/10/2015 01:21:49 PM	Marino, Samuel	Object Definition Test removed		Test Complete Red.zip	
33	04/10/2015 01:21:29 PM	Marino, Samuel	Object Definition Test removed		Test Complete Red.zip	
21	04/10/2015 01:19:48 PM	Marino, Samuel	Object Definition Test added		Test Complete Red.zip	
21	04/10/2015 01:16:02 PM	Marino, Samuel	Object Definition Test added		Test Complete Red.zip	

#### Imported Change History

Use the search filter above the Imported Change History table to pinpoint a specific package import,

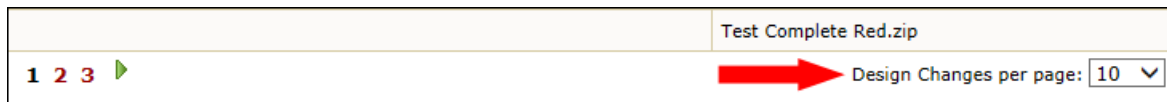
or to view changes made during a specific time range. Click the icons to select "from" and "to" dates, or type a date directly into the date fields.

1. Find changes made by an individual by selecting a TeamConnect user from the Changed By drop-down list, or search by description of the imported package by typing directly into the Description text box.
2. If the imported package contained text in the **Notes** column, you can search for that package by matching the text in the Notes text box.

3. Click **Apply Filters** to search using the updated criteria, or **Reset Filters** to start over. Results display above the search filter.
4. Further refine your search by toggling the direction of the lists using the directional arrows (↕) in each of the column headers. The first click causes the column to list items in ascending order; click again for descending order.

At the bottom of the Imported Change History table, dictate how many design changes you would like to see per page by making a selection from the drop-down list (shown below).

Navigate through multiple pages of design changes using the number hyperlinks or the ▶ icon to the left of the drop-down.



#### 3.1.16.4.3 Troubleshooting and Best Practices

- When creating a Custom Action or Scheduled Action Rule (Object Definition>Rules) within setup for an [Address Book](#) as the recipient, the address book must first be created in the import instance before the actions assigned to the address book can be successfully imported into the new instance. To create the address book, click the **Contacts** tab at the top of the TeamConnect application, then select the **New** drop-down and click **New Address Book**. Enter the same name as the address book for which the custom action or scheduled action rule is being created, then click **Save**.
- Workflow: Make configuration changes only on development server(s). If a problem is discovered in staging or production, correct this problem in development, re-export the changes, then import the changes again. This process takes more time, but it ensures that the same content is deployed and tested across all environments.
- Export: Build and test small export packages before assembling large ones. Once you can verify that they are importing accurately, you can consider bundling them into larger packages for export.
- Import: Skip errors when importing to a non-production environment. It is more efficient to troubleshoot errors found during import after completion, rather than stopping at every error. Refer to the results and warning icons to locate the steps that failed, and test again once they are fixed. However, when importing changes in production, it is best practice to stop immediately when an error occurs, as errors should generally not be expected for this type of environment.
- Users: Log in as a "Special User" to avoid the possibility of two users importing at the same time. The **Design Change Tool** does not prevent two users from simultaneously performing imports on the same server, which could result in unpredictable behavior and conflicting changes. It is best practice to define a design change group and assign a single user with permissions to the design import section of the application. If you have a package that is in the process of importing changes and another user attempts to import changes, both imports will fail.
- If a design change has dependencies (or rules on which a custom object depends) on artifacts on the **Export** instance, but not on the **Import** instance, the dependencies will need to be in the same package. If any package is imported without the dependencies either existing in the **Import** instance's design, or in the package itself, the import process will result in an error.

- If your [search results preferences](#) is set to display less records per page than the number of design changes you have made, the changes may not appear in their entirety. Avoid this by going to **Preferences** and entering a larger number (i.e. 999) in the "Number of records per result page" text box, located under Search Results.
- ☐ A brief description of the XML package tags:

Tag	Description
Operation type	The design change will be either an update, insertion, or deletion.
number	This is the display order of the change.
entityName	The type of change (e.g. rule condition, object data field, user system setting, etc.).
exportDate	The date and time the export was committed.
description	This is the description the user entered for the package.
Change description	A short description of the design change included in the package. This is the same description found on the <b>Export Design Changes</b> page in the <b>Description</b> column.
type	The category that the design change falls into. This is the same description found on the <b>Export Design Changes</b> page in the <b>Type</b> column.
time	The time the design change was made.
tcVersion	The TeamConnect application version (update) on which the change was made.
objUniqueCode	A unique code required for identification that was entered by the user at the time of design change creation. This code is related to an object and its object rights, and ensures that proper rights are assigned to the correct object, despite order of import or object creation.
notes	Any notes appended to the change by the user appear here.
id	The ID number of the design change as it appears on the <b>Export Design Change</b> page.
date	The date on which the design change was made.

Tag	Description
changedBy	The name of the user who made the design change.

#### 3.1.16.4.4 Artifacts Captured for Custom Designs

At the bottom of this page is a table outlining the custom design changes that are captured in the TeamConnect Configuration Transfer Utility environment. Custom design changes are made in TeamConnect Setup.

**Note:** CTU does not capture Reporting artifacts.



Click on the highlighted link for more information on the artifact and how to make captures for exporting and importing.

Artifact	Category
<a href="#">Assignee Roles</a>	<a href="#">Object Definition XXXX</a>
<a href="#">Audit Appenders</a>	Audit Appender
<a href="#">Categories</a>	Object Definition XXXX
<a href="#">Custom Blocks</a>	Object Definition XXXX
<a href="#">Custom Fields</a>	Object Definition XXXX: Custom Field
<a href="#">Custom Look-up Tables</a>	Custom Look-up Table
<a href="#">Custom Look-up Table Items</a>	Custom Look-up Table
<a href="#">Custom Tools</a>	Custom Tool
<a href="#">Default object views</a>	Settings
Global Navigation	Global Navigation
<a href="#">Group accounts</a>	Group Account
Group account tool access	Group Account

Artifact	Category
Group account object copy items	Group Account
<a href="#">Home pages</a>	Home Page
<a href="#">Home page panes</a>	Home Page
<a href="#">Home page pane content items</a>	Home Page
Line item adjustment reasons	Object Definition \$LNI
Line item display items	Object Definition \$LNI
<a href="#">Line item expense unit price items</a>	Object Definition \$LNI
Line item task codes	Object Definition \$LNI
<a href="#">Notification templates</a>	Notifications
<a href="#">Object definitions</a>	Object Definition XXXX
<a href="#">Object definition display attributes</a>	Object Definition XXXX
Object definition files (automated actions)	Object Definition XXXX: Automated Action
Object definition files (automated qualifiers)	Object Definition XXXX: Automated Qualifier
Object definitions files (screens)	Object Definition XXXX: Screen
<a href="#">Object definition unique attributes</a>	Object Definition XXXX
<a href="#">Object definition phases</a>	Object Definition XXXX
<a href="#">Object definition phase transitions</a>	Object Definition XXXX
Object definition batch view items	Object Definition XXXX
Object definition field configuration	Object Definition XXXX
<a href="#">Object definition custom messages</a>	Object Definition XXXX

Artifact	Category
<a href="#">Object views</a>	Object Definition XXXX: Object View
<a href="#">Object view tabs</a>	Object Definition XXXX: Object View
<a href="#">Object view tab blocks</a>	Object Definition XXXX: Object View
<a href="#">Portal panes</a>	Portal Pane
<a href="#">Portal pane content items</a>	Portal Pane
<a href="#">Routes</a>	Route
<a href="#">Route stops</a>	Route
<a href="#">Route stop members</a>	Route
<a href="#">Route notifications</a>	Route
<a href="#">Rules</a>	Object Definition XXXX : Rule
<a href="#">Rule action parameters</a>	Object Definition XXXX : Rule
<a href="#">Rule qualifier parameters</a>	Object Definition XXXX : Rule
<a href="#">Rule conditions</a>	Object Definition XXXX : Rule
Rule history description items	Object Definition XXXX : Rule
Rule history fields	Object Definition XXXX : Rule
<a href="#">Search Views</a>	Object Definition XXXX : Search View
<a href="#">Search view criteria items</a>	Object Definition XXXX : Search View
<a href="#">Search view result items</a>	Object Definition XXXX : Search View
<a href="#">System appenders</a>	System Appender
<a href="#">System custom messages</a>	Settings

Artifact	Category
System folders (every folder under System except Design Versions and every folder under HTTPRoot)	System File
<a href="#">Templates</a>	Object Definition XXXX : Template
<a href="#">Template fields</a>	Object Definition XXXX : Template
<a href="#">Template sub templates</a>	Object Definition XXXX : Template
<a href="#">Template sub template fields</a>	Object Definition XXXX : Template
Template related templates	Object Definition XXXX : Template
<a href="#">Template categories</a>	Object Definition XXXX : Template
<a href="#">Wizards</a>	Object Definition XXXX : Wizard
Wizard rule condition	Object Definition XXXX : Wizard
<a href="#">Wizard pages</a>	Object Definition XXXX : Wizard
<a href="#">Wizard page components</a>	Object Definition XXXX : Wizard
<a href="#">Wizard page actions</a>	Object Definition XXXX : Wizard
<a href="#">Wizard page rules</a>	Object Definition XXXX : Wizard

### 3.1.16.5 Troubleshooting

This section is organized into the following areas:

- [Warnings](#)
- [General Troubleshooting Steps](#)
- [Error Messages and Performance Issues](#)

#### 3.1.16.5.1 Warnings

This section provides general information about some elements and objects that can be imported using the Design Import tool.

- [Design Import Restrictions](#)



- [Source Design With No Design Component](#)

#### 3.1.16.5.1.1 Design Import Restrictions

- Currently, the design upgrade file is created only for updating a TeamConnect instance design from an older design to newer design. If you need to revert a QA or production TeamConnect instance to an older design version, there are two possible methods:
  - If the older design version is captured in an earlier snapshot, use the method described in [Reverting to a Previous Snapshot](#).
  - If there is no earlier snapshot, you must make the design changes manually on the development TeamConnect instance, take a snapshot of the current design, and create a corresponding design upgrade file.
- This release of the Design Import tool does not support importing user accounts.
- Deleting System Settings is not supported during importing.
- Please be aware that when you import specific object types, some child elements are deleted from the destination TeamConnect (QA or production) and replaced by the child elements from the source TeamConnect design (from the development instance).

Please see the following table for specific details:

When Importing and Updating the Following Objects:	The Following Logic Will Apply:
Route stops	The source design's memberList child elements will replace those of the destination design.
Home pages	The source design's contentList child elements will replace those of the destination design.
Portal panes	The source design's contentList child elements will replace those of the destination design.
Rules	The source design's conditionList, historyDescriptionList, and historyFieldList child elements will replace those of the destination design.
Wizards	The source design's pageList and child elements will replace those of the destination design.

#### 3.1.16.5.1.2 Source Design With No Design Component

Generally, if the source design (from the development TeamConnect instance) does not include a design component (defined at the XML element level) and the destination design includes the corresponding design component, the import deletes that component from the destination TeamConnect database (for example, QA or production TeamConnect).

For example, if a Dispute object definition is being imported and the source XML file contains definitions for four phases, and the destination XML file contains definitions for five phases (including the four phases from the source database plus an additional phase), the import results in the deletion of the unique phase from the TeamConnect destination database.

This deletion would appear in the log file, but that log entry is not tagged with any specific error message.

#### 3.1.16.5.2 General Troubleshooting Steps

This section provides a general flow for locating the source of DMT issues within a design "diff." In addition, a few examples of errors you could find in the DMT log file and troubleshooting tips are provided.

1. Click the **View Log** link to open the dmt.log file.
2. If a DMT operation is successful, the last line of the log file will contain a success message. Otherwise, the end of the log file will contain a message that the operation failed, followed by detailed messages about each error.
3. Search for "FATAL" in the log to find issues that must be resolved for the import (or snapshot creation or upgrade file generation) to complete.
4. Afterward, search for "ERROR" in the log for additional issues.

#### 3.1.16.5.2.1 Logging

DMT creates a new log file any time it begins an import, export, or diff operation. The name of the log file is always **dmt.log**. Any previous log file is renamed to

```
dmt.import.log.YYYY-MM-DD-HH-MM
```

where the file extension represents the date and time the previous file was last modified. (Note that the string `import` in the example above may instead say `export` or `diff`, depending upon the operation that was performed.) DMT retains up to 10 log files. The oldest file is deleted when creating a new one would exceed the limit of 10.

#### 3.1.16.5.3 Sample Error Messages and Tips

The following are examples of error messages you might find in the log file. Specific details may differ for your designs/diffs but you can apply these to troubleshooting concepts for your specific scenarios.

- **Error message:**

```
FATAL - ImExCustomLookupTables.import: caused exception importing custom lookup
table=LOOK1UP.xml com.mitratesoft.teamconnect.foundation.TCException: Cannot
delete YDetailLookupItem because it still has 4 EProjDetailObjValues
referencing it
```

- a. To locate the corresponding \*.XML source file, from your diff path, navigate to:

```
diff\Lookup Tables\Custom\LOOK1UP.xml
```

where *diff* is the path to the diff

If the issue were in a system lookup table, you could find the \*.XML file in *diff*\Lookup Tables\System\LOOK1UP.xml

- b. If you open the source \*.XML file, the first line should end like:

```
"opcode="delete"
```

This is the command that tries to delete lookup table items.

- c. It is recommended to rename the \*.XML file (like \*.XML.old) so the DMT will skip the file during import but you can later track changes that might need to be made manually.

- **Error message:**

```
FATAL - ImExCategory.import: exception unmarshalling.Please enter a unique name.
```

```
com.mitratesoft.teamconnect.foundation.TCException: Please enter a unique name.
```

- a. To locate the corresponding \*.XML source file, look in the dmt.log file at the events just above (before) the FATAL message. A sample excerpt follows:

```
DEBUG - ImExCategory.importCategory: key=OTHE
DEBUG - ImExCategory.importCategory: itemKey=PR01_OTHE, opCode=insert
DEBUG - ImExCategory.importCategory: inserting category for
itemKey=PR01_OTHE, parent=PR01
DEBUG - ImExCategory.importCategory: inserting category for itemName=OTHER,
key=OTHE, order=0
```

- b. The first four characters of the itemKey, PRO1\_OTHE, or **PRO1**, indicates the object for which a duplicate category was found.
- c. From your diff path, navigate to:

```
diff\Object Definitions\PRO1\categories.xml
```

where *diff* is the path to the diff

- d. Open the categories.xml file and search for "PR01\_OTHE"

For example, the context could look like:

```
<Category displayOrder="" name="Other" treePosition="PR01_OTHE"
isActive="true" opcode="insert" />
```

- e. Generally you must verify that you do not create a category that already exists in the target instance. It is recommended to comment out the corresponding line from the categories.xml file so DMT will skip insertion of the duplicate category.

- **Error message:**

```
FATAL - ImExCustomFields.import: caused exception for filename=PR01_0112.xml
```

```
com.mitratesoft.teamconnect.foundation.TCException: Cannot delete Detail Field
because it still has 1 EProjDetailTextValue referencing it
```

- a. To locate the corresponding \*.XML source file, from your diff path, navigate to:

```
diff\Object Definitions\PRO1\Custom Fields\PRO1_0112.xml
```

where *diff* is the path to the diff

- b. If you open the source \*.XML file, the first line should end like:

```
opcode="delete"
```

This is the command that tries to delete the detail field.

- c. It is recommended to rename the \*.XML file like \*.XML.old so the DMT will skip the file during import but you can track changes that might need to be made manually.

#### 3.1.16.5.4 Error Messages and Performance Issues

This section contains some limitations in the Design Snapshot tool or Design Import tool. For most of these issues, workarounds are provided. If a workaround is not provided and you need more information, please contact Mitratech support.

- [BUILD FAILED message](#)
- [DMT script errors](#)

##### 3.1.16.5.4.1 BUILD FAILED message

**Issue:** An attempt to create a design snapshot, generate a design upgrade file, or import a design upgrade file results in an error message.

**Workaround:** If you are creating design snapshots from, or importing a design upgrade file to, a TeamConnect instance using an Oracle database, verify that you have the latest ojdbc driver file from Oracle in the DMT \lib directory. Also verify that your class path (environment variable) is not pointing to another directory that might contain a different version of the driver (for example, `C:\j2sdk1.5.0_6\jre\lib\ext`).

##### 3.1.16.5.4.2 DMT script errors

**Issue:** If you are troubleshooting a Design Snapshot tool or Design Import tool operation, and the log contains error messages like the following:

```
java.sql.SQLException: Data size bigger than max size for this type: 23862
at oracle.jdbc.dbaccess.DBError.throwSQLException(DBError.java:134)
at oracle.jdbc.dbaccess.DBError.throwSQLException(DBError.java:179)
```

**Workaround:** Find the latest database driver. If you are using an Oracle thin JDBC driver, you may receive a similar error message when trying to import a design that is 4KB or larger. If you are using an Oracle OCI driver and try to import a design that is 4KB or larger, the entire design will not be imported successfully resulting in functionality limitations.

## 3.1.17 About Document Generator

Document Generator is a tool that automatically generates customized letters and other documents from XML templates. You may design these documents according to your organization's needs and requirements.

Your TeamConnect users may have the need to write various types of documents on a frequent basis. For example, in the insurance industry, correspondence such as a claim notification needs to be sent whenever an insured client is involved in an automobile accident. Most likely, the same letter always needs to be sent, with certain information in the letter being entered specific to the client.

For another example, information such as addresses, involved parties, and possible violations of the law are different for each client. Instead of having to manually type the same document each time a

client is involved in an accident, a user can create customized letters from a template located in TeamConnect's Documents area.

To become a proficient document template designer, you need to invest some time in learning and understanding how the tool works and what it needs in order to generate a letter.

### 3.1.17.1 Document Generator Basics

Document templates give your users a quick and easy way to generate documents using data from TeamConnect. In order to create a document template, you must learn how to use the various tags available to automatically retrieve the correct data.

***Important:** In your design process, keep in mind that, while using Document Generator, the user must finish working with the document created during the current session. If the user navigates away from Document Generator before finishing, all work done on the document is lost.*

You may even define screens in which the user selects relevant data and manually enters values when generating a document. This definition becomes a part of a document template.

### Generated Letter Example

The following image is an example of a finished letter that was created by a user with the Document Generator. Everything from the date, to the salutation, to the Contact phone numbers, are dependent upon a document template.

Johnson, Mitch Claim  
06-02-0074

Mon, Jul 15, 02

Mitch Johnson  
105 Anaheim Street  
Long Beach, CA 90745

Dear Mr. Johnson,

ClientGuard was informed of your recent auto accident that occurred on 06/02/02. We are currently processing this claim and according to our records, the following individuals have filed claims regarding your auto accident:

Russel Emerson  
Maria Cortez  
Katherine Frost

The following passengers in your vehicle are claiming injury as a result of this accident:  
Katherine Frost

From the police report, we have determined that you were charged with the following:  
Drunk Driving

For the charges listed above, you were in violation of the following California statutes:

**Cal. Veh. Code §23152. Driving Under Influence of Alcohol or Drugs**

*(a) It is unlawful for any person who is under the influence of any alcoholic beverage or drug, or under the combined influence of any alcoholic beverage and drug, to drive a vehicle.*

In addition, based on the violations listed above, you were also guilty of breaking the following statutes:

**Cal. Veh. Code §23103. Reckless Driving**

*(a) Any person who drives any vehicle upon a highway in willful or wanton disregard for the safety of persons or property is guilty of reckless driving.*

The balance due for this claim and the services provided by ClientGuard is a total of \$1,500 dollars.

If you have any questions regarding your remaining balance, please contact any of the following claim associates here at ClientGuard:

Anita Diaz	Claim Associate	(310) 555-2293
Nathan Dean	Supervisor	(310) 555-4589

If you have any additional information you would like to include with your claim, please contact Nathan Dean (310) 555-4589.

Sincerely,

Jane Smeeth  
Claim Associate

For information about our services, visit us at [www.ClientGuard.com](http://www.ClientGuard.com).

js  
July 15, 2002

Document Generator - Finished Letter

## Object Model Structure for Identifying Data

It is important to know the areas in TeamConnect from where Document Generator can retrieve information. Knowing where to retrieve information, and the object for which you are creating the template, helps you select the appropriate fields, sub-objects, related objects, XML files, and so on, in order to design a functional document template.

TeamConnect's Object Model is your guide for identifying data that must be retrieved by your template. TeamConnect data is organized in a structure called the Object Model. Each object in the Object Model has its own attributes, and the objects have specific relationships with one another through certain attributes.

When you construct the tags that identify the data that will be inserted into generated documents, you use the relationships and object attribute names that are found in the Object Model to "navigate" to the data you need.

See [Object Model: Read This First](#) and the reference tables in later appendixes, for explanations of each attribute in the Object Model. You may use these references to help you construct the tags in your template. The reference tables familiarize you with the relationships between objects.

You may also find it helpful to become familiar with the Object Model by exploring it using the Object Navigator tool. For more details about using Object Navigator, see [Using Object Navigator](#).

## Object Dependency

Each document template used by Document Generator is based on a single object. Upon finishing a document template, you must upload the template into the Document Templates folder of a specific Object Definition within the Documents area of TeamConnect.

The object to which the template belongs is the point of origin for all tags that you use in the template to retrieve data. You must be able to identify the data that you want to be populated in the document through relationships between objects in the Object Model.

## Information Retrieval

Depending on what you specify in the document template, Document Generator is capable of retrieving information from the following sources:

- **TeamConnect Object Records** Document Generator can get information from custom fields, system fields, sub-objects, related objects, user accounts, and the system date.
- **Document Generator User** Document Generator can get information that the user manually enters into input pages or selects from filter pages. These screens appear in the user interface of Document Generator and are created by you within the document template.
- **Other XML files** Document Generator can get information found in XML files stored in the Documents area of TeamConnect.

## Document Template Structure

The document template is composed of three major sections. Each of these three sections are created with Document Generator tags that tell TeamConnect exactly what information to include in the generated document. The three sections of the document template and their respective tags are:

- **Content** Provides the necessary data that appears in the generated document, such as text that you write, and specific information extracted from TeamConnect system and custom fields.

The content section of the document template is made up of:

- Text
- Document Generator tags

- **Header** Provides data that asks the user, through filter and input pages, to make a choice (for example, choose a record, item, or enter information) that will be reflected in the generated document.

The header section of the document template is composed of the following Document Generator tags:

- `tc:header`
- `tc:filter`
- `tc:input`
- `tc:inputForm`

- **Document** The content and header of the document template are enclosed within the following tag:
  - `tc:document`

This tag has attributes that specify what application to launch in order to open the generated document, as well as the name of the template as it will appear to the user in Document Generator.

## Overview of Template Creation and Usage

When a user creates a document from the Document Generator, the procedure is very simple and straightforward. After invoking the Document Generator, the user is prompted by the user interface of the Document Generator to manually enter information into input pages, such as initials, paragraphs of client-specific information, and so on, and/or to select already existing TeamConnect information from filter pages, such as a Contact's name and address. The information that the user enters/selects appears in the final generated document.

Document Generator templates are created and put into use in the following way:

1. You (the Solution Developer) write a preliminary version of a template, or a preliminary version of the template is provided to you by an individual who is familiar with the business requirements for the template. This document includes placeholders identifying where data must be dynamically populated when a user generates a document based on the template.



2. You go through the template, replacing each placeholder with actual tags that can be read by the Document Generator, and then upload it to TeamConnect.
3. You test the template to insure that the tags used in the template are retrieving the correct data.
4. When the end user wants to generate a document based on the template, he or she will probably be working with a particular record. The user goes to that record's **Documents** block (or tab), opens the Document Generator tool, selects the template, enters any information required by the template, and clicks a button to generate the document.
5. The Document Generator reads through the template and retrieves the correct data to replace the tags in the template, and generates a document.
6. The user can then view, save, or print the generated document.

### 3.1.17.2 Creating Templates: Process Outline

There are multiple steps involved in creating the content, header, and document sections of the document template. These steps are in order within this document.

#### To create a document template

1. Write the document template text, with placeholders for all tags you will need. These placeholders will be replaced later with Document Generator tags. This step is described in detail in [Writing Document Template Text](#).
2. Replace placeholders with Document Generator tags that will retrieve data. This step is described in detail in [Document Generator Tags Summary](#).
3. Convert the file to XML and create user interface filter and input pages in the header section.  
  
If the document requires the user to select any data or enter text into the document, you must create filter and input pages that will be displayed for the user when Document Generator is invoked, by including the appropriate tc:filter and tc:input tags within the header section of the document template. This step is described in detail in [Using Document Template Headers](#).
4. Specify the application that will launch and open the generated document.  
  
After completing the previous steps, you will need to finalize the template by enclosing all of the text and tags in the appropriate outer tags. This step is described in detail in [Completing the Document Template](#).
5. Test the XML code of the document template.  
  
Before uploading the document template, you must test it to see if the XML code is well-formatted and fully functional. Internet Explorer, or any other XML viewing application, can be used for this testing.
6. Upload the document template.  
  
Upload your document template into the proper folder within the Documents area of TeamConnect. This step is described in detail in [Uploading Document Templates](#).
7. Test the document template within TeamConnect.

After uploading the document template to TeamConnect, test the template to verify that the tags in the template retrieve data correctly by generating a document. This step is described in detail in [Testing Document Generator Templates](#).

### 3.1.18 Using Document Template Headers

The header section of a document template is where you define filter pages and input pages for the end user. A filter page prompts the user who is generating a document to select records, or possibly sub-objects, from which data should be retrieved. An input page prompts the user to enter information that will be included in the generated document.

These pages are defined using the following tags:

- `tc:filter`

See [Defining Filter Pages](#).

- `tc:input`

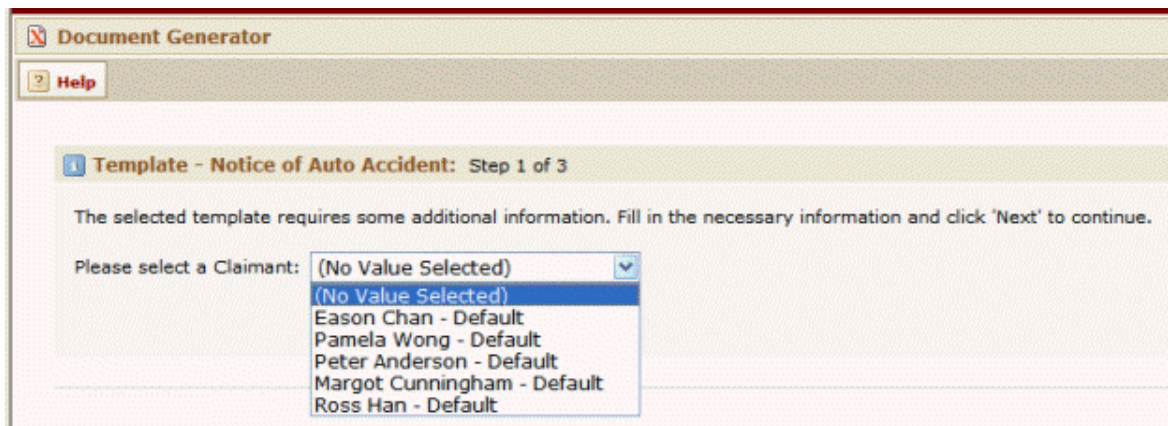
See [Defining Pages for User Input](#).

The header section of the document template may consist of zero, one or many instances of the `tc:filter` and `tc:input` tags, depending on how many different records need to be identified in order to retrieve data, as well as how many different pieces of information the user needs to input manually.

The `tc:filter` and `tc:input` tags used in the header section to define filter and input pages must have corresponding `tc:filter` and `tc:input` tags in the content section, which refer to the filter and input pages you defined. The `tc:filter` and `tc:input` tags in the content section specify the data to be retrieved by TeamConnect.

#### 3.1.18.1 Defining Filter Pages

You must create a filter page to allow Document Generator users to select one or more records containing additional information not available in the current object record. [The `tc:filter` Tag Attributes table](#) shows the result in the end-user interface.



Example: Document Generator - Filter Page

To create a filter page, use the `tc:filter` tag in the header section of the document template as follows:

- In the header section, the tc:filter tag is used to identify an object record from which the Document Generator should retrieve data.
- Each tc:filter tag that you use in the content section must reference a filter that is defined in the header section.

### tc:filter Tag Attributes

The tag attributes noted in the following table are the basic tags you need to create any type of filter page.

In addition to these tag attributes, all tc:filter tags in your header must use at least one other tag attribute to identify an object or sub-object.

#### tc:filter Tag Attributes

Required ?	Tag attribute	Possible attribute value	Description
Yes	name	"FilterName"	Specify a name that uniquely identifies the filter. A name that indicates which object the filter points to is helpful.  Important: <ul style="list-style-type: none"> <li>• This attribute value must exactly match the select="" attribute value of the corresponding tc:filter tag you entered in the content section.</li> </ul> You can either define the filter first or write the content section first and define the filter later, but you must use the same name of the filter in both places. <ul style="list-style-type: none"> <li>• Using spaces in the name of the filter is not recommended.</li> </ul>
No	hidden	"yes" or "no"	Indicate whether the filter should be displayed to the user as a filter page, or applied automatically in the background. By default, hidden="no".  When hidden="yes", you must provide the attribute list="yes" in the content section of the template where this filter is referenced. For details, see <a href="#">tc:filter</a> .
Yes	label	"UserInstructions"	Enter a label that appears as instructions for the user within the filter page.
Yes	displayString	"Data;Data;Data"	Specify how the selections appears for the user within the filter page. Multiple data items must be separated by semicolons (;).

			<p>For example, displayString="Contact/FirstName;Contact/Name;DefaultCategory/Name"</p> <p><b>Note:</b> You can only include system fields in the displayString. Custom fields cannot be included.</p>
<b>Yes</b>	displayFormat	"* * _ *"	<p>Specify the format of the information displayed by the displayString by typing an asterisk (*) for each data item, and typing any other desired characters, such as hyphens and spaces.</p> <p>Each data item retrieved by the displayString attribute must be represented by an asterisk (*) in the displayFormat attribute.</p>
<b>No</b>	multi	"yes" or "no"	<p>Type multi="yes" to allow users to select multiple items simultaneously from within the filter page.</p> <p>By default, only one item may be selected. You do not have to type multi="no" to allow only one selection in the filter page.</p> <p><b>Caution:</b> Do not use multi="yes" when referencing the filter as a main filter. For details, see <a href="#">Main Filters and Sub-Filters</a>.</p>
<b>Each attribute with an asterisk (*) is required</b>	*searchCondition	searchCondition identifies a related object. For details, see <a href="#">Filters for Related Objects</a> .	
	qualifier (used with searchCondition)		
	*test	test identifies a sub-object or a single object record selected in a system field. For details, see <a href="#">Filters for Sub-objects and Single Records</a> .	
	*entity  criteria (used with entity)	entity identifies an object that is not related to the main object. For details, see <a href="#">Filters for Non-Related Objects</a> .	

No	usedBy	usedBy and objectFrom allow an object identified in one filter to be based on the object identified in another filter. This is a way of navigating to the object record that you want the user to select by using filter pages. For details, see <a href="#">Main Filters and Sub-Filters</a> .
No	objectFrom	

### 3.1.18.1.1 Filters for Related Objects

When you want a filter page to identify an object that is directly related to either the main object or to an object identified by another filter, use the searchCondition tag attribute of tc:filter. This tag attribute is used to provide the user with a list of related objects.

The tag attributes for identifying related objects are described in the following table. Use these tag attributes in combination with the basic tag attributes in [The tc:filter Tag Attributes table](#).

#### searchCondition Tag Attribute of tc:filter

Tag attribute	Possible attribute value	Description
searchCondition	"RelatedObject"	<p>Type the name of the related object for which you want to define a filter. Type name of the object as follows:</p> <p>Appointment Account Expense History Involved Task Project (for child custom objects and embedded objects only)</p> <p><b>Note:</b> You must include the qualifier attribute, if you are defining a filter page to retrieve child objects.</p>
qualifier	"Application_ProjObjectDefinition_UniqueCode=UCOD"	<p>This tag attribute is required when you are defining a filter page for a child custom object, child account, or an embedded object.</p> <p>This attribute value identifies the child object using the following values:</p> <ul style="list-style-type: none"> <li>Path of the object attribute that uniquely identifies the child or embedded object, which is</li> </ul>

		<p>Application_ProjObjectDefinition_UniqueCode.</p> <ul style="list-style-type: none"> <li>• Unique code of the child or embedded object, which can be copied from the <b>Object Definition List</b> in the user interface.</li> </ul>
--	--	--

Let's see how the tc:filter tag and its attributes were used in the content and header sections in order to create the filter page shown in [the tc:filter Tag Attributes table](#). In this example, the tc:filter tag is defining a filter page where the user selects a record belonging to the Involved object, which is a related object of the main object Claim.

Notice how the select attribute value in the content section below exactly matches the name attribute value in the header section.

**Header**

```
<tc:filter name="InvolvedParty" searchCondition="Involved"
hidden="no" label="Please select Claimant" displayString="Contact/
FirstName;Contact/Name;DefaultCategory/Name" displayFormat="* * -
*" />
```

**Content**

```
<tc:filter select="InvolvedParty">
<tc:data select="Contact/FirstName" />
<tc:data select="Contact/Name" />
</tc:filter>
```

You may need to create a filter page where the user selects a child custom object of the main object. In the following example, the filter allows the user to select a child Claim record belonging to a Policy record (the main object). In the header, the qualifier tag attribute of tc:filter specifies which child object of the Policy object that the filter page must display.

**Header**

```
<tc:filter name="SelectedClaim" searchCondition="Project"
qualifier="Application_ProjObjectDefinition_UniqueCode=CLAM"
label="Select the appropriate claim"
displayString="Name;NumberString" displayFormat="* - *" />
```

**Content**

```
<tc:filter select="SelectedClaim">
<tc:data select="NumberString" />
<tc:data select="Name" />
</tc:filter>
```

For more examples of how searchCondition can be used, see [Main Filters and Sub-Filters](#). Also, see [Document Template Samples](#).

### 3.1.18.1.2 Filters for Sub-objects and Single Records

You may want users to select one or more sub-objects from which to retrieve data in a document. For example, you may want the user to select one address from the list of addresses belonging to a contact. In order to provide a filter page where the user selects a sub-object, use the test attribute in the tc:filter tag.

The test attribute can also identify a single record that is selected in a system field. Although you normally do not need a filter page that only gives the user one item to select, this becomes necessary when using tc:filter tags to navigate to the list of related records or sub-objects you want to provide to the user. For example, you may need a filter page where the user selects an Involved (related object) of the parent custom object record Policy (a single record). You will learn more about this in [Main Filters and Sub-Filters](#).

The test tag attribute accepts a path as its value. The path must meet the following requirements:

- It must be constructed of TeamConnect object attributes, according to the relationships between object attributes found in the Object Model.
- It must end with an object attribute that identifies either a single record selected in a system field of a record, or a sub-object list.

The following table describes how to use the test tag attribute for tc:filter. Use this tag attribute in addition to the basic tag attributes noted in [the tc:filter Tag Attributes table](#).

**test Tag Attribute of tc:filter**

Tag attribute	Possible attribute value	Description
test	"AttributePath" (attributes separated by underscores _ )	Use test to specify the path of object attributes separated by underscores ( _ ) that identifies the object attribute that you want the user to select in the filter page. The object attribute identified by the path may be one of the following types: <ul style="list-style-type: none"> <li>• A sub-object from a sub-object list.</li> </ul> <p><b>Tips:</b></p> <ul style="list-style-type: none"> <li>○ <i>Sub-object attributes contain the word "List" in the name of the object attribute (such as AssigneeList, AddressList, or LineItemList).</i></li> <li>○ <a href="#">Object Model: Read This First</a> (and the additional reference tables it points to) describes all of the sub-objects for each main TeamConnect object.</li> </ul> <ul style="list-style-type: none"> <li>• An object attribute that identifies a single record (such as the Parent or Contact for a custom object record).</li> </ul> <p>Displaying a single record in a filter page is useful when you need to navigate to a list you want the user to view. For more details about this scenario, see <a href="#">Single-Option Filters</a>.</p>

*Tip: An object attribute that identifies a single record has the data type "object" in this document.*

## Sample Values for Test Attribute

If you are identifying a sub-object list with the test attribute, the attribute and value might look like any of the following examples:

- test="AssigneeList" (in a custom object)
- test="AddressList" (in a contact)
- test="Contact\_AddressList" (in a contact-centric custom object)
- test="Project\_Contact\_PhoneList" (in an expense record, to its project's contact-centric field's phone numbers)
- test="Parent\_AssigneeList" (in a child custom object, to its parent's assignees)

If you are identifying a single record that is selected in a system field with the test attribute, the attribute and value could look like any of the following examples:

- test="Parent" (in a child custom object)
- test="Parent\_Contact" (in a child custom object, to its contact-centric parent)
- test="Contact" (in a custom object, to the contact in its contact-centric field)
- test="Project" (in an expense record)

**Header** `<tc:filter name="UserSelectedAssignee" test="AssigneeList" displayString="User/Contact/FirstName;User/Contact/Name;Type/Name" displayFormat="* * - *" label="Please select an assignee." />`

**Content** `<tc:filter select="UserSelectedAssignee">  
     <tc:data select="User/Contact/FirstName" />  
     <tc:data select="User/Contact/Name" />  
 </tc:filter>`

In the TeamConnect, the filter defined in the header tests the AssigneeList sub-object of the main object and displays a list of assignees in the filter page. In the content section, the first and last name of the assignee is generated in the document.

**Header** `<tc:filter name="SelectedAddress" test="Contact_AddressList" label="Please select the address of the Insured to which this letter should be mailed." displayString="Street;City;State;PostalCode" displayFormat="* *, * *" />`

**Content** `<tc:filter select="SelectedAddress">  
     <tc:data select="Street" />`



```
<tc:data select="City" />, <tc:data select="State"/> <tc:data
select="PostalCode" />
</tc:filter>
```

In the TeamConnect, the test attribute of tc:filter in the header identifies the list of addresses for the contact selected in the contact-centric field (Insured) of the main object record. In the content section, the address selected by the user appears in the generated document.

**Header** `<tc:filter name="ParentPolicy" test="Parent" label="Please select the parent Policy below and click Next." displayString="Name" displayFormat="*" usedBy="1" />`

In this example of a header only, the filter identifies the Policy selected as the parent in the main object record (the child Claim). This filter will be useful for then allowing the user to select the Policy's Involved parties in a subsequent filter. You will see how this is done in [Single-Option Filters](#).

### 3.1.18.1.3 Filters for Non-Related Objects

When you want the user to select a record that is unrelated to the main object record, use tc:filter in the header section to define a filter page where the user will make this selection. To create the filter page for an unrelated object, use the tag attributes described in [the tc:filter Tag Attributes table](#) as well as those described in the following table.

**Filter Tag Attributes - Creating Filter Pages for Non-Related Objects**

Tag attribute	Possible attribute value	Description
entity	"EntityName"	Type the name of the object from which you want the user to select a record. The user will be able to select from all available records, regardless of whether or not they are related to the main object record.  For example, using entity="Account" allows the user to select any available Account.  If the entity is a custom object, you must type entity="Project" and use the criteria tag attribute to identify the unique code of the custom object. For example: criteria="objectDefinition="SDTX".
criteria	Use any of the following components in the criteria attribute value: <ul style="list-style-type: none"> <li>category='DefaultCategoryName' (the name of the default category of the records you want available in the filter page)</li> </ul>	When using the entity tag attribute, you can specify criteria for which object records will be available to the user to select using the criteria tag attribute. This prevents all records in the system from being made available in the drop-down list in the filter page.

<ul style="list-style-type: none"> <li>• objectDefinition='ProjectUniqueCode' (if the non-related object is a project, use this component to provide the unique code of the project whose records you want available in the filter page)</li> <li>• name='RecordName' (the name of the record you want available in the filter page)</li> <li>• numberstring='RecordNumber' (the number of the record you want available in the filter page)</li> </ul>	<p>You can use any combination of the four components. For each component, you include the value that is desired in order for that record to be available to the user in the filter page.</p> <p>When using the criteria tag components, you must use the following conventions:</p> <ul style="list-style-type: none"> <li>• Component values are enclosed by single quotes.</li> <li>• Type a comma between components when using more than one.</li> <li>• Do not type any spaces between items or between commas in this tag attribute.</li> <li>• Enclose all components within a set of double quotes, as collectively they are a single attribute value.</li> </ul> <p>For example:</p> <pre>criteria="category='AutoDriver',objectDefinition='VEHI'"</pre>
---	--

**Header** `<tc:filter name="SelectAccount" entity="Account" criteria="category='Office Reserve'" hidden="no" label="Please select an Account" displayString="Name;Number" displayFormat="* - *" />`

**Content** `<tc:filter select="SelectAccount">  
     <tc:data select="Name" />  
     <tc:data select="DefaultCategory/Name" />  
 </tc:filter>`

In this example, all Account records from the database that have the default category Office Reserve are listed in the filter page, so that the user generating the document can select an account. In the content section, the name and default category of the account that the user selected are retrieved and included in the generated document.

#### 3.1.18.1.4 Main Filters and Sub-Filters

There are situations where you may want the user to select a record from which data must be retrieved, and this record is not directly related to the main object record. Instead, it is related to a related object of the main object. Or perhaps, you want the user to select a sub-object from a record that is related to the main object. You might even need the user to select a related object or sub-object of a record that is not related to the main object.

For example, if your main object is Policy, you may want the user to select parties Involved (a related object) in the Claim (a child custom object of Policy). This requires two filters: one to specify the Claim and one to specify the Involved.

Another example: you may want the user to select an address (a sub-object) of the contact selected in the Insured field (the contact-centric field) of a Policy (the main object).

These relationships sound complicated but they are not unusual. All of these situations require the use of multiple tc:filter tags, where one filter references another filter to indicate the desired relationship.

These filters are referred to as a main filter and a sub-filter. A sub-filter is a filter that references another filter, called a main filter, to indicate which object owns the sub-object that you are identifying.

## Points To Remember

You must know the following points about main filters and sub-filters:

- The main filter must be written before all of its sub-filters in the header section, so that the user first selects the record on which the sub-filters are based.
- The main filter must allow the user to only select one record. You cannot use the multi="yes" tag attribute in the main filter tag.
- The main filter can use searchCondition, test, or entity to identify the object record on which the sub-filter is based.
- The sub-filter can use searchCondition or test to identify a related object or a sub-object of the record identified in the main filter.

The following table describes the tag attribute that is used in the main filter. Use this tag attribute in addition to the basic tag attributes noted in [the tc:filter Tag Attributes table](#).

**Tag Attribute for Main Filter**

Tag attribute	Possible attribute value	Description
usedBy	"Integer"	Type the number of sub-filters that link to the main filter.  <i>Note: When the usedBy attribute is used, the objectFrom attribute must be used in a sub-filter.</i>

The following table describes the attribute needed for tc:filter in the header section when defining the sub-filter. Use this tag attribute in addition to the basic tag attributes noted in [the tc:filter Tag Attributes table](#).

**Tag Attribute for Sub-Filter**

Tag attribute	Possible attribute value	Description
objectFrom	"MainFilterName" "	Type the exact name that you used for the main filter.

## Related Object of a Related Object (Involved parties of Claim)

In this example, notice how the searchCondition of the main filter indicates which object it is pointing to: the Claim records of the Policy. The sub-filter is referencing the main filter, and the searchCondition attribute of the sub-filter is pointing to the Involved Parties of the Claim.

```

Header Main filter <tc:filter name="SelectedClaim" searchCondition="Project"
qualifier="Application_ProjObjectDefinition_UniqueCode=CLAM"
label="Select the appropriate claim"
displayString="Name;NumberString" displayFormat="* - *"
usedBy="1" />

Sub-filter <tc:filter name="ClaimInvolvedParties"
searchCondition="Involved" label="Please select the parties
involved in this Claim" displayString="Contact/
FirstName;Contact/Name;DefaultCategory/Name" displayFormat="*
* - *" multi="yes" objectFrom="SelectedClaim"/>

Content <tc:filter select="SelectedClaim">
<tc:data select="NumberString" />
<tc:data select="Name" />
</tc:filter>
<tc:filter select="SelectedInvolvedParties">
<tc:data select="Contact">
<tc:data select="FirstName" />
<tc:data select="Name" />
</tc:data>
<tc:data select="DefaultCategory/Name" />
</tc:filter>

```

## Sub-Object of a Related Object (Addresses of Involved Contact)

In this example, notice how the searchCondition of the main filter indicates which object it is pointing to: the Contact records of the Involveds. The sub-filter is referencing the main filter, and the test attribute of the sub-filter is pointing to the AddressList of the Contacts.

```

Header Filter <tc:filter name="SelectedInvolvedContact"
searchCondition="Involved_Contact" usedBy="1"
displayString="FirstName;Name" label="Please select an
involved" />

Sub-filter <tc:filter name="InvolvedContactAddress"
objectFrom="SelectedInvolvedContact" test="AddressList"
displayString="Type/Name;Street;City" displayFormat="* - *"
label="Please Select the Involved's Address." />

Content <tc:filter select="SelectedInvolvedContact">
<tc:data select="FirstName" />
<tc:data select="Name" />
</tc:filter>

```

```
<tc:filter select="InvolvedContactAddress">
  <tc:data select="Type/Name" />
  <tc:data select="Street" />
  <tc:data select="City" />
</tc:filter>
```

## Single-Option Filters

In certain circumstances, you will need to provide a filter page that only gives the user one record to select. This is necessary when you want the user to make a selection from a related object or sub-object of an object that is not the main object. For example, suppose you have a contact-centric custom object, Claim, and you want the user to select an address belonging to the contact that is selected in the contact-centric field for the Claim.

In this situation, you have to provide the user with a filter page where the user selects the contact record. This is the main filter, and it must be the only available option. In the next filter page, the sub-filter, the user selects the desired address from the list of addresses for that contact.

The main filter, which only provides one selection to the user, uses the test tag attribute of tc:filter to identify a single record, such as:

- The parent custom object
- The contact in the contact-centric field
- The custom object record with which a task record is associated

A scenario like this is demonstrated in the following example:

<b>Header</b>	<b>Filter</b>	<pre>&lt;tc:filter name="RetrieveParent" test="Parent" label="Please select the parent policy below." displayString="Name" displayFormat="*" usedBy="1" /&gt;</pre>
	<b>Sub-filter</b>	<pre>&lt;tc:filter name="SelectedInvolveds" searchCondition="Involved" label="Please select the parties involved in the parent policy" displayString="Contact/ FirstName;Contact/Name;DefaultCategory/Name" displayFormat="* * - *" objectFrom="RetrieveParent" /&gt;</pre>
<b>Content</b>		<pre>&lt;tc:filter select="SelectedInvolveds"&gt;   &lt;tc:data select="Contact"&gt;     First Name: &lt;tc:data select="FirstName" /&gt;     Last Name: &lt;tc:data select="Name" /&gt;&lt;/tc:data&gt;     Role: &lt;tc:data select="DefaultCategory/Name" /&gt; &lt;/tc:filter&gt;</pre>

In TeamConnect, the sub-filter indicates that it can display the Involved records of the parent Policy selected by the user in the main filter. In the content section, information about the Involved party, which the user selected in the filter page, appears in the generated document.

### 3.1.18.2 Defining Pages for User Input

Whenever you need to include information in a document that must be typed by the user who is generating the document, you must use an input tag. In the end-user interface of Document Generator, `tc:input` tags nested within the header section of the document template define input fields. These fields request text information from the user, which can then be placed directly into the generated document.

For example, your users may need to generate letters in which a paragraph explaining the status of a claim must be typed by the user, while the rest of the letter is automatically generated. In this case, you have to create an input page where the user types the explanation for the generated document.

**Note:** When nesting the `tc:input` or `tc:inputForm` tags within the `tc:header` tag, you must make sure that each `tc:input` tag in the content section corresponds to a tag in the header section.

The following list shows the different types of input tags you may use in the header section of the document template:

<code>tc:input</code>	Requires the user to enter text in a single-entry field within one input page in the end-user interface of the Document Generator. For details, see <a href="#">Creating User Input Fields</a> .
<code>tc:inputForm</code>	Requires the user to enter the appropriate information in multiple-entry fields within one input page in the end-user interface of the Document Generator. For details, see <a href="#">Displaying Multiple Entry Fields in One Input Page</a> .

#### 3.1.18.2.1 Creating User Input Fields

The `tc:input` tag creates a single entry field within one input page (see [the single simple text field example image](#)). Each entry field will appear on a separate page.

The following image provides a list of the attributes and possible attribute values that can be used with the `tc:input` tag within the header section of the document template.

**tc:input attributes in Header Section**

Tag attribute	Attribute value	Description
<code>name</code>	"FieldName"	Specify a name that uniquely identifies the field you are creating.  <b>Important:</b> This attribute value must exactly match the <code>select=""</code> attribute value of the corresponding <code>tc:input</code> tag you entered in the content section. The <code>name</code> attribute must always be used when the <code>tc:input</code> tag is nested within the header section.

label	"FieldLabel"	Enter a label that will display instructions for the user within the input page.
size	"FieldSize"	Specify the maximum size, in characters, of the entry field within the input page. This is the length of the field but does not control the number of characters the user may enter.
maxChars	"MaxNumCharacters"	Specify the maximum number of characters that the user can enter into the input field.  <b>Note:</b> <i>This attribute is not available when the input field is scrollable (when multi attribute is set to "yes").</i>
multi	"yes"	Enter "yes" as the attribute value to allow users to enter information into a scrollable text field within the input page (see <a href="#">the single scrollable text field example image</a> ). This is useful when a large amount of text needs to be entered by the user.  <b>Note:</b> <i>By default, users will always enter information into a one-line text field (see <a href="#">the single simple text field example image</a>). If you would like users to have only a text field, simply do not use the multi attribute, there is no need to enter "no" as the attribute value.</i>
cols	"ColumnWidth"	Specify the width of the columns of the scroll text field. The number you enter represents the number of columns. This attribute is used in conjunction with the multi attribute.
rows	"RowLength"	Specify the length of rows in the scroll text field. The number you enter represents the number of rows. This attribute is used in conjunction with the multi attribute.

Let's see how the tc:input tag and its attributes were used in the content and header sections in order to create the input page with a simple text field shown in the following image. Notice that the select attribute value in the content section exactly matches the name attribute value in the header section.

**Header** `<tc:input name="UserInitials" label="Enter your initials here." size="15" />`

**Content** `<tc:input select="UserInitials" />`

**Example: Document Generator - Input Page - Single Simple Text Field**

Next, you may check how the `tc:input` tag and its attributes were used in the content and header sections in order to create the input page with a scrollable text field shown in the following image.

**Header** `<tc:input name="ClaimStatus" label="Enter the status of the claim." size="50" multi="yes" cols="45" rows="2"/>`

**Content** `<tc:input select="ClaimStatus" />`

**Example: Document Generator - Input Page - Single Scrollable Text Field**

### 3.1.18.2.2 Displaying Multiple Entry Fields in One Input Page

The `inputForm` tag lets you display multiple entry fields within one input page (see [the multi-entry field example image](#) for example). This task is accomplished by nesting `tc:input` tags within a `tc:inputForm` tag. You can nest multiple `tc:input` tags as needed. Each `tc:input` tag can use any of the attributes described in [the tc:input attributes in Header Section table](#).

#### `tc:inputForm` attributes in Header Section

Tag attribute	Attribute value	Description
name	"FormName"	Specify a name that uniquely identifies the input page you are creating. <ul style="list-style-type: none"> <li>• Within the <code>tc:inputForm</code> tag, you must nest each <code>tc:input</code> tag.</li> <li>• In the content section, the value for the <code>select</code> attribute of the corresponding <code>tc:input</code> tag must be the name attributes of both the <code>tc:inputForm</code> and <code>tc:input</code> in the header section, separated by a period (.).</li> </ul>



label	"FormLabel"	Enter a label that will display instructions for the user within the input page.
-------	-------------	--

Let's see how the `tc:inputForm` tag and its attributes were used in the content and header sections in order to create the input page shown in [the single simple text field example image](#). Each `tc:input` tag is nested within the `tc:inputForm` tag. Notice that the `select` attribute value in the content section contains the name attribute value of both the `tc:inputForm` and `tc:input` tags in the header section, separated by a period (.).

**Header**

```
<tc:inputForm name="UserInput" label="Enter the following information">
  <tc:input name="UserInitials" label="Initials" size="3"
    maxChars="3" />
  <tc:input name="LetterDate" label="Date letter will be sent"
    size="15" />
</tc:inputForm>
```

**Content**

```
<tc:input select="UserInput.UserInitials" />
<tc:input select="UserInput.LetterDate" />
```

Example: Document Generator - Input Page - Multi-Entry Field

### 3.1.19 Using Document Generator Templates

The content section of a document template consists of the following major components:

- Text**                      The general information that is the same across all documents created from the same template. The text you include in the content of the document template is static information that never changes and appears in the generated document exactly as you typed and formatted it in the original template.
- Document Generator Tags**      Retrieve data from TeamConnect records. The document template tags you include in the document template are dynamic information that always change according to the data the tags retrieved from TeamConnect records. You learn more about Document Generator tags in [Document Generator Tags](#). The tags are replaced with specific data in the generated document.

### 3.1.19.1 Writing Document Template Text

The first step in creating a document template is to write the text that will appear in the generated document. This text can be written in almost any word processing program that supports RTF (Rich Text Format). Microsoft Word™ and Corel WordPerfect™ are highly recommended. You can also use Microsoft WordPad™.

#### About Placeholders

While writing the text of the document template, you should use placeholders within the text to mark for yourself where to insert tags later. This method of developing your document template will help you divide the tasks of composing the document and coding it.

Each placeholder should mark where one piece of data should be inserted. For example, if you want the template to retrieve the first name of an Involved Contact for a Claim project, you would enter a placeholder in your template like this:

```
<First Name of Involved Contact for this Claim>
```

#### Tips on Formatting and Creating Placeholders

- Create placeholders that identify as specifically as possible what data should be retrieved, especially if you will be delivering the document to another individual for converting it to XML.
- All the formatting that you use in creating the text and placeholders of the document content, including line returns, will be carried over to the generated document. If you enter a line return in a series of nested tags, a line return will be added in the final generated document. For example, in [the text and placeholders image](#), notice how the placeholder tags are on different lines. This is how the retrieved data will appear in the final generated document.
- You can also place tags within the header and footer sections of a Microsoft Word document, and/or within a table, as long as you make sure that the flow of the text in the word processing application places the tags in the correct order.

The following image is an example of text and placeholders that were created in Microsoft Word. This particular template was used in creating the claim notification letter shown in [the finished letter image](#). Notice how the layout of the template is designed according to the desired format of the final document.

<Claim Name>  
<Claim Number>

<Date of Letter>

<Insured's First Name> <Insured's Last Name>  
<Insured's Street Address>  
<Insured's City>, <Insured's State> <Insured's Zip Code>

<Insured's Salutation>

ClientGuard was informed of your recent auto accident that occurred on <Date of Accident>. We are currently processing this claim and according to our records, the following individuals have filed claims regarding your auto accident:  
<List all claimants here>

The following passengers in your vehicle are claiming injury as a result of this accident:  
<Involved Injured Passenger Contact's First Name> <Involved Injured Passenger Contact's Last Name>

From the police report, we have determined that you were charged with the following:  
<Drunk Driving custom field in Claim>  
<Speeding custom field in Claim>  
<Reckless Driving custom field in Claim>  
<Loud Music custom field in Claim>

For the charges listed above, you were in violation of the following California statutes:  
<If Drunk Driving is true, import DrunkStatute.xml here>  
<If Speeding is true, import SpeedingStatute.xml here>  
<If Loud Music is true, import LoudMusicStatute.xml here>

In addition, based on the violations listed above, you were also guilty of breaking the following statutes:  
<If any of the above are true, then import the file RecklessStatute.xml here>

The balance due for this claim and the services provided by ClientGuard is a total of <Subtract Amount Paid custom field from Amount Due custom field> dollars.

If you have any questions regarding your remaining balance, please contact any of the following claim associates here at ClientGuard:

<Assignee first name> <Assignee last name>	<Role>	<Phone Number>
--	--------	----------------

If you have any additional information you would like to include with your claim, please contact  
<Supervisor Assignee first name> <Supervisor Assignee last name> <Phone Number>

Sincerely,

<User who created letter>  
<User Title>

<Any additional comments>

<Initials of user who created letter – input by user>  
<Date letter will be sent – input by user>

#### Document Generator Template - Text and Placeholders

### 3.1.19.2 Document Generator Tags

Document Generator tags are the tags used to extract data from existing TeamConnect records. You will replace the placeholders you created in the content section (see [the text and placeholders image](#)) with the actual Document Generator tags. These tags tell the Document Generator exactly what data to retrieve.

The following points will help you understand and work with Document Generator tags:

- All Document Generator tags begin with the prefix tc:.
- Tags, tag attributes, and tag attribute values are case sensitive.
- Within a tag, always capitalize the first letter of each object attribute, even though it is not capitalized when you are working in other areas of TeamConnect or with other TeamConnect tools. This is true for all Document Generator tags.
- The tags are designed so that you have a great deal of flexibility in how you want to use them. Any tag can be nested within almost any other tag, depending on what you want to accomplish. This also means that many of the tags of the same type can be nested within each other.
- Standards for writing well-formed XML apply when you are writing Document Generator tags in your document template.

#### 3.1.19.2.1 Document Generator Tags Summary

After you have written the desired text for your document template, you must replace the placeholders you created with the actual Document Generator tags that will retrieve the necessary data.

Document Generator tags can be classified into the following main groups:

- **Tags that navigate**

Navigating tags help you identify where data should come from by "navigating" through TeamConnect's Object Model.

- **Tags that retrieve data**

Once you have navigated to the appropriate object in the Object Model using navigation tags, these tags retrieve the data.

The following table summarizes the Document Generator tags that must be used depending on the type of data you would like to include in the generated document.

**Document Generator Tags**

To retrieve data from	Use these tags	Used to navigate	Used to retrieve data	Where to learn more
<b>System Fields</b>	tc:data Use this tag to navigate through system fields or to retrieve data from a system field.	x	x	<a href="#">tc:data</a>

<b>Custom Fields</b>	<code>tc:detail</code> Use this tag to navigate through custom fields or to retrieve data from a custom field.	<b>x</b>	<b>x</b>	<a href="#">tc:detail</a>
<b>Sub-objects</b> (user-selected)	<code>tc:filter</code> Use this tag to navigate to sub-objects which are selected by the user, from which to retrieve data.  For example, use this tag to identify an address selected by the user to be included in a document.  <b>Note:</b> <i>You need to define the filter in the header section. You will learn about the header section later.</i>	<b>x</b>		<a href="#">Filters for Sub-objects and Single Records</a>
<b>Sub-objects</b> (non user-selected)	<code>tc:loop</code> Use this tag to navigate to sub-objects. For example, the list of assignees for a project is a sub-object.  When you use this tag, the selection of data retrieved is predefined in the template, so the data is identified automatically without intervention by the user.	<b>x</b>		<a href="#">tc:loop</a>
<b>Related Objects</b> (user-selected)	<code>tc:filter</code> Use this tag to navigate to related objects when you want the user to have the ability to select a related object from which to retrieve data.  For example, use this tag to let the user select which Involved of a project to include in a document.  <b>Note:</b> <i>You will need to define the filter in the header section. You will learn about the header section later.</i>	<b>x</b>		For referencing a filter in the content of the document template: <a href="#">tc:filter</a>  For defining filters: <a href="#">Defining Filter Pages</a>
<b>Related Objects</b> (non user-selected)	<code>tc:search</code> Use this tag to navigate to related objects. For example, if the main object is a custom object, use it to navigate to	<b>x</b>		<a href="#">tc:search</a>

	<p>the Involves or child objects of the custom object.</p> <p>When you use this tag, the data is identified automatically without intervention by the user because the selection of data retrieved is predefined by you in the template.</p>			
<b>Current Date</b>	<p><code>tc:date</code></p> <p>Use this tag to get the current system date.</p>		<b>x</b>	<a href="#">tc:date</a>
<b>Current User</b>	<p><code>tc:user</code></p> <p>Use this tag to get information about the current user.</p>		<b>x</b>	<a href="#">tc:user</a>
<b>XML Files</b>	<p><code>tc:file</code></p> <p>Use this tag to import content from a different file into the generated document. You can use this tag to include content only under certain circumstances by using it with the <code>tc:conditional</code> tag.</p>		<b>x</b>	<a href="#">tc:file</a>
<b>Conditional Content</b>	<p><code>tc:conditional</code></p> <p>Use this tag to dynamically change the content of the generated document depending on certain conditions.</p>			<a href="#">tc:conditional</a>
	<p><code>tc:block</code></p> <p>Use this tag to group conditional tags together so that more than one condition will trigger the same content to be generated.</p>			<a href="#">tc:block</a>
<b>Calculating Two Numerical Values</b>	<p><code>tc:compute</code> and <code>tc:operand</code></p> <p>Use this tag to make a calculation using two number fields or a number field and a value that you provide in the template.</p>			<a href="#">tc:compute</a> <a href="#">tc:operand</a>
<b>End-User Input</b>	<p><code>tc:input</code></p> <p>Use this tag to include text that is entered by a user in an input page.</p>			<a href="#">tc:input</a>

	<p><b>Note:</b> You will need to define the input page in the header section of the document template. You will learn about the header section later.</p>			
--	---	--	--	--

### 3.1.19.2.2 tc:data

The tc:data tag is used to do both of the following actions:

- Retrieve data from TeamConnect system fields.
- Point the Document Generator to different object records from which to retrieve data by nesting tc:data tags, thereby "navigating" to the system fields you need.

### What Can Be Retrieved with tc:data

The tc:data tag can retrieve data from the following areas:

- **Main object record** Use the tc:data tag independently to retrieve single instances of data from TeamConnect system fields of the main object record.
- **Sub-objects** Nest the tc:data tag within one of the following tags:
  - tc:loop tag, to automatically retrieve data from sub-objects.
  - tc:filter tag, to have the user select the sub-object from which to retrieve the data.

These tags retrieve data from TeamConnect system fields of sub-objects, such as project assignees and categories. For more information, see:

  - [tc:loop](#) or [tc:filter](#)
  - [Filters for Sub-objects and Single Records](#)
- **Related objects** Nest the tc:data tag within one of the following tags:
  - tc:search tag, to automatically retrieve data from related objects.
  - tc:filter tag, to have the user select the related object record from which to retrieve the data.

These tags retrieve data from TeamConnect system fields of related objects (such as involved, appointment, child custom objects, embedded custom objects and child accounts). For more information, see [tc:search](#) and [tc:filter](#).
- **Objects not related to the** Nest the tc:data tag within the tc:filter tag to retrieve data from objects that are not related to the main object. For more information, see [Filters](#)

main object [for Non-Related Objects.](#)

- System object related to current object through a system field** Nest the tc:data tag within another tc:data tag when a system field relates the current object to a system object, in order to retrieve data from the related system object, such as contact. For an example, see [Nesting to Navigate](#).
- Custom object record selected in a custom field** Nest the tc:data tag within the tc:detail tag to retrieve data from system fields of a record selected in a custom field of type Custom Object. For an example, see [Nesting to Navigate](#).

## Points To Remember

When using tc:data, you must know the following concepts:

- The tc:data tag will retrieve data from the specified system field of the current object. When the tc:data tag is not nested, it points to data from the main record for which the document is being generated.
- The tc:data tag requires one tag attribute, select. It can also accept several other tag attributes, which are used to format the data item that the tag retrieves.
- When using tc:data to retrieve data, the object attribute you are retrieving with tc:data cannot be of the data type Object. An object is actually a collection of attributes that identify data, so it cannot itself be inserted into a generated document. If you attempt to do this, the generated document will contain a large amount of nonsensical code. If you need specific information from that object, you should nest tc:data tags in order to navigate to the specific data you need.

*Tip: [Object Model: Read This First](#) and the additional reference tables it points to indicate the data type of each object attribute in the **Data Type** column of all tables. You can see which attributes have the data type **Object** and avoid retrieving them in a template.*

- If you need to retrieve data from a custom field, use tc:detail. For more information, see [tc:detail](#).

## Required tc:data Tag Attribute

Whether using the tc:data tag on its own to retrieve data from the currently selected record, or if nesting the tc:data tag to retrieve data from sub-objects or related objects, the select attribute must always be used with the tc:data tag.

Required tc:data Tag Attribute

Tag attribute	Attribute value	Description
select	"ObjectAttribute"	Specify the system field from which the Document Generator will retrieve data by typing the name of the object attribute from the Object Model, making sure to capitalize the first letter of the name.



The object attributes that correspond to all system fields are found in the [Object Model: Read This First](#) and in the additional reference tables it points to.

**Example:** `<tc:data select="NumberString" />`

The select attribute takes a system field's name as its value. In the example above, NumberString is the object attribute that corresponds to the system field, **Number**. Therefore, the data retrieved by the Document Generator will be the number of the record selected by the user.

### Optional tc:data and tc:detail Tag Attributes

The following attributes are optional attributes that can be used within the tc:data and tc:detail tag. Each optional attribute is applicable to a specific field type (such as number, date, text). These tags control the format of the data that is generated in the document.

**Caution:** If you try to assign an attribute to a field that is not of the appropriate type, TeamConnect is not able to generate the document and displays an error.

Optional Tag Attributes for tc:data and tc:detail

Tag attribute	Applicable field types	Possible attribute values	Default	Description	Example
numberFormat	Number	Examples: #,###.00 #,##0.00 ####.## ####.### ####	If the format of the digit was specified with #, the number will drop all trailing and unnecessary zeros.  If the format of the digit was specified with a 0, the digit will be displayed as it was typed.	Specifies the format of number fields, including the placement of the comma and decimal point.  Using a 0 instead of # tells the Document Generator to display the digit, even if it is a 0. When generating financial information, you should use a format such as #,##0.00.	<code>&lt;tc:data select="Unit Price" numberFormat="#,##0.00" /&gt;</code>
percentage	Number	###% ##0.0#%	1/100	Specifies that the number should be displayed as a percentage, and	<code>&lt;tc:data select="CompletedPercent" percentage="###%" /&gt;</code>

				<p>whether decimals should be included.</p> <p>Using a 0 instead of # tells the Document Generator to display the digit, even if it is a 0.</p>	
dateFormat	Date	See <a href="#">the Possible Values for dateForm</a> at <a href="#">Tag Attribute table</a> .	MM/dd/yyyy	Specifies the display format of date and time values.	<pre>&lt;tc:data select="DueOn" dateFormat="dd/MM/yyyy" /&gt;</pre>
showTime	Date	yes no	no	<p>Shows the time next to the date. Only necessary if the value for dateFormat does not display the time.</p> <p><b>Note:</b> <i>This tag should only be used if dateFormat is not specified.</i></p>	<pre>&lt;tc:data select="DueOn" showTime="yes" /&gt;</pre>
timeZone	Date	yes no	no	<p>Shows the time zone (in parenthesis) next to the date. The time zone is from the current user's settings.</p> <p><b>Note:</b> <i>This tag attribute is only necessary if the value for dateFormat does not display the time zone.</i></p>	<pre>&lt;tc:data select="DueOn" showTime="yes" timeZone="yes" /&gt;</pre>

case	Text	upper lower	Unless an attribute value is specified, the text appears as you typed it.	Specifies that the text should be displayed in all uppercase or all lowercase letters.	<tc:data select="Location" case="upper" />
true/ false	Check- Box	Any text value.	For true, the custom field label.  For false, blank.	Specifies the display values in the document for the true and false states of a check-box.	<tc:data select="IsAutoPost " true="Automatic posting is enabled" false="Automatic Posting is disabled" />

## dateFormat Tag Attribute Values

The following values are the available values for the **dateFormat** attribute.

### Possible Values for dateFormat Tag Attribute

Value	Example output
yyyy.MM.dd G 'at' hh:mm:ss z	2001.12.08 AD at 03:08:17 PDT
MMMM d, yyyy	December 8, 2001
EEE, MMM d, yy	Tue, Dec 8, 01
hh:mm:ss	03:08:00
h:mm a	3:08 PM
hh 'o clock' a, zzzz	03 o clock PM, Pacific Daylight Time
K mm a, z	3 08 PM, PDT
yyyy.MMMMM.dd GGG hh.mm aaa	2001.December.08 AD 03:08 PM
MM/dd/yyyy	12/08/2001
<b>Note:</b> This is the default value of this tag attribute.	
dd/MM/yyyy	08/12/2001

yyyy/dd/MM

2001/08/12

The date and time formats can be combined to create the desired result. For example, you can combine MM/dd/yyyy with h:mm a (dateFormat="yyyy/dd/MM h:mm a") to display the date as 12/08/2001 3:08 PM.

## Navigating Within tc:data Tag

In tc:data tags, you can navigate throughout the Object Model to retrieve data from other objects without opening and closing separate tags for each object. If you only want to retrieve one data item from an object, rather than multiple items, it may be easier to write one tag that points to the data you need rather than nesting to navigate.

A path to a data item, or object attribute, that exists in a different object is retrieved in a single tag by writing a path to it using the object attributes that serve as relationships between objects. These object attributes are noted in [Object Model: Read This First](#) and in the additional reference tables it points to, with arrows (-->).

Follow these guidelines when navigating to an attribute within a tc:data tag:

- Separate each object attribute in the path with a forward slash (/).
- Capitalize the first letter of all object attributes in the path.
- Close the tag to point the Document Generator back to the previous object that it was pointing to.

For example:

```
<tc:data select="Parent/Contact/
FirstName" />
```

Retrieves the first name of the Contact selected for the parent record's contact-centric field.

```
<tc:data select="Project/Name" />
```

Retrieves the name of the Project record to which a Task record is related.

```
<tc:data select="MainAssignee/Type/
Name" />
```

Retrieves the name of the main assignee's role for a record.

```
<tc:data select="MainAssignee/User/
Contact/FirstName" />
```

Retrieves the first name of the main assignee for a Project record.

### 3.1.19.2.3 tc:detail

The tc:detail tag retrieves data from a TeamConnect custom field. You can use tc:detail to retrieve custom field values from the currently selected record, or you can nest it, thereby "navigating" to retrieve values from objects that you specify.

Keep the following points in mind when working with this tag:

- To retrieve custom fields of related objects, nest the tc:detail tag within the tc:search or tc:filter tag (for details, see [tc:search](#) or [tc:filter](#)).
- To retrieve custom fields from a system or custom object to which a custom field links, nest the tc:detail tag within the tc:data tag (for details, see [Nesting to Navigate](#)).
- The tc:detail tag uses the same attributes as the tc:data tag for formatting the data retrieved by the tag. See [the Optional Tag Attributes for tc:data and tc:detail table](#) for a list of additional attributes.

The following table provides a description of the required attribute of tc:detail.

**Required tc:detail Tag Attribute**

Tag attribute	Attribute value	Description
select	"CategoryTreePosition/ FieldName"	Specify the full category tree position and the field name of the custom field from which TeamConnect will retrieve data.  The full category tree position includes the unique code of the object (the root) and the tree position of each category in the tree, separated by underscores ( _ ).

**Example:** `<tc:detail select="CLAM_AUTO/Personal Injury" />`

In the previous example, **Personal Injury** is a custom field found in a Claim record in the category **Auto**. Since this custom field is a check-box, the label of the check-box is the data extracted and inserted into the generated document if the check-box is checked. Additional categories on the same level are other types of claims, for example, **General Liability** and **Home**.

In this example, CLAM is the unique code of the custom object **Claim**. If you are referencing a custom field in the top node (root) of a custom object's category tree, the tree position of the custom field is the unique code of the object.

The options in the resulting drop-down list in Document Generator appear to the end user as follows:

```

Claim
  Auto
    Personal Injury
  General Liability
  Home
  
```

## Custom Field Display Formats

By default, data retrieved from each type of custom field appears in a particular format in the generated document. However, you can modify the way certain data types appear using the tag attributes in [the Optional Tag Attributes for tc:data and tc:detail table](#).

The following table indicates how each type of data will be displayed by default.

**Custom Field Display Formats in Generated Document**

Custom field type	Display format of values in generated document
<b>Text</b>	Exact data entered in field.
<b>Number</b>	Exact data entered in field.
<b>Date</b>	MM/dd/yyyy
<b>Check-Box</b>	For <b>true</b> , the custom field label. For <b>false</b> , blank.
<b>Memo Text</b>	Exact data entered in field.
<b>Custom Object</b>	The title of the custom object record. The exact format depends on the <b>Automatic Numbering Pattern</b> defined in the custom object definition. This pattern may be any of the following: <ul style="list-style-type: none"> <li>• Name only</li> <li>• Number only</li> <li>• Name - Number</li> <li>• Number - Name</li> </ul>
<b>Involved</b>	The full name of the contact who is the involved, in the following format: <code>Name, FirstName</code>
<b>List</b>	The name of the lookup table item.

#### 3.1.19.2.4 tc:loop

The tc:loop tag is used to navigate to sub-objects of the current object. The tc:loop tag does not retrieve any data by itself. It only specifies the sub-object from which TeamConnect will retrieve data. Essentially, it navigates from the current object to the sub-object that you specify.

When you close the tc:loop tag, you return to the object to which the sub-object belongs. However, while inside the tc:loop tag, you can continue to navigate to other objects by nesting additional tags. You can also nest tc:loop within other tags as needed, such as tc:filter or tc:conditional.

To retrieve data from system fields of the sub-object, you must nest the tc:data tag within the tc:loop tag.

When nesting the tc:data tag within the tc:loop tag, the following data can be retrieved:

- Data from all instances of a sub-object.

For example, you may want to include all the names and phone numbers of all the Assignees for an auto claim. For details, see [Retrieving Data from All Instances of Sub-object](#).

- Data from only instances of the sub-object that match a criterion that you define.

For example, you may want to include only the name and phone number of Assignees with a certain role, such as Supervisor. For details, see [Retrieving Data with tc:loop Using Qualifiers](#).

## Automatically Selected vs. User-Selected Sub-objects

The tc:loop tag is useful for selecting instances of a sub-object from which data should be retrieved automatically in a document template. However, there are some cases where it is necessary for the user who is generating a document to select a sub-object from which to retrieve data.

For example, your document template may require that the user select an assignee, address, or phone number. In order to provide the user with a list of instances of a sub-object, you would not use tc:loop. Instead, you would use tc:filter to define a filter page where the user would make the appropriate selection. You will learn more about how to do this in [Filters for Sub-objects and Single Records](#).

### 3.1.19.2.4.1 Retrieving Data from All Instances of Sub-object

To retrieve data from all instances of a sub-object, regardless of the type, you only need to use the select attribute with the tc:loop tag and then nest the tc:data tag within the tc:loop tag.

For example, you can use this method to list all Assignees for a project, regardless of the assignee role (type). The Document Generator will print the data you specify for all of the assignees of the project.

#### Usage of tc:loop - Retrieving Data from All Instances of Sub-object

Tag attribute	Attribute value	Description
select	"SubObjectList"	<p>For the value of the select attribute, type the name of the object attribute that links to the desired sub-object. For example, if you want the Assignees sub-object for a Project, type AssigneeList.</p> <p>See <a href="#">Object Model: Read This First</a> and the additional reference tables it points to, for a complete list of sub-objects and to find the object attribute names that identify them. Typically, attributes that link to sub-objects have the suffix "List" (AssigneeList, PhaseList, and so on).</p> <p><b>Note:</b> The tc:data tag must be nested within the tc:loop tag in order to retrieve any data.</p>

**Example:**

```
<tc:loop select="AssigneeList">
  <tc:data select="User/Contact/FirstName" />
  <tc:data select="User/Contact/Name" />
  <tc:data select="Type/Name" />
</tc:loop>
```

```
<tc:data select="User/Contact/DefaultPhone/PhoneString"/>
</tc:loop>
```

The select attribute of tc:loop takes a sub-object, such as an assignee or a category, as its value. Data values from the system fields of the sub-object specified by tc:loop are retrieved by tc:data tags nested within tc:loop.

In the example given, the data retrieved by TeamConnect is the first and last name, title, and phone number of all the users who are Assignees. The sub-object Assignee is specified by the tc:loop tag. The first and last name, title, and phone number of all the users are specified by the nested tc:data tags. Notice that this information is really stored in the user's Contact record, so the tags are actually navigating to the Contact record.

#### 3.1.19.2.4.2 Retrieving Data with tc:loop Using Qualifiers

To retrieve data from certain instances of the sub-object that meet a certain criterion, rather than all instances of the sub-object, you must use the qualifier attribute with tc:loop. Again, you must also nest the tc:data tag within the tc:loop tag.

For example, you can use this method to retrieve data only from assignees that have the role Supervisor within a project record. If one or more assignees are found to match your qualifier, their data will be retrieved and inserted in the generated document. The Document Generator will automatically retrieve data only from those records that match the criterion, whether there are zero, one, or multiple records that match.

**Caution:** *The qualifier attribute can only handle one qualifier.*

For most sub-objects, the Type attribute is the most useful characteristic for filtering the data that is retrieved. For assignees, the role is actually the Type attribute of the assignee sub-object in the Object Model.

It is possible to use other object attributes from the sub-object, such as a number or text field, but these values are rarely useful for setting a criterion for which sub-objects to retrieve.

The following table indicates the tags and attributes that you would use to set up a tc:loop that retrieves a qualified list of sub-object data in a document template.

**Usage of tc:loop - Retrieving Data with Qualifier**

Tag attribute	Attribute value	Description
select	"SubObjectList"	<p>For the value of the select attribute, type the name of the object attribute that links to the desired sub-object. For example, if you want the Assignees sub-object for a Project, type AssigneeList.</p> <p>See <a href="#">Object Model: Read This First</a> and the additional reference tables it points to, for a complete list of sub-objects and to find the object attribute names that identify them.</p>



		<p><b>Note:</b> The <i>tc:data</i> tag must be nested within the <i>tc:loop</i> tag in order to retrieve any data.</p>
qualifier	"PathToQualifier=Value"	<ul style="list-style-type: none"> <li>PathToQualifier is a path of object attributes, separated by underscores, that qualifies the sub-object list. This path is built according to the logic of the Object Model.</li> </ul> <p>The last object attribute in this path must be of one of the following data types:</p> <ul style="list-style-type: none"> <li>integer</li> <li>string</li> <li>date</li> </ul> <p>For example, you can use the type of the sub-object as a qualifier by typing <i>Type_TreePosition</i> for <i>PathToQualifier</i>.</p> <p>The data types can be found in the <a href="#">Object Model: Read This First</a> and related tables.</p> <ul style="list-style-type: none"> <li>Value is the data that must exist in the selected attribute in order for the Document Generator to retrieve that instance of the sub-object. If the value does not match the value that you specify, the sub-object will not be retrieved.</li> </ul> <p>See <a href="#">Qualifier Tag Attribute Syntax</a> for details about the value of this attribute.</p> <p><b>Note:</b> The Value can be a ; delimited string to provide OR logic. However, only one instance of the sub-object will be returned, rather than all matching instances.</p>

The following table indicates the syntax for the qualifier attribute value.

**Qualifier Tag Attribute Syntax**

Object attribute of sub-object used as qualifier	Syntax	Description
Type attribute	qualifier="Type_TreePosition=UCOD_TREE"	The Type attribute of a sub-object is the most commonly-used attribute for setting a

		<p>criterion for which sub-objects to retrieve.</p> <ol style="list-style-type: none"> <li>1. Replace UCOD with the four-character unique code of the object that TeamConnect will retrieve data from. <ul style="list-style-type: none"> <li>○ If the sub-object is a system lookup table, type the unique code of the system lookup table.</li> <li>○ If the sub-object is DetailList (categories), PhaseList, or AssigneeList, type the unique code of the object to which it belongs. This may be found in the <b>General</b> tab of the object definition.</li> </ul> </li> <li>2. Replace TREE with the full tree position of the selected item within the sub-object. The tree position should be in all caps, with each level separated with an underscore. For example: <pre>AAAA_BBBB_CCCC</pre> <p>If the desired type is in the Root of the sub-object, then the tree position is simply the four-character unique code of the sub-object.</p> </li> </ol>
<p>Any attribute that identifies a number or text value</p> <p><b>Note:</b> <i>Not useful as a criterion.</i></p>	<pre>qualifier="PathToAttribute=ExactValue"</pre>	<p>When using a number or text field to qualify your tc:loop tag, simply type the value that you want to use as the criterion for which sub-objects to retrieve as the Value of the tag attribute. For example, from the sub-object JContAddress:</p> <pre>qualifier="State=CA" qualifier="Version=1"</pre> <p><b>Note:</b> <i>For number values, you must verify the type of number that is stored in the database and follow that format (float, integer, and so on).</i></p>
<p>Any attribute that identifies a date value</p>	<pre>qualifier="PathToAttribute=YYYY-MM-DD"</pre>	<p>Use this syntax for the date value when using a date value to qualify the sub-object. For example, from the sub-object JContRate:</p> <pre>qualifier="EffectiveTo=2003-02-26"</pre>

<p><b>Note:</b> <i>Not useful as a criterion.</i></p>		
---	--	--

The following examples show how qualifiers can be used in the tc:loop tag.

**Example 1:**

```
<tc:loop select="AssigneeList"
  qualifier="Type_TreePosition=CLAM_SPRV">
  <tc:data select="User/Contact/FirstName" />
  <tc:data select="User/Contact/Name" />
  <tc:data select="User/Contact/DefaultPhone/ PhoneString" />
</tc:loop>
```

In the example above, the qualifier attribute takes the unique code of the object and the tree position of the selected item within the sub-object as its value. CLAM is the unique code of the selected object, Claim, and SPRV is the tree position of the project assignee. The data retrieved by TeamConnect is the first and last name, and phone number of only the type of assignee (specifically, assignee role) that is specified by the qualifier attribute.

**Example 2:**

```
<tc:loop select="DetailList" qualifier="Category_Parent_Name=Claim">
  <tc:data select="Category/Name" />
</tc:loop>
```

This is an example of using a text value as a qualifier. In this example, which is for a custom object named Claim, the qualifier attribute makes the Document Generator retrieve only those categories added to the record that have Claim as the parent category. This property means that any categories on the level directly beneath Root which are added to the record are retrieved. However, the category Claim itself is not retrieved.

Also, any categories added to the record that are on the second, third, fourth levels, and so on, beneath the Root are not retrieved, because their Parent category is not Claim. The data being retrieved with the tc:data tag is the name of each category that meets the qualifier.

This property is useful when you want the generated document to include only the major categories of a record and not all of the sub-categories.

**Example 3:**

```
<tc:loop select="PhoneList" qualifier="Type_TreePosition=PHON_BUS1">
  <tc:data select="PhoneString" />
</tc:loop>
```

In the previous example, the qualifier attribute makes the Document Generator retrieve only the phone numbers for a contact that are of the type Business 1 (BUS1). The tc:data tag retrieves the actual phone number.

Notice that the value for the qualifier attribute is the unique code of the lookup table (PHON), followed by an underscore ( ) and the tree position of the phone type being retrieved (BUS1).

## 3.1.19.2.5 tc:date

The tc:date tag retrieves the current system date, at the time the document is generated. By default, the date will be displayed in the following format:

MM/dd/yy

To display the current system date in another format, use the format tag attribute.

You can also nest the tc:user tag, since it works the same regardless of which other tags you have outside of it. The following table describes the possible tag attributes for tc:date.

**Usage of tc:date**

Tag attribute	Attribute value	Description
format	"DateAttribute"	Specify the format in which the date and time will display in the end-user's generated document.  See <a href="#">the Possible Values for dateFormat Tag Attribute table</a> for a list of date formats.
showTime	"yes" or "no"	Shows the time next to the date.  <b>Note:</b> This tag should only be used if format is not specified
timeZone	"yes" or "no"	Shows the time zone next to the date. Only necessary if the value for format does not display the time zone.  <b>Note:</b> The time zone is obtained from the current user's settings.

**Example:** `<tc:date format="EEE, MMM d, yy"/>`

In the example given, the date will be inserted in the generated document and formatted by Day, Month Date, Year, as in Friday, Jun 28, 2002.

## 3.1.19.2.6 tc:user

The tc:user tag retrieves data from the user account of the current user who is generating the document. You can nest tc:data tags to retrieve the necessary data from that user. You can also nest the tc:user tag, since it works the same regardless of which other tags you have outside of it.

This tag points to the YUser object table. See [Object Model: Read This First](#) to learn about what data you can retrieve about the current user, either from YUser or from an object that is related to it.

**Example:** `<tc:user>  
<tc:data select="Contact">  
<tc:data select="FirstName" />`

```

    <tc:data select="Name" />
    <tc:data select="Title" />
  </tc:data>
</tc:user>

```

In the example given, the data retrieved by TeamConnect is the first name, last name, and job title of the user who generates the document, which is all located in the contact record associated with that user.

#### 3.1.19.2.7 tc:file

The `tc:file` tag retrieves the text of a separate XML file and places that text in the end-user's generated document. The file you are retrieving must be uploaded to a specific directory in TeamConnect's Documents area. This tag can be nested as needed in a document template.

You can use this tag to retrieve content that is being used by many different templates, instead of including the same content directly in each template. This tag is also useful if the text in the XML file should be included only under certain circumstances. You can use this tag in conjunction with the `tc:conditional` tag to accomplish this.

##### Usage of `tc:file`

Tag attribute	Attribute value	Description
select	"FileFolder/ FileName"	Specify the name of the folder in which the XML file is located and the name of the file.  This path is relative to the document template's location in TeamConnect's Documents area. This means that all supporting XML files that are retrieved by a document template must be stored in a subfolder of the <b>Document Templates</b> folder where the template is located.

**Example:** `<tc:file select="Statutes/CalStatutes.xml" />`

The `select` attribute takes the name of the folder in which the XML file is located and the name of the XML file as its value. In the example given, `Statutes` is the name of the folder and `CalStatutes` is the name of the XML file. The content of the `CalStatutes` file will be included as text in the generated document.

## Preparing XML Files Referenced by `tc:file`

You may wish to retrieve text from a file with a DOC extension (a Microsoft Word™ file) and insert that text into the generated document. However, only an XML file can be extracted with the `tc:file` tag and inserted into the generated document.

### To convert the DOC file into an XML file

1. Open the DOC file with your word processing application (i.e. Microsoft Word™).
2. Format the text exactly as you would like the information to appear in the generated document (enter all returns, indentations, line formatting, font size, and so on).
3. Save the file with an RTF extension and close your word processing application.
4. Find the RTF file you just saved and open it with a text editor such as Microsoft Notepad™.
5. At the very beginning of the document, insert the following tags, with a line return after each, in the following order:

```
<?xml version="1.0" encoding="UTF-8"?>
<tc:document version="1.0" mime-type="MimeType" name="TemplateName"
xmlns:tc="http://www.w3.org/1999/XSL/Transform">
```

For information about the attribute values "MimeType" and "TemplateName" see [the Document Tag Attributes table](#).

```
<tc:content>
```

**Caution:** Do not place a line return after `tc:content`. If the source code does not start immediately after this tag, TeamConnect encounters an error when trying to generate the document.

6. At the very end of the document, insert the following tags, in the following order:

```
</tc:content>
</tc:document>
```

7. Save the file with an XML extension.
8. Open the file in Internet Explorer or an HTML editor to test whether it is well-formed.
9. Create a new folder within the **Document Templates** folder of the object for which your template is created.
10. Upload the XML file to the folder you just created.

The contents of the XML file will be included in the generated document as long as the XML file and its folder are in the correct directory in the **Documents** area of TeamConnect.

#### 3.1.19.2.8 tc:conditional

The `tc:conditional` tag is used to specify a condition that must be met in order for any nested content within the `tc:conditional` tag, including static text as well as tags, to be included in the generated document. If the condition specified by the tag is met, then anything nested within `tc:conditional` is generated. If the condition is not met, then the nested content is ignored.

The following points summarize the basics about `tc:conditional`:

- `tc:conditional` can be nested within almost any other tags. For example, you can nest it within `tc:filter` or `tc:search`. For an example of nesting `tc:conditional`, see [Retrieving Specific Type of Related Object with tc:conditional](#).
- You can nest `tc:conditional` tags within each other, if necessary.

- There are two main tag attributes for tc:conditional: compare and test. You must always use either one or the other when you use tc:conditional. However, you cannot use both.
- There are two other tag attributes that you can use in combination with compare or test, which are described in the following table.

The following table describes all of the tag attributes that may be used with tc:conditional.

**Tag Attributes of tc:conditional**

Tag attribute	Attribute value	Description
compare	<pre>"Detail[TreePosition]/ DetailValue[FieldName],Value"</pre> <p>or</p> <pre>"System[SubObject],Value"</pre> <p><b>Note:</b> Do not type a space before or after the forward slash (/) or comma.</p>	<p>This tag attribute compares the value from the record with the value specified in the tag. If the value in the record is the same as the value you specify here in the tag, then the condition is met.</p> <p><b>Custom fields:</b></p> <ol style="list-style-type: none"> <li>1. Replace TreePosition with the actual category tree position of the custom field.</li> <li>2. Replace FieldName with the name of the desired custom field.</li> <li>3. Replace Value with the value that is expected for the condition to be met.</li> <li>4. For more details, see <a href="#">Using tc:conditional to Compare Custom Field Values</a>.</li> </ol> <p><b>Sub-objects:</b></p> <ol style="list-style-type: none"> <li>1. Replace SubObject with the name of the sub-object. Do not type List at the end of the name of the sub-object (AddressList, etc.). For example: <code>Address, Assignee, Email, Fax, InetAddress, Phase, Phone, Relation, Resource, Skill</code></li> <li>2. Replace Value with the Name of the sub-object type that you want to look for.</li> </ol> <p>For more details, see <a href="#">Using tc:conditional to Compare Sub-objects</a>.</p>
test	<pre>"Detail[UCOD_TREE]"</pre> <p>(testing for a category)</p> <p>or</p> <pre>"ObjectAttribute"</pre>	<p>You can use this tag attribute for checking whether a value exists in system fields only (including categories). The test tag attribute cannot be used for custom fields.</p> <p><b>Testing for a category:</b></p> <p>Use the test attribute with this syntax to test whether a category exists (has been added) in the current object record. Replace UCOD_TREE with the tree position of the category you want to test.</p>

		<p>For example, if you want to test whether the Office (OFFI) category exists for Expense, replace UCOD_TREE with EXPE_OFFI.</p> <p><b>Testing for an object attribute value:</b></p> <p>Use this tag attribute to test whether a value exists for the object attribute you specify. If it exists, then any content within tc:conditional is generated.</p> <p>You can use this tag attribute to test:</p> <ul style="list-style-type: none"> <li>• Whether a value exists for almost any type of object attribute. Types of data you can test include: Date, Text, Number, Object, Memo Text, and so on.</li> </ul> <p><i><b>Note:</b> You cannot use this tag attribute to test a boolean value such as a check-box.</i></p> <ul style="list-style-type: none"> <li>• Whether there are any instances of a sub-object (for example, whether there is at least one Assignee added to a record)</li> <li>• Whether a category has been added to a record.</li> </ul> <p><i><b>Important:</b> When you are using test to see if there is a value and that value is of the data type object (such as a parent record), the tc:conditional tag actually navigates to that object. This means that if you nest any tags within tc:conditional, you must write them relative to the object that you have navigated to.</i></p>
condition	"greater" "less" "equal" (default)	<p>You can use this attribute in conjunction with the compare attribute when setting a tc:conditional tag for a number or date field.</p> <ul style="list-style-type: none"> <li>• Use greater to indicate that the condition will be met when the number or date value from the system is greater than the value specified in the compare attribute.</li> <li>• Use less to indicate that the condition will be met when the number or date value from the system is less than the value specified in the compare attribute.</li> </ul> <p>The default value for this attribute is equal. When you do not include the condition attribute in the tag, it is assumed that the Document Generator should consider the condition to be met when the value in the system is equal to the value you specify in the compare attribute value.</p>



negate	"yes" or "no" (default)	This attribute reverses the logic of tc:conditional. You can use negate with either compare or test.  When you add this attribute to the tag with the attribute value equal to "yes," then the Document Generator will only generate what is nested within tc:conditional when the condition is NOT met.
--------	-------------------------------	--

#### 3.1.19.2.8.1 Using tc:conditional to Compare Custom Field Values

The compare attribute of tc:conditional is useful for comparing a custom field to a value that you specify in the tag. If the custom field has the value that you specify, then the condition is found to be true, and the content nested within the tc:conditional tag is generated.

You can use compare with the following types of custom fields. For each type of custom field, you must provide the value of the compare tag attribute in the specified format.

##### Tag Attribute Value Formats for Comparing Custom Fields

Custom field type	Format of value	Example tag
Check-box	true or false	<pre>&lt;tc:conditional compare="Detail[CLAM]/ DetailValue[PoliceInvolved],true "&gt;</pre>
Number	Value must be typed as an integer (without decimals or commas)	<pre>&lt;tc:conditional compare="Detail[CLAM_AUTO]/ DetailValue[ClaimEstimate],10000 " condition="greater"&gt;</pre>
Text	Exact text that you want to compare	<pre>&lt;tc:conditional compare="Detail[CLAM]/ DetailValue[LossLocation],Near Residence"&gt;</pre>
List	The name of the lookup table item.	<pre>&lt;tc:conditional compare="Detail[CLAM]/ DetailValue[Fatality],Yes"&gt;</pre> <b>Caution:</b> <i>If the lookup table item name has a comma in it, then this tag will not function properly.</i>
Date	YYYY-MM-DD	<pre>&lt;tc:conditional compare="Detail[CLAM]/ DetailValue[AwardDate],2003-02- 26"&gt;</pre>

```

Example <tc:conditional compare="Detail[CLAM]/DetailValue[DrunkDriver],true">
:           <tc:file select="Statutes/DrunkDriverStatute.xml" />
           </tc:conditional>
           <tc:conditional compare="Detail[CLAM]/DetailValue[Speeding],true">
             <tc:file select="Statutes/SpeedingStatute.xml" />
           </tc:conditional>

```

In this example, the custom fields **DrunkDriver** and **Speeding** are check-boxes within the project Claim. The values specified for each is true, meaning that if a check-box is checked, then the XML file specified for that field will be included in the generated document. For example, if the **DrunkDriver** check-box is checked, the XML file **DrunkDriverStatute** is included in the generated document since the true condition that you specified was met.

#### 3.1.19.2.8.2 Using tc:conditional to Compare Sub-objects

The compare attribute of tc:conditional can also be used to check whether an instance of a certain type of sub-object exists. If the type of sub-object that you specify in the tag has been added to the record, then the condition is found to be true, and the content nested within the tc:conditional tag is generated.

You can use the compare attribute to check any sub-object that has a Type object attribute. For example, Address, Assignee, Phone, and Email all have a Type.

```

Example: <tc:conditional compare="System[Assignee],Adjuster">Please contact
your claim adjuster for more information.</tc:conditional>

```

In this example, if the Assignee List in the record contains an assignee of the type Adjuster, then the text inside the tc:conditional tag is generated. **Adjuster** is an example of the type of assignee.

#### 3.1.19.2.8.3 Using tc:conditional to Test for Object Attribute Values

The test attribute of tc:conditional is used to test whether a condition is true. You can test any system fields (object attributes) belonging to an object, EXCEPT for check-boxes. A condition is true when a value exists for that attribute. For example, you can test whether a custom object record such as Claim has a parent selected in the **Parent** system field.

You can also test the list of sub-objects added to a record to check whether any instances of that sub-object exist. If at least one instance of the sub-object is found, then the condition is true, and the content nested within the tc:conditional tag is generated. For example, if there is at least one Address added to a Contact, then the condition is true.

Unlike the compare tag attribute, the test attribute does not check the type of the sub-object. Any instance found proves the condition to be true.

```

Example: <tc:conditional test="MainAssignee">
Please contact the following associate with any questions you might
have about your claim.
           <tc:data select="User">
           <tc:data select="Contact">
           <tc:data select="FirstName" />

```

```
<tc:data select="Name" />
</tc:data></tc:data>
</tc:conditional>
```

In this example, the text and tags that are nested within `tc:conditional` will only be inserted into the generated document if a Main Assignee exists for the Claim record.

#### 3.1.19.2.8.4 Testing for a Category

You can use the test tag attribute to test whether a certain category has been added to a record. This is a useful way to include content in a generated document only when the main object record has a certain category added to it.

**Example**

```
<tc:conditional test="Detail[CLAM_HOME]">
:   We have the following information about your loss in our records:
      Caused by natural disaster:
      <tc:detail select="CLAM_HOME/NaturalDisaster" />
      Date of Loss:
      <tc:detail select="CLAM_HOME/LossDate" format="" />
      Summary of Loss from your written statement:
      <tc:detail select="CLAM_HOME/LossSummary" />
      Estimated Dollar Amount of Loss:
      <tc:detail select="CLAM_HOME/LossEstimate"
      numberFormat="#,##0.00" />
</tc:conditional>
```

In this example, the text and custom fields contained within the `tc:conditional` tag will only be inserted into the generated document if the category `CLAM_HOME` has been added (exists) in the Claim record.

#### 3.1.19.2.9 tc:block

The `tc:block` tag is used to group `tc:conditional` tags. The `tc:conditional` tags on their own can each have nested content that is only included in the generated document when the condition is true. However, when nested within a `tc:block` tag, multiple `tc:conditional` tags are handled differently.

The advantage of `tc:block` is to generate the same content, such as retrieving another XML file, when at least one of a group of conditions are met. For example, if one or more of five check-boxes is checked, then retrieve an XML file. Each `tc:conditional` tag that is nested within `tc:block` must have exactly the same content nested within it, so that no matter which condition is found to be true, the Document Generator generates the same content.

When `tc:conditional` tags are nested within `tc:block`, the Document Generator reads through each `tc:conditional` tag and checks the condition. As soon as it encounters a `tc:conditional` tag with a true condition, it generates that tag's nested content. It then ignores the rest of the `tc:conditional` tags in the block.

The following points will help you use `tc:block`:

- If you have several tc:conditional tags to include in a document template but they have different content that they will generate, then do NOT nest them within a tc:block tag. The purpose of using tc:block is to allow the same content to be generated, no matter which condition is met.
- You cannot nest any tags within a tc:block tag except for tc:conditional tags. Other tags can be nested within those tc:conditional tags, but they cannot be nested directly within the tc:block tag.
- Like tc:conditional, you can nest tc:block as needed within other tags.

**Example:**

```
<tc:block>
  <tc:conditional compare="Detail[CLAM]/
  DetailValue[DrunkDriver],true">
    <tc:file select="Statutes/RecklessStatute.xml" />
  </tc:conditional>
  <tc:conditional compare="Detail[CLAM]/DetailValue[Speeding],true">
    <tc:file select="Statutes/RecklessStatute.xml" />
  </tc:conditional>
</tc:block>
```

*In this example, the custom fields DrunkDriver and Speeding are specified. The conditions for these check-boxes are each set to true.*

**Note:** *The tag nested within each tc:conditional tag is exactly the same. If only one of these fields is checked, then the XML file named RecklessStatute.xml will be retrieved and included in the generated document.*

#### 3.1.19.2.10 tc:compute

The tc:compute tag can add, subtract, multiply, or divide two numeric values, depending on the mathematical operator specified.

When using tc:compute, you must know the following concepts:

- Each value can be either a system field, custom field, or value specified within the template, but they must both be numerical values. Otherwise, TeamConnect will display an error and the document will not generate properly.
- You can nest tc:compute as needed within other tags.
- In order to specify which two values to compute, you nest tags within tc:compute that point to numerical values.
- The tc:compute tag can be described as "Field1 (+, -, \*, /) Field2". For example, for subtracting, the second nested field is subtracted from the first nested field. For dividing, the first field is divided by the second field.

The following table describes the tag attributes for tc:compute.

##### Usage of tc:compute

Tag attribute	Attribute value	Description
method	"Operator"	Enter one of the following to specify the mathematical operation between the specified number fields:  "+" for addition  "-" for subtraction  "*" for multiplication  "/" for division
format	"format"	Specify the desired number format that will be displayed in the generated document.  For a description of possible attribute values, see the description for numberFormat in <a href="#">the Optional Tag Attributes for tc:data and tc:detail table</a> .

**Example:**

```
<tc:compute method="-" format="###0.00">
  <tc:detail select="CLAM/ClaimEstimate" />
  <tc:detail select="CLAM/ClaimPaid" />
</tc:compute>
```

In the example given, the compute method specified is subtraction "-". CLAM/ClaimEstimate and CLAM/ClaimPaid are number fields for the project Claim. Since CLAM/ClaimEstimate is specified first, CLAM/ClaimPaid will be subtracted from CLAM/ClaimEstimate to give the resulting value in the generated document.

To compute a value from one TeamConnect system or custom field and a value specified within the template, use tc:operand in conjunction with tc:compute. See [tc:operand](#) for details.

#### 3.1.19.2.11 tc:operand

The tc:operand tag can be nested within tc:compute to specify a static value to compute with a value from a number field.

**Example:**

```
<tc:compute method="*" format="###0.00">
  <tc:detail select="CLAM/ClaimEstimate" />
  <tc:operand value=".2" />
</tc:compute>
```

This example calculates a value by multiplying the value in the Claim Estimate field by .2. The purpose of this example is to find a percentage of a value in TeamConnect and insert this percentage value into a generated document.

## 3.1.19.2.12 tc:search

The tc:search tag is used to automatically retrieve data from related objects of the current object. When you want to retrieve data from related objects automatically, rather than having the user select the related objects from which data should be retrieved, use tc:search.

For example, if you are creating a document template for Policy and you want to automatically retrieve data from its child Claim records, you would use tc:search to find the desired Claim records and get data from them.

You can retrieve data from the following related objects by using tc:search to navigate to them:

- A system object that is directly related to the current object, such as its involved or history.

**Note:** Documents are not directly related to an object. Therefore, you will not be able to retrieve information or content about a document using tc:search. You will need to use tc:filter instead. For information, see [Filters for Non-Related Objects](#).

- A child custom object of the current custom object.
- An embedded object of the current custom object.
- Child accounts of the current account.

### Points To Remember

When using tc:search, you must know the following points:

- Unlike tc:filter, when you use tc:search you cannot guarantee that data will be retrieved from only one record. The Document Generator is "searching" for records according to your tag, similar to a search screen in the user interface.
- If you use tc:search without any tc:conditional tags, then the data will be retrieved from all instances of a related object and included in the generated document. For example, you may want to include the all of the claims filed against a particular policy.
- By nesting the tc:conditional tag within tc:search, you can specify a criterion for the related object records from which to automatically retrieve data.
- You can nest tc:search within other tags as needed to navigate to the appropriate object. For example, after navigating to the Contact object through the contact-centric field in a custom object (by using tc:data), you can use tc:search to get that contact's History records.

### When NOT to Use tc:search

You must not use tc:search when:

- You want the user to select which related object record(s) from which data will be retrieved. Instead, use tc:filter.
- You want to retrieve data from a record that is not related to the main object record in any way. Instead, use tc:filter so that the user can select the record.
- You want to retrieve data or content from a document record. Instead, use tc:filter to select a non-related record.

- You want to retrieve data from sub-objects. Instead, use tc:loop (to automatically identify the sub-objects) or tc:filter (to have the user select the sub-objects).

The following table describes the tag attributes for tc:search.

**tc:search Tag Attributes**

Tag attribute	Attribute value	Description
link	"ObjectName"	<p>Type the name of the object to which you want the Document Generator to navigate from the current object. The name of the object should be typed as shown:</p> <p>Appointment Account Expense History Involved Task Project (for child custom objects and embedded objects only)</p> <p><b>Note:</b> You must include the qualifier attribute, if you are using tc:search to retrieve child objects.</p>
isForChild	"yes" or "no"	<p>With a value of "yes," this attribute tells the Document Generator that you are retrieving records that belong to one of the following:</p> <ul style="list-style-type: none"> <li>A child object of the current object.</li> <li>An embedded object of the current object.</li> </ul> <p>Incidentally, this means that the object you are navigating to resides in the same (current) table in the database.</p> <p>By default, "no" is the value for this tag attribute. Make sure to use "no" when you are identifying an object that is not a child of the current object.</p>
qualifier	"Application_ProjObjectDefinition_UniqueCode=UCOD"	<p>This tag attribute is required when you are using tc:search for a child custom object, child account, or an embedded object.</p> <p>This attribute value identifies the child object using the following values:</p> <ol style="list-style-type: none"> <li>Path of the object attribute that uniquely identifies the child or embedded object, which is Application_ProjObjectDefinition_UniqueCode.</li> </ol>

- |  |  |  |
|--|--|--|
|  |  | 2. Unique code of the child or embedded object, which can be copied from the Object Definition List in the user interface. |
|--|--|--|

The following example illustrates how tc:search can be used.

**Example 1:**

```
<tc:search link="Project" isForChild="yes"
qualifier="Application_ProjObjectDefinition_UniqueCode=CLAM">
Claim Information (Child of Policy)
  Claim Name: <tc:data select="Name" />
  Claim Number: <tc:data select="NumberString" />
</tc:search>
```

In the example above, tc:search is retrieving only child object records for Claim, since Policy may have other child custom objects besides Claim.

Rather than including data from all related object records that are identified by tc:search, you may want to include only data from those records that meet certain criteria. For example, instead of the names of all the Involved Contacts added to a policy, you may want to include only the names of Involved Contacts that have a certain category, such as Additional Driver.

**Example 2:**

```
<tc:data select="Parent">
Parent (Policy) Information
  Number: <tc:data select="NumberString" />
  Name: <tc:data select="Name" />
  Policy Holder: <tc:data select="Contact"><tc:data
select="FirstName" /> <tc:data select="Name" /></tc:data>
  Additional Drivers: <tc:search link="Involved"><tc:conditional
test="Detail[PLIV_ADDR]"><tc:data select="Contact"><tc:data
select="FirstName" />
<tc:data select="Name" />
</tc:data></tc:conditional></tc:search>
</tc:data>
```

In this example, the document template is for the Claim object. The template retrieves data from the Involved of the parent of Claim, which is Policy. First, the template navigates to the parent object. Once the template is pointing at Policy, tc:search is used to identify the Involved records. The tc:conditional tag is then used to retrieve data only from those Involved records with the Additional Driver category (role) added to them.

### 3.1.19.2.13 tc:filter

You can define a filter for the document template so that when the user launches the Document Generator, a filter page opens. The user can then select a record, such as a claimant (an Involved), from a drop-down list. Data from this record can then be retrieved in the content of the document.

In the content section of a document template, tc:filter is used to switch focus from the current object record to the record that will be selected by the user in a filter page. Once this object is selected as the current object, you can retrieve data from it using other tags.



For example, if your document template is for a custom object named Claim, you may use tc:filter to retrieve data from involved, appointment, task, history, child objects, or any other objects that are related to Claim and which have been selected by the user in filter pages.

In the header section, you will need to specify how the filter page will appear in the end-user interface of the Document Generator. You will learn more about this in [Defining Filter Pages](#).

## When to Use tc:filter

Use tc:filter in the content section when you want to retrieve data from any of the following records or sub-objects which have been selected by the user in filter pages:

- A related object. The related object can be any of the following:
  - Child custom objects
  - Embedded objects
  - Child accounts (if the main object is an account)
  - Related system objects (for example, a project's involved, appointment, or history records), except document.

- An object that is not directly related to the main object.

For example, the main object of your **template** is Policy, whose child object is Claim. You may want the user to select the involved parties of the related Claims.

- A sub-object.

For example, you may want the user to select an Address, Assignee, or Email Address. You can define filter pages that enable users to select these sub-objects.

- An object that is not related to the main or current object, such as Documents.

For example, you can use tc:filter to retrieve data from a Contact that the user selects, when that Contact is not related to the main object record.

## When NOT to Use tc:filter

You would not use tc:filter in the content section under the following circumstances:

- When you do not want the user to select the related records or sub-objects. If this is the case, use tc:search or tc:loop to automatically identify the records and retrieve the correct data.
- When the relationship from the main object to the desired object is found in a custom field.

For example, if Party and Claim are custom objects and there is a custom field in Claim of type Custom Object that is used for selecting a Party record, then you would navigate to the Party record using tc:detail.

- When the relationship from the main object to the desired object is directly found in a system field.

For example, from a Claim record, its parent Policy record is found in the system field Parent. Therefore, tc:data is the tag used to get to the data in the Policy record.

You would use `tc:filter`, however, if you want the user to select an object that is related to the record selected in the system field. For example, if you wanted to use records selected by the user from the Involved records of the Policy mentioned above, this would require the use of `tc:filter`.

- When you want to retrieve data from a sub-object and you do not want the user to select an instance of the sub-object.

For example, if you want to navigate to the Contact record of Assignees and automatically list the names of all assignees from their Contact records (instead of having the user select an Assignee), then you would use `tc:loop`.

## Points To Remember

When using `tc:filter`, you must know the following concepts:

- The `tc:filter` tag does not actually retrieve data and insert it in the generated document. `tc:filter` is used for navigating to the correct object in TeamConnect to identify the data you need. In order to retrieve data from that object, you use other Document Generator tags, such as `tc:data` and `tc:detail`.
- You can nest other tags within `tc:filter` as needed in order to navigate to the desired objects.
- You must always define the filter using a `tc:filter` tag in the header of the document template. In the content section, you use `tc:filter` again to reference the filter you defined that identifies the record(s) from which you want to retrieve data. You will learn more about this in [Defining Filter Pages](#).
- The filter that you define in the header section of the document template can allow the user to select one or multiple related object records.
- Use `tc:filter` when you need to retrieve data from a sub-object that is selected by the user. Then, in the header section, use the `tc:filter` tag to define that filter as a filter page.
- You can refer to the following for samples of document templates using `tc:filter`:
  - [Automatically Retrieving Data from Related Objects](#).
  - [Including Information When Certain Conditions Are Met](#).

The following table describes the attributes of the `tc:filter` tag when you are using it in the content section of the document template.

**tc:filter Tag Attribute**

Required?	Tag attribute	Attribute value	Description
Yes	<code>select</code>	"FilterName"	Type the unique name for the filter that will appear in the filter page in the end-user interface of the Document Generator. This is the name that you define for the filter in the header of your document template.

			<p><b>Tip:</b> If you have not yet defined the filter in the header section of the document template, you can choose a name for it now and use this name later when you define the filter. For more details, see <a href="#">Defining Filter Pages</a>.</p> <p><b>Important:</b> The <code>tc:filter</code> tag does not retrieve any data by itself; you must nest other tags within the <code>tc:filter</code> tag to retrieve data.</p>
<b>No</b>	list	"yes" or "no"	By default, this attribute is set to "no". You must set <code>list="yes"</code> when the filter you are using is a hidden filter. For details, see <a href="#">Defining Filter Pages</a> .

**Example:**

```
<tc:filter select="InvolvedParty">
  <tc:data select="Contact">
    <tc:data select="FirstName" />
    <tc:data select="Name" />
  </tc:data>
</tc:filter>
```

In the example given, the data retrieved by TeamConnect will be the first and last name of the Contact, who is an Involved Party of the Claim record. The user will choose which Involved Party's data should be retrieved when invoking the Document Generator. InvolvedParty is the filter chosen. The filter that you select here is defined in the header section of the document template.

You can find more examples using `tc:filter` in [Document Template Samples](#).

#### 3.1.19.2.14 tc:input

The `tc:input` tag retrieves text that is entered by the user who invokes the Document Generator. The user will enter text within fields that you will create by defining them in the header section of the document template. The information retrieved by the `tc:input` tag is not data within any TeamConnect records.

When using the `tc:input` tag in the content section of the document template, you reference a corresponding `tc:input` tag in the header section of the document template where you define the input page.

You can nest `tc:input` where needed in the content section. It is not affected by the outer tags where it is nested.

The following table indicates how to use `tc:input` in the content section to retrieve the selection made by the user in the input page.

##### Usage of `tc:input`

Tag attribute	Attribute value	Description
select	"InputName" or "InputForm.InputName"	<p>Type the unique name for the text field in which the user will enter information in an input page.</p> <p>The name that you type here in the content section of the document template must correspond to an input field that you define in the header.</p> <p>If the field is defined as one of multiple fields in one input page, then the InputName must reflect which input page contains the field, as shown.</p>

**Example:** `<tc:input select="UserInitials" />`

In the example given, UserInitials is the input field chosen. When the document is generated, the text that the user entered in this field in the input page will be inserted in place of this tag.

The input field is defined in the document template's header section. For details about defining input pages, see [Defining Pages for User Input](#). See also the example letter in [the Document Generator Template Page Two image](#) for examples of using tc:input.

### 3.1.19.3 Nesting to Navigate

To retrieve the data you need in a document template, you must utilize the built-in relationships among the objects in the Object Model. These relationships are reflected in the tags you write in the document template as either nested tags or paths written within a tag attribute.

Nesting tags in a document template is equivalent to navigating throughout the Object Model to build paths to the data you want to retrieve. Each time you nest a tag, you are "traversing" between two objects in the Object Model using a relationship that exists between them. When you open a tag but do not close it, you point the Document Generator to the object to which that particular item has a relationship. The next tag you use refers to that object. When you are finished using data and relationships from that object, close the nested tag to return to the object that you started from.

Nesting to navigate can start with any piece of data that identifies another object. System fields, custom fields, related objects, and sub-objects can all have relationships to other objects. Therefore, you can even nest a tag after pointing to a system field or a custom field.

In order to find out whether a field points to another object, or how two objects are related, see [Object Model: Read This First](#). Information here indicates whether an object attribute points to another object by indicating the data type Object in the Data Type column of each table.

For example, suppose you have two custom objects named Claim and Policy, where:

- Claim is a child of Policy.
- Policy is contact-centric.
- Claim has a custom field of type Custom Object that is used for selecting another claim as the Master Claim.

If the following tags were in a document template for the Claim object, they would retrieve data from the policy that is the parent of the master claim:

```
<tc:detail select="CLAM/MasterClaims">
  <tc:data select="Parent">
  <tc:data select="Contact">
  <tc:data select="FirstName" />
  <tc:data select="Name" />
  <tc:loop select="AddressList" qualifier="Type_TreePosition=ADDR_BUS1">
  <tc:data select="Street" /> <tc:data select="City" />
  <tc:data select="State" /> <tc:data select="PostalCode" />
  </tc:loop>
  <tc:data select="SsOrTaxNumberString" />
  <tc:data select="DriverLicense" />
</tc:data>
</tc:data>
</tc:detail>
```

The following table provides a detailed explanation of each line of code in this example.

Line #	Example	Description
1	<code>&lt;tc:detail select="CLAM/MasterClaims"&gt;</code>	<p>This <code>tc:detail</code> tag points to the custom field <code>MasterClaims</code> in the current claim.</p> <ul style="list-style-type: none"> <li>Since the tag is not closed, the Document Generator navigates to the Claim record selected in this custom field, known as the master claim.</li> <li>Also, since the tag is not closed, nothing is printed in the generated document.</li> </ul>
2	<code>&lt;tc:data select="Parent"&gt;</code>	<p>This <code>tc:data</code> tag points to the system field where the parent record is selected.</p> <ul style="list-style-type: none"> <li>Since the tag is not closed, the Document Generator navigates to the parent record, which is a policy.</li> <li>Also, since the tag is not closed, nothing is printed in the generated document.</li> </ul>
3	<code>&lt;tc:data select="Contact"&gt;</code>	<p>This <code>tc:data</code> tag points to the system field where a contact is selected for a contact-centric record.</p>

		<ul style="list-style-type: none"> <li>• Since the tag is not closed, the Document Generator navigates to the Contact record.</li> <li>• Also, since the tag is not closed, nothing is printed in the generated document.</li> </ul>
4	<code>&lt;tc:data select="FirstName" /&gt;</code>	<p>This tc:data tag points to the system field where the contact's first name is stored.</p> <ul style="list-style-type: none"> <li>• The tag is closed, so the Document Generator does not navigate but remains in the Contact object.</li> <li>• Also, since the tag is closed, the value of the field is printed in the generated document.</li> </ul>
5	<code>&lt;tc:data select="Name" /&gt;</code>	<p>This tc:data tag points to the system field where the contact's last name is stored.</p> <ul style="list-style-type: none"> <li>• The tag is closed, so the Document Generator does not navigate but remains in the Contact object.</li> <li>• Also, since the tag is closed, the value of the field is printed in the generated document.</li> </ul>
6	<code>&lt;tc:loop select="AddressList" qualifier="Type_TreePosition=ADDR_BUS1"&gt;</code>	<p>This tc:loop tag points to the Address sub-object of the Contact object.</p> <ul style="list-style-type: none"> <li>• The qualifier attribute limits the sub-object to addresses of type BUS1 only.</li> <li>• Since the tag is not closed, the Document Generator navigates to the list of addresses added to the contact record.</li> <li>• Also, since the tag is not closed, nothing is printed in the generated document.</li> </ul>
7	<code>&lt;tc:data select="Street" /&gt; &lt;tc:data select="City" /&gt; &lt;tc:data select="State" /&gt; &gt;&gt;tc:data select="PostalCode" /&gt;</code>	<p>These tc:data tags point to the system field where the street, city, state and postal code are stored for the address.</p> <ul style="list-style-type: none"> <li>• Each tag is closed, so the Document Generator does not navigate but remains in the Address sub-object.</li> </ul>

		<ul style="list-style-type: none"> <li>Also, since each tag is closed, the value of the field is printed in the generated document.</li> </ul>
8	<code>&lt;/tc:loop&gt;</code>	This is the closing tag for tc:loop. When this tc:loop is closed, the Document Generator navigates back from the Address sub-object to the Contact object.
9	<pre>&lt;tc:data select="SsOrTaxNumberString" /&gt; &lt;tc:data select="DriverLicense" / &gt;</pre>	<p>These tc:data tags point to the system fields where the contact's social security number and driver license number are stored.</p> <ul style="list-style-type: none"> <li>Each tag is closed, so the Document Generator does not navigate but remains in the Contact object.</li> <li>Also, since each tag is closed, the value of the field is printed in the generated document.</li> </ul>
10	<code>&lt;/tc:data&gt;</code>	This is the closing tag for the tc:data tag that we left open on line 3. When this tc:data tag is closed, the Document Generator navigates back from the Contact to the Policy object.
11	<code>&lt;/tc:data&gt;</code>	This is the closing tag for the tc:data tag that we left open on line 2. When this tc:data tag is closed, the Document Generator navigates back from the Policy object to the Claim object.
12	<code>&lt;/tc:detail&gt;</code>	This is the closing tag for the tc:detail tag that we left open on line 1. When this tc:detail tag is closed, the Document Generator navigates back from the Master Claim selected in the custom field to the starting Claim.


#### 3.1.19.4 Locating Object Attribute Names

As you construct tags in your document template, you will identify items using the select tag attribute. The name of the item that you identify in a tag directly corresponds to the object attributes in the TeamConnect Object Model. Each object's attributes are listed in [Object Model: Read This First](#) and in the additional tables it points to.

When you identify an object attribute in a tag attribute, you must type the name of the object attribute correctly in the tag. For example, the first letter of the object attribute must be upper case when you type the object attribute name in a document template tag.

The following procedure will help you find and type the object attribute names correctly as you construct tags in your document template.

#### To find the correct object attribute for a tag

1. Open an online version of this document in Adobe Acrobat®.
2. In the Contents pane on the left, find [Object Model: Read This First](#). Additional related collections of reference tables follow this entry.
3. Press the **V** key on your keyboard to turn on Acrobat's Select Text Tool, or click the Select Text Tool icon  on the toolbar.
4. In the Bookmarks pane on the left, find and click the name of the table for the current object you are working with. For example, if the latest tag in your document template is pointing to a project record, go to the TProject table.
5. In the object table, find the attribute you need.

*Tip: Read the **Comments**, **Field in User Interface**, and other columns to help identify the correct object attribute.*

6. Double-click the name of the attribute in the **TeamConnect Attribute** column to select it.
7. Open your document template and place your cursor where you want to include a reference to an object attribute in a tag.
8. Use the **Paste Special** command to paste the name of the object attribute as **Unformatted Text**.
9. Change the first letter of the object attribute to upper case.

*Important: The Document Generator will not recognize the tag attribute value unless it is capitalized.*

10. If the object attribute has an arrow after it, delete the arrow or whatever character gets pasted in place of the arrow.
11. Delete any extra spaces before or after the attribute name.
12. Finish the tag as needed.

#### 3.1.19.5 Inserting Tags and Preparing RTF Files

Based on the previous sections, you should be able to insert the correct document template tags to retrieve data. You will need to replace all of your placeholders with the correct tags.

After inserting all of the tags into your document template, your file must be converted to RTF, if you have not already been saving it as an RTF file.



### To insert tags and prepare the RTF file

1. With your word processing application, open the document template text that you previously created.
2. Replace all of the placeholders with their corresponding document template tags. For more information about how to identify the data that you need from the Object Model, see [Locating Object Attribute Names](#).
3. Format the text and the document template tags exactly as you would like the information to appear in the generated document (specifically, enter all line returns, indentations, line formatting, font size, and so on).

The [Document Generator Template Page One image](#) is an example of the text from [the Document Generator Template - Text and Placeholders image](#) that has been replaced with corresponding Document Generator tags. Notice that the formatting (line returns, indentation, spacing, tables, and so on) used in [the Document Generator Template Page One image](#) is reflected in the generated document shown in [the Document Generator Template - Text and Placeholders image](#).

4. Save the content with the extension RTF and close your word processing application.
5. Find the RTF file you just saved and open it with a text editor such as Microsoft Notepad™.

**Caution:** Do not use Microsoft WordPad™ to edit your RTF file. You must be able to see all formatting as text in your text editor.

6. After opening an RTF file that was created with Microsoft Word™, all double quotes (") that were entered in Word may be displayed as \94 in the text editor. Similarly, all single quotes (') may be displayed as \92.
  - a. Find all instances of \94 that appear within the source code and replace with double quotes (").
  - b. Find all instances of \92 that appear within the source code and replace with single quotes (').
7. Save the file.

This completes the content section of the document template. See [the Document Generator Template Page One image](#) for an example of a content section with Document Generator tags.

You must now build the header section and document tags of the template around this source code, as described in [Using Document Template Headers](#) and [Completing the Document Template](#).

```
<tc:data select="Name" />
<tc:data select="NumberString" />
```

```
<tc:date format="EEE, MMM d, yy"/>
```

```
<tc:filter select="SelectInvolvedContact"><tc:data select="FirstName" /> <tc:data select="Name" /></tc:filter>
<tc:filter select="InvolvedContactAddress"><tc:data select="Street" />
<tc:data select="City" />, <tc:data select="State" /> <tc:data select="PostalCode" /></tc:filter>
```

```
<tc:filter select=" SelectInvolvedContact "><tc:data select=" Salutation" />, </tc:filter>
```

ClientGuard was informed of your recent auto accident that occurred on <tc:detail select="CLAM/LossDate" />. We are currently processing this claim and according to our records, the following individuals have filed claims regarding your auto accident:

```
<tc:filter select="involvedinaccident" list="yes"><tc:data select="Contact/FirstName" /> <tc:data select="Contact/Name" /></tc:filter>
```

The following passengers in your vehicle are claiming injury as a result of this accident:

```
<tc:filter select="passenger" qualifier="IVPT_DEFA_INPS" list="yes">
<tc:data select="Contact/FirstName" /> <tc:data select="Contact/Name" /></tc:filter>
```

From the police report, we have determined that you were charged with the following:

```
<tc:detail select="CLAM/DrunkDriver" />
<tc:detail select="CLAM/Speeding" />
<tc:detail select="CLAM/RecklessDriving" />
<tc:detail select="CLAM/LoudMusic" />
```

For the charges listed above, you were in violation of the following California statutes:

```
<tc:conditional compare="Detail[CLAM]/DetailValue [DrunkDriver],true">
<tc:file select="Statutes/DrunkStatute.xml" /></tc:conditional>
<tc:conditional compare="Detail[CLAM]/DetailValue [Speeding],true">
<tc:file select="Statutes/SpeedingStatute.xml" /></tc:conditional>
<tc:conditional compare="Detail[CLAM]/DetailValue [LoudMusic],true">
<tc:file select="Statutes/LoudMusicStatute.xml" />
</tc:conditional>
```

In addition, based on the violations listed above, you were also guilty of breaking the following statutes:

```
<tc:block>
<tc:conditional compare="Detail[CLAM]/DetailValue [DrunkDriver],true">
<tc:file select="Statutes/RecklessStatute.xml" /></tc:conditional>
<tc:conditional compare="Detail[CLAM]/DetailValue [Speeding],true">
<tc:file select="Statutes/RecklessStatute.xml" /></tc:conditional>
<tc:conditional compare="Detail[CLAM]/DetailValue [LoudMusic],true">
<tc:file select="Statutes/RecklessStatute.xml" />
</tc:conditional>
</tc:block>
```

The balance due for this claim and the services provided by ClientGuard is a total of <tc:compute method="-" format="####.###"><tc:data select="ClaimEstimate" /><tc:data select="ClaimPaid" /></tc:compute> dollars. Please send your payment to ClientGuard as soon as possible.

```
<tc:input select="UserInitials" />
<tc:input select="LetterDate" />
```

```
<tc:data select="Name" />
<tc:data select="NumberString" />
```

If you have any questions regarding your remaining balance, please contact any of the following claim associates here at ClientGuard: <tc:loop select="AssigneeList">

<pre>&lt;tc:data select="User/Contact/FirstName" /&gt; &lt;tc:data select="User/Contact/Name" /&gt;</pre>	<pre>&lt;tc:data select="Type/Name" /&gt;</pre>	<pre>&lt;tc:data select="User/Contact/DefaultPho ne/PhoneString" /&gt;</pre>
---	---	--

```
</tc:loop>
```

If you have any additional information you would like to include with your claim, please contact <tc:loop select="AssigneeList" qualifier="Type\_TreePosition=CLAM\_SPRV"> <tc:data select="User/Contact/FirstName" /> <tc:data select="User/Contact/Name" /> <tc:data select="User/Contact/DefaultPhone/PhoneString" />. </tc:loop>

Sincerely,

```
<tc:user>
<tc:data select="Contact/FirstName" /> <tc:data select="Contact/Name" />
<tc:data select="Contact/Title" />
</tc:user>
```

```
<tc:input select="AdditionalComments" />
```

```
<tc:input select="UserInitials" />
<tc:input select="LetterDate" />
```

### 3.1.20 Completing the Document Template

After you have included all of the necessary tags in the content of the document template, you must do the following operations to finish creating the template:

- Create the document tags.
- Create the template header.
- Convert the template to XML.
- Upload the template.
- Test the template.

After you have finished these operations, your template is completed and ready for the end user.

#### 3.1.20.1 Creating Document Tags

The document tag of a document template specifies all the necessary background information such as the name of the template within the Document Generator and the mime type used to open the generated document.

Every document template must have the tc:document tag. One of the final steps you complete as you finalize your document template is adding this tag.

The following table provides a list of the attributes and possible attribute values that must be used with the tc:document tag.

**Document Tag Attributes**

Tag attribute	Attribute value	Description
version	"1.0"	Specify the XML version (currently 1.0).
mime-type	"MimeType"	Specify the application that is launched by TeamConnect to open the generated document. The mime-type is for a specific word processor or RTF format. For example, the mime type for Microsoft Word is application/msword, while the mime type for RTF is application/rtf.  For mime types msword and rtf, the mime type must match the file type of the template itself. For example, if the mime type is msword, the document template must be a Word (.DOC) file.
name	"TemplateName"	Specify the name of the template. This is the name that the user will see when he/she selects templates to generate documents from them.

		<ul style="list-style-type: none"> <li>Do not include a file extension in the name. The file extension, such as .doc, is inserted automatically when a document is generated from the template, based on the mime type.</li> <li>The template name must be unique among templates for the object.</li> </ul>
xmlns:tc	"http://www.w3.org/1999/XSL/Transform"	Specifies what name space the prefix "tc" is bound to (used for parsing).

The following is an example of a tc:document opening tag in a document template. The mime type indicates that Microsoft Word is the desired output format and that the name of the template that the user will see is "Notice of Auto Accident."

**Document Section**

```
<tc:document version="1.0" mime-type="application/msword"
name="Notice of Auto Accident" xmlns:tc="http://
www.w3.org/1999/XSL/Transform">
```

### 3.1.20.2 Creating Header and Converting Template to XML

Once you have finished inserting tags in the content section of your document template and have identified what filters and input pages you need to define in the header, you are ready to finalize the template by creating the header and converting the file to XML.

**Important:** Before completing the following process, verify that you have completed [Inserting Tags and Preparing RTF Files](#).

#### To create the header section and convert the template to XML

- Open the document template, which is an RTF file, in a text editor or HTML editor, such as EditPlus or Notepad™.
- At the beginning of the document, type the following:

```
<tc:content>
```

**Caution:** Do not place a line return after this tag. If the source code for the document content does not start immediately after the tc:content tag, TeamConnect will encounter an error when trying to generate this document.

- At the end of the document, type the corresponding end tag:

```
</tc:content>
```

**Caution:** Do not place a line return before this tag.

- Save the document as a new file with an XML extension.

5. With a text editor such as Notepad or HTML editor such as EditPlus, open the Document Content XML file.
6. At the beginning of the document before the opening tc:content tag, insert the following tags, with a line return after each, in the following order:

- a. `<?xml version="1.0" encoding="UTF-8"?>`
- b. `<tc:document version="1.0" mime-type="MimeType" name="TemplateName" xmlns:tc="http://www.w3.org/1999/XSL/Transform">`

For more details about the attributes for the document tag, such as the mime type and template name, see [Completing the Document Template](#).

- c. `<tc:header>`
7. Nest the appropriate tc:filter and tc:input tags after the tc:header tag. If your document template does not require any user input, then you will not nest any tags within the tc:header tag.
8. Close the header section by inserting the following tag after the last tc:filter or tc:input tag.

```
</tc:header>
```

**Important:** You must include the open and close tc:header tags whether or not you are using tc:filter and tc:input.

9. At the end of the document after the closing tc:content tag, insert the following tag:

```
</tc:document>
```

This completes the template.

10. Save the XML document.
11. Test your XML document by opening it in Internet Explorer to make sure that your coding has no errors.

After you save the XML document, it is ready to be uploaded into TeamConnect as a document template. For more information, see [Uploading Document Templates](#).

## Header Example

The following is an example of a header. This header includes the filter and input tags necessary for the claim notification letter shown in the examples throughout this document. For example, see [the document generator template page two image](#).

```
<tc:header><tc:filter name="SelectInvolvedContact"
searchCondition="Involved_Contact" hidden="no" label="Please select an Involved
Party" displayString="FirstName;Name" displayFormat="* *" />
  <tc:filter name="InvolvedContactAddress" objectFrom="SelectedInvolvedContact"
test="AddressList" displayString="Type/Name;Street;City" displayFormat="* - *
*" label="Please Select the Involved's Address."/>
<tc:filter name="involvedinaccident" searchCondition="Involved"
hidden="yes" />
<tc:filter name="passenger" searchCondition="Involved" hidden="yes"
displayString="DefaultCategory/TreePosition" />
```

```
<tc:input name="UserInitials" label="Enter your initials here." size="15" />
<tc:input name="LetterDate" label="Date letter will be sent" size="15" />
<tc:input name="AdditionalComments" label="Enter any additional comments."
size="50" maxChars="400" multi="yes" cols="45" rows="2"/>
</tc:header>
```

### 3.1.20.3 Uploading Document Templates

Now that you have created the content, header, and document sections of your document template and resolved all errors, you can now upload the document template into the Documents area of TeamConnect to make available to users.

#### To upload document templates into TeamConnect

1. Open the appropriate object definition and go to the object's folder.
2. Open the **Document Templates** folder.

**Important:** Remember that the template must be uploaded to the Custom or System Object on which all of its tags are based. For more information, see [Object Dependency](#).

3. Click the **Upload File** button on the toolbar. The **Upload New File** window opens.
4. Click the **Browse** button in the **Upload New File** window and find the document template you would like to upload.
5. Click **Upload File**.

The XML file you created is immediately available to the user when invoking the Document Generator.

The [document generator template image](#) is an example of a finished letter that was created with the Document Generator. Compare the finished letter to the template shown in [the document generator template page two image](#). Notice the data that each tag in [the document generator template page two image](#) retrieved and inserted into the generated document in [the document generator template image](#).

**Caution:** Once you have chosen the directory that will hold your document template, you should not rename, delete, or move that directory. You should also not rename delete or move files within that directory. If you do these operations, the associated documents will no longer work correctly for end users at runtime.

### 3.1.20.4 Testing Document Generator Templates

After uploading a document template, you must ensure that it is retrieving data correctly by using it to generate documents with sample data that has been entered into the system. You will need to make sure that the records you use to test your template have the data that is being utilized by the template.

While verifying that your template is functioning as you have intended, use the following checklist to help you identify possible issues. For more information, see [Troubleshooting Document Templates](#) for commonly encountered issues.

Keep in mind the following tips and cautions:

- There are no errors when you try to generate the document.
- When you view the generated document, all of the tags retrieve data from the correct sources (for example, user input and TeamConnect data). No tag is skipped unless there is no data for the tag to retrieve.
- When you view the generated document, there are no pieces of tags remaining in the document.
- If the template contains any tc:conditional tags, the content inside the tc:conditional tag is being generated only when the condition is met, and not when the condition is NOT met.
- If the template contains any tc:filter tags, the tags are retrieving data from all related records, according to the qualifier, or according to the user's selection, depending on how you have defined it.
- If the template contains any tc:search tags, the tags are retrieving data from the correct related records.
- If the template contains any tc:loop tags, the tags are retrieving data from all of the correct sub-objects.
- Templates that request fields with null values or empty strings (spaces without characters or numbers) will display in the Missing Data screen.
- If the template contains any tc:conditional tags to test or compare a value that is either null or an empty string, the logic will return false. The Missing Data screen for that document will not display the field for the compared value.
- If the template contains any tc:filter tags, and if there is no selection available that meets the filter criteria, that selection and its associated system and custom fields will not be included in the Missing Data list.

### 3.1.21 Using EasyDocs

EasyDocs is a method of creating document templates, which has been built upon the Document Generator feature in TeamConnect. EasyDocs allows you to create Document Generator templates without having to manually code the XML tags. Instead, you create an RTF file with mail merge fields and then perform mapping to TeamConnect data in the user interface, using a Data Mapping tool. See [Locating the Data Mapping Tool](#) for more information.

EasyDocs helps speed up the process of creating document templates by eliminating much of the potential for human error, because you do not have to manually code the XML tags.

To get started with EasyDocs, review the following explanations:

- [EasyDocs vs. Document Generator: Functional Scope](#)
- [Creating a Document Template: 5-Step Process Overview](#)
- [Required TeamConnect User Group Rights](#)



### 3.1.21.1 EasyDocs vs. Document Generator: Functional Scope

EasyDocs provides most, but not all, of the functions of Document Generator. If EasyDocs does not provide the data that you need to populate in the generated document, you may need to create document templates using Document Generator XML tags, as described in [About Document Generator](#).

The following table lists the available XML tags in Document Generator and the equivalent merge field type in EasyDocs.

**EasyDocs vs. Document Generator**

Type of data to retrieve	Document Generator tag	EasyDocs merge field type	Limitations in EasyDocs
System field	tc:data	Basic merge field or automatically mapped merge field	
Custom field	tc:detail	Basic merge field	
Fields automatically retrieved from a sub-object	tc:loop	loop@	EasyDocs cannot retrieve a list of categories added to the record (detailList).
Current date	tc:date	Basic merge field, mapped to today's date	
Current user	tc:user	Basic merge field, mapped to current user	
Content from another file	tc:file	Basic merge field, mapped to file	
Conditional content	tc:conditional	if@	
Fields from a related object retrieved automatically without user interaction	tc:search	loop@	
Fields from a related object, sub-object, or unrelated object, according to user selection	tc:filter	filter@	EasyDocs cannot retrieve data from unrelated objects.  Example: A drop-down list of contacts that are unrelated to the current

			record cannot be provided to the end user.
Text entered by user at the time of document generation	tc:input	Basic merge field, mapped to user input	
Several text entry fields in one user input screen	tc:inputForm	Not available in this version	
Conditional content based on multiple conditions	tc:block, used with tc:conditional	Not available in this version	
Computed numeric value	tc:compute	Not available in this version	
Static value used for tc:compute calculation	tc:operand	Not available in this version	

#### 3.1.21.1.1 Creating a Document Template: 5-Step Process Overview

EasyDocs provides a graphical user interface for creating Document Generator templates.

This is how a template is created and configured:

[Step 1: Create RTF with Merge Fields.](#)

[Step 2: Upload RTF File to TeamConnect.](#)

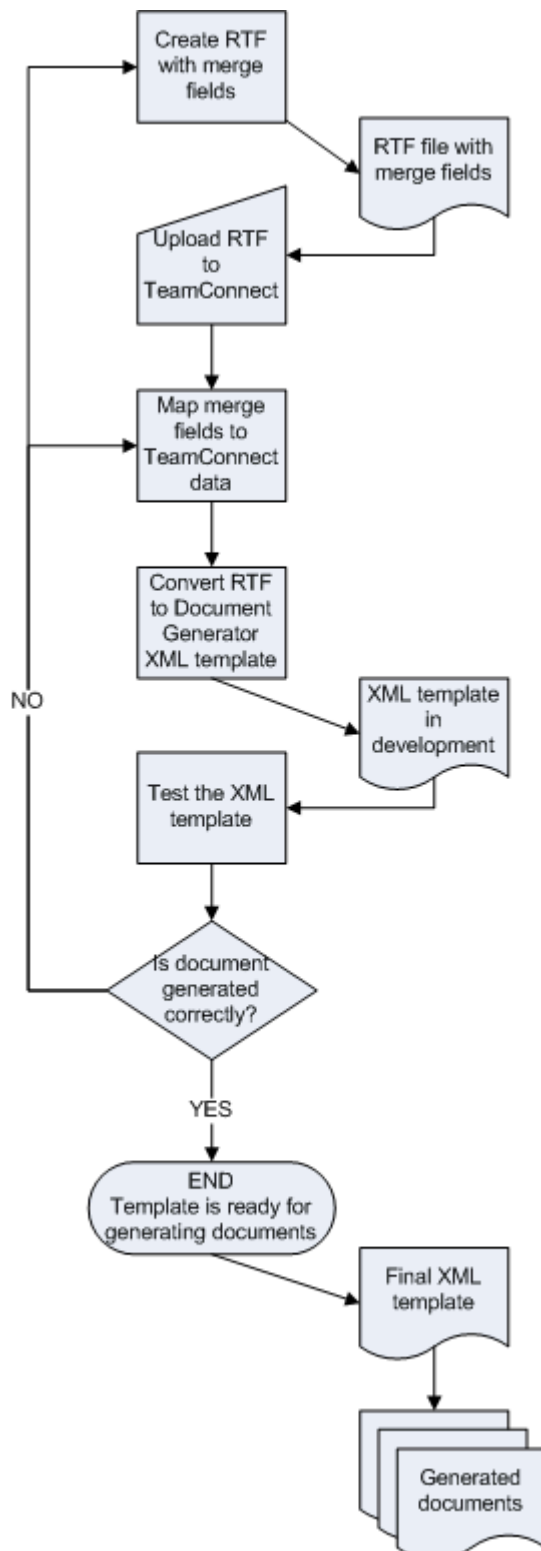
[Step 3: Map RTF Merge Fields to TeamConnect Data.](#)

[Step 4: Convert RTF File to a Document Generator Template.](#)

[Step 5: Test Your Template.](#)

While configuring a template, you will most likely need to cycle through these steps several times until all of the merge fields and mapping are configured properly.

The following flowchart provides a visual depiction of this process.

**EasyDocs Template Creation Process**

### 3.1.21.1.2 Required TeamConnect User Group Rights

In order to perform the mapping of an RTF to TeamConnect data and to test a document template, you must belong to a user group that has the appropriate rights assigned to it by an administrator. These include, but are not necessarily limited to:

- For Documents:
  - Read
  - Update
  - Create
  - Perform Template Data Mapping
  - Generate New XML Based Document
- For the record type with which the document template is associated (such as Matters):
  - Read
  - Create
  - Update
  - Also make sure you have rights to its categories and sub-objects (such as assignees).
- For all related records with which the main record type is associated, if data from these records needs to be retrieved in the document (such as involved parties, child or embedded project records, appointments, tasks, and so on):
  - Read
  - Create
  - Update
  - Also make sure you have rights to its categories and sub-objects.

Please ensure that your group has these rights given to it, at a minimum, before proceeding with the instructions in this document.

### 3.1.21.2 Step 1: Create RTF with Merge Fields

The first step for creating a Document Generator template through EasyDocs is to create a Rich Text Format file (RTF) using Microsoft Word or Corel WordPerfect that includes mail merge fields. The RTF file will become the template in TeamConnect.

**Important:** Make sure that you have the appropriate rights assigned to you, as listed in [Required TeamConnect User Group Rights](#).

#### To create an RTF with merge fields

1. Create a Word or WordPerfect document that you want to convert to a Document Generator template.

2. Insert a separate field of the type "MergeField" for each spot in your document where you would like to retrieve data from TeamConnect.

For example, if you want a contact name to appear in a letter, you must insert two merge fields, one for the first name and one for the last name, because this is how the name is stored in TeamConnect.

Be sure to read the tips in step 3 to avoid errors in the fields that you insert.

**Note:** Merge fields appear in your document as field names surrounded by angle brackets, such as <<assignee>>. However, you cannot simply type such names directly into your document. You must use the special menu commands and procedures in Word or WordPerfect to create merge fields.

Here is an example of inserting a merge field in Microsoft Word 2002:

- Position your cursor at the point in your document where you want the merge field to appear.
- Choose menu items Insert | Field.
- A dialog appears like that shown in [the merge field in Microsoft Word 2002 image](#). Choose "MergeField" from the list box on the left. Type a "Field name" for your merge field where prompted. Click OK. The merge field appears in the document.

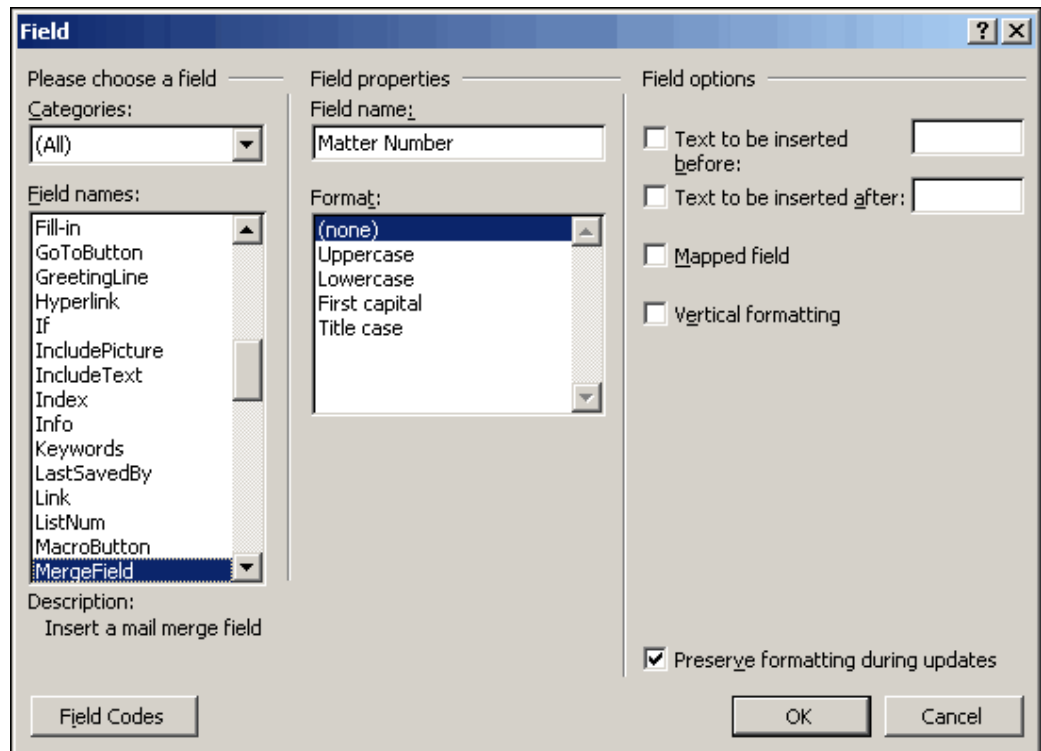
The [simple field names RTF sample image](#) shows an example of an RTF with merge fields that map to TeamConnect data. Most of the merge field names are designed to be easy to read. If you use simple, easy names you must manually use Object Navigator, later, to map each merge field name to its proper data attribute.

The [advanced field names RTF sample image](#) shows a similar RTF, but this one uses precise syntax rules to create more complex names for most of the fields. The advantage of using this special syntax is that the data mapping procedure is done automatically for fields that have names in this syntax. If you already know the relationships between your objects and you know the syntax rules, you can save time by typing field names in this syntax.

Both RTF files use the special "filter@" and "loop@" tags that will be explained later in this chapter. Both RTF files produce exactly the same output (shown in [the sample output from EasyDocs image](#)). The only difference between them is the use of special syntax in some of the field names in [the advanced field names RTF sample image](#).

For more information about how to set up your merge fields in the RTF, see the following:

- [Basic Merge Fields](#)
- [Automatically Mapped System Fields](#)
- [Automatically Mapped Custom Fields](#)
- [loop@ Merge Fields for Automatically Selecting Related or Sub-objects](#)
- [filter@ Merge Fields for User-selected Related or Sub-objects](#)
- [if@ Merge Fields for Conditional Text](#)



Creating a merge field in Microsoft Word 2002

3. While inserting merge fields, keep the following in mind:
  - Do not insert two merge fields immediately next to each other. You must type a space or other text between merge fields to avoid errors.
  - Merge field names can include these characters: "A-Z", "a-z", "0-9", ".", and "\_". Do not include spaces in merge field names.
  - Do not include other types of fields in the RTF file that are possible in Word or WordPerfect, such as the current date. TeamConnect only recognizes mail merge fields. To insert the current date, create a basic merge field, which you can later map to the current date.
  - If you use the same merge field name more than once in an RTF to identify the same piece of information, you only have to map it once in TeamConnect.
  - If you need to include multiple merge fields that identify similar data, make sure to give them distinctly different names, because later they will all be listed in a single drop-down list when you are mapping them.
4. Save the RTF file.
5. Upload the RTF file as described in [Step 2: Upload RTF File to TeamConnect](#), after completing your merge fields. If you need to, you can make changes after uploading the file and mapping the merge fields that you created so far.

«Date»

Matter Name: «Matter Name»  
Matter Number: «Matter Number»  
Main Assignee: «Main Assignee First Name» «Main Assignee Last Name»  
Main Assignee Phone: «Main Assignee Phone»

Contact-centric field's contact:

«Contact First Name» «Contact Last Name»  
«Contact Address Street»  
«Contact Address City», «Contact Address State» «Contact Address Postal Code»

Dear «Contact Prefix» «Contact Last Name»,

The following Involved Parties of type Outside Counsel are involved in this matter:

«filter@Involved Parties»«Involved First Name» «Involved Last Name», «Involved Role»  
«filter@»

Here is a list of all Involved Parties, regardless of type:

«loop@Involved Parties»«Involved First Name» «Involved Last Name», «Involved Role»  
«loop@»

Sincerely,

«Current User First Name» «Current User Last Name»

«Current User Initials»

**Sample RTF File (simple field names)**

«Date»

Matter Name: «@name»  
Matter Number: «@numberString»  
Main Assignee: «@mainAssignee.firstName» «mainAssignee.name»  
Main Assignee Phone: «MainAssigneePhone»

I

Contact-centric field's contact:

«@contact.firstName» «@contact.name»  
«@contact.defaultAddress.street»  
«@contact.defaultAddress.city», «@contact.defaultAddress.state»  
«ContactAddressPostalCode»

Dear «@contact.prefix» «@contact.name»,

The following Involved Parties of type Outside Counsel are involved in this matter:

«filter@InvolvedParties»«InvolvedFirstName» «InvolvedLastName», «InvolvedRole»  
«filter@»

Here is a list of all Involved Parties, regardless of type:

«loop@InvolvedParties»«InvolvedFirstName» «InvolvedLastName», «InvolvedRole»  
«loop@»

Sincerely,

«@user.firstName» «@user.name»

«@user.userName»

**Sample RTF File (advanced field names)**



October 9, 2007

Matter Name: Frank Brown - Eligibility  
Matter Number: Case THX-1138  
Main Assignee: Jake Ashby  
Main Assignee Phone: 323-909-5582

Contact-centric field's contact:  
Frank Brown  
2408 Glendale Blvd.  
Glendale, CA 91014

Dear Mr. Brown,

The following Involved Parties of type Outside Counsel are involved in this matter:  
Nathan Becker, Outside Counsel

Here is a list of all Involved Parties, regardless of type:  
Nathan Becker, Outside Counsel  
Thomas Jones, Claimant

Sincerely,  
Fred Dobbs

fred

---

The sample output from the EasyDocs templates in the previous images

#### 3.1.21.2.1 Basic Merge Fields

The purpose of merge fields in a template is to identify specific elements of TeamConnect data that should be retrieved when a document is being generated.

Basic merge fields are merge fields that do not need to follow any specific naming convention. You can specify any name when you define them in the RTF. Each merge field can be mapped to a simple data element in TeamConnect, such as a number, date, text, or list field.

Basic merge fields can be used to identify each of the following types of information:

- Data from system fields (for example, fields in the **General** block of the record).

If you want to map system fields automatically, you should follow a naming convention instead. See [Automatically Mapped System Fields](#).

- Data from custom fields (fields that are specifically created for your TeamConnect implementation and belong to specific Categories).

If you want to map custom fields automatically, you should follow a naming convention instead. See [Automatically Mapped Custom Fields](#).

- Text that the user needs to enter manually at the time of generating a document with this template.
- The current date.
- Text from another file.
- Attributes of the current user who is generating the document (usually, the current user's first and last name)

Also see [Predefined Merge Field Codes](#) for predefined mapping to the current user.

For example, if a document template is based on a matter, you may want to retrieve the matter name and matter number. To do this, you can create merge fields with names like **MatterName** and **MatterNumber**, and this will help you to recognize them when you are mapping them in TeamConnect.

The following table provides examples of basic merge fields for a document template that is based on a project (such as Matter). You would need to map each of these fields manually after uploading the RTF into TeamConnect.

**Examples of Basic Merge Fields**

Data to be retrieved	Example merge field name	Sample output
Project name (system field)	<<MatterName>>	Company X vs. Company Y
Project number (system field)	<<MatterNumber>>	MATT-1234
Involved party selected in a custom field of type Involved	<<OCAttorneyFirstName>> <>> <<OCAttorneyLastName>> >	Kerry Goldman
Date from a custom field	<<LossDate>>	November 1, 2006
Dollar amount from a custom field of type Number	<<EstimatedExpenses>>	\$4,000.00
Text that the user provides when generating the document	<<SecretaryInitials>>	jh
Current date	<<Today>>	November 21, 2006
Content from file (Such as LetterInfo.doc)	<<InfoFile>>	"This is a paragraph of information that is shared among several templates and

		is therefore managed as a separate file."
The current user's name	<pre>&lt;&lt;currentUserFirstName&gt;&gt; &lt;&lt;currentUserLastName&gt;&gt;</pre>	Jane Smith

### 3.1.21.2.2 Automatically Mapped System Fields

Using predefined merge field codes to automatically map system fields can save time and reduce errors in the configuration of your document template. When you open the RTF in the Data Mapping tool, TeamConnect does a lot of the mapping for you. See [Locating the Data Mapping Tool](#) for more information.

There are two ways to automatically map system fields:

- For certain fields, there is a predefined merge field code that will be recognized and automatically mapped. See [Predefined Merge Field Codes](#).
- You can build the path of the system field in the merge field name. If you name the merge field with a valid path, it will be auto-mapped. See [Naming Merge Fields for Auto-mapping System Fields](#).

You can also use predefined merge field codes to automatically map custom fields. See [Automatically Mapped Custom Fields](#).

#### 3.1.21.2.2.1 Predefined Merge Field Codes

The following table provides a list of merge field codes that automatically map to certain values, mostly having to do with assignees. These predefined codes make it easy to quickly map merge fields to assignees' contact information. This technique can also be used for custom fields of type Involved, as described in [Automatically Mapped Custom Fields](#).

**Note:** The technique behind the merge field codes for people is that the colon (:) is replacing the navigation.user.contact or .contact.

#### Predefined Merge Field Codes

Object record type	System field	Predefined merge field code
Any record type (matters, tasks, involved parties, and so on)	Current date  <b>Note:</b> The date will be formatted using the format specified in System Settings. If you want to use a different format, do not automap the date.	<<@date>>

	First name and last name of current user from his/her contact record	<<@user:firstName>> <<@user:name>>
	Name of user who created the record	<<@createdBy:firstName>> <<@createdBy:name>>
Task	Task assignee's first name and last name  Note: These merge fields can be used inside of filter@ or loop@ merge fields that identify task records	<<@currentAssignee:firstName>> <<@currentAssignee:name>>
Any project (such as Matters)	Main assignee's name information from contact record	<<@mainAssignee:prefix>> <<@mainAssignee:firstName>> <<@mainAssignee:name>> <<@mainAssignee:suffix>> <<@mainAssignee:title>>
	Main assignee's contact ID	<<@mainAssignee:numberString>>
	Main assignee's default address from contact record	<<@mainAssignee:street>> <<@mainAssignee:city>> <<@mainAssignee:state>> <<@mainAssignee:postalCode>>
	Main assignee's default email address	<<@mainAssignee:email>>
	Main assignee's default phone	<<@mainAssignee:phone>>
	Main assignee's default fax	<<@mainAssignee:fax>>
	Main assignee's default Internet address	<<@mainAssignee:inetAddress>>
	Main assignee's default rate	<<@mainAssignee:defaultRateValue>>
	Main assignee's Social Security number or Tax ID	<<@mainAssignee:ss>>

## 3.1.21.2.2.2 Naming Merge Fields for Auto-mapping System Fields

You can cause system fields to be mapped automatically by following a naming convention in your merge field names:

```
@attributePath
```

where `attributePath` is the object attribute path that points from the starting object to the system field.

Object attributes are listed in [Object Model: Read This First](#). To identify a system field that is related to the main object, create a path of attribute names that leads to the needed system field, delimited by periods (.).

Keep the following points in mind:

- If the system field is within the main object on which the template is based (such as the project number), then the path only contains one attribute with no periods.
- The final attribute cannot end with an arrow (->) in the attribute name as shown in [Object Model: Read This First](#) (or in related reference tables it points to) or the Object Navigator. For example, to select a contact, you must continue the path from `contact->` to identify the name or some other attribute of the contact.
- You cannot create a path through an attribute that ends with `List->`, because these are related and sub-objects. For these, you must use `loop@` or `filter@` merge fields.
- System field names (known as object attributes) are case sensitive. Make sure to match the exact name and capitalization of the field name.

The following are examples of merge fields in a document template that is based on a project (such as Matter) that would be mapped automatically.

**Examples of System Field Automatic Mapping**

System field	Required merge field name	Sample output
Project name	<<@name>>	Company X vs. Company Y
Project number	<<@numberString>>	MATT-1234
Parent project name	<<@parent.name>>	Sorrelson Dispute
Opened On date of project	<<@openedOn>>	November 21, 2006
Name of contact selected in contact-centric field of project	<<@contact.firstName>> <<@contact.name>>	Joy Ferguson
Main assignee first name, last name	<<@mainAssignee.contact.firstName>> <<@mainAssignee.contact.name>>	Michael Ferguson
Main assignee's default phone number	<<@mainAssignee.contact.defaultPhone.phoneString>>	323-900-1700

Involved party's default role (when used inside of a filter@ merge field)	<<@defaultCategory.name>>	OC Attorney
Project's current phase	<<@currentPhaseType.name>>	Intake

### 3.1.21.2.3 Automatically Mapped Custom Fields

You can use predefined merge field codes to automatically map custom fields. This can save you time and reduce errors in the configuration of your document template. When you open the RTF in the Data Mapping tool, TeamConnect does a lot of the mapping for you. See [Locating the Data Mapping Tool](#) for more information.

Note that when a field is automatically mapped, the formatting is set to default options. For example, an automatically mapped date value will be formatted according to the default date format set in System Settings.

#### To automatically map a merge field to a custom field

1. Create a merge field with the following merge field name:

```
@Detail[categoryTreePosition]/customFieldName
```

where `categoryTreePosition` is the tree position of the category to which the custom field belongs, and `customFieldName` is the name (not the label) of the custom field.

Examples:

```
@Detail[CLMS_DEPA]/LiJudgeSettlemtDate
```

```
@Detail[PRTP_0013]/PartyDate1
```

2. If the custom field is of type `Involved`, and you want to retrieve an attribute of the `Involved`'s contact record, you can append a colon (:) and the name of the attribute, as described in [the Predefined Merge Field Codes table](#).

For example, if `InvolvedOCAttorney` is a custom field of type `Involved` where an involved OC attorney is identified:

```
@Detail[MATT_LITI]/InvolvedOCAttorney:name
```

```
@Detail[MATT_LITI]/InvolvedOCAttorney:firstName
```

```
@Detail[MATT_LITI]/InvolvedOCAttorney:phone
```

```
@Detail[MATT_LITI]/InvolvedOCAttorney:street
```

```
@Detail[MATT_LITI]/InvolvedOCAttorney:city
```

```
@Detail[MATT_LITI]/InvolvedOCAttorney:state
```

```
@Detail[MATT_LITI]/InvolvedOCAttorney:postalCode
```

3. If the custom field is of type `Custom Object`, and you want to retrieve an attribute of the `Custom Object` record such as the name or number, you can append a period (.) and the name of the attribute.

For example, if `Litigation` is a custom field of type `Custom Object` where a `Litigation` is identified:

```
@Detail[MATT_LITI]/Litigation.name
```

```
@Detail[MATT_LITI]/Litigation.numberString
```

- If the custom field belongs to a related object of the main record for which the document template is created, nest the merge field within filter@ or loop@ merge fields. Otherwise, it does not need to be nested.

If you followed these steps correctly, the custom field is mapped automatically when you open the RTF in the Data Mapping tool in TeamConnect. See [Locating the Data Mapping Tool](#) for more information.

**Tip:** If you prefer not to map the custom field automatically, try mapping it manually in the Data Mapping tool, as described in [Custom Field Mapping](#).

#### 3.1.21.2.4 loop@ Merge Fields for Selecting Related or Sub-objects

If you need to include fields from sub-objects or related objects in the document template, and you do not want the user generating the document to select them from a list, then use loop@ in your merge field codes in the RTF file.

**Note:** For users of Document Generator, loop@ corresponds to the Document Generator tag tc:loop for sub-objects and tc:search for related objects.

The Data Mapping tool provides you with additional options for filtering the available sub-objects or related objects. For example, you may want to get the business address, not just any address listed for the contact. See [Locating the Data Mapping Tool](#) for more information.

In some cases, even with criteria that you specified, multiple sub-objects or related objects may be found. For example, a contact may have more than one business address. In this case, data from all of them will be listed in the generated document.

For details about how to perform mapping, see [loop@ Merge Field Mapping](#).

#### Related and Sub-objects Available Through filter@ or loop@

Main object for which document template is defined	Related objects/Sub-objects	Object attribute in the Object Navigator
Any object (Contact, Matter, and so on)	History	historyList->
Project (such as Matters)	Assignees	assigneeList->
	Project relations (where related project is on left side of relationship)	leftRelationList->
	Project relations (where related project is on right side of relationship)	rightRelationList->

	relationship)	
	Involved	involvedList->
	Child or embedded projects	childList->
	Tasks	taskList->
	Appointments	appointmentList->
	Accounts (budgets)	accountList->
	Expenses	expenseList->
Contact	Addresses	addressList->
	Phone numbers	phoneList->
	Email addresses	emailList->
	Internet addresses	inetAddressList->
	Fax numbers	faxList->
	Territories	territoryList->
	Rates	rateList->
	Skills	skillList->
	Contact relations (where related contact is on the left side of relationship)	leftRelationList->
	Contact relations (where related contact is on the right side of relationship)	rightRelationList->
Invoice	Line items	lineItemList->



#### 3.1.21.2.4.1 When NOT to Use loop@

Do not use loop@ under the following circumstances:

- If you want the user to manually select which sub-object's or related object's information to include in the generated document, you should use filter@. For details, see [filter@ Merge Fields for User-selected Related or Sub-objects](#).

If you want the user to manually select a sub-object or related object only when more than one is found, you should also use filter@.

- If you want to retrieve the address (or other sub-object) of the contact in a contact-centric project, you cannot use loop@. Instead, it is recommended to retrieve the default address of that contact, using a basic merge field that is mapped to an object field.
- If you want to get a sub-object that is set as "default," such as the default category, default address, or default phone number, you can do this with a basic merge field or an auto-mapped merge field, instead of a loop@ merge field. For examples, see [Automatically Mapped System Fields](#).
- If you want to get the main assignee's name, you can do this with a basic merge field or with an automatically mapped field. See [Automatically Mapped System Fields](#).

#### 3.1.21.2.4.2 Inserting loop@ Merge Fields

##### To set up loop@ merge fields for sub-objects or related objects

1. In the RTF, insert a merge field with a name that starts with loop@ followed by the text of your choice.
2. Insert (or nest) basic merge fields for the data that you need to retrieve from the sub-object or related object, formatted the way you want them to appear in the generated document.
3. Complete the loop with a merge field named simply loop@.

For example:

The local office is located at the following address:

```
<<loop@partyAddress>><<street>>  
<<city>>, <<state>> <<zip>>  
<<loop@>>
```

This would generate the following:

The local office is located at the following address:

```
124 North Woodranch Road  
Glen Oaks, CA 94551
```

For details about how to map these merge fields to TeamConnect data, see [loop@ Merge Field Mapping](#).

### 3.1.21.2.5 filter@ Merge Fields for User-selected Related or Sub-objects

Using filter@ merge fields causes the document generator to pause at runtime and prompt the user to make a selection before the generator continues.

If you want the user to select related objects or sub-object(s) from which data must be retrieved, use filter@ merge fields. The [Related and Sub-objects Available Through @filter or @loop table](#) provides a list of related objects and sub-objects that the user can select for including data in the generated document.

When you use filter@ to define merge fields, you specify which related object records or sub-objects the user can select while generating the document. Data is then pulled into the document from the user-selected related records or sub-objects.

You can also nest a filter within another filter. For example, if you need the user to select a child project, and then select an assignee, you would nest one set of filter@ merge fields within another.

**Note:** filter@ corresponds to the Document Generator tag `tc:filter`.

#### 3.1.21.2.5.1 When NOT to Use filter@

You should not use filter@ in the following circumstances:

- If you want to retrieve sub-object or related object data automatically, and you do not want the user to select which sub-object or related object to retrieve, use loop@. See [loop@ Merge Fields for Automatically Selecting Related or Sub-objects](#).
- If you want to get information from the parent record of a project, use an automatically mapped merge field. See [Automatically Mapped System Fields](#).
- If you want to get information from a record that is not related to the record for which a document is being generated, this is currently not possible with EasyDocs. Instead, you must create a Document Generator template manually using XML tags, as described in [About Document Generator](#).

#### 3.1.21.2.5.2 Inserting filter@ Merge Fields

##### **To set up filter@ merge fields for related or sub-objects**

1. In the RTF, insert a merge field with a name that starts with filter@ followed by the text of your choice.
2. Insert (or nest) basic merge fields for the data that you need to retrieve from the sub-object, formatted the way you want them to appear in the generated document.
3. Complete the filter with a merge field named simply filter@.

For example:

```
The following parties are participants in this matter:  
<<filter@Participants>><<@contact.firstName>> <<@contact.name>>,  
<<@defaultCategory.name>> <<@Detail[PRTP]/AgencyNumber>>  
<<filter@>>
```

This would generate the following, if several related objects are retrieved:

```
The following parties are participants in this matter:
```

```
Reingold & Silverman, Outside Counsel Firm A-3433
```

```
John Silverman, Outside Counsel Attorney A-3434
```

```
Susan Yessler, Judge/Arbiter A-3420
```

For details about how to map these merge fields to TeamConnect data, see [filter@ Merge Field Mapping](#).

#### 3.1.21.2.6 if@ Merge Fields for Conditional Text

If you need the document generator template to include certain content only under a certain condition, use if@ merge fields.

The conditional content can include document content, such as text, and any other types of merge fields (basic, automatically mapped, loop@, and filter@ merge fields).

An if@ merge field can do any of the following:

- Check if a certain category has been added.
- Check if a system field has a value or not.
- Check if a custom field has a certain value.
- Check if a sub-object of a certain type has been added.

You define what the condition is that you need to check when you map the merge field to TeamConnect data.

**Note:** *if@ corresponds to the Document Generator tag tc:conditional.*

#### To set up if@ merge fields to identify conditional content

1. In the RTF, insert a merge field with a name that starts with if@ followed by the text of your choice.
2. Insert (or nest) basic merge fields for the content that you need to be generated only when the condition is met, formatted the way you want it to appear in the generated document.
3. Complete the filter with a merge field named simply if@.

For example:

```
<<if@LitigationCategoryAdded>> If the condition is true, you will see the content of a related file.
```

```
Content of another file:
```

```
<<ImportedFileContent>>
```

```
<<if@>>
```

If true (that is, if the Litigation category exists), then the output in the generated document would be as follows:

```
If the condition is true, you will see the content of a related file.
```

```
Content of another file:
```

"This is a paragraph of information that is shared among several templates and is therefore managed as a separate file."

For details about how to map the if@ merge field to define the condition, see [if@ Merge Field Mapping](#).

### 3.1.21.3 Opening the Document Templates Folder

To upload RTF files to TeamConnect, and then map RTF merge fields to TeamConnect data, you must first locate the Documents Template folder.

The Document Templates folder can contain the following, as shown in [the Document Templates Folder Example image](#):

- **RTF files**—The original files upon which the document templates are based.
- **DM files**—Files containing mapping information for an RTF file.

***Caution:** Do not modify or delete these files. They are automatically generated by the Data Mapping tool to store the mapping information, and should not be opened, modified, or touched.*

- **XML files**—Actual document templates used by the system when a user generates documents.

***Caution:** Do not modify or delete these files, unless they are created manually without the use of EasyDocs and have no corresponding RTF and DM files.*

- **Sub-folders**—Folders within the Document Templates folder for the object definition which can contain additional templates or files referenced by templates.

***Caution:** Once you choose a directory to hold your document template, do not rename, delete, or move that directory. Also, do not rename, delete, or move files within the directory. If you do, the associated documents no longer work correctly for end users at runtime.*

## Locating the Data Mapping Tool

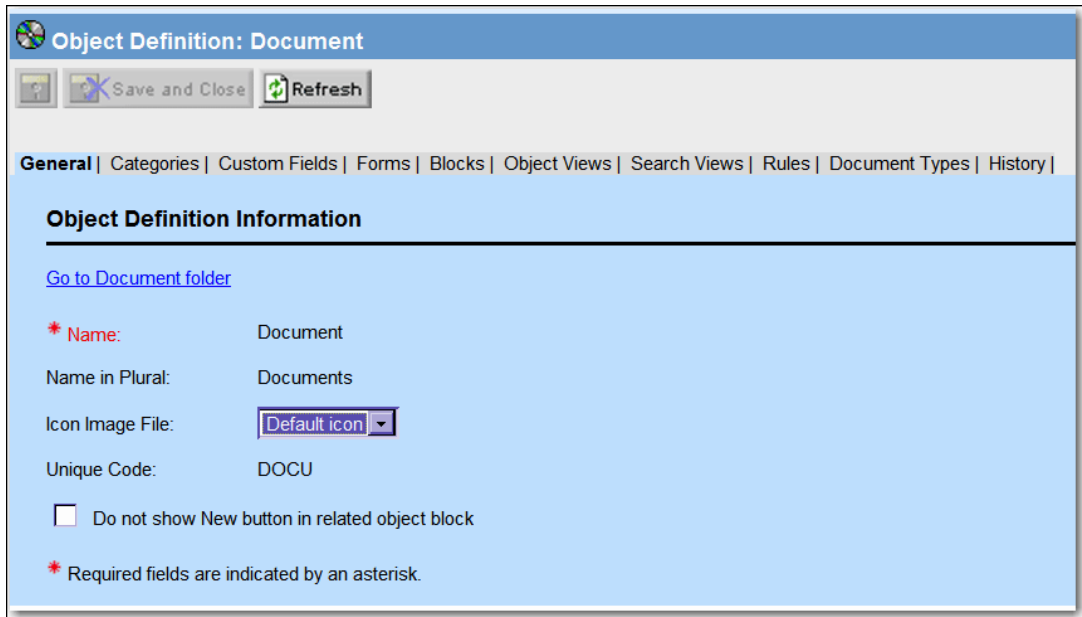
The Document Templates folder also contains the icon that opens the Data Mapping Tool.



Document Templates Folder Example

### To open the Document Templates folder

1. In TeamConnect, click the **Setup** link.  
The Designer opens.
2. In the **Go to** drop-down list, select **Object Definitions**.  
The Object Definition list view opens.
3. Click the Document object definition.  
The Document tab view opens to the **General** page.



**Object Definition: Document**

Save and Close Refresh

General | Categories | Custom Fields | Forms | Blocks | Object Views | Search Views | Rules | Document Types | History |

**Object Definition Information**

[Go to Document folder](#)

\* Name: Document

Name in Plural: Documents

Icon Image File: Default icon

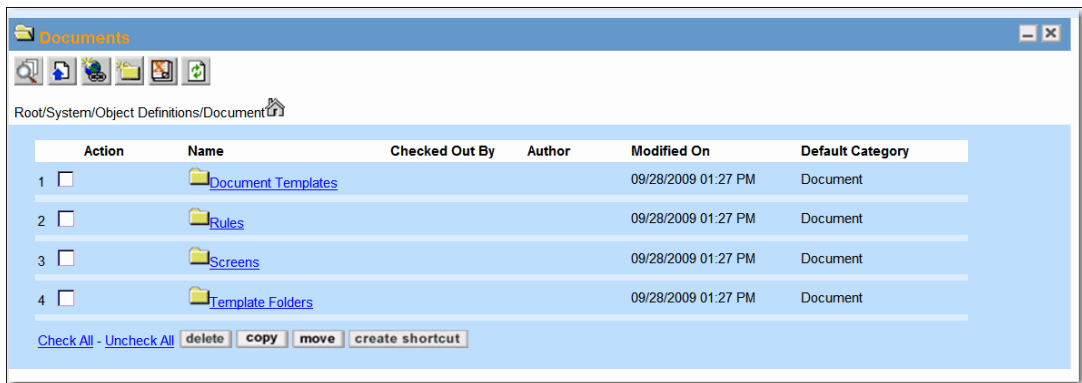
Unique Code: DOCU

Do not show New button in related object block

\* Required fields are indicated by an asterisk.

Document Tab View

4. Click the **Go to Document folder** link.
5. The Document page opens and the **Document Templates** folder is listed.



Documents

Root/System/Object Definitions/Document

Action	Name	Checked Out By	Author	Modified On	Default Category
1 <input type="checkbox"/>	<a href="#">Document Templates</a>			09/28/2009 01:27 PM	Document
2 <input type="checkbox"/>	<a href="#">Rules</a>			09/28/2009 01:27 PM	Document
3 <input type="checkbox"/>	<a href="#">Screens</a>			09/28/2009 01:27 PM	Document
4 <input type="checkbox"/>	<a href="#">Template Folders</a>			09/28/2009 01:27 PM	Document

Check All - Uncheck All delete copy move create shortcut

Documents Page

6. To open the Document Templates folder, click the **Document Templates** link in the **Name** column.

The **Document Templates** folder opens.







Document Templates Folder

Once you open the **Document Templates** folder, you can view the contents or select one of the following icons to perform a task.

## Document Templates Folder Icons

Icon	Task
	Upload File
	Search Documents
	Create New Hyperlink

	Create Folder
	Launch Data Mapping Tool
	Launch Document Generator
	Refresh

#### 3.1.21.4 Step 2: Upload RTF File to TeamConnect

Use this procedure to upload an RTF file to TeamConnect once it is ready to be mapped.

After uploading an RTF for the first time, you can modify it as needed while you are setting up mapping, for example, to modify your merge fields or document formatting. Use the **Check Out** and **Check In** buttons on the **Documents** screen to upload a modified version of the RTF. For details, see [Modifying and Deleting Template Files](#).

#### To upload an RTF file into TeamConnect

1. Open the appropriate object definition and go to the object's folder.
2. Open the **Document Templates** folder. See [Opening the Document Templates Folder](#) for more information.

*Tip: You can create subfolders inside of the **Document Templates** folder to organize your templates and then upload templates to those subfolders.*

3. Click **Upload File** on the toolbar.

The **Upload New File** window opens.

4. Click **Browse** in the **Upload New File** window and find the RTF file that you want to upload.
5. Click **Upload File**.

The **Upload New File** window closes.

The RTF is listed on the screen, as shown in [the Document Templates Folder Example](#)

[image](#), with the **Not Data Mapped** icon  next to it. This icon indicates that no mapping



has been saved for the RTF. After you perform data mapping, the icon is removed (even if the mapping is only partially completed).

After you upload the RTF file, you can map the merge fields in the RTF file to fields in TeamConnect.

### 3.1.21.5 Step 3: Map RTF Merge Fields to TeamConnect Data

Once you upload your RTF with merge fields into TeamConnect, it becomes available to be mapped in the Data Mapping tool. See [Locating the Data Mapping Tool](#) for more information.

**Important:** In order to successfully map merge fields, you must be familiar with the Object Navigator. See [Using Object Navigator](#).

The techniques for mapping merge fields to TeamConnect data are described in the following procedures in this section:

- [Opening an RTF in the Data Mapping Tool](#)
- [System Field Mapping](#)
- [Custom Field Mapping](#)
- [loop@ Merge Field Mapping](#)
- [filter@ Merge Field Mapping](#)
- [if@ Merge Field Mapping](#)
- [Mapping to Current Date](#)
- [Mapping to Current User](#)
- [Mapping to a File](#)
- [Defining a User Input Screen](#)

#### 3.1.21.5.1 Opening an RTF in the Data Mapping Tool

##### To open an RTF in the Data Mapping tool

1. Navigate to the **Document Templates** folder or sub-folder where you uploaded the RTF. See [Opening the Document Templates Folder](#) for more information.

2. Click **Launch Data Mapping Tool** .

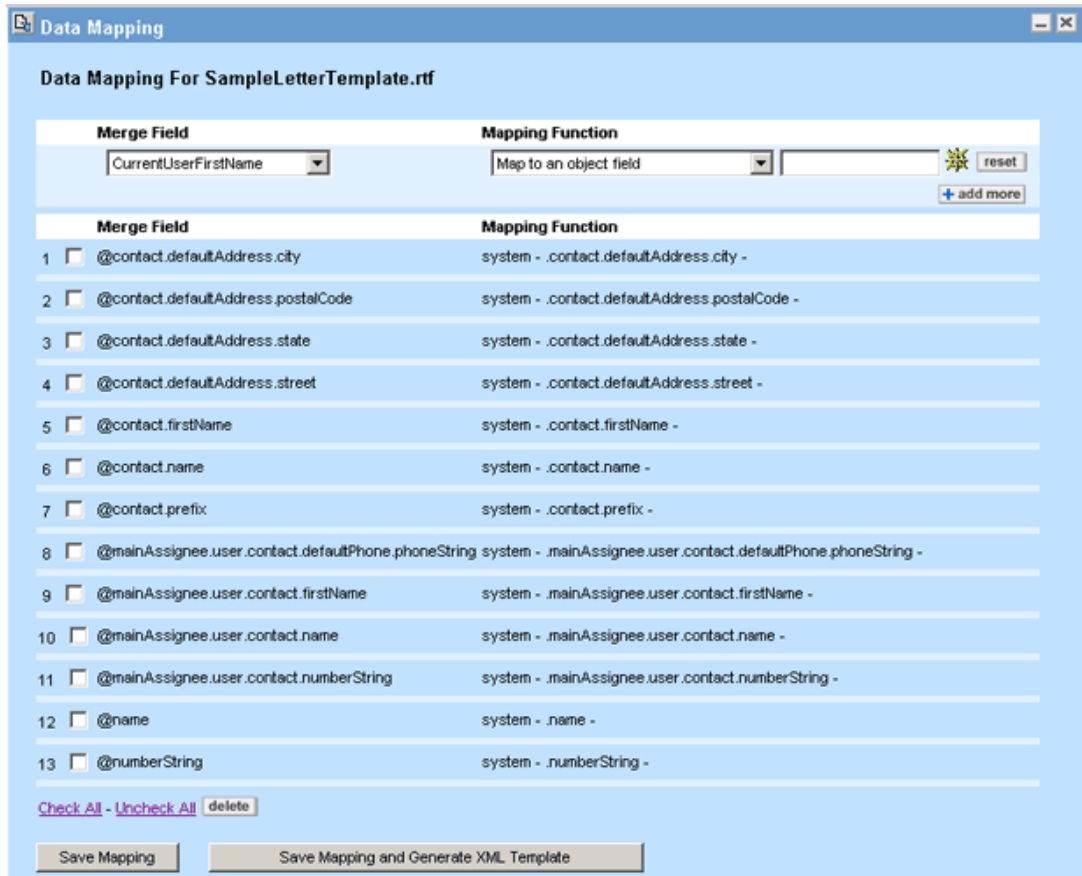
The **Data Mapping** screen opens, with a list of all RTFs in the current **Document Templates** folder that contain unmapped merge fields.

3. If you do not see the button, make sure that you are in the HTML version of the **Documents** screen. Also, make sure that you have the right to the tool assigned to you, as listed in [Required TeamConnect User Group Rights](#).

- Click the name of the RTF that you want to map.

The mapping screen of the Data Mapping tool opens, as shown in the following image.

If you configured any merge fields to be automatically mapped to system fields, as described in [Automatically Mapped System Fields](#), then the successfully mapped fields are already listed on the lower part of the **Data Mapping** screen.



**Data Mapping Tool**

### 3.1.21.5.2 System Field Mapping

There are two ways to map a merge field to a system field.

- If you correctly name the merge field as described in [Automatically Mapped System Fields](#), then the mapping is done automatically when you open the RTF in the Data Mapping tool.
- If you created the merge field using your own name for it, you must perform the mapping in the Data Mapping tool.

See [Locating the Data Mapping Tool](#) for more information.

#### To map a basic merge field to a system field

1. Open the RTF in the Data Mapping tool. See [Opening an RTF in the Data Mapping Tool](#).
2. Select the merge field that you want to map to a system field from the **Merge Field** drop-down list.

The option **Map to an object field** is selected automatically as the **Mapping Function**.

3. Click the **Object Navigator** icon .

The Object Navigator opens to the T-table of the object on which your template is based.

For example, if the document template is for a custom object, such as Matter, then the Matter table appears.

4. Navigate to the system field value that you want to retrieve in the document using **traverse**.

For commonly-used system field mapping, see [the Navigation to Commonly-used System Fields table](#).

**Important:** Do not traverse through any attributes whose names end with List->. To get values from related objects or sub-objects, you must use filter@ or loop@ merge fields.

5. Click ok when you reach the system field value. The object attribute cannot end with (->).
6. If necessary, select the formatting options provided in the Data Mapping tool for your selection.

For example, if you are mapping the merge field to a field that stores a number, then the tool provides formatting options for numeric values.

7. Click **add more**.

The merge field is mapped, and the mapping is listed in the lower section of the Data Mapping screen.

8. Click **Save Mapping** to save the mapping.

Or, click **Save Mapping and Generate XML Template** to test the fields that you mapped. For details, see [Step 5: Test Your Template](#).

The following table provides examples of how to navigate to certain commonly-used system field values for Document Generator templates if your starting object is a custom object (project), such as Matter.

Navigation to Commonly-used System Fields

System field	Navigation steps	Resulting path
<b>Note:</b> These examples are for a document template that is based on a custom object.		
Project Name	1 name (ok)	.name


Project Number	1 numberString (ok)	.numberString
Project's Default Category	1 defaultCategory-> (traverse) 2 name (ok)	.defaultCategory.name
Contact-centric contact's last name	1 contact-> (traverse) 2 name (ok)	.contact.name
Contact-centric contact's first name	1 contact-> (traverse) 2 firstName (ok)	.contact.firstName
Contact-centric contact's default address	Must have a separate merge field for each field in the address. Example navigation: 1 contact-> (traverse) 2 defaultAddress-> (traverse) 3 street (ok)	Examples: <ul style="list-style-type: none"> <li>• contact.defaultAddress.street</li> <li>• contact.defaultAddress.city</li> <li>• contact.defaultAddress.state</li> <li>• contact.defaultAddress.postalCode</li> </ul>
Main assignee's last name	1 mainAssignee-> (traverse) 2 user-> (traverse) 3 contact-> (traverse) 4 name (ok)	.mainAssignee.user.contact.name
Main assignee's first name	1 mainAssignee-> (traverse) 2 user-> (traverse) 3 contact-> (traverse) 4 firstName (ok)	.mainAssignee.user.contact.firstName
Main assignee's role	1 mainAssignee-> (traverse) 2 type-> (traverse) 3 name (ok)	.mainAssignee.type.name
Current Phase	1 currentPhaseType-> (traverse) 2 name (ok)	.currentPhaseType.name

A system field from the parent record of the project	1 parent-> (traverse) 2 Then, navigate as described for the needed parent system field.	Examples: <ul style="list-style-type: none"> <li>• <code>.parent.numberString</code></li> <li>• <code>.parent.mainAssignee.user.contact.name</code></li> </ul>
Name of an assignee with a certain role	Map a <code>loop@</code> merge field to <b>assigneeList-&gt;</b> , as described in <a href="#">loop@ Merge Field Mapping</a> .  Or, map a <code>filter@</code> merge field to <b>assigneeList-&gt;</b> if you want the user to select the assignee. See <a href="#">filter@ Merge Field Mapping</a> .	
A system or custom field from a child record of the project (or from an embedded object record)	Map a <code>filter@</code> merge field to <b>childList-&gt;</b> , as described in <a href="#">filter@ Merge Field Mapping</a> .	

### 3.1.21.5.3 Custom Field Mapping

You can use this procedure to retrieve custom field values.

#### To map a merge field to a custom field

1. Open the RTF in the Data Mapping tool. See [Opening an RTF in the Data Mapping Tool](#).
2. Select the merge field that you want to map to a custom field from the **Merge Field** drop-down list.  
The option **Map to an object field** is selected automatically as the **Mapping Function**.
3. Click the **Object Navigator** icon .  
The Object Navigator opens to the T-table of the object on which your template is based.  
For example, if the document template is for a custom object, such as Matter, then the Matter table appears.
4. Select **detailList->** and click **traverse**.  
The list of categories for the object appears.
5. Select the category to which the custom field belongs.
6. Click **traverse**.  
The list of custom fields for the selected category appears.
7. Select the name of the custom field.
8. If the custom field is of type Involved or Custom Object, click **traverse** and select the attribute that you want to use in the document (such as the project numberString).
9. Click **ok**.

The Object Navigator closes and the selected path appears in the Object Navigator field. One or more additional fields appear, which provide formatting options for the type of the custom field (such as date or number formatting options).

10. Select the appropriate formatting as you want it to appear in the generated document.
11. Click **add more**.

The merge field is mapped, and the mapping is listed in the lower section of the **Data Mapping** screen.

#### 3.1.21.5.4 loop@ Merge Field Mapping

Before mapping loop@ merge fields, make sure that you properly set up the merge fields that are nested within your two loop@ merge fields in the RTF, as described in [loop@ Merge Fields for Automatically Selecting Related or Sub-objects](#).

There are two steps to mapping sub-objects or related objects to be automatically retrieved:

- [Mapping loop@ Merge Field to Related Object or Sub-object](#)
- [Mapping Merge Fields Nested within loop@](#)

The **Merge Field** drop-down list will not display the basic merge fields that are nested in your loop@ merge fields in until you map your loop@ merge field to a sub-object or related object.

##### 3.1.21.5.4.1 Mapping loop@ Merge Field to Related Object or Sub-object

When you map the main loop@ merge field to a sub-object or related object, you are simply specifying the sub-object from which you want to automatically retrieve data.

#### To map a loop@ merge field to a sub-object or related object

1. Open the RTF in the Data Mapping tool. See [Opening an RTF in the Data Mapping Tool](#).
2. Select the loop@ merge field that you want to map to a sub-object from the **Merge Field** drop-down list.

The option **Map to an object field** is selected automatically as the **Mapping Function**.

3. Click the **Object Navigator** icon .

The Object Navigator opens to the T-table of the object on which your template is based.

For example, if the document template is for a custom object, such as Matter, then the **Matter** table appears.

4. Select the sub-object or related object attribute of the main object to which you want to map the loop@ merge field.

For assistance identifying the object attribute that represents the sub-object or related object that you need, see [the Related and Sub-objects Available Through filter@ or loop@ table](#).

5. Click **ok**.

The Object Navigator window closes and a second Object Navigator field appears, labeled **Qualifier**.

- (Sub-objects only) If necessary, use the **Qualifier** field to identify a system field that you want to use to filter the list of sub-objects or related objects that are retrieved in the generated document.

Typically, a list of sub-objects must be filtered by the sub-object type. For example, you may want to automatically list all assignees with the role Attorney.

To select the type as the filter, use this navigation in the Object Navigator:

- (Any)**-> (traverse)
- type**-> (ok)

The Object Navigator window closes and the path appears in the Qualifier field. A drop-down list of types (or roles) appears on the Data Mapping screen.

- Select the necessary type to use as a filter.
- (Related objects only) If you are mapping loop@ to involved parties (involvedList->), you would typically use the **defaultCategory** as the qualifier. For child projects (childList->), the qualifier should be the specific custom object records you want displayed. For more assistance, see the steps in [Mapping filter@ Merge Field to a Related or Sub-object](#).
  - Click **add more**.

The merge field is mapped, and the mapping is listed in the lower section of the **Data Mapping** screen.

The **Merge Field** drop-down list now includes the merge fields that are nested within the loop@ merge fields in the RTF. They are ready to be mapped to sub-object data.

#### 3.1.21.5.4.2 Mapping Merge Fields Nested within loop@

After you map the main loop@ merge field, the nested merge fields become available in the Data Mapping tool.

#### To map nested merge fields to sub-object or related object data

- Select one of the basic merge fields belonging to the loop@ from the **Merge Field** drop-down list.

The option **Map to an object field** is selected automatically as the **Mapping Function**.

- Click the **Object Navigator** icon .

The Object Navigator opens, with the sub-object or related object already selected in the first table.

- Navigate to the attribute of the sub-object or related object that you want to retrieve in the generated document.

The [Sub-object Nested Merge Field Mapping Examples table](#) lists examples of sub-object attributes that you may want to retrieve within loop@ and how to navigate to them.

- Click **ok** to close the Object Navigator.

The selected path appears in the Object Navigator field.

- If necessary, select the formatting options provided in the Data Mapping tool for your selection.

For example, if you are mapping the merge field to a field that stores a number, then the tool provides formatting options for numeric values.

- Click **add more**.

The merge field is mapped, and the mapping is listed in the lower section of the **Data Mapping** screen.

- Click **Save Mapping** to save the mapping.

The merge field is mapped, and the mapping is listed in the lower section of the **Data Mapping** screen.

**Sub-object Nested Merge Field Mapping Examples**

Sub-object	System field	Navigation steps	Resulting path
assigneeList->	Assignee last name	1 Select the assignee role (traverse) 2 user-> (traverse) 3 contact-> (traverse) 4 name (ok)	<code>.user.contact.name</code>
	Assignee first name	1 Select the assignee role (traverse) 2 user-> (traverse) 3 contact-> (traverse) 4 firstName (ok)	<code>.user.contact.firstName</code>
	Assignee role	1 Select the assignee role (traverse) 2 type-> (traverse) 3 name (ok)	<code>.type.name</code>
	Assignee's default phone number	1 Select the assignee role (traverse) 2 user-> (traverse)	<code>.user.contact.defaultPhone.phoneNumberString</code>



		3 contact-> (traverse) 4 defaultPhone-> (traverse) 5 phoneString (ok)	
detailList->	Category name	1 name (ok)	.name
addressList->	Street	1 Select the address type (traverse) 2 street (ok)	.street
	City	1 Select the address type (traverse) 2 city (ok)	.city
	State	1 Select the address type (traverse) 2 state (ok)	.state
	Zip	1 Select the address type (traverse) 2 postalCode (ok)	.postalCode
phoneList->	Phone number	1 Select the phone type (traverse) 2 phoneString (ok)	.phoneString
faxList->	Fax number	1 Select the fax type (traverse) 2 faxString (ok)	.faxString

#### 3.1.21.5.5 filter@ Merge Field Mapping

Every filter@ merge field will cause the letter generator to pause at runtime and prompt the user to select one or more choices from a list. The mapping determines what values that list will contain.

Before mapping filter@ merge fields, make sure that you properly set up the merge fields that are nested within your two filter@ merge fields in the RTF, as described in [filter@ Merge Fields for User-selected Related or Sub-objects](#).

There are two steps to mapping related or sub-objects to be selected by the user:

- [Mapping filter@ Merge Field to a Related or Sub-object](#)

- [Mapping Merge Fields Nested within filter@](#)

The **Merge Field** drop-down list will not display the basic merge fields that are nested in your filter@ merge fields in until you map your filter@ merge field to a related object or sub-object.

#### 3.1.21.5.5.1 Mapping filter@ Merge Field to a Related or Sub-object

When you map the main filter@ merge field to a related object or sub-object, you are simply specifying the related object (or sub-object) that you want the user to be able to select while generating a document with this template.

#### To map a filter@ merge field to a related or sub-object

1. Open the RTF in the Data Mapping tool. See [Opening an RTF in the Data Mapping Tool](#).
2. Select the filter@ merge field that you want to map to a related or sub-object from the **Merge Field** drop-down list.

The option **Map to an object field** is selected automatically as the **Mapping Function**.

3. Click the **Object Navigator** icon .

The Object Navigator opens to the T-table of the object on which your template is based.

For example, if the document template is for a custom object, such as Matter, then the Matter table appears.

4. Select the related object or sub-object of the main object to which you want to map the filter@ merge field.

For assistance identifying the object attribute that represents the related object or sub-object you need, see [the Related and Sub-objects Available Through filter@ and loop@ table](#).

5. Click **OK**.

The Object Navigator window closes and a second Object Navigator field appears, labeled **Qualifier**. This field is used to identify a condition that you want to use to limit the list of related objects or sub-objects that the user can select, if needed.

Do not specify an attribute that traverses to yet another object. You may specify simple attributes (number, text, date, boolean) or you may specify a list. If you specify a simple attribute, go to step 7.

6. For all lists, select **(Any)->** (traverse)

Your next selection depends on what kind of list you chose. In most cases, you would select **defaultCategory->** (OK). For assigneeList, you would instead select **type->** (OK). For childList, you would select **application->** (OK), then select the child or embedded record type that you want the user to see at runtime.

For example, if you are retrieving involved parties through involvedList->, you may want the list to include only involved parties with a certain role.

To select the involved role as the **Qualifier**, use this navigation:

- a. **(Any)->** (traverse)

b. **defaultCategory**-> (OK)

Object Navigator closes and the path is displayed in the Qualifier field. A drop-down list of roles is displayed on the Data Mapping screen.

c. Select the necessary role to use as a filter for which parties should appear.

7. Type the **Label** that you want the user to see when he or she is prompted to select the related or sub-object(s) when generating the document.
8. Select **Allow multiple selection** if you want the user to be able to select multiple items from the list when generating the document.
9. Select **Skip filter screen if only one record is found** if you want the end user to not have to make a selection when only one related or sub-object record is found. If this option is not selected, then the end user would see a drop-down list containing only one option and would have to click next.
10. Click **add more**.

The merge field is mapped, and the mapping is listed in the lower section of the **Data Mapping** screen.

The **Merge Field** drop-down list now includes the merge fields that are nested within the filter@ merge fields in the RTF. They are ready to be mapped to related object or sub-object data.

Once you generate the template, a user who generates a document with the template will be prompted to select a record before generating the document.

#### 3.1.21.5.5.2 Mapping Merge Fields Nested within filter@

After you map the main filter@ merge field, the nested merge fields become available in the Data Mapping tool.

#### To map nested merge fields to related or sub-object data

1. Select one of the basic merge fields belonging to the filter@ from the **Merge Field** drop-down list.

The option **Map to an object field** is selected automatically as the **Mapping Function**.

2. Click the **Object Navigator** icon .

The Object Navigator opens, with the sub-object or related object already selected in the first table.

3. Navigate to the attribute of the related object or sub-object that you want to retrieve in the generated document.

The [Sub-object Nested Merge Field Mapping Examples table](#) lists examples of sub-object attributes that you may want to retrieve within filter@ and how to navigate to them.

The [Related Object Nested Merge Field Mapping Examples table](#) lists examples of related object attributes that you may want to retrieve within filter@ and how to navigate to them.

- Click **ok** to close the Object Navigator.

The selected path appears in the Object Navigator field.

- If necessary, select the formatting options provided in the Data Mapping tool for your selection.

For example, if you are mapping the merge field to a field that stores a number, then the tool provides formatting options for numeric values.

- Click **add more**.

The merge field is mapped, and the mapping is listed in the lower section of the **Data Mapping** screen.

- Click **Save Mapping**.

#### Related Object Nested Merge Field Mapping Examples

Related object	Data to retrieve	Navigation steps	Resulting path
Any related object	Custom field value	<ol style="list-style-type: none"> <li>1 (Any)-&gt; (traverse)</li> <li>2 detailList-&gt; (traverse)</li> <li>3 Select the category of the custom field (traverse)</li> <li>4 Select the name of the custom field (ok)</li> </ol>	<p>Example:</p> <pre>.detailList(LAWY).detailDateValueList(MyDate2).detailValue</pre>
Involved parties	Contact last name	<ol style="list-style-type: none"> <li>1 (Any)-&gt; (traverse)</li> <li>2 contact-&gt; (traverse)</li> <li>3 name (ok)</li> </ol>	<code>.contact.name</code>
	Contact first name	<ol style="list-style-type: none"> <li>1 (Any)-&gt; (traverse)</li> <li>2 contact-&gt; (traverse)</li> <li>3 firstName (ok)</li> </ol>	<code>.contact.firstName</code>
	Involved party's default role	<ol style="list-style-type: none"> <li>1 (Any)-&gt; (traverse)</li> <li>2 defaultCategory-&gt; (traverse)</li> <li>3 name (ok)</li> </ol>	<code>.defaultCategory.name</code>
Histories	Description (history text)	<ol style="list-style-type: none"> <li>1 text (ok)</li> </ol>	<code>.text</code>


Child projects (such as Participants) or embedded projects	Child project name	1 (Any)-> (traverse) 2 name (ok)	.name
	Child project number	1 (Any)-> (traverse) 2 numberString (ok)	.numberString
	Contact-centric contact's last name	1 (Any)-> (traverse) 2 contact-> (traverse) 3 name (ok)	.contact.name
	Contact-centric contact's first name	1 (Any)-> (traverse) 2 contact-> (traverse) 3 firstName (ok)	.contact.firstName
Appointments	Begins On dates	1 (Any)-> (traverse) 2 startOn (ok)	.startOn
Tasks	Due on date	1 (Any)-> (traverse) 2 dueOn (ok)	.dueOn

#### 3.1.21.5.6 if@ Merge Field Mapping

The steps for mapping an if@ merge field depend on what condition you want to check:

#### To use if@ to check if a category has been added to the record


**Note:** You can check only categories of the main record for which a document is being generated.

1. Open the RTF in the Data Mapping tool. See [Opening an RTF in the Data Mapping Tool](#).
2. Select the if@ merge field that you want to map from the **Merge Field** drop-down list.  
The option **Map to an object field** is selected automatically as the **Mapping Function**.
3. Click the **Object Navigator** icon .  
The Object Navigator opens to the T-table of the object on which your template is based.  
For example, if the document template is for a custom object, such as Matter, then the Matter table appears.
4. Select **detailList->** and click **traverse**. Click **ok**.

The Object Navigator window closes and the path appears in the Object Navigator field. A drop-down list of categories appears.

5. Select the category that you want to check for existence in the record to determine whether the content within the if@ merge fields should be generated.
6. Select **Negate** if you want the content to be generated when the selected category is NOT present in the record. Otherwise, leave the check-box unchecked.
7. Click **add more**.
8. Click **Save Mapping**.


#### To use if@ to check if a system field is populated

1. Open the RTF in the Data Mapping tool. See [Opening an RTF in the Data Mapping Tool](#).
2. Select the if@ merge field that you want to map from the **Merge Field** drop-down list.  
The option **Map to an object field** is selected automatically as the **Mapping Function**.
3. Click the **Object Navigator** icon .  
The Object Navigator opens to the T-table of the object on which your template is based.  
For example, if the document template is for a custom object, such as Matter, then the Matter table appears.
4. Navigate to the system field that you want to check for a value, and click **ok**.  
The Object Navigator window closes and the path appears in the Object Navigator field.
5. Select **Negate** if you want the content to be generated when the selected system field is NOT populated in the record. Otherwise, leave the check-box unchecked.
6. Click **add more**.
7. Click **Save Mapping**.

#### To use if@ to check if a custom field has a certain value

This method is most useful for custom fields of type List, Number, or Check Box.

**Note:** You can check only custom fields belonging to the main record for which the document is being generated.

1. Open the RTF in the Data Mapping tool. See [Opening an RTF in the Data Mapping Tool](#).
2. Select the if@ merge field that you want to map from the **Merge Field** drop-down list.  
The option **Map to an object field** is selected automatically as the **Mapping Function**.
3. Click the **Object Navigator** icon .

The Object Navigator opens to the T-table of the object on which your template is based.

For example, if the document template is for a custom object, such as Matter, then the Matter table appears.

4. Select **detailList->** and click **traverse**.
5. Select the category of the custom field, and click **traverse**.
6. Select the custom field whose value you want to check and click **ok**.

The Object Navigator window closes and the path appears in the Object Navigator field.

7. Select the value that you want to check for, depending on the field type. For example, if you want to check whether a certain lookup item is selected in the field, select that lookup item.
8. Select **Negate** if you want the content to be generated when the specified custom field does NOT have the specified value. Otherwise, leave the check-box unchecked.
9. Click **add more**.
10. Click **Save Mapping**.

#### To use if @ to check if a sub-object of a certain type has been added

You can use this mapping to check whether a sub-object of a certain type has been added, and thus specify whether content should be generated.

For example, if a certain type of assignee has been added, you may want to list the assignee's contact information in the generated document.

**Note:** You can check only sub-objects of the main record for which a document is being generated.

1. Open the RTF in the Data Mapping tool. See [Opening an RTF in the Data Mapping Tool](#).
2. Select the if@ merge field that you want to map from the **Merge Field** drop-down list.

The option **Map to an object field** is selected automatically as the **Mapping Function**.

3. Click the **Object Navigator** icon .

The Object Navigator opens to the T-table of the object on which your template is based.

For example, if the document template is for a custom object, such as Matter, then the Matter table appears.

4. Select the sub-object that you want to check (such as **assigneeList->**) and click **ok**.

The Object Navigator window closes and the path appears in the Object Navigator field.

5. Select the type that you want to check for.
6. Select **Negate** if you want the content to be generated when the specified sub-object type is NOT added. Otherwise, leave the check-box unchecked.
7. Click **add more**.

8. Click **Save Mapping**.

#### 3.1.21.5.7 Mapping to Current Date

A basic merge field can be set to the current date in the document when the user generates it. You can also specify the format of the date information.

**Note:** *This mapping function corresponds to the Document Generator tag `tc:date`.*

#### To map a merge field to the current date

1. Open the RTF in the Data Mapping tool. See [Opening an RTF in the Data Mapping Tool](#).
2. Select the basic merge field that you want to map to the current date from the **Merge Field** drop-down list.
3. Select **Map to today's date** as the **Mapping Function**.  
Several additional fields for setting the date format appear.
4. Select the date format from the drop-down list.
5. Select **Show Time** if you want the time of the document generation to appear in the generated document, along with the date.
6. Select **Show Timezone** if you want the time zone of the current user to appear in the generated document, along with the date and time.
7. Click **add more**.  
The merge field is mapped, and the mapping is listed in the lower section of the **Data Mapping** screen.
8. Click **Save Mapping**.

The mapping is saved for the template.

#### 3.1.21.5.8 Mapping to Current User


If you want to retrieve information about the current user who is generating the document, such as his or her name, you can do this by mapping a basic merge field to the current user.

**Note:** *This mapping function corresponds to the Document Generator tag `tc:user`.*


#### To map a merge field to the current user's first name

1. Open the RTF in the Data Mapping tool. See [Opening an RTF in the Data Mapping Tool](#).
2. Select the basic merge field that you want to map to the current user from the **Merge Field** drop-down list.



3. Select **Map to current user** as the **Mapping Function**.
4. Click the **Object Navigator** icon .  
The Object Navigator opens to the YUser table.
5. Select **contact->** and click **traverse**.
6. Select **firstName** and click **ok**.  
The Object Navigator closes.
7. Click **add more** to add the mapping.

#### To map a merge field to the current user's last name

1. Open the RTF in the Data Mapping tool. See [Opening an RTF in the Data Mapping Tool](#).
2. Select the basic merge field that you want to map to the current user from the **Merge Field** drop-down list.
3. Select **Map to current user** as the **Mapping Function**.
4. Click the **Object Navigator** icon .  
The Object Navigator opens to the YUser table.
5. Select **contact->** and click **traverse**.
6. Select name and click **ok**.  
The Object Navigator closes.
7. Click **add more** to add the mapping.

#### 3.1.21.5.9 Mapping to a File

You can map a merge field to a file, so that the content of that file is imported into the generated document in place of the merge field. This is useful when the content is shared among multiple document templates.

If necessary, you can include merge fields that need to be mapped to TeamConnect data. In many cases, however, the external file contains only static text.

Mapping to a file requires two steps:


1. [Converting RTF File to XML File](#)—Any merge fields in the file must be mapped, and the file must be converted from RTF to XML.
2. [Mapping to XML File](#)

## 3.1.21.5.9.1 Converting RTF File to XML File

**To convert an RTF file to an XML file in TeamConnect**

1. Create a sub-folder in the **Document Templates** folder where the main RTF of your template is stored.

**Important:** You cannot map to a file unless it is stored in a sub-folder of the **Document Templates** folder of the template.

2. In the sub-folder you created, upload the RTF file that you want to import into your template.
3. Click Launch Data Mapping Tool .

The Data Mapping tool screen opens, where you can convert the RTF to XML.

4. Click the name of the RTF that you want to convert.
5. The mapping screen of the Data Mapping tool opens.

If the file contains no merge fields, the text (**No Item Selected**) appears. If the file does contain merge fields that need to be mapped, they are listed in the **Merge Field** drop-down list.

6. If necessary, map all of the merge fields.
7. Click **Save Mapping and Generate XML Template**.

A message appears that the generation is complete. The XML file is stored in the same folder as its RTF. You can now map a merge field in the main template to this file.

## 3.1.21.5.9.2 Mapping to XML File

If you already uploaded and converted the RTF to XML, you can map a merge field to it.

**Note:** This mapping function corresponds to the Document Generator tag `tc:file`.

**To map a merge field to an external XML file**

1. Open the main RTF in the Data Mapping tool. See [Opening an RTF in the Data Mapping Tool](#).
2. Select the **Merge Field** that you want to map to a file.
3. Select **Map to a file** from the **Mapping Function** drop-down list.
4. Select the file to which you want to map the merge field.
5. Click **add more**.

The merge field is mapped, and the mapping is listed in the lower section of the **Data Mapping** screen.

6. Click **Save Mapping**.

### 3.1.21.5.10 Defining a User Input Screen

When you map a merge field to user input, the Document Generator automatically creates a user input screen, so that the user generating the document can enter text to be included in place of the merge field.

**Note:** *This mapping function corresponds to the Document Generator tag `tc:input`.*

#### To define a user input screen

1. Open the main RTF in the Data Mapping tool. See [Opening an RTF in the Data Mapping Tool](#).
2. Select the **Merge Field** that you want to map to a file.
3. Select **Define an input screen for user input** from the **Mapping Function** drop-down list.

Additional fields appear for you to define the appearance of the input screen.

4. Enter the appropriate information to define the input screen:
  - **Screen Label**—Instructions to the user about what input to provide in the text box.
  - **Max Size**—Set the maximum number of characters for the input field, if you want the field to be only one line with no scrolling. This setting sets the length of the field, in characters, and also limits the text input to that number of characters.
  - **Show Scroll Bar**—Select this option if you want the user to enter text in a scrolling text box instead of a one-line text box.
  - **Rows**—When **Show Scroll Bar** is checked, this field sets the number of text rows of the text box that will be displayed. It does not limit the number of characters.
  - **Columns**—When **Show Scroll Bar** is checked, this field sets the width of the text box in the display, in characters. It does not limit the number of characters.

5. Click **add more**.

The merge field is mapped, and the mapping is listed in the lower section of the **Data Mapping** screen.

6. Click **Save Mapping**.

Now, once you generate the template, a user who generates a document with the template will be prompted to enter text first before generating the document.

### 3.1.21.6 Step 4: Convert RTF File to a Document Generator Template

After you map some or all of the merge fields in your RTF file, you are ready to convert the RTF to a Document Generator template.

If you only mapped some of the merge fields, you can convert the RTF to a template in order to test your mapping. It is useful to do this several times in order to check your progress. Or you can wait until all merge fields are mapped to test your template.

#### To convert an RTF file to a Document Generator template

1. Open the RTF that you want to convert in the Data Mapping tool. For details, see [Opening an RTF in the Data Mapping Tool](#).
  2. Click **Save Mapping and Generate XML Template**.  
A message appears, indicating that the XML template has been generated.
  3. Close the Data Mapping tool.  
The XML file has been placed in the same directory as the RTF file.
- Now you are ready to test the template as described in [Step 5: Test Your Template](#).

### 3.1.21.7 Step 5: Test Your Template

After mapping and generating a document template, you must ensure that it is retrieving data correctly by generating test documents.

**Note:** Make sure that you have the appropriate rights, as listed in [Required TeamConnect User Group Rights](#).

#### To test a Document Generator template

1. Select one of the following options to browse to the Document Templates folder:
  - Using the **Documents** tab—Click the **Documents** tab. Click **More Actions**, and then click **Generate Documents**.
  - Using the **Documents** page for a record—Open the appropriate record and click the **Documents** link in the left pane. In the toolbar, Click the **Document Generator** icon.  
The **Document Generator** screen opens and displays a list of available templates.
2. Click the Select button for the template that you want to test.
3. Make any appropriate selections and enter text input as defined by your template.  
For example, your template may require the user to select related objects or sub-objects.
4. When the document generation is completed, click **Preview Document**.
5. Open or save the file.
6. Use the [Successful Document Generation Checklist](#) to verify that the document has been generated correctly.
7. If the document is correct, click **Save Document**.

For more information, see [Generating Documents Using TeamConnect Data](#).

### Successful Document Generation Checklist

Use the following checklist when testing your generated document:

- Make sure that you have adequate test data to use. For example, if the same contact records or data values are used repeatedly, or if values are missing, it may be difficult to distinguish whether data is being retrieved from the correct fields.
- Make sure that the document generation process prompts you for all user input, related object selections, and sub-object selections.
- View the generated document and make sure that there are no unpopulated merge fields.
- Make sure that all of the merge fields retrieve data from the correct sources (for example, user input and TeamConnect data). No merge field is skipped unless there is no data to retrieve from the specified source.
- If the template contains any if@ merge fields, make sure that the content inside the if@ merge fields is being generated only when the condition is met, and not when the condition is NOT met.
- If the template contains any filter@ merge fields, make sure that they are retrieving data from all related records selected by the user.
- If the template contains any loop@ merge fields, make sure that they are retrieving data from all of the correct related or sub-objects.

For more troubleshooting assistance, see [About Document Generator](#) and [Troubleshooting Document Templates](#).

### 3.1.21.8 Modifying and Deleting Template Files

This section contains the following sub-sections:

- [Modifying an Existing RTF File](#)
- [Modifying Existing Data Mapping](#)
- [Deleting Template Files](#)

#### 3.1.21.8.1 Modifying an Existing RTF File

You may need to modify the RTF of a template after you upload it, for example, to adjust the formatting or the merge field names, or to add or remove merge fields. You can modify an RTF without losing your mapping.

**Caution:** Do not delete the RTF from the **Document Templates** folder, or you will lose your mapping and will have to re-map all of the merge fields in the RTF. Deleting the RTF will also automatically delete the XML template.

#### To modify an existing RTF File

1. Navigate to the **Document Templates** folder or sub-folder where the RTF is stored in the **Documents** area of TeamConnect.
2. Check out the RTF. (Do not delete it, or you will lose all of the mapping.)
3. Modify the RTF as needed. If you modify merge fields, keep in mind:

- If you change the name of a merge field, it will no longer be mapped in the Data Mapping tool.  
The exception is if you change a merge field from a basic merge field with a custom name to an automatically mapped merge field (for example, @contact.name).
- If you remove a merge field, the mapping will automatically be removed in the Data Mapping tool.
- If you add a merge field, you will need to map it (unless it is an automatically mapped field).

4. Check in the RTF in the **Document Templates** folder or sub-folder.
5. Open the RTF in the Data Mapping tool (see [Opening an RTF in the Data Mapping Tool](#)).
6. If any additional mapping is needed, perform the data mapping.
7. Click **Save Mapping and Generate XML Template**.

The Document Generator template is generated, with the new formatting and/or data mapping.

#### 3.1.21.8.2 Modifying Existing Data Mapping

You can modify the mapping of an existing template that has been created using EasyDocs.

##### **To modify the data mapping of an existing document template:**

1. Open the RTF of the document template in the Data Mapping tool. See [Opening an RTF in the Data Mapping Tool](#).

If there are any unmapped merge fields in the RTF, they are listed in the **Merge Field** drop-down list.

2. Map any unmapped merge fields.
3. To modify the mapping of an already mapped merge field:
  - a. Select the mapped merge field's check-box and click **delete**.
  - b. Click **OK** to confirm deletion of the mapping.  
The merge field appears as an unmapped merge field in the **Merge Field** drop-down list.
  - c. Map the merge field.

4. Click **Save Mapping and Generate XML Template** when you are finished modifying the mapping.

The template now contains the revised mapping and is ready for you to test it.

### 3.1.21.8.3 Deleting Template Files

As in any Documents folder in TeamConnect, you have the option to delete files in the **Document Templates** folder (if you have the rights assigned to you).

Keep the following in mind when you consider deleting template files:

- If you delete an RTF from a Document Templates folder or sub-folder, both the DM mapping file and the XML template file (if you generated one) are also deleted and no longer available to end users for generating documents.
- If you delete an XML template file, the RTF and any completed data mapping are retained.

However, the document template will not be available for generating documents, because the RTF file alone does not provide a document template to end users.

You can generate a new XML template for an RTF using the Data Mapping tool.

- Deleting the DM file itself is not recommended. If you delete this file:
  - You will lose all data mapping between the RTF merge fields and TeamConnect data.
  - The **Not Data Mapped** icon will appear next to the RTF file, indicating that there is no mapping saved.
  - The XML template will not be deleted, and users can continue to generate documents using the template.
  - If you need to make any changes to the template, you will have to re-map the merge fields using the Data Mapping tool. See [Locating the Data Mapping Tool](#) for more information.

## 3.1.22 Troubleshooting Document Templates

The following table lists common issues you may encounter when creating document templates and how to resolve these issues:

**Troubleshooting Document Templates**

Issue	Resolution
Error when viewing XML file in Internet Explorer:  Only one top level element is allowed in an XML document.	Make sure that the <code>tc:document</code> tag at the top is not closed and that at the very end of the document it includes the closing <code>tc:document</code> tag.
Error when viewing XML file in Internet Explorer:  A string literal was expected, but no opening quote character was found.	Search and replace the following string with double quotes in your XML file.  Replace \’94  with  "

<p>Error when viewing XML file in Internet Explorer:</p> <p>Reference to undeclared namespace prefix: 'tc'. Error processing resource.</p>	<p>You have not provided the <code>tc:document</code> tags and <code>tc:header</code> tags that are necessary for the template.</p>
<p>The web browser displays an error saying that the opening tag does not match the close tag.</p> <p>For example, <code>tc:content</code> does not match the opening tag <code>tc:data</code>. I've looked at my code and everything seems to be correct.</p>	<p>If you created your original document in Microsoft Word, it might be because the first tag on the very first line of your document was stored in the Title.</p> <ol style="list-style-type: none"> <li>1. Open Microsoft Word and select File &gt; Properties.</li> <li>2. Check the Title field found in the Summary tab. If the Title field includes a tag, remove it.</li> <li>3. Convert the RTF file to XML and try again.</li> <li>4. If the title is not the problem, verify again that you have closed all of your tags in the correct sequence of nesting.</li> </ol>
<p>When generating the document I get an error message similar to:</p> <p>[*Error in Contact*]</p>	<p>When converting your document to XML, there may have been some issues that caused extra white space in the file. Review your XML file and make sure the Document Generator tags do not have any spaces or carriage returns inside them.</p> <p>Example #1: <code>&lt;tc:data select="Contact"&gt;</code></p> <p>Example #2: <code>&lt;tc:data select="Contact"&gt;</code></p> <p>To resolve this issue find <code>&lt;tc</code> and check for any spaces or carriage returns inside the tag. Remove the spaces that were added while converting to XML so that it is on one line.</p>
<p>When generating the document I get an error message similar to:</p> <p>[*Error in DetailField: CONT_EMPL/HireDate*]</p> <p>[*Error in numberString*]</p>	<ul style="list-style-type: none"> <li>• Make sure that the tree position is correct. The tree position must include the entire path (for example, Parent_Child).</li> <li>• Make sure that the field name is correct.</li> <li>• For system fields, make sure the first letter of the attribute name is uppercase.</li> </ul>
<p>When generating the document I get an error message similar to:</p> <p>No data provided for letter content.</p>	<p>There is probably a typographical error in your <code>tc:document</code> tag. Make sure the entire tag is typed correctly. For example, make sure that the <code>xmlns</code> attribute value is correct.</p> <p>This message also appears when there are errors in other tags. Make sure that all of your tags are typed correctly.</p>



<p>When I opened the generated document, tags were replaced by empty data and there were no errors, although there is data that should have been included. What is causing the problem?</p>	<ul style="list-style-type: none"> <li>• Make sure that the field corresponding to the tag has a value in the record. When a field has null value, the Document Generator will replace the tag with nothing.</li> <li>• Make sure that you (or the user generating the document) have rights to the objects, categories, and other data that you are retrieving in the template. Any data to which you do not have rights will not appear in the generated document.</li> <li>• Make sure that the tag and its attributes do not contain any extra characters (for example, {}).</li> </ul> <p><b>Example:</b> <code>&lt;tc:detail select="CLAM/{ClaimEstimate}" /&gt;</code></p> <ul style="list-style-type: none"> <li>• Read the Issues in this table regarding errors when generating documents. If the setting is turned off to include errors in the generated document, then it may appear that data is not being retrieved when there is an error in a tag.</li> <li>• Your RTF file may be corrupted. To test it, create a new template only containing document generator tags (no text) and attempt to generate the new letter. If this template works, your old RTF file is corrupted and it should be replaced.</li> </ul>
<p>I uploaded the template, but when I try to test it by clicking on the Document Generator button, the template is not there.</p>	<ul style="list-style-type: none"> <li>• The template will not display unless it is well-formed. Make sure to open your template in a web browser before uploading it.</li> <li>• Make sure you are in an object record where that template is available when trying to test your template.</li> <li>• Make sure that you uploaded the template to the correct folder in the Documents area: <ul style="list-style-type: none"> <li>○ The template must be in the Document Templates folder for the correct Object Definition.</li> <li>○ The template must not be in a subfolder of the Document Templates folder. Only supporting XML files for a template may be in a subfolder.</li> </ul> </li> </ul>
<p>When I test my document template using Document Generator, the data from custom fields is not being inserted into the document.</p>	<ul style="list-style-type: none"> <li>• Make sure that you are testing using records where the data has been populated.</li> <li>• Make sure that you have included both the category tree position and the custom field name in the attribute value of the tag.</li> </ul>

	<ul style="list-style-type: none"> <li>• Make sure the category tree position represents the entire path to the category (for example, Parent_Child_Grandchild).</li> </ul>
<p>How do you find the tree position for custom and system lookup tables? For example tree position for address.</p>	<p>Each table found in the <b>Lookup Tables</b> screen has its own unique code.</p> <ul style="list-style-type: none"> <li>• For custom lookup tables, the unique code is defined specifically for your implementation of TeamConnect. It is located next to the name of the custom lookup table.</li> <li>• For system lookup tables, the unique code is defined by Mitrtech.</li> </ul>
<p>When I open the generated document, the RTF source code appears instead of the actual document content.</p>	<p>This occurs when there is a line return or space after the opening <code>tc:content</code> tag. Make sure to include the RTF source code immediately after the opening <code>tc:content</code> tag.</p>
<p>When I test my document template using Document Generator, I get a format error similar to:</p> <p>[*Error in format: MMMMM-dd-YY*]</p>	<p>There is an error in the format that you have provided for a date field. For example, in this error, the Y's for the year are capitalized. Remember that the <code>dateFormat</code> tag attribute for date fields is case-sensitive.</p>
<p>When I test my document template using Document Generator, there is a huge chunk of code directly in the generated document where I used a tag to retrieve data.</p>	<p>You may have a <code>tc:data</code> tag that is trying to retrieve a data item of the data type "object." An object is not a valid type of data to retrieve in a document template. You may need to continue to navigate, for example, to the name attribute of the object.</p> <p>In <a href="#">Object Model: Read This First</a> or in the reference tables it points to, locate the object attribute and find its data type in the Data Type column. If it is of type "object," then you cannot retrieve it using a <code>tc:data</code> tag. Refer to the object table to which that object attribute links to find the data that you need to retrieve in the document template.</p>
<p>When I test my document template using Document Generator, I get an error similar to one of the following:</p> <p>[*Error in numberString*]</p> <p>[*Error in parent*]</p> <p>[*Error in name*]</p>	<p>Notice that all of these errors specify an object attribute. The Document Generator is not recognizing them because the first letter is not capitalized. Make sure that every time you specify an object attribute in a tag, the first letter is capitalized.</p>

<p>A document is not generated correctly, and other means of troubleshooting fail.</p>	<p>If the debug level is activated, error information is written to <a href="#">the debug log</a>. You may check this file for details on the problem.</p>
<p>The document template is generated successfully, but the letter generator hangs after selecting a document template.</p>	<p>At least one XML template is not well formed.</p> <ol style="list-style-type: none"> <li>1. Check for any un-escaped reserved characters (e.g. `&amp;`) in the XML template. If an un-escaped reserved character is found, please make the appropriate change in the XML template and test whether this resolves the issue.</li> <li>2. Check whether there are any begin/end tags missing from the XML template. For a migrated letter template, the missing begin/end tags are usually "filter@" or "loop@". Insert the missing begin/end tag mapping into the DM file and then use either Data Mapping Tool or EasyDocs Template Generator to re-generate the XML template.</li> </ol> <p>Here is the syntax and example for end-tag mapping:</p> <pre>&lt;MergeField name="filter@"&gt;&lt;operation&gt;filter&lt;/operation&gt; &lt;ownerPath&gt;&lt;MERGEFIELD begin tag name::~/&gt;ownerPath&gt; &lt;dataPath&gt;&lt;/dataPath&gt; &lt;conditionValue&gt; &lt;/conditionValue&gt; &lt;formatOption&gt;&lt;/formatOption&gt; &lt;/MergeField&gt; &lt;MergeField name="filter@"&gt;&lt;operation&gt;filter&lt;/operation&gt; &lt;ownerPath&gt;filter@Name0.2::~/&gt;ownerPath&gt; &lt;dataPath&gt;&lt;/dataPath&gt; &lt;conditionValue&gt; &lt;/conditionValue&gt;</pre>
<p>The XML template is generated successfully and the XML template is well formed, but launching Document Generator results in <code>java.lang.StringIndexOutOfBoundsException</code>.</p>	<p>Open the XML template and check whether element <code>&lt;BASE_TREEPOSITION&gt;</code> or <code>&lt;TREEPOSITION&gt;</code> exists. If either one of them exists in the XML, replace the corresponding <code>&lt;BASE_TREEPOSITION&gt;</code> or <code>&lt;TREEPOSITION&gt;</code> with the appropriate TeamConnect Object Definition unique code or category tree position in the XML template. If you replace the unique code / tree position in the DM file, please use either Data Mapping Tool or EasyDocs Template Generator to regenerate the XML template.</p> <p>If the XML template does not contain either <code>&lt;BASE_TREEPOSITION&gt;</code> or <code>&lt;TREEPOSITION&gt;</code>,</p>

please report this issue to TeamConnect support for further assistance.

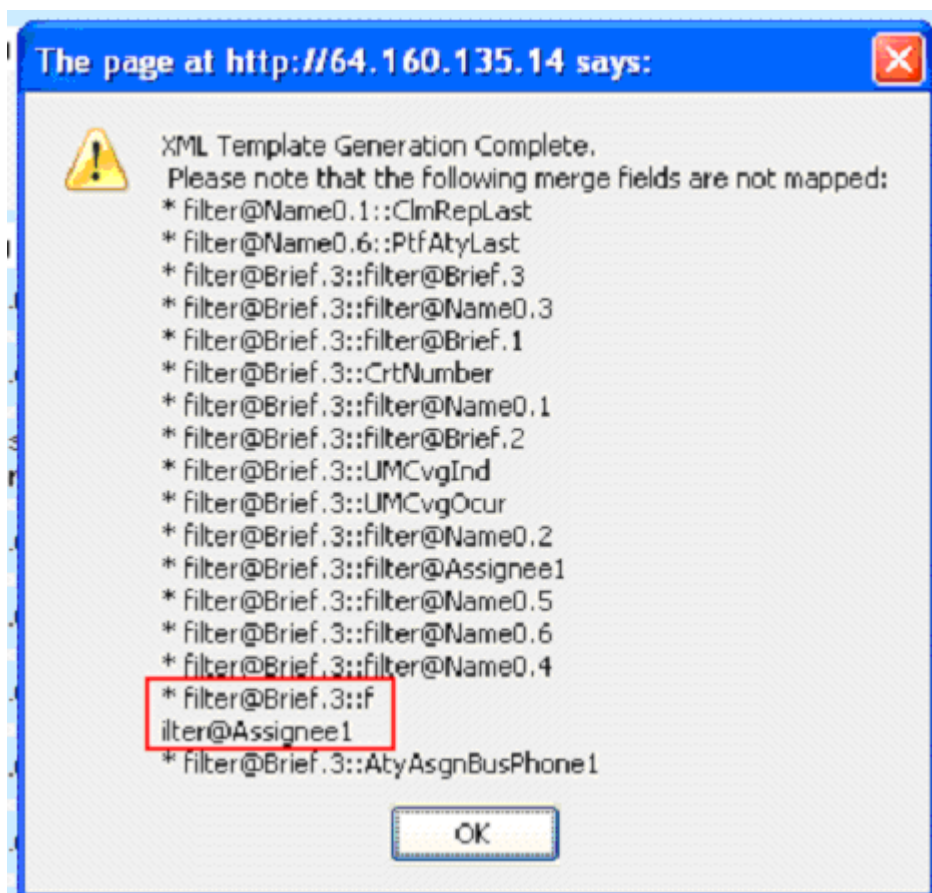
## Additional Errors and Workarounds

If a line break is encountered in an XML file, in the middle of a token, it may cause an error message to be displayed. To fix the problems you must edit the XML file, removing the incorrect line breaks.

If the token is a merge field name, as shown below:

```
1523 \par }\pard \ltrpar\ql
    \li0\ri0\keep\keepn\widctlpar\tqr\tx9180\wrapdefaut\aspalpha\aspnum\fauto\adjustright\rin0\lin0\itap0\pararsi
    {\field\*\fldinst {\rtch\fcsl \af0 \ltrch\fcso \lang1024\langfel024\noproof\insrsid5716630 MERGEFIELD f
1524 filter@Assignee1}\* MERGEFORMAT)}\fldrslt {\rtch\fcsl \af0 \ltrch\fcso \lang1024\langfel024\noproof\insrsid57
    \abfilter@Assignee1\hb}}\sectd \pszl\pgrstart\linex0\headery432\footery432\titlepg\sectdefaultcl\sectrsid1.
```

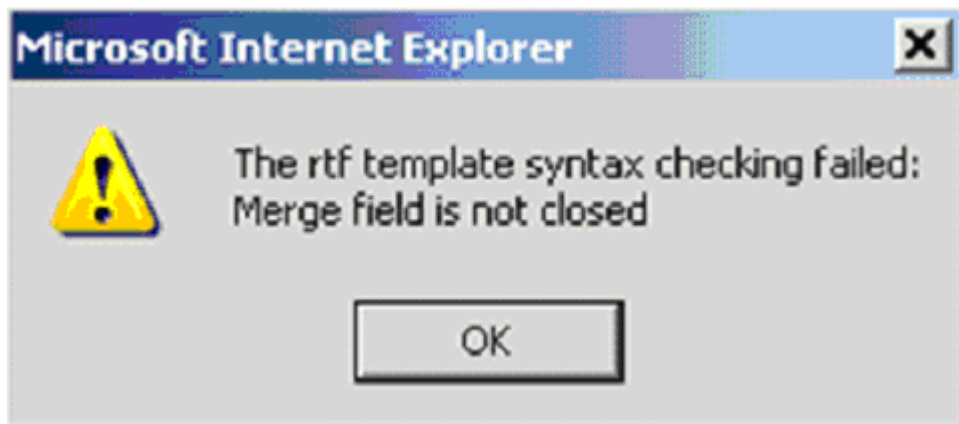
...then the resulting error message will look like this:



If the token containing the line break is a keyword instead, as shown below:

```
114 \af0\afs20\alang1025 \ltrch\fcso \fs24\lang1033\langfel033\cgrid\langmp1033\langfenp1033 {\field
    \ltrch\fcso \insrsid10383591 MERGEFI
115 ELD letterHead }}\fldrslt {\rtch\fcsl \af0 \ltrch\fcso
    .....
```

...then a different set of error messages will be displayed, as shown below:



### 3.1.23 Document Template Samples

Use the samples contained here as models for constructing tags in your document templates.

The following conventions are used in these samples:

- Bold** On the left of the sample, indicates what the code is doing or what data is being retrieved in that section of code.
- Within the sample, indicates static text that would be in the content of the document.
- Code Indicates tags that would be in the content of the document.
- In some cases, comments are also written in code.

#### 3.1.23.1 Nesting to Navigate

The following is an example of how to nest tags within system fields of data type 'Object' to retrieve the necessary information from the main object. In this example, the main object is claim, which is a custom object.

As you will see, to retrieve data from a field of type 'Object' you must specify the data that you want to retrieve by nesting the appropriate tags.

- Contact-Centric** `<!-- The contact-centric field is used to navigate to the contact record. It is not used to retrieve specific data,`

<b>Field</b>	<p>such as the insured contact's first and last name. Therefore, we nested additional tags to retrieve the first and last name of the insured. --&gt;</p> <p><b>Contact:</b> <code>&lt;tc:data select="Contact"&gt;&lt;tc:data select="FirstName" /&gt; &lt;tc:data select="Name" /&gt;&lt;/tc:data&gt;</code></p>
<b>Main Assignee</b>	<p>&lt;!-- By nesting, we are able to navigate to the contact card of the main assignee to get the main assignee's name and phone number. This example also shows how to use tc:loop to get the business phone number as opposed to the default phone number. --&gt;</p> <p><b>Main Assignee:</b> <code>&lt;tc:data select="MainAssignee"&gt;</code></p> <p><b>First Name:</b> <code>&lt;tc:data select="User"&gt;&lt;tc:data select="Contact"&gt;&lt;tc:data select="FirstName" /&gt;</code></p> <p><b>Last Name:</b> <code>&lt;tc:data select="Name" /&gt;</code></p> <p><b>Business Phone Numbers:</b> <code>&lt;tc:loop select="PhoneList" qualifier="Type_TreePosition=PHON_BUS1"&gt;&lt;tc:data select="PhoneString" /&gt;&lt;/tc:loop&gt;</code></p> <p><code>&lt;/tc:data&gt;&lt;/tc:data&gt;&lt;/tc:data&gt;</code></p>
<b>Created By</b>	<p><b>Created By:</b> <code>&lt;tc:data select="CreatedBy"&gt;&lt;tc:data select="Contact/FirstName" /&gt; &lt;tc:data select="Contact/Name" /&gt;&lt;/tc:data&gt;</code></p>
<b>Current Phase Type</b>	<p>&lt;!-- By itself, the object attribute CurrentPhaseType would not retrieve the name of the current phase. In order to do that, we needed to first navigate to WObjdPhaseType object to retrieve the actual name of the phase. The same is true for DefaultCategory. --&gt;</p> <p><b>Current Phase Type:</b> <code>&lt;tc:data select="CurrentPhaseType"&gt;&lt;tc:data select="Name" /&gt;&lt;/tc:data&gt;</code></p>
<b>Default Category</b>	<p><b>Default Category:</b> <code>&lt;tc:data select="DefaultCategory"&gt;&lt;tc:data select="Name" /&gt;&lt;/tc:data&gt;</code></p>
<b>Parent Policy</b>	<p>&lt;!-- To retrieve information about the parent policy, we used the object attribute Parent to navigate to the parent policy and then nested the appropriate tags to retrieve specific data about the parent policy. --&gt;</p> <p><b>Parent Information:</b> <code>&lt;tc:data select="Parent"&gt;</code></p> <p><b>Name:</b> <code>&lt;tc:data select="Name" /&gt;</code></p> <p><b>Number:</b> <code>&lt;tc:data select="NumberString" /&gt;</code></p> <p><b>Main Assignee Info:</b> <code>&lt;tc:data select="MainAssignee"&gt;&lt;tc:data select="User/Contact/FirstName" /&gt; &lt;tc:data select="User/Contact/Name" /&gt;&lt;/tc:data&gt;</code></p> <p><b>Assignees:</b> <code>&lt;tc:loop select="AssigneeList"&gt;&lt;tc:data select="User/Contact/FirstName" /&gt; &lt;tc:data select="User/Contact/Name" /&gt; &lt;tc:data select="Type/Name" /&gt;&lt;/tc:loop&gt;</code></p> <p><code>&lt;/tc:data&gt;</code></p>

### 3.1.23.2 Retrieving Data through Contact-Centric Fields

The following sample is going to retrieve data from a contact-centric system field found in a custom object definition.

The first tc:data tag navigates to the Contact object using the **contact** attribute of TProject. From there, the nested tags retrieve the following types of data about the contact.

- System fields
- Custom fields
- Sub-objects
- Related objects

Also, after navigating back to the main object (Claim) by closing the tc:data tag that navigated to the contact, the template retrieves data from a related object, Involved, using a filter defined in the header.

#### Creating Filter Screen for Related Objects

```
<?xml version="1.0" encoding="UTF-8"?>
<tc:document version="1.0" mime-type="application/msword" name="Contact-Centric Custom Fields"
xmlns:tc="http://www.w3.org/1999/XSL/Transform">

<!-- The header section identifies the filter screen
that will be displayed to the end-user. The filter
screen will display the Involved parties associated to
the Claim. The related object records in the filter
screen will be displayed in a drop-down list. -->
<tc:header>
<tc:filter name="SelectInvolvedParties"
searchCondition="Involved" label="Please select the
Involved Party" displayString="Contact/
FirstName;Contact/Name;DefaultCategory/Name"
displayFormat="* * - *" />
</tc:header>
<tc:content><tc:data select="Contact">
First Name: <tc:data select="FirstName" />
Last Name: <tc:data select="Name" />
```

#### Retrieving Data from Custom Fields

```
<!-- The following tags retrieve data from the custom
fields within the Contact record. These custom fields
are defined for the category Employee. -->
Hire Date: <tc:detail select="CONT_EMPL/HireDate" />
Salary: <tc:detail select="CONT_EMPL/Salary" />
Employee Type: <tc:detail select="CONT_EMPL/
EmployeeType" />
Comments: <tc:detail select="CONT_EMPL/Comments" />
Citizen: <tc:detail select="CONT_EMPL/Citizen"
true="Yes" false="No" />
Book: <tc:detail select="CONT_EMPL/Book"><tc:data
select="Name" /></tc:detail>
```

### Retrieving All Instances of the Sub-object Address

```
<!-- The following tags retrieve all of the addresses
added to the contact record. In this case, the tc:loop
tag is used because we want to retrieve all of the
addresses added to the contact record, not just a
specific type of address (which would require the
qualifier attribute) or one that the user selected
(which would require tc:filter to be used instead of
tc:loop). -->
```

```
Address Information <tc:loop select="AddressList">
```

```
Street: <tc:data select="Street" />
```

```
Country: <tc:data select="CountryItem/Name" />
```

```
</tc:loop>
```

### Automatically Retrieving All Instances of the Related Object History

```
<!-- The following tags retrieve all of the history
records created for the contact record. In this case,
the tc:search tag is used because we want to retrieve
all history records that were created for the contact
record, not just a specific history record (which would
require the qualifier attribute or tc:conditional) or
one that the user selected (which would require
tc:filter to be used instead of tc:search). -->
```

```
History Information <tc:search link="History">
```

```
Archived On: <tc:data select="ArchivedOn" />
```

```
Default Category: <tc:data select="DefaultCategory/Name"
/>
```

```
Notes: <tc:data select="Text" />
```

```
</tc:search></tc:data>
```

### Retrieving the User-Selected Related Object Involved

```
<!-- The following tags retrieve data pertaining to the
involved party that the user selected while generating
the letter. In this case, the tc:filter tag is used
because we want to retrieve the data pertaining to the
involved party that the user selected. -->
```

```
Involved Information <tc:filter
select="SelectInvolvedParties"><tc:data
select="Contact">
```

```
First Name: <tc:data select="FirstName" />
```

```
Last Name: <tc:data select="Name" /> </tc:data>
```

### Automatically Retrieving All Instances of the Sub-Object Category

```
<!-- The following tags retrieve the categories added
to the involved party record. In this case, the tc:loop
tag is used because we want to retrieve all of the
categories added to the user selected record. -->
```

```
Role: <tc:loop select="DetailList"><tc:data
select="Category/Name" />
```

```
</tc:loop></tc:filter></tc:content></tc:document>
```



### 3.1.23.3 Automatically Retrieving Data from Related Objects

The following sample shows the different methods that can be used to automatically retrieve data from object records that are related to the main object (Claim). These samples retrieve data from:

- System fields from the main object (Claim) and its related object (Involved Parties)
- Sub-objects of the main object (Claim) and its related object (Involved Parties)

**Claim Name:** `<tc:data select="Name" />`

**Claim Number:** `<tc:data select="NumberString" />`

**Insured:** `<tc:data select="Contact"><tc:data select="FirstName" />  
<tc:data select="Name" /> </tc:data>`

#### Retrieving All Instances of a Sub-object

`<!-- The following tags retrieve data from all of the individuals assigned to the claim record. In this case, the tc:loop tag is used because we want to retrieve all of the assignees added to the claim record, not just a specific type of assignee (which would require the qualifier attribute) or one that the user selected (which would require tc:filter to be used instead of tc:loop). -->`

**Assignee Information** `<tc:loop select="AssigneeList"> <tc:data select="User"> <tc:data select="Contact">`

**First and Last Name:** `<tc:data select="FirstName" /> <tc:data select="Name" />`

**Phone Number:** `<tc:data select="DefaultPhone/PhoneString" />`

**Job Title:** `<tc:data select="Title" />`

`</tc:data></tc:data></tc:loop>`

#### Retrieving Specific Type of Related Object with tc:conditional

`<!-- The following is an example of how tc:conditional is used as a qualifier to retrieve the involved parties who are claimants. This will retrieve all of the involved records that have the role 'Claimant' added.-->`

**Claimant Information** `<tc:search link="Involved"><tc:conditional test="Detail[CLIN_CLMT]"><tc:data select="Contact">`

**First Name:** `<tc:data select="FirstName" />`

**Last Name:** `<tc:data select="Name" />`

**Social Security #:** `<tc:data select="SsOrTaxNumberString" />`

**Driver's License:** `<tc:data select="DriverLicense" />`

**Company Name:** `<tc:data select="Company"><tc:data select="Name" />`

**Company Address:** `<tc:data select="DefaultAddress"><tc:data select="Street" /> <tc:data select="City" /> <tc:data select="State" /> <tc:data select="PostalCode" /></tc:data></tc:data>`

**Business Phone:** `<tc:loop select="PhoneList" qualifier="Type_TreePosition=PHON_BUS1"><tc:data select="PhoneString" /> </tc:loop>`

**Business Email:** `<tc:loop select="EmailList" qualifier="Type_TreePosition=MAIL_BUS1"><tc:data select="EmailString" /></tc:loop>`

`</tc:data></tc:conditional></tc:search>`

**Retrieving Specific Type of Related Object Using qualifier**

```
<!-- The following is an example of how the qualifier attribute is used to retrieve involved records that have the default role `Injured Passenger' selected. -->
```

**Injured Information** <tc:search link="Involved"

```
qualifier="DefaultCategory_TreePosition=CLIN_CLMT_INPA"><tc:data select="Contact">
```

**First Name:** <tc:data select="FirstName" />

**Last Name:** <tc:data select="Name" />

**Social Security #:** <tc:data select="SsOrTaxNumberString" />

**Driver's License:** <tc:data select="DriverLicense" />

**Company Name:** <tc:data select="Company"><tc:data select="Name" />

**Company Address:** <tc:data select="DefaultAddress"><tc:data select="Street" /> <tc:data select="City" /> <tc:data select="State" /> <tc:data select="PostalCode" /></tc:data></tc:data>

**Retrieving Specific Types of Sub-objects**

```
<!-- By using the qualifier attribute of tc:loop, we will retrieve the contact record's business phone number(s) and email address(es). -->
```

**Business Phone:** <tc:loop select="PhoneList" qualifier="Type\_TreePosition=PHON\_BUS1"><tc:data select="PhoneString" /> </tc:loop>

**Business Email:** <tc:loop select="EmailList" qualifier="Type\_TreePosition=MAIL\_BUS1"><tc:data select="EmailString" /></tc:loop>

```
</tc:data></tc:search>
```

### 3.1.23.4 Retrieving Data through Custom Fields of Type Custom Object

The following sample is used to retrieve data from the parent record of the Claim identified in a custom field of type Custom Object (MasterClaims).

```
<tc:detail select="CLAM/MasterClaims">
```

```
<tc:data select="Parent">
```

**Master Claim's Parent (Policy) Information**

**Number:** <tc:data select="NumberString" />

**Name:** <tc:data select="Name" />

**Categories List:** <tc:loop select="DetailList">

```
<tc:data select="Category/Name" />
```

```
</tc:loop>
```

**Default Category:** <tc:data select="DefaultCategory/Name" />

**Policy Effective Date:** <tc:detail select="PLCY/EffectiveDate" />

**Good Driver?:** <tc:detail select="PLCY/GoodDriver" />

**Policy Postal Code:** <tc:detail select="PLCY/PolicyZip" />

**Policy State:** <tc:detail select="PLCY/PolicyState" />

```

Comments: <tc:detail select="PLCY/Comments" />
</tc:data> </tc:detail>

```

### 3.1.23.5 Retrieving Data through Custom Fields of Type Involved

The following sample is used to retrieve data from the contact record linked to the project through the Involved custom field.

```

Additional Driver Info <tc:detail select="PLCY/AdditionalDriver">
<tc:data select="Contact">
First Name: <tc:data select="FirstName" />
Last Name: <tc:data select="Name" />
Home Address Information <tc:loop select="AddressList"
qualifier="Type_TreePosition=ADDR_BUS1"><tc:data select="Street" /><tc:data
select="City" /><tc:data select="State" /><tc:data select="PostalCode" /></
tc:loop>
Social Security #: <tc:data select="SsOrTaxNumberString" />
Driver's License: <tc:data select="DriverLicense" />
</tc:data></tc:detail>

```

### 3.1.23.6 Retrieving Data from Child Objects

There are two tags used to retrieve data from child objects:

- `tc:search`, used to automatically retrieve data from child object records.
- `tc:filter`, used to define a screen where the end user selects the child object record(s) to be included in the document.

#### Automatically

The following sample is used to retrieve data from all child claims of a parent policy. In addition, it retrieves the vehicles damaged in the accident of each claim. Vehicle is an embedded object of Claim. Notice that `tc:search` is nested within another `tc:search` tag.

#### Policy Information

```

Policy Number: <tc:data select="NumberString" />
Insured: <tc:data select="Contact"><tc:data
select="FirstName" /> <tc:data select="Name" /></tc:data>

```

#### Retrieving Specific Type of Child Records with `tc:conditional`

```

<!-- The tc:conditional tag is used to retrieve child
claim records of type `Auto'. -->
<tc:search link="Project" isForChild="yes"
qualifier="Application_ProjObjectDefinition_UniqueCode=CLA
M"> <tc:conditional test="Detail[CLAM_AUTO]">

```

#### Claim Information (Child of Policy)

**Claim Name:** <tc:data select="Name" />

**Claim Number:** <tc:data select="NumberString" />

**Assignees:** <tc:loop select="AssigneeList" qualifier="IsActive=1"><tc:data select="User"> <tc:data select="Contact">

### Name, Date Assigned

```
<tc:data select="FirstName" /> <tc:data select="Name" /></tc:
data></tc:data>, <tc:data select="AssignedOn" />
</tc:loop>
```

```
<tc:data select="MainAssignee">
```

**Main Assignee:** <tc:data select="User"><tc:data select="Contact"><tc:data select="FirstName" /> <tc:data select="Name" /></tc:data></tc:data></tc:data>

<!-- Retrieving vehicles damaged in each claim -->

```
<tc:search link="Project" isForChild="yes"
qualifier="Application_ProjObjectDefinition_UniqueCode=VEHI">
```

### Vehicle Information <!-- Embedded Object of Claim -->

The following vehicles were involved in the this claim:

**VIN:** <tc:detail select="VEHI/VIN" />

**Make:** <tc:detail select="VEHI/MakeModel" />

**Year:** <tc:detail select="VEHI/Year" />

**Damage:** <tc:detail select="VEHI/Damage" />

```
</tc:search></tc:conditional></tc:search>
```

## User Selected

This sample utilizes the tc:filter tag to retrieve data from the user selected child record(s). This sample retrieves data from:

- System fields from policy record.
- User-selected assignee(s) of the policy record.
- User-selected claim record.
- User-selected assignee(s) of the selected claim record.
- User-selected involved contact(s) of the selected claim record.

```
Header <?xml version="1.0" encoding="UTF-8"?>
<tc:document version="1.0" mime-type="application/msword" name="Policy,
Claim and Vehicle Information - Filter" xmlns:tc="http://
www.w3.org/1999/XSL/Transform">
<tc:header>
<tc:filter name="SelectedPolicyAssignee" test="AssigneeList"
label="Please select the assignee to contact in regards to the Policy"
```

```

displayString="User/Contact/FirstName;User/Contact/Name;Type/Name"
displayFormat="* * - *" />
<tc:filter name="Claims" searchCondition="Project"
qualifier="Application_ProjObjectDefinition_UniqueCode=CLAM"
label="Select the appropriate claim" displayString="Name;NumberString"
displayFormat="* - *" usedBy="2" />
<tc:filter name="SelectedClaimAssignee" test="AssigneeList"
label="Please select the assignee to contact in regards to the Claim"
displayString="User/Contact/FirstName;User/Contact/Name;Type/Name"
displayFormat="* * - *" multi="yes" objectFrom="Claims"/>
<tc:filter name="SelectedInvolvedParties" searchCondition="Involved"
label="Please select the parties involved in this Claim"
displayString="Contact/FirstName;Contact/Name;DefaultCategory/Name"
displayFormat="* * - *" multi="yes" objectFrom="Claims"/>
</tc:header>

```

**Content**

```
<tc:content>
```

```
Policy Number: <tc:data select="NumberString" />
```

```
Insured Information <tc:data select="Contact">
```

```
First Name: <tc:data select="FirstName" />
```

```
Last Name: <tc:data select="Name" /> </tc:data>
```

```

<!-- The following tags retrieve data from the Assignee that the user
selected. In this case, the tc:filter tag is used because we want to
retrieve data from the sub-object that the user selected, not just all
of the users assigned to the policy record (which would require the
tc:loop) or specific type of assignee (which would require the
qualifier attribute). -->

```

```

Assignee Information <tc:filter
select="SelectedPolicyAssignee"><tc:data select="User"><tc:data
select="Contact">

```

```
First Name: <tc:data select="FirstName" />
```

```
Last Name: <tc:data select="Name" /></tc:data></tc:data>
```

```
Role: <tc:data select="Type/Name" /></tc:filter>
```

```

<!-- The following tags retrieve data from the user selected claim
record(s), which is a child object of Policy. In addition, we nested
other tags to retrieve data pertaining to the related objects and sub-
objects of Claim. -->

```

```
<tc:filter select="Claims">
```

```
Claim Number: <tc:data select="NumberString" />
```

```
Assignees <tc:filter select="SelectedClaimAssignee"><tc:data
select="User"><tc:data select="Contact">
```

```
First Name: <tc:data select="FirstName" />
```

```
Last Name: <tc:data select="Name" /></tc:data></tc:data>
```

```
Assigned On: <tc:data select="AssignedOn" />
```

```
Role: <tc:data select="Type/Name" /></tc:filter>
```

```

Main Assignee: <tc:data select="MainAssignee"><tc:data
select="User"><tc:data select="Contact"><tc:data select="FirstName" />
<tc:data select="Name" /></tc:data></tc:data></tc:data>

```

```

<!-- The following tags retrieve data from the involved contact
selected by the user. -->
Involved Parties <tc:filter select="SelectedInvolvedParties"><tc:data
select="Contact">
First Name: <tc:data select="FirstName" />

Last Name: <tc:data select="Name" /></tc:data>
Roles: <tc:data select="DefaultCategory/Name" /></tc:filter></
tc:content></tc:document>

```

### 3.1.23.7 Using Qualifiers to Filter Sub-objects

This sample shows how you can use the qualifier attribute of tc:loop to refine the data that you want to retrieve. It also shows how you can include different types of system fields as a qualifier.

This template is retrieving all of the active assignees within the selected claim. For each active assignee, it is retrieving:

- The tasks that they perform where the rate is equal to \$100.00.
- Their skill level as an adjuster.
- Their address within the city of Los Angeles.
- The contacts that have a vendor relationship with them.

```

Boolean (Check Box) <tc:loop select="AssigneeList" qualifier="IsActive=1">
<tc:data select="User"> <tc:data select="Contact">
Name: <tc:data select="FirstName" /> <tc:data select="Name" /></
tc:data></tc:data>
Role: <tc:data select="Type/Name" />
Date Assigned: <tc:data select="AssignedOn" />
<tc:data select="User"> <tc:data select="Contact">

Number <tc:loop select="RateList" qualifier="RateAmount=100">Rate
Information
Task Type <tc:data select="TaskCategory/Name" />
Rate <tc:data select="RateAmount" />
Effective From <tc:data select="EffectiveFrom" />
Effective To <tc:data select="EffectiveTo" />
</tc:loop>

List (Lookup Table) <tc:loop select="SkillList"
qualifier="Type_TreePosition=SKIL_ADST">Skill Information
Skill <tc:data select="Type/Name" />
Level of Expertise <tc:data select="SkillLevel" />
</tc:loop>

```

**Text**

```
<tc:loop select="AddressList" qualifier="City=Los Angeles">Address
Information
Type <tc:data select="Type/Name" />
Street <tc:data select="Street" />
City <tc:data select="City" />
State <tc:data select="State" />
Postal Code <tc:data select="PostalCode" />
Country <tc:data select="CountryItem/Name" />
Current On <tc:data select="CurrentOn" />
</tc:loop>
```

**Object (Another Record)**

```
<tc:loop select="RightRelationList"
qualifier="RightContact_DefaultCategory_TreePosition=CONT_VNDR">Conta
ct Relations
<tc:data select="RightContact">
Name: <tc:data select="FirstName" /> <tc:data select="Name" /></
tc:data>
Relationship: <tc:data select="Type/Name" />
<tc:data select="RightContact">
Business Address: <tc:loop select="AddressList"
qualifier="Type_TreePosition=ADDR_BUS1">
Street <tc:data select="Street" />
City <tc:data select="City" />
State <tc:data select="State" />
Postal Code <tc:data select="PostalCode" />
</tc:loop>
</tc:data>
</tc:loop>
```

### 3.1.23.8 Using tc:conditional

There are many different ways to use `tc:conditional` to retrieve data, include static text, and/or import text from other files.

#### 3.1.23.8.1 Comparing Custom Fields

The following are examples of how `tc:conditional` can be used to compare a custom field value with the value specified in the tag to include static text and retrieve data.

**Date**

```
<tc:conditional condition="greater" compare="Detail[CLAM]/
DetailValue[AwardDate],2003-01-01">Date Awarded: <tc:detail
select="CLAM/AwardDate" /></tc:conditional>
```

**Number**

```
<tc:conditional condition="less" compare="Detail[CLAM]/
DetailValue[ClaimEstimate],10000">Claim Estimate: <tc:detail
select="CLAM/ClaimEstimate" /></tc:conditional>
```

<b>List</b>	<pre>&lt;tc:conditional compare="Detail[CLAM]/ DetailValue[Fatality],Yes"&gt;Unfortunately there was a fatality.&lt;/ tc:conditional&gt;</pre>
<b>Check Box</b>	<pre>&lt;tc:conditional compare="Detail[CLAM]/ DetailValue[PoliceInvolved],true"&gt;Police Involved: &lt;tc:detail select="CLAM/PoliceOfficer" /&gt;&lt;/tc:conditional&gt;</pre>
<b>Text</b>	<pre>&lt;tc:conditional compare="Detail[CLAM]/DetailValue[LossLocation],Near Residence"&gt;Loss Location: &lt;tc:detail select="CLAM/LossLocation" /&gt;&lt;/ tc:conditional&gt;</pre>

### 3.1.23.8.2 Comparing Sub-objects

The following example will check to see if an assignee with the role 'Supervisor' is added to the claim. If so, it will include the information nested within the `tc:conditional` tag. If not, nothing will be included.

```
<tc:conditional compare="System[Assignee],Adjuster">Please contact your claim
adjuster <tc:loop select="AssigneeList"
qualifier="Type_TreePosition=CLAM_ADST"><tc:data select="User"><tc:data
select="Contact"><tc:data select="FirstName" /> <tc:data select="Name" /> at
<tc:data select="DefaultPhone/PhoneString" /></tc:data></tc:data>.</tc:loop></
tc:conditional>
```

The following example will loop through all of the assignees within the claim record to see if any of the assignees have the skill 'Adjuster' in their contact card. If so, it will include the information nested within the `tc:conditional` tag. If not, nothing will be included.

```
<tc:loop select="AssigneeList" qualifier="Type_TreePosition=PHON_BUS1"><tc:data
select="User"><tc:data select="Contact"><tc:conditional
compare="System[Skill],Adjuster">Please contact <tc:data select="FirstName" />
<tc:data select="Name" /> at <tc:loop select="PhoneList"
qualifier="PHON_BUS1"><tc:data select="PhoneString" /></tc:loop> if you have any
questions regarding your claim.</tc:conditional></tc:data></tc:data></tc:loop>
```

### 3.1.23.8.3 Testing for Existing Data

The following samples show how `tc:conditional` is used to see if data exists for system fields or sub-objects. If data exists, the nested information is included. If not, nothing is included.

#### Testing System Fields

The following example will test to see if the claim is closed. If so, it will include the static text and the date that the claim was closed.

```
<tc:conditional test="ClosedOn">Your claim was closed on <tc:data select="ClosedOn" /
>.</tc:conditional>
```

The following example will test to see if the claim is linked to a policy. If so, it will include the static text and the policy number.



```
<tc:conditional test="Parent">This claim is in regards to Policy Number <tc:data
select="NumberString" />.</tc:conditional>
```

## Testing Sub-objects

The following example will test to see if any assignees are assigned to the claim. If so, it will include the static text and loop to find all the assignees with the role `Adjuster` and bring in the nested information.

```
<tc:conditional test="AssigneeList">Please contact your claim adjuster <tc:loop
select="AssigneeList" qualifier="Type_TreePosition=CLAM_ADST"><tc:data
select="User"><tc:data select="Contact"><tc:data select="FirstName" /> <tc:data
select="Name" /> at <tc:data select="DefaultPhone/PhoneString" /></tc:data></
tc:data></tc:loop>if you have any questions.</tc:conditional>
```

The following example will test to see if the claim includes the category `Auto`. If so, it will include the static text in the letter and retrieve data from the specified system and custom fields.

```
<tc:conditional test="Detail[CLAM_AUTO]">We have the following information about your
auto accident.
```

**Date of Accident:** <tc:detail select="CLAM\_AUTO/AccidentDate" />

**Loss Location:** <tc:detail select="CLAM/LossLocation" />

**Claimant:** <tc:detail select="CLAM\_AUTO/Claimant" />

**Policy Number:** <tc:data select="Parent"><tc:data select="NumberString" /></
tc:data></tc:conditional>

### 3.1.23.8.4 Including Information When Certain Conditions Are Met

The following sample shows how `tc:conditional` can be used to include the appropriate pronoun based on the gender of an individual. In this sample, the pronoun he/she or his/her is based on whether the claimant is male or female.

```
<tc:document version="1.0" mime-type="application/msword"
name="Conditional" xmlns:tc="http://www.w3.org/1999/XSL/Transform">
```

**Header** <tc:header>

```
<tc:filter name="UserSelectedClaimant" searchCondition="Involved"
displayString="Contact/FirstName;Contact/Name;DefaultCategory/Name"
displayFormat="* * - *" label="Please select the Claimant"/>
</tc:header>
```

**Content** <tc:content>

```
<tc:data select="Contact"><tc:data select="FirstName" /> <tc:data
select="Name" />
```

```
<tc:loop select="AddressList"><tc:data select="Street" />
```

```
<tc:data select="City" />, <tc:data select="State" /> <tc:data
select="PostalCode" /></tc:loop>
```

```
Dear <tc:data select="Prefix" /> <tc:data select="FirstName" />
<tc:data select="Name" /></tc:data>
```

**We were notified of your recent auto accident with** <tc:filter select="UserSelectedClaimant"><tc:data select="Contact"><tc:data select="FirstName" /> <tc:data select="Name" />.

<tc:conditional compare="Detail[CONT]/DetailValue[Gender],Male">He</tc:conditional><tc:conditional compare="Detail[CONT]/DetailValue[Gender],Female">**She**</tc:conditional></tc:data></tc:filter>

**has reported that the accident occurred on** <tc:detail select="CLAM\_AUTO/AccidentDate" />.

**Based on** <tc:filter select="UserSelectedClaimant"><tc:data select="Contact"><tc:conditional compare="Detail[CONT]/DetailValue[Gender],Male">his</tc:conditional><tc:conditional compare="Detail[CONT]/DetailValue[Gender],Female">her</tc:conditional></tc:data></tc:filter> **report the following people were involved in the auto accident.**

<tc:search link="Involved"><tc:data select="Contact"><tc:data select="FirstName" /> <tc:data select="Name" /></tc:data> - <tc:data select="DefaultCategory/Name" />

</tc:filter>

**Sincerely,**

<tc:loop select="AssigneeList" qualifier="Type\_TreePosition=CLAM\_ADST"><tc:data select="User"><tc:data select="Contact"><tc:data select="FirstName" /> <tc:data select="Name" /></tc:data></tc:data></tc:loop>

</tc:document>

### 3.1.24 Using Tag Attributes

This appendix provides examples of XML tag attributes in TeamConnect that are commonly used to format fields and their labels on the screen. The attributes are grouped by the type of custom field each attribute affects. For details on the field and label tags, [Block Tags](#).

#### Field Tags

The field tags that can be formatted with the help of attributes include:

**Form** <TCFIELD NAME="FieldName" attributes />

**Block** <tc:field name="FieldName" category="TreePosition" attributes />

<tc:wizardParameter name="ParameterName" attributes />

To include an attribute in one of these tags, replace attributes with the attributes and their values. For example, to restrict the length of a field (MomMaidenName) to 20 characters, you would use the maxlength attribute:

<TCFIELD NAME="MomMaidenName" maxlength="20" />

#### Label Tags

Unlike fields that can have a variety of formatting attributes, field labels may have only a the formatting attributes listed in [Field Label Tag Attributes](#). The label tags are:

```
Form    <TCLABEL NAME="FieldName" attributes />

Block   <tc:label name="FieldName" category="TreePosition" attributes />

        <tc:wizardParameterLabel name="ParameterName" attributes />
```

To include an attribute in one of these tags, replace attributes with the attributes and their values. For example, if you do not want the settings defined for required fields to be applied to a field (MomMaidenName), you would use the `notUseRequiredFieldStyle` attribute, as in the following example:

```
<TCLABEL NAME="MomMaidenName" notUseRequiredFieldStyle="true" />
```

### 3.1.24.1 Field Label Tag Attributes

You can use the following attributes with custom field and parameter labels:

- [notUseLabelStyle](#)
- [notUseRequiredFieldStyle](#)
- [irequired](#)
- [style](#)

#### **notUseLabelStyle**

Set this attribute to "true" if you do not want the **Label Text** style defined in the system settings or [user preferences](#) for both Edit and Read-only mode to be applied to a custom field or parameter label.

You can then use a style sheet or the `style` attribute to define the font color and size of the label. If you do not use a style sheet or the `style` tag, the field label is set to the **General Text** settings defined in system settings or user preferences.

```
Form    <TCLABEL NAME="LastReviewDate" notUseLabelStyle="true"/>

Block   <tc:label name="LastReviewDate" category="EMPL" notUseLabelStyle="true" /
        >

        <tc:wizardParameterLabel name="LastReviewDate" notUseLabelStyle="true" />
```

#### **notUseRequiredFieldStyle**

Set this attribute to "true" if you do not want the **Required Field** style defined in system settings or user preferences to be applied to a label of a required custom field or parameter.

You can then use your style sheet or the `style` tag to define the font color and size of the label. If you do not use a style sheet or the `Style` tag, the field label is set to the **General Text** settings defined in system settings or user preferences.

**Form**     <TCLABEL NAME="LastReviewDate" notUseRequiredFieldStyle="true" />

**Block**     <tc:label name="LastReviewDate" category="EMPL"  
notUseRequiredFieldStyle="true" />

          <tc:wizardParameterLabel name="LastReviewDate"  
notUseRequiredFieldStyle="true" />

### isFieldRequired

Set this attribute to "true" if you want to make a Label Only field required, for example, when it is associated with another required field. you can also use this attribute to make the label of a custom field required when this field is made required through a rule or its attribute is included in the unique ID pattern for custom object records.

**Form**     <TCLABEL NAME="LastReviewDate" isFieldRequired="true" />

**Block**     <tc:label name="LastReviewDate" category="EMPL" isFieldRequired="true" />

          <tc:wizardParameterLabel name="LastReviewDate" isFieldRequired="true" />

For details on system settings, see [Working with Admin Settings](#).

**Note:** To learn more about the style sheets, refer to an HTML or XML reference guide.

### 3.1.24.2 Tag Attributes for All Field Types

#### forceNotEditable

Set this attribute to "true" to display custom fields as Read-only text. For example:

**Form**     <TCFIELD NAME="LastReviewDate" forceNotEditable="true"/>

**Block**     <tc:field name="LastReviewDate" category="EMPL"  
forceNotEditable="true" />

          <tc:wizardParameter name="LastReviewDate" forceNotEditable="true" />

#### style

The `style` attribute enables you to apply specific properties such as color, font, text alignment to a field. The `style` attribute also applies to the field label tags (see [Label Tags](#)).

#### Points to remember:

- Properties defined in the `style` attribute override those defined in the system settings, user preferences, or by the `style` tag.

- You can apply all style properties available in cascading style sheets.
- When defining the `style` attribute you must adhere to the following requirements:
  - Include the attribute value within braces (`{}`).
  - Separate each style with a semicolon (`;`).
  - Separate each property and its value with a colon (`:`).
- The style applies to both Edit and Read-only modes.

For example:

**Form**     `<TCFIELD NAME="LastReviewDate" style="{text-align:right;font-style:italic}" />`

**Block**     `<tc:field name="LastReviewDate" category="EMPL" style="{text-align:right;font-style:italic}" />`

`<tc:wizardParameter name="LastReviewDate" style="{text-align:right;font-style:italic}" />`

## tabIndex

By default, the tabbing order is set to tab from left to right, top to bottom. You can, however, customize the tabbing order to meet the needs of your users by using the `tabIndex` attribute with each field on the screen. Specify the order for each field, beginning with 1.

### Points to remember:

- If this attribute is used, it must be added to every field tag in your form or custom block.
- Fields must not have duplicate `tabIndex` values.
- Applying `tabIndex` to the following field types affects the way the user navigates to their corresponding buttons.
  - *Custom Object* and *Involved* field types, which have **Find** and **Open** buttons.
  - *Date* field type, which has a **Calendar** button.

The user is unable to navigate to these buttons by using the TAB key. He or she is required to use the mouse.

- Applying `tabIndex` to a Date field that has the Time field displayed, affects the way the user navigates to the time field. The user is unable to navigate to the time field by using the TAB key. He or she is required to use the mouse.
- The **Detail Forms** system block or custom block that includes the `tabIndex` attribute must be added as a separate tab within an object view.

This is because the system blocks do not have the `tabIndex` attribute. A screen that includes multiple blocks where they contain this attribute and others do not, prevents the user from using the TAB key through the entire screen.

- You can include multiple custom blocks to a tab within an object view, if the tab order is sequential and adheres to all of the requirements noted above.

For example:

**Form**     `<TCFIELD NAME="LastReviewDate" tabIndex="1" />`

**Block**     `<tc:field name="applicationEntity.LastReviewDate" category="EMPL"
tabIndex="1" />`

`<tc:wizardParameter name="LastReviewDate" tabIndex="1"/>`

### 3.1.24.3 Number Field Tag Attributes

#### format

Controls the format of the number field to either dollar, decimal or percent. The following values can be entered:

- DOLLAR**—Sets the format of the number field to include the appropriate currency sign, a decimal point, and commas. For example, 42000 is formatted to \$42,000.00 after the user saves the record.

**Form**     `<TCFIELD NAME="AnnualSalary" format="DOLLAR" />`

**Block**     `<tc:field name="AnnualSalary" category="EMPL" format="DOLLAR" />`

`<tc:wizardParameter name="AnnualSalary" format="DOLLAR" />`

This attribute displays the symbol of the system default currency in all object records except contacts, where it displays the symbol of the currency selected for that contact.

When used for custom fields of type Number in invoices that are entered in the original currencies different from the system default currency, this attribute will not reflect the currency symbol of the invoice.

- DECIMAL**—Sets the format of the number field to include a decimal point and commas. The number of zeros after the decimal point depends on the values of two other attributes:
  - `minimumFractionDigits` is used in cases where you wish to show digits to the right of the decimal point even when the rightmost decimal place(s) contain zero.
  - `maximumFractionDigits` is used in cases where you wish to store no more than a specified number of decimal places. If the user types more than the specified number of decimals, the result is rounded to the specified number of decimals.

If either of these attributes is not specified, the attribute's default number of decimal places is 2.

The value of `maximumFractionDigits` must not be specified as less than the value of `minimumFractionDigits`.

**Form**     `<TCFIELD NAME="Salary" format="DECIMAL" />`

**Block** `<tc:field category="MATT" name="Rate" format="DECIMAL" minimumFractionDigits="3" maximumFractionDigits="4" />`

For example, if the user types 42000 into field "Salary", above, the user input 42000 is formatted to 42,000.00 (default 2 decimal places) after the user saves the record.

An additional example: In the "Rate" field shown above, if the user types 37.1, it is formatted at save time to 37.100 (the minimum of three decimals applies in this case.) If the user types 37.10046, the field is formatted to 37.1005 (the maximum of four decimals applies in this case, and rounding is used.)

- **NONE**—Sets the format of the number field to use no formatting, such as a decimal point and commas. For example, if a number field is set to NONE, the user input 1501082006 is not formatted with commas and appears exactly as it is entered.

**Form** `<TCFIELD NAME="VoucherNumber" format="NONE" />`

**Block** `<tc:field name="VoucherNumber" category="VNUM" format="NONE" />`

`<tc:wizardParameter name="VoucherNumber" format="NONE"/>`

- **PERCENT**—Sets the format of the number field to round the decimal to the nearest integer and to include the percent sign. For example, if a number field is set to PERCENT the user input 45.75 is formatted to 46% after the user saves the record.

**Form** `<TCFIELD NAME="RaisePercent" format="PERCENT" />`

**Block** `<tc:field name="RaisePercent" category="EMPL" format="PERCENT" />`

`<tc:wizardParameter name="RaisePercent" format="PERCENT" />`

## size

Controls the width of number fields by specifying the number of numeric characters for the field. For example:

**Form** `<TCFIELD NAME="Salary" size="10" />`

**Block** `<tc:field name="Salary" category="EMPL" size="10" />`

`<tc:wizardParameter name="Salary" size="10" />`

## maxlength

Limits the maximum number of characters a user can enter in number fields. Set the attribute value to a number to represent the maximum number of numeric characters allowed. For example:

**Form** `<TCFIELD NAME="Salary" size="10" maxlength="10" format="DOLLAR" />`

**Block** `<tc:field name="Salary" category="EMPL" size="10" maxlength="10" format="DOLLAR" />`

`<tc:wizardParameter name="Salary" size="10" maxlength="10" format="DOLLAR" />`

### **onFocus="javascript:blur();"**

Prevents users from entering data into the number fields. This attribute is used with business rules and JavaScript. For example:

**Form** `<TCFIELD NAME="Percent" format="PERCENT" onFocus="javascript:blur();" />`

**Block** `<tc:field name="Percent" category="EMPL" format="PERCENT" onFocus="javascript:blur();" />`

`<tc:wizardParameter name="Percent" format="PERCENT" onFocus="javascript:blur();" />`

The same rules apply to including multiple attributes for a tag. For example, the custom field Salary can have the attributes `format`, `size`, and `maxlength`. See the example in [maxlength](#).

#### **3.1.24.4 Text Field Tag Attributes**

##### **size**

Sets the width for text fields. Set the attribute value to a number to represent the width of the field in characters. For example:

**Form** `<TCFIELD NAME="LastEmployer" size="50" />`

**Block** `<tc:field name="LastEmployer" category="EMPL" size="50" />`

`<tc:wizardParameter name="LastEmployer" size="50" />`

##### **maxlength**

Sets the maximum number of characters for text fields. Set the attribute value to a number to represent the maximum number of alphanumeric characters allowed. For example:

**Form** `<TCFIELD NAME="Age" size="3" maxlength="3" />`

**Block** `<tc:field name="Age" category="EMPL" size="3" maxlength="3" />`

`<tc:wizardParameter name="Age" size="3" maxlength="3"/>`



### 3.1.24.5 Memo Text Field Tag Attributes

#### rows

Sets the height of memo text fields by specifying the number of rows for the memo text field. For example:

Form        `<TCFIELD NAME="Comments" rows="10" />`

Block       `<tc:field name="Comments" category="EMPL" rows="10" />`  
`<tc:wizardParameter name="Comments" rows="10" />`

#### cols

Sets the width of memo text fields by specifying the number of columns for the memo text field. Enter a number to represent the width. For example:

Form        `<TCFIELD NAME="Comments" cols="125" />`

Block       `<tc:field name="Comments" category="EMPL" cols="125" />`  
`<tc:wizardParameter name="Comments" cols="125" />`

#### wrap

Sets how line breaks are managed in memo text fields. The following values can be entered:

- OFF—Restricts the browser from wrapping text, which forces the user to do it manually.
- SOFT or VIRTUAL—Eliminates all the line break symbols from the element's value after the form is submitted. `SOFT` is only supported by Netscape Navigator browsers. `VIRTUAL` is only supported by Microsoft Internet Explorer browsers.
- HARD or PHYSICAL—Maintains all the line break symbols in the element's value when the form is submitted. `HARD` is only supported by Netscape Navigator browsers. `PHYSICAL` is only supported by Microsoft Internet Explorer browsers.

For example:

Form        `<TCFIELD NAME="Comments" wrap="OFF" />`

Block       `<tc:field name="Comments" category="EMPL" wrap="OFF" />`  
`<tc:wizardParameter name="Comments" wrap="OFF" />`

### 3.1.24.6 Date Field Tag Attributes

#### showTime

Displays the time field to the right of date fields. Set the attribute value to "true" to have the time field appear next to the date field. For example:

Form        <TCFIELD NAME="DateHired" showTime="true" />

Block       <tc:field name="DateHired" category="EMPL" showTime="true" />  
             <tc:wizardParameter name="DateHired" showTime="true" />

#### timeUnderDate

Displays the time field directly below the date fields. Set the attribute value "true" to have the time field appear below the date field. For example:

Form        <TCFIELD NAME="DateHired" showTime="true" timeUnderDate="true" />

Block       <tc:field name="DateHired" category="EMPL" showTime="true"  
             timeUnderDate="true" />  
  
             <tc:wizardParameter name="DateHired" showTime="true"  
             timeUnderDate="true" />

**Important:** You must include the showTime attribute to use timeUnderDate.

#### size

Controls the width of date fields by specifying the number of characters for the field. For example:

Form        <TCFIELD NAME="DateHired" size="8" />

Block       <tc:field name="DateHired" category="EMPL" size="8" />  
             <tc:wizardParameter name="DateHired" size="8" />

### 3.1.24.7 List Field Tag Attributes

#### type

Turns a list field, which by default is a drop-down list, into a radio button list. Enter "radio" to set the list field as a radio button list. For example:

Form        <TCFIELD NAME="PerformanceReview" type="radio" />

Block `<tc:field name="PerformanceReview" category="EMPL" type="radio" />`

**Note:** You can define wizard parameters of type List as drop-down, radio, or check box lists (see [Defining Parameters](#)).

## size

Controls the height of list fields by specifying the number of rows in the list to be displayed. For example:

Form `<TCFIELD NAME="Gender" size="20" />`

Block `<tc:field name="Gender" category="EMPL" size="20" />`  
`<tc:wizardParameter name="Gender" size="20" />`

## allowNullValue

Includes a null value as the first item in a drop-down list. Enter "true" to include a null value in the list. That way, no item is selected by default.

**Important:** The `allowNullValue` attribute must be set to use `nullValueDisplayString`.

## nullValueDisplayString

Specifies the first item displayed in a drop-down or radio button list. Enter a message that should appear as the first item in the list, such as **Choose One**. For example:

Form `<TCFIELD NAME="Gender" allowNullValue="true"`  
`nullValueDisplayString="Please Select One" />`

Block `<tc:field name="Gender" category="EMPL" allowNullValue="true"`  
`nullValueDisplayString="Please Select One" />`  
`<tc:wizardParameter name="Gender" allowNullValue="true"`  
`nullValueDisplayString="Please Select One" />`

## forceSelect and forceSearch

If `forceSelect=true`, the field will display a drop-down list when used as a search view filter criterion. If `forceSearch=true`, the field will instead display a search module field with auto-suggest functionality.

### 3.1.24.8 Custom Object Field Tag Attributes

#### size

Controls the width of custom object fields by specifying the number of characters for the field. For example:

Form        `<TCFIELD NAME="ParentPolicy" size="10" />`

Block        `<tc:field name="ParentPolicy" category="EMPL" size="10"/>`  
              `<tc:wizardParameter name="ParentPolicy" size="10" />`

#### display

Sets whether the field for selecting a related custom object appears as a search module or a drop-down list.

For example, if only a few records are available, displaying a drop-down list is sufficient. If many records might be available for the user to select, then the field should be displayed as a search module.

If a qualifier has been defined for the custom field, this qualifier applies no matter which display type is used. However, if a search view has been selected for the field, this search view is only used if the field appears as a search module. For details about these settings for a custom field of type Custom Object, see [Custom Object Field Type](#).

The display type of custom fields of type Custom Object is initially determined by the default selection mechanism of the custom object, which is selected on the **General** tab of the custom object definition.

- Set the value to "dropdownlist" to make the field a drop-down list.
- Set the value to "searchmodule" to make the field a search module.

For example:

Form        `<TCFIELD NAME="ParentPolicy" display="dropdownlist" />`

Block        `<tc:field name="ParentPolicy" category="CLAM" display="dropdownlist" />`  
              `<tc:wizardParameter name="ParentPolicy" display="dropdownlist" />`

### 3.1.24.9 Involved Field Tag Attributes

#### size

Controls the width of Involved fields by specifying the number of characters for the field.

**Note:** Among wizard parameter types, the type corresponding to custom fields of type *Involved* is known as *Contact*.

For example:

Form      `<TCFIELD NAME="Claimant" size="10" />`

Block     `<tc:field name="Claimant" category="CLAM" size="10" />`

`<tc:wizardParameter name="Claimant" size="10" />`

### 3.1.25 Using Object Navigator

Object Navigator is a user interface tool you can use to specify a path to particular attributes in TeamConnect's object model for creating custom designs. End users generally do not have rights to Object Navigator or need to use it.

Many of TeamConnect design features require you to specify paths in the object model. For example, when defining a naming pattern for a custom object, you can use Object Navigator to specify the value of a field to be used in record names. Or you may need to specify a path when designing an EasyDocs document template.

TeamConnect's object model is comprised of attribute tables. Attributes tables are linked together to form a complex web of relationships. You can use Object Navigator to traverse these relationships and create a path that points to a specific field in the user interface.

To use Object Navigator, you need:

- An understanding of the object model in relation to the end-user interface
- Knowledge of attributes and paths
- A familiarity with Object Navigator and object models ([Object Model: Read This First](#)).

The following table describes where Object Navigator is located in TeamConnect.

**Features That Use Object Navigator**

Feature	Tab	Option that uses Object Navigator
Conditions	General	Defining expressions for conditions that you save for all object definitions
Custom Object Definition	Name	Automatic naming of custom object records using a pattern of object attributes
Any Object Definition	Custom Fields	Creating a custom field of type Custom Object and defining a qualifier for that field

Rules	Qualifier	Defining qualifiers to specify the desired conditions that trigger a rule
	Action	Defining actions for Audit rules to specify a description of the triggering event or to capture data in custom fields
Routes	Stops	Selecting a <b>User With Role</b> or <b>User Path</b> to add as a stop member in a route
Search Views	Filter Display	Selecting object attributes to be used as search qualifiers
	Results Display	Selecting object attributes to be used as display keys
Templates	Fields	Assigning dynamic or static-dynamic values to fields with the help of the operators containing the " <b>value of path</b> " options
	Sub-objects	
	Related Objects	
Wizards	Page Actions	Assigning dynamic or static-dynamic values to fields with the help of the operators containing the " <b>value of path</b> " and " <b>value of parameter</b> " options

### 3.1.25.1 Attributes

Attributes are the building blocks of objects. Each attribute in the object model is mapped to a column in TeamConnect's database. Attributes act like placeholders for the actual data that users enter. Most attributes appear as fields in the user interface. For example, the **currentPhaseType** attribute is represented by a field labeled **Current Phase**.

Each object in TeamConnect consists of multiple attributes, most of which appear in the object's main table in Object Navigator. For example, attributes for custom objects appear in the **TProject** table.



The following items can be displayed in Object Navigator:

- **(End of Path) Attributes**—Identify data end values and do not link to other tables. When selected, they are the end of the path.

For example, `createdOn` identifies the date a record was created.

- **Bridges**—Bridges are items you use to traverse (navigate the relationships) between tables in the object model to build paths. You cannot use them as an end of a path.



Bridges always have an arrow (->) next to their name, and their names typically have the **List** suffix. For example, `assigneeList->` links to the **JProjAssignee** table, which lists attributes about project assignees.

A bridge will only link to one other table. However, it is possible for multiple attributes to be bridges that link to the same table.

- **Bridge Attributes**—Bridge attributes are both attributes and bridges. Like bridges, they have an arrow (->) next to their name and can be used to traverse to another table. Because they also represent data, you can use bridge attributes to end a path.



For example, **defaultCategory->** identifies an object's default category, but it also links to the **WObjdCategory** table, which contains attributes that represent specific information about the category.

For a complete list of TeamConnect attributes, including which attributes can be used to traverse tables, end paths, or both, see [Object Model: Read This First](#).

### 3.1.25.2 Paths

A path is a specific route to the location of an attribute in the object model. It is a sequence of object attributes separated by periods (.) that identifies the location of the needed value. For example: `.mainAssignee.type` is a path that identifies the role of the project's main assignee.

Depending on the starting object table, you can distinguish two types of paths:

- **Simple**—An attribute in the current object table (the starting table) without traversing other tables.

For example, if you want a rule to check which user created a record, you could select the **createdBy->** bridge attribute in the current object table without traversing to another table. TeamConnect follows the resulting `.createdBy` path to find the value.


- **Complex**—A sequence of attributes that traverse various object tables, using the relationships between them (bridges and bridge attributes). If the required value resides in a different table, you must build a complex path to it.

For example, if you want to include the plaintiff's name in the record naming pattern of the custom object **Litigation**, you could navigate to **Plaintiff**, a custom field of type **Involved**:


**detailList-> Litigation-> Plaintiff**

The resulting path appears as:


```
.detailList(Litigation).detailInvlValueList(Plaintiff).detailValue.contact.
```

After you have used Object Navigator to create a path to an attribute, the path appears as text in the field associated to the **Object Navigator** icon .

### 3.1.25.3 Object Navigator Window

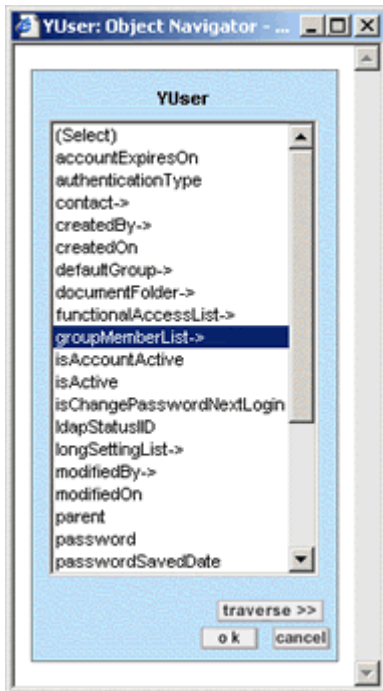
You can use Object Navigator with Designer screens that contain a field for identifying an attribute in the object model. These fields display the **Object Navigator** icon .

Some screens always display the **Object Navigator** icon, such as the **Qualifier** tab in rule screens. Other screens only display the **Object Navigator** icon when certain selections are made. For example, the **Fields** section of the **Templates** screen only displays the **Object Navigator** icon when particular operators are selected, such as **Attribute** in the **Value** field.

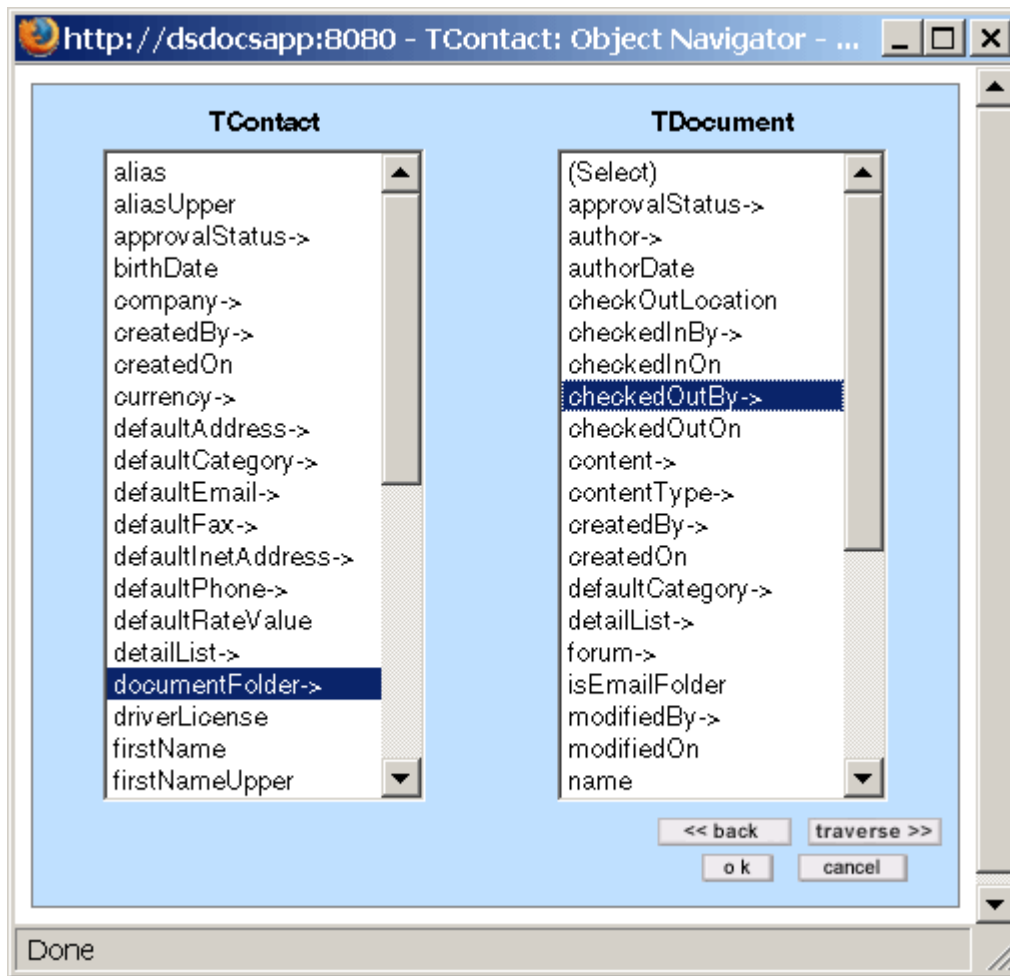
To start Object Navigator, click the **Object Navigator** icon  next to the field for which you want to define an attribute path. Object Navigator appears in a separate window that displays the starting table in the object model from which your path will originate. You can create a simple path by selecting an attribute in the starting table or a complex path by selecting a bridge or bridge attribute and traversing to open the next table. For details, see [Creating Paths](#).

Object Navigator displays the active table on the right, each table you have traversed on the left, and highlights each attribute that you add to the path.





Object Navigator Window with Only  
a Starting Table



Object Navigator Window Showing Traversed Starting Table and Current Table

The following table describes the items in the **Object Navigator** window.

#### Object Navigator Window

Item	Description
<b>Table name</b>	Name of traversed tables or current table.  The table names of system objects appear as they are defined in the object model, such as <b>TAppointment</b> . The starting tables of custom objects are named after the custom object, such as <b>Dispute</b> .
<b>Current table</b>	Displays the available attributes and bridges in the currently selected table. Select an attribute or bridge to add it to your path.
<b>Traversed table</b>	Displays a traversed table with the selected bridge or bridge attribute highlighted in the table.
<b>traverse</b>	First select a bridge or a bridge attribute and then click <b>traverse</b> to:

	<ul style="list-style-type: none"> <li>• Add the bridge or bridge attribute to your path</li> <li>• Open the table to which the bridge points</li> </ul>
<b>back</b>	<p>Click to go back to the previous table. When you click <b>back</b>, the selection is removed from the path. This button is only displayed after you have traversed at least one table.</p> <p>You can click <b>back</b> and <b>traverse</b> as needed to select different attributes or to traverse to different tables.</p>
<b>ok</b>	After you have selected your final attribute, click <b>ok</b> to complete the path.
<b>cancel</b>	Click to close Object Navigator and cancel the path.

### 3.1.25.3.1 Starting Tables

The first table that you see when you open Object Navigator is the starting table, which is the point from which your path originates.

Specifying your starting table is important because it corresponds to a table in the object model. [Object Model: Read This First](#) explains each table's attributes and where any bridges and bridge attributes lead. Using this information about your starting table, you can make selections that navigate to the desired attribute.

The starting table typically depends on the object for which you are building the path. For example, if you are defining rules, templates, search views, or wizards for an object, the starting table is the main table of that object (usually with the prefix **T**). There are **T** tables for eleven system objects, including Involved (for example, **TInvolved**).

For custom objects, the starting table is **TProject**. The **TProject** starting table is labeled with the name of the custom object, such as **Matter** or **Claim**, unless the specific custom object is not known. If you are building a path for a current user, the starting table is **YUser**.

The following table provides a list of starting tables that appear in Object Navigator in different screens. If you are not sure which attribute, bridge, or bridge attribute to select from these tables, refer to [Object Model: Read This First](#).

**Object Navigator Starting Tables**

Screen	Tab	Object drop-down list options	Starting table
<b>Conditions</b>	<b>General</b>	<b>Current Object</b>	T-table of the object you are working with, such as <b>TAccount</b> .
		<b>Current User</b>	<b>YUser</b>

<b>Custom Object Definition</b>	<b>Name</b>	N/A	<b>TProject</b> (labeled with the name of the current custom object for which you are creating the naming pattern, such as Dispute).
<b>Any Object Definition</b>	<b>Custom Fields</b> , for custom fields of type Custom Object	<b>this.</b>	<b>TProject</b> (labeled with the name of the custom object selected for the field's definition).
		<b>container.</b>	T-table of the object you are working with, such as <b>TAppointment</b> .
<b>Rules</b>	<b>Qualifiers</b>	<b>Current Object</b>	T-table of the object you are working with, such as <b>TAccount</b> .
		<b>Current User</b>	<b>YUser</b>
<b>Routes</b>	<b>Stops</b>	N/A	T-table of the object with which the route is associated, such as <b>TTask</b> .
<b>Templates</b>	<b>Fields</b>	N/A	T-table of the object you are working with, such as <b>TTask</b> .
	<b>Sub-Objects</b>	N/A	J-table of the sub-object, except categories, such as <b>JContAddress</b>  EDetail table for categories, such as <b>EContDetail</b> .
	<b>Related Objects</b>	N/A	T-table of the related object you are adding to the template, such as <b>TAccount</b> .
<b>Wizards</b>	<b>Page Actions</b>	<b>Current Object</b>	T-table of the object you are working with, such as <b>TAccount</b> .  <b>TProject</b> (labeled with the name of the custom object) is the starting table for custom objects. In Object Navigator, it is always displayed with the name of the

			custom object, such as <b>Case</b> , <b>Matter</b> , and <b>Policy</b> , except in cases where it is representing all available custom objects.
		<b>Related Objects</b> (from template)	T-table of the related object added to the associated template, such as <b>TAppointment</b> .
		<b>Sub-Objects</b> (from template)	J-table of the sub-object, such as <b>JProjAssignee</b> .
	<b>Qualifiers</b> (accessed by <b>Page Transitions</b> ) <b>Parameters</b> of type:	Lookup table items	The corresponding L-table of the item, such as <b>LContPhoneType</b> .
		Category	<b>WObjdCategory</b>
		Custom object	<b>TProject</b> (labeled with the name of the custom object for which you are creating the wizard).
		User	<b>YUser</b>
		Group	<b>YGroup</b>
		Account	<b>TAccount</b>
		Contact	<b>TContact</b>

#### 3.1.25.3.2 Filtering Sub-objects and Related Objects

If an attribute links to a list of sub-objects or related objects, you have the option of incorporating a filter into the path. The filter could be the sub-object's role or related object's category. For example, you may want to filter matter assignees by role or involved parties by category (role). This is done with the use of a special intermediary table that appears in Object Navigator but is not actually part of the object model.

In this intermediary table between the starting table and the related or sub-object table, you can do either of the following:

- Choose the specific item for the filter.
- Select (Any)->, meaning that you do not want the list to be filtered.

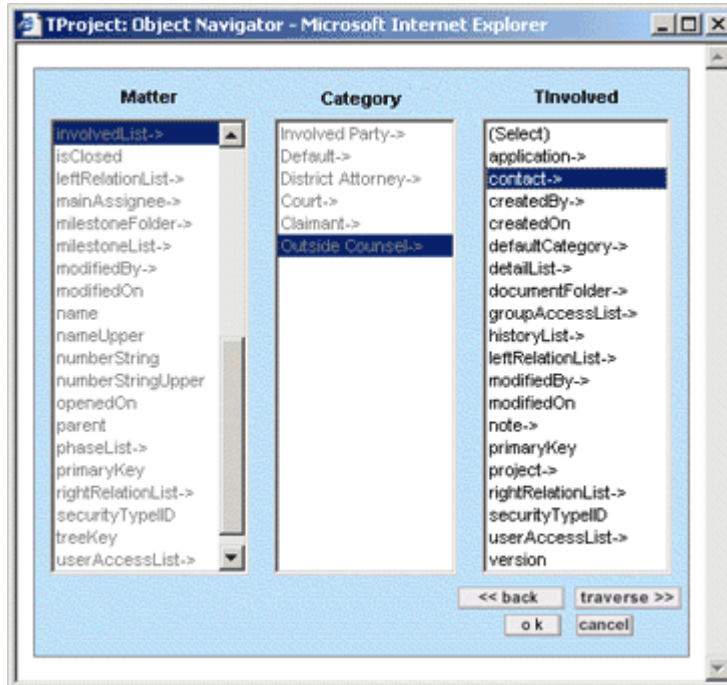
For example, to create a path from a matter to its involved parties' contact records which have the role Outside Counsel, you could specify the following:

**involvedList-> Outside Counsel-> contact-> (ok)**

This would result in the following path:

```
.involvedList (Outside Counsel) .contact
```

The following image shows what this path looks like in Object Navigator:



**Object Navigator Related Object Filtering Example**

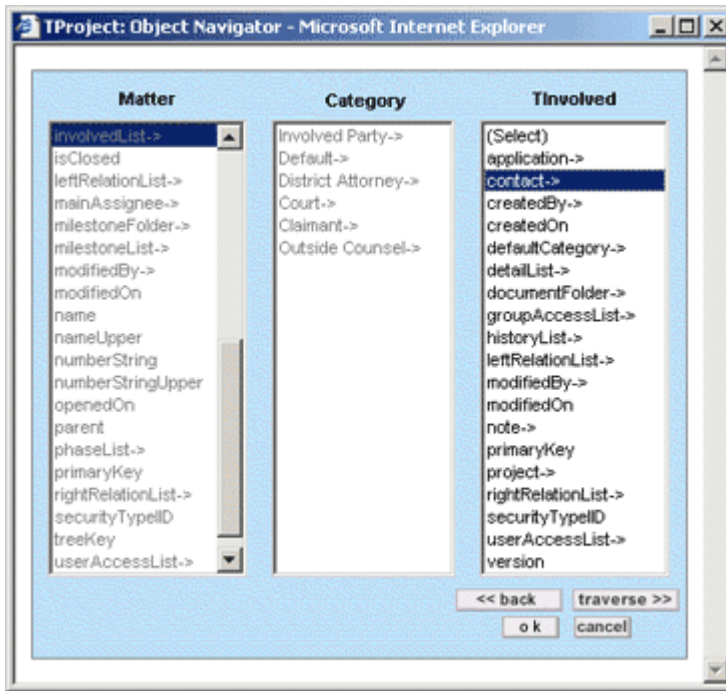
If you want to create a path from a matter to its involved parties without filtering them by their role, you could make the following selections:

**involvedList-> (Any)-> contact-> (ok)**

This would result in the following path:

```
.involvedList (Any) .contact
```

The following image shows what this path looks like in Object Navigator:



**Object Navigator Unfiltered Related Object Example**

The third table, which represents the involved parties, is the same whether you filter the list with an involved role or not. From the TInvolved table, you can access the contact information for the involved parties, such as their phone numbers or rates, through the contact-> attribute.

The following table indicates which attributes lead to intermediary, filtering tables in Object Navigator and the table to which the attribute links in the object model.

**Object Navigator Attributes That Link to Intermediary Tables**

Object table	Attribute	Leads to this intermediate table in the user interface	Then leads to this table in the object model
TProject	accountList->	Category	TAccount
	appointmentList->	Category	TAppointment
	assigneeList->	Assignee Role	JProjAssignee
	childList->	Child Object	TProject (labeled with the custom object's name)
	expenseList->	Category	TExpense
	involvedList->	Involved Role	TInvolved

	taskList->	<b>Category</b>	<b>TTask</b>
	rightRelationList->	<b>Relation Type</b>	<b>JProjRelation</b>
	leftRelationList->	<b>Relation Type</b>	<b>JProjRelation</b>
<b>TContact</b>	addressList->	<b>Address Type</b>	<b>JContAddress</b>
	emailList->	<b>Email Type</b>	<b>JContEmail</b>
	faxList->	<b>Fax Type</b>	<b>JContFax</b>
	phoneList->	<b>Phone Type</b>	<b>JContPhone</b>
	inetAddressList->	<b>Internet Address Type</b>	<b>JContInetAddress</b>
	skillList->	<b>Skill Type</b>	<b>JContSkill</b>
	rateList->	<b>Task Category</b>	<b>JContRate</b>
	rightRelationList->	<b>Relation Type</b>	<b>JContRelation</b>
	leftRelationList->	<b>Relation Type</b>	<b>JContRelation</b>
<b>TInvolved</b>	rightRelationList->	<b>Relation Type</b>	<b>JInvlRelation</b>
	leftRelationList->	<b>Relation Type</b>	<b>JInvlRelation</b>
<b>TAppointment</b>	resourceList->	<b>Resource Type</b>	<b>JApptResource</b>

### 3.1.25.3.3 Creating Paths

Before you create a path with Object Navigator, make sure that you are familiar with the following:

- Object Navigator and its components
- The concept of your starting table
- The location of the end of path attribute that you need
- Attributes, bridges, and bridge attributes the path requires


For details about starting tables, attributes, and the TeamConnect object model, see the [Object Model: Read This First](#).



- Necessary settings in the screen where you are creating a path (for example, wizards, templates and rules)

*Tip: If you know the attributes to include in the path, you can type the path into the field. Separate each attribute with a period (.)*

### To create a path with Object Navigator

1. In the screen where you want to create a path, click the **Object Navigator**  icon. Object Navigator opens, displaying the starting table.
2. If you know how to reach the necessary attribute, build the path and go to step 6.
3. If you are not sure which attribute to select, find the your starting table in [Object Model: Read This First](#).

*Tip: For custom objects, look up the **TProject** table in [Object Model: Read This First](#).*

4. Look through the **Comments** column of the selected table in [Object Model: Read This First](#) and determine whether there is an attribute in the current object table that contains the value you need. If you find the necessary attribute or bridge attribute in current table, select it in Object Navigator and go to step 6.
5. If you do not find the attribute in the current table:
  - a. Find the bridge that can link you to the appropriate table. If necessary, look up the tables to which the bridges link in [Object Model: Read This First](#) to see if they contain the attribute you need.
  - b. Select the bridge and click **traverse**.
  - c. A new table appears to the right of the starting table.
  - d. Repeat steps 2 through 4 until you have navigated to the necessary attribute.

*Tip: If you have traversed more than three tables but have not found the necessary attribute, you might be on the wrong track. Click **cancel** and start over.*

6. Click **ok**. Object Navigator closes and the path that you have created appears in the field.
7. To complete the settings for the path, make other selections required in the screen you are using.

The path is completed and displayed in the field. You can click the Object Navigator icon again and edit the path if necessary. Object Navigator opens, returning to the table where you left off. You can then select a different attribute, click back, or traverse to a different table.

### 3.1.25.4 Tips for Navigating Objects

This section provides tips for creating paths with Object Navigator and lists examples of common bridges and paths.

#### Practical Tips for Building Paths

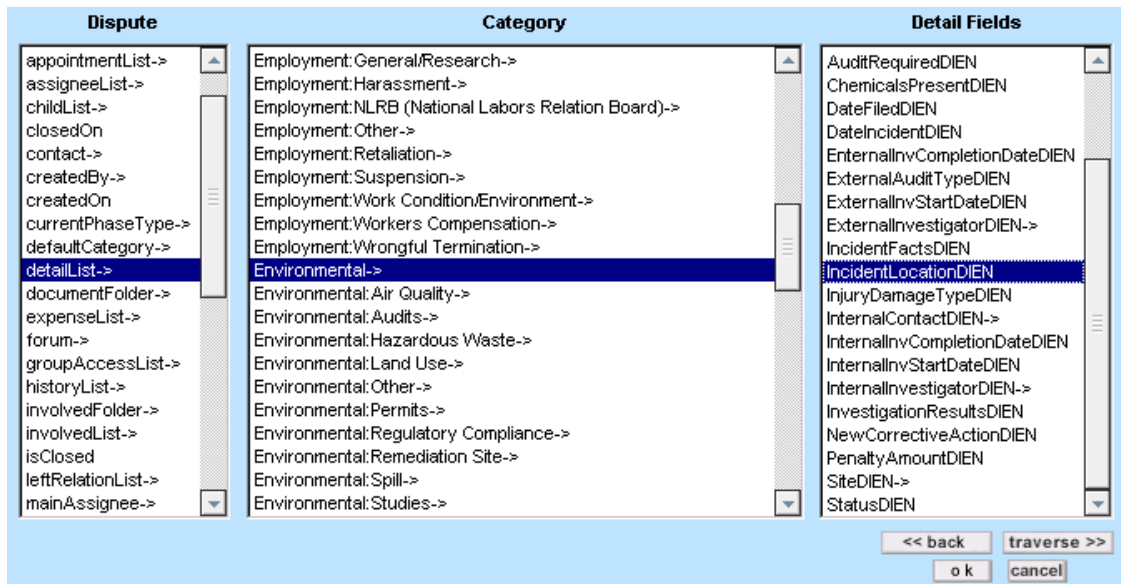
- The field label can help you determine which attribute to select. For example, the `currentPhaseType` attribute corresponds to the Current Phase field. However, there are two exceptions to this general rule:
  - The `type->` attribute (instead of "role") represents project assignee roles in the `JProjAssignee` table.
  - The `owner->` attribute (instead of "task" or "project") for task or project assignees (in the `JProjAssignee` and `JTaskAssignee` tables) identifies assignees for a task or project.
- To create a path, you typically do not have to traverse more than one or two tables. If you have traversed over three tables and still cannot find the required attribute:
  - You may have selected the wrong bridge. If so, click back or cancel and start again. For assistance, check [Object Model: Read This First](#).
  - The required attribute may not be accessible from the current object table. If so, you may have to write a Java action for the desired operation.
- When constructing a path, make sure that the end of path value type matches the value type of the field whose value you are setting.

For example, if you are setting a value for a number field and the path ends with a date, you cannot add the path. Or, if you are building a path that identifies a user, the end of path attribute must be in the `YUser` table.
- In most cases, `displayOrder`, `version`, and `primaryKey` are not useful.
- In certain screens, attributes with the suffix "List" cannot be traversed, so they are not displayed with an arrow (->). This indicates that the related table is not applicable.

#### Common Bridges and Bridge Attributes

- The bridge attribute `contact->` in the `TProject` table represents the contact selected for contact-centric custom objects.
- Items that have the "List" suffix are bridges that typically link to sub-objects, such as a project's assignees.
- To specify a user group, follow the `.groupMemberList.group` path from the `YUser` table.
- If you need to identify a field containing a system lookup table item, such as the field containing the role of a project assignee, your last selection in the path should be the bridge attribute `type->`. For example:
  - `.assigneeList.type`
  - `.rightRelationList.type`
  - `.addressList.type`

- Custom fields created for an object can be reached by traversing the bridge detailList-> to the Category table, which links you to the list of all custom fields created for the selected category as shown in the following example:



Example of Traversing to Custom Fields

See also the example of a complex path in [Paths](#).

### Commonly Used Paths

The following table lists the origins and destinations of common paths and the selections in Object Navigator used to build them. For more information on specific object model tables and attributes, see [Object Model: Read This First](#).

#### Common Object Navigator Paths

From this object	To this attribute	Path
Any <b>T-table</b> (any object)	Custom fields of the current object	1 detailList-> (traverse) 2 categoryName-> (traverse) 3 customFieldName (ok)
	A record's default category	<b>defaultCategory-&gt;</b> (ok)
	Any record category	1 detailList-> (traverse) 2 categoryName-> (ok)
	The main assignee of a project that is selected in a custom field of type Custom Object	1 detailList-> (traverse) 2 categoryName-> (traverse)

		<p>3 detailFieldName-&gt; (traverse)</p> <p>4 mainAssignee-&gt; (traverse)</p> <p>5 user-&gt; (ok)</p>
	The user who created a record	<b>createdBy</b> -> (ok)
<b>TExpense</b>	The total amount of an expense	<b>totalAmount</b> (ok)
	The main assignee of a related project	<p>1 project-&gt; (traverse)</p> <p>2 mainAssignee-&gt; (traverse)</p> <p>3 user-&gt; (ok)</p>
<b>TInvoice</b>	The main assignee of a project identified in an invoice line item	<p>1 lineNumber-&gt; (traverse)</p> <p>2 project-&gt; (traverse)</p> <p>3 mainAssignee-&gt; (traverse)</p> <p>4 user-&gt; (ok)</p>
	The net total of a line item	<p>1 lineNumber-&gt; (traverse)</p> <p>2 netTotal (ok)</p>
	The task category of a line item	<p>1 lineNumber-&gt; (traverse)</p> <p>2 taskCategory-&gt; (ok)</p>
	The expense category of a line item	<p>1 lineNumber-&gt; (traverse)</p> <p>2 expenseCategory-&gt; (ok)</p>
	The category of a vendor	<p>1 vendor-&gt; (traverse)</p> <p>2 detailList-&gt; (traverse)</p> <p>3 categoryName-&gt; (ok)</p>
	The date of the invoice	<b>invoiceDate</b> (ok)
<b>TProject</b> (any custom object)	An assignee with a specific role	<p>1 assigneeList-&gt; (traverse)</p> <p>2 &lt;role&gt;-&gt; (ok)</p>
	A user assigned to a project with a specific role	1 assigneeList-> (traverse)

		2 <role>-> (traverse) 3 user-> (ok)
	The current phase of a project	<b>currentPhaseType-&gt;</b> (ok)
	The user who is the project's main assignee	1 mainAssignee-> (traverse) 2 user-> (ok)
	The role of a project's main assignee	1 mainAssignee-> (traverse) 2 type-> (ok)
	The main contact of a contact-centric project	<b>contact-&gt;</b> (ok)  <i>Note: Only available when the custom object is contact-centric.</i>
	Custom fields of a parent project of the current object	1 parent-> (traverse)  <i>Note: Only available when there is a parent-child relationship.</i>  2 detailList-> (traverse) 3 categoryName-> (traverse) 4 customFieldName (ok)
<b>YUser</b>	The user's group membership	1 groupMemberList-> (traverse) 2 group-> (ok)
	The user's default group	<b>defaultGroup-&gt;</b> (ok)

### 3.1.26 Object Model: Read This First

This appendix introduces a series of appendixes with reference tables to aid in the use of the TeamConnect object model. These tables provide information about objects, attributes and relationships in TeamConnect's object model, which you can use to create custom designs.

These appendixes do not describe the objects that construct the administrative and design functionality in TeamConnect, or TeamConnect's database structure. There are many parallels between the database structure and the object model, but they are distinct.

The appendixes are:

- [Object Model: Projects](#)
- [Object Model: Contacts](#)
- [Object Model: Appointments](#)
- [Object Model: Accounts](#)
- [Object Model: Expenses](#)
- [Object Model: Tasks](#)
- [Object Model: Invoices](#)
- [Object Model: Histories](#)
- [Object Model: Documents](#)
- [Object Model: Common](#)

### 3.1.26.1 Objects and Object Model

Each table in these appendixes corresponds to an object in the object model. You may use these tables to understand the structure that organizes data in TeamConnect. For example, use these tables when you are doing any of the following actions:

- Using TeamConnect's Object Navigator tool in the user interface while creating rules, templates, wizards, wizard page transition rules, or automated object naming patterns based on attribute values.
- Creating templates using Document Generator or EasyDocs.
- Selecting attributes from a list (rather than from Object Navigator), such as when creating qualifiers in a search view or setting an object's Unique ID to a pattern of certain attributes.

### 3.1.26.2 Legacy Names and Newer Names

Some attribute names were changed in TeamConnect version 3.4. The new names are used when developing custom blocks with XML. The legacy names are used when working with search views, rules, Object Navigator, and other TeamConnect features. When an attribute has both a legacy name and a newer name, that fact is noted in its description.

### 3.1.26.3 Intended Audience

These appendixes are primarily intended for solution developers who are involved in the process of customizing TeamConnect to meet the needs of an organization. The user of these appendixes is expected to have the following skills and knowledge:

- Conceptual understanding of:
  - Object-oriented programming
  - Relational databases
- Analytical skills

- Understanding of the business needs of the organization for which you are developing a solution

#### 3.1.26.4 Object Model Conventions

- Each table in this documentation includes all of the attributes available in Object Navigator for the corresponding object. This includes relationships that do not exist in the database.
- Attributes are listed in alphabetical order within each table.
- You can use the grouping of tables in the left pane to understand the relationships between the various tables. The tables are grouped in the left pane according to the main objects to which they are most closely associated (such as Account, Project, or Task). Tables that are administrative and are not closely associated with one particular object are listed in the **Common** section.
- Each main object's tables are listed in the following order:
  - a. T-table
  - b. J-tables and their corresponding R-tables
  - c. Category and Custom Field tables (E-tables)
  - d. User and group access tables
- Object (table) names are provided as links, so that if you are viewing this documentation electronically, you can quickly navigate to the table for the corresponding object.
- Items that have the data type **object** are treated as objects, rather than foreign keys (as they are treated in the database) in the descriptions that are listed in the **Comments** column of each table. This is why the **Comments** column does not indicate that an attribute stores the ID or primary key of the object, but instead indicates that the attribute identifies the object itself. This convention is followed in order to lessen confusion for users who are not concerned with the way the database actually works.

For example, in the table **TInvolved**, the attribute **project-->** is described as the project to which the involved record belongs, rather than the primary key of the project to which the involved record belongs.

##### 3.1.26.4.1 Enumerations

Enumerations are represented in the Object Model by `typeIID` (Internal ID) attributes that can only have static, predefined values. The predefined values of these enumerations are listed in parentheses in the **Comments** column of the table for each such attribute. In some queries you may wish to use the integer that represents each value. In other situations, such as embedding report tags into custom blocks, you will need to use the Java constant name for the value instead of the integer.

As an example, consider attribute `typeIID` in entity `TInvoice`. It has four possible values:

Integer	Java constant	Text shown to end user (US English locale)
0	STANDARD	Standard

1	CREDIT_NOTE	Credit Note
2	ACCRUAL	Accrual
3	SHADOW	Shadow Invoice

The text displayed to the end user for an enumeration may vary depending upon the locale that is being used for the session. However, the Java constant names will always be those described in the Comments for the attribute, regardless of locale.

If you are querying the actual database value, you would look for one of the integers (0 through 3). If you are referencing this attribute in connection with a report tag in a custom block you would use the Java constant name that corresponds to the value you are looking for.

The datatype shown for enumerations in the Object Model tables is **Enum**. This usually represents an integer datatype in the underlying database (although some enumerations are character datatypes), extended with predefined values and Java constant names.

**Note:** All enumerations in this Object Model have attribute names ending in "IID". There is further information about enumerations in .XCT files that are provided in the installation media; see [Viewing XCT Files](#).

#### 3.1.26.4.2 Object Model Terms

You must understand the meaning of the following terms in order to work with this document:

- **Object** is used to refer to objects in the object model. This is a different definition for this term than for other TeamConnect documentation. The distinction is that in TeamConnect's GUI, only the main objects are treated as objects (for example, System Objects, Custom Objects, Related Objects, and so on). In the object model, this term is more broad.
- **Object Navigator** is a UI tool that allows you to navigate through tables that represent objects in the object model. This makes it easier for you to select attributes when necessary for the task you are completing. For more details about Object Navigator, see [Using Object Navigator](#).
- **Attributes** are the building blocks of objects, and represent the actual data that end user enter into TeamConnect. They also represent relationships between objects. For example, every Custom Object consists of over a dozen attributes, all of which are listed in the main object table for Project: mainAssignee, createdOn, createdBy, closedOn, defaultCategory, and so on. All of the attributes for an object are listed in its corresponding table in the document.
- **End of path** If an attribute is an **end of path**, it can be selected at the end of a path in Object Navigator. An end of path contains some type of value, such as a number, date, or text.

If an item cannot be selected as the ending of a path, it is a "bridge" and exists in the object model for the purpose of establishing relationships to other objects.

- **Bridge** If an attribute is a Bridge, it allows you to traverse to other tables in Object Navigator that correspond to objects in the object model. Some bridges correspond to database columns and represent a specific data value. However, not all bridge attributes exist in the database. If an item is marked as a bridge but not an end of path, it does not represent a database column. It



represents a one-to-many relationship and exists in the object model for the purpose of establishing relationships to other tables, and can never be selected as an end of path.

- **Table** is used to refer to objects, for three reasons:
  - Each table in this document represents an object in the object model. This format is used to convey the information in a logical and visual manner.
  - Object Navigator represents objects by displaying them as tables in the UI.
  - All of the objects in this document have corresponding tables in the database, with corresponding columns for most attributes.

#### 3.1.26.4.3 Descriptions of Columns in Tables

The columns in each table in this document are provided to give you a good understanding of each attribute. The following columns are included in each table:

<b>TeamConnect Attribute</b>	Name of the attribute as it appears in Object Navigator.
<b>Field in UI</b>	System field in TeamConnect's end-user UI (as provided by default with the system) that corresponds to the attribute. The location of the field in the UI is indicated in parentheses. Not all attributes are represented by fields in the UI.  This document does not attempt to reflect any customization of System Fields in the UI that has been performed for your system.
<b>End of path</b>	Indicates whether the item can be selected at the end of a path in Object Navigator. See <a href="#">Object Model Terms</a> for details.
<b>Bridge</b>	Indicates whether the item is a bridge. This is also indicated by an arrow (-->) at the end of the attribute name. See <a href="#">Object Model Terms</a> for details.
<b>Links to object table</b>	If the item is a bridge, this column indicates the object in the object model (or, the table in Object Navigator tool) to which the bridge provides a relationship.  <b>Note:</b> <i>If the item is ALSO an attribute, then the primary key of this object is the value that the attribute stores in the database.</i>  <b>Tip:</b> <i>When viewing this document electronically, you can click the name of the table to go directly to that object's attribute descriptions in this document.</i>
<b>Data type</b>	Indicates the data type of the attribute.

	<ul style="list-style-type: none"> <li>• If the data type is <b>string</b>, the character limit is indicated in parentheses ().</li> <li>• If the data type is <b>longString</b>, the limit is imposed by the database. In Oracle, the maximum size of a longString value is 4 GB. In SQL, the maximum size is 1 GB.</li> <li>• If the data type is <b>datetime</b>, there is an additional property to consider. Some datetime attributes are <b>time-zone-independent</b>. This means that the value in the database for this attribute is retrieved and displayed as is, regardless of the local time zone of the user who entered that value, or the time zone of the user who is now viewing that value. Other attributes are time-zone-dependent. Their value is stored in the database but is adjusted during retrieval to compensate for the time zone of the user who is viewing the value.  In this documentation, the phrase "time-zone-independent" will appear for attributes that possess this property.  Custom datetime fields that are created by solution developers are configurable. The solution developer can choose whether or not to make the field time-zone-independent.</li> </ul> <p>• Bridges are complex data types, and are marked in this documentation with the data type <b>object</b>, except when the bridge links to a list of records rather than a single record.</p>
<p><b>Comments</b></p>	<p>Provides a description of the item. For beginners, this column may also provide suggestions for using the item.</p> <p>For items that are marked <b>End of path</b> and <b>Bridge</b>, two bulleted descriptions are given in this column. These bullets describe the two ways that you can use this item.</p> <p>For example:</p> <p><b>8</b> The first item is marked with this bullet and describes the item's function when selected as an end of path. For example, selecting user--&gt; as the end of your path in the table JProjAssignee gives you the user who is being identified as an assignee of the project.</p> <p>--&gt; The second item is marked with this bullet and describes the contents of the table to which the item links, if you use the item as a bridge. For example, selecting user--&gt; as a bridge in the table JProjAssignee gives you access to the table YUser, which contains that user account's definition information.</p> <p>These two descriptions are provided in order to help you determine the following:</p> <ul style="list-style-type: none"> <li>• Whether or not the item should be included as part of your path.</li> <li>• Whether the item is the attribute you are trying to reach and should therefore be selected as the end of your path.</li> </ul>

	<ul style="list-style-type: none"> <li>Whether you need to use the item as a bridge, and link to the table to which the bridge provides access.</li> </ul>
<b>Commonly used in (Rules, Templates/Wizards)</b>	<p>These columns have been introduced primarily to help you learn to build a path using Object Navigator. Because Object Navigator displays all existing attributes, regardless of their function, there are many selections available that serve no practical purpose when you are creating Rules, Templates, Wizards, and Wizard Rules. You can use these columns to help you distinguish between useful and non-useful selections.</p> <p>For example, in most cases, the Version attribute is not considered a commonly used attribute for creating Rules, Templates, or Wizards.</p>

### 3.1.26.5 Object Model Properties

The organization of the object model follows several conventions. Becoming familiar with these conventions will help you understand the structure of the object model. The following sections also contain reference material, such as the unique codes of TeamConnect objects.

#### 3.1.26.5.1 Object Table Prefixes

Object tables are named with a one-letter prefix according to their function in the object model. The table below indicates the various types of tables and their prefixes.

Prefix	Description	Examples
<b>T</b>	<p>T-tables are the 11 main tables for TeamConnect objects. In Object Navigator, they are almost always the starting point of navigation.</p> <p>A T-table contains the basic information for each record of its type that is created. In other words, it contains instances of an object definition. T-tables own specific instances of custom fields (E-tables), categories (E-tables), and sub-objects (J-tables), all of which store items added to the record in batch screens.</p> <p>T-tables are not owned by any other kind of table. However, a T-table might have a relationship to another T-table. For example, T-History is related to all of the other T-tables, because History records can be created for each kind of record.</p>	<ul style="list-style-type: none"> <li><b>TAccount</b></li> <li><b>TAppointment</b></li> <li><b>TContact</b></li> <li><b>TDocument</b></li> <li><b>TExpense</b></li> <li><b>THistory</b></li> <li><b>TInvoice</b></li> <li><b>TInvolved</b></li> <li><b>TProject</b></li> <li><b>TTask</b></li> </ul>
<b>J</b>	<p>J-tables are owned by T-tables (or, occasionally, R-tables). J-tables are also</p>	<ul style="list-style-type: none"> <li><b>JApptAttendee</b></li> <li><b>JContAddress</b></li> </ul>

	<p>known as sub-objects.</p> <p>The values stored in J-tables are usually obtained from lookup tables (L-tables). In the UI, items in a J-table are added to a record through a batch screen. For example, assignees added to a project in a batch screen are sub-objects of the project.</p>	<ul style="list-style-type: none"> <li>• <b>JContEmail</b></li> <li>• <b>JProjAssignee</b></li> <li>• <b>JTranDetail</b></li> </ul>
<b>L</b>	<p>L-tables store the main lookup table information for system lookup tables.</p>	<ul style="list-style-type: none"> <li>• <b>LContFaxType</b></li> <li>• <b>LCountryItem</b></li> <li>• <b>LProjAssigneeType</b></li> <li>• <b>LContSkillType</b></li> </ul>
<b>E</b>	<p>E-tables are used with every T-table throughout the object model to store the following:</p> <ul style="list-style-type: none"> <li>• specific categories that have been added to records</li> <li>• specific custom field values added to records</li> <li>• specific user or group security settings for records</li> </ul>	<ul style="list-style-type: none"> <li>• <b>EHistDetail</b> (contains categories that have been added to History records)</li> <li>• <b>EHistDetailNumbValue</b> (contains values that have been added in custom fields of the type Number, in History records)</li> <li>• <b>EHistUserAccess</b> (contains user security settings for History records)</li> </ul>
<b>R</b>	<p>There are only four R-tables in the object model. Each of the R-tables has a unique purpose (shown at right).</p> <p>Like T-tables, R-tables are not owned by any other table. However, they do not contain all of the attributes found in a T-table. For example, they do not have custom fields or categories associated with them, and they do not belong to an object definition.</p>	<ul style="list-style-type: none"> <li>• <b>RTransaction</b> (stores instances of financial transactions)</li> <li>• <b>RDistribution</b> (stores contact groups, that is, Address Books)</li> <li>• <b>RApproval</b> (stores instances of approvals)</li> <li>• <b>RForum</b> (stores any forums created for records)</li> </ul>
<b>Y*</b>	<p>Y-tables are system tables. They typically contain values that correspond to</p>	<ul style="list-style-type: none"> <li>• <b>YGroup</b></li> <li>• <b>User</b></li> </ul>

	administration objects, such as groups and users. They are owned by Z-tables.  Custom lookup table information is also stored in Y-tables.	<ul style="list-style-type: none"> <li>• <b>YDetailLookupTable</b></li> <li>• <b>YDetailLookupItem</b></li> </ul>
<b>W*</b>	W-tables store object definition information. The components related to object definition are stored in these tables. They are owned by Z-tables.  Categories and custom field definitions for all objects are also stored in W-tables.	<ul style="list-style-type: none"> <li>• <b>WObjdCategory</b></li> <li>• <b>WObjdDetailField</b></li> </ul>
<b>Z*</b>	Z-tables own W-tables and Y-tables. They contain specific information pertaining to system design, such as object, template, wizard, and search view definition information.	<b>ZObjectDefinition</b>
<b>U*</b>	U-tables are administrative tables that are owned by W-tables.	<b>UGrupMember</b>
<p>* Most tables that are named with this prefix are not typically used when creating rules, templates, wizards, or object naming patterns, although there are certain exceptions. For example, <b>UGrupMember</b> is used in rules that check whether the current user is a member of a certain group.</p>		

#### 3.1.26.5.2 Unique Codes of System Objects and Lookup Tables

Each system object and each system lookup table has a four-character unique code. You will need to use these unique codes for certain tasks such as writing document template for Document Generator or writing automated qualifiers or actions for rules.

The following table lists the unique codes and database table names for system objects.

System Object	Object Table Name	Database Table Name	Unique Code
<b>Account</b>	TAccount	T_ACCOUNT	ACCT
<b>Appointment</b>	TAppointment	T_APPOINTMENT	APPT
<b>Contact</b>	TContact	T_CONTACT	CONT
<b>Document</b>	TDocument	T_DOCUMENT	DOCU
<b>Expense</b>	TExpense	T_EXPENSE	EXPE

<b>History</b>	THistory	T_HISTORY	HIST
<b>Invoice</b>	TInvoice	T_INVOICE	INVC
<b>Invoice Line Item</b>	JInvcLineItem	J_INVC_LINE_ITEM	LNIS
<b>Task</b>	TTask	T_TASK	TASK

The following table lists the unique codes and database table names for all system lookup tables.

<b>System Lookup Table</b>	<b>Object Table Name</b>	<b>Database Table Name</b>	<b>Unique Code</b>
<b>Activity Item</b>	LTaskActivityItem	L_TASK_ACTIVITY_ITEM	ACTI
<b>Address Type</b>	LContAddressType	L_CONT_ADDRESS_TYPE	ADDR
<b>Area Item</b>	LApptAreaItem	L_APPT_AREA_ITEM	AREA
<b>Contact Relation Type</b>	LContRelationType	L_CONT_RELATION_TYPE	CONR
<b>Country Item</b>	LCountryItem	L_COUNTRY_ITEM	COUN
<b>Currency Item</b>	LCurrencyItem	L_CURRENCY_ITEM	CURR
<b>Email Type</b>	LContEmailType	L_CONT_EMAIL_TYPE	MAIL
<b>Fax Type</b>	LContFaxType	L_CONT_FAX_TYPE	FAXX
<b>Internet Address Type</b>	LContInetAddressType	L_CONT_INET_ADDRESS_TYPE	INET
<b>Phone Type</b>	LContPhoneType	L_CONT_PHONE_TYPE	PHON
<b>Project Assignee Type</b>	LProjAssigneeType	L_PROJ_ASSIGNEE_TYPE	Unique code of the corresponding custom object definition.
<b>Project Relation Type</b>	LProjRelationType	L_PROJ_RELATION_TYPE	PRJR

<b>Resource Type</b>	LApptResourceType	L_APPT_RESOURCE_TYPE	RESO
<b>Skill Type</b>	LContSkillType	L_CONT_SKILL_TYPE	SKIL
<b>Territory Type</b>	LContTerritoryType	L_CONT_TERRITORY_TYPE	TERR

### 3.1.26.5.3 Viewing XCT Files

To verify information about the attributes in any table, such as the data type or equivalent database column or table name, you can view the XCT file corresponding to the table. The XCT files provide mapping between the database and the object model. These files are used by Object Navigator to obtain the correct attributes. In the XCT files, the external name of an attribute or table is its name in the TeamConnect database.

The XCT files are located on the TeamConnect installation media in the **utilities/objectmodel** directory. This directory contains two folders: **simpletypes** and **complextypes**.

- **utilities/objectmodel/complextypes** contains the XCT files for all objects in the object model.
- **utilities/objectmodel/simpletypes** contains the XCT files for `typeIID` attributes. These object attributes are enumerations that are permitted to have only certain static values that have been predefined. Each file in the **simpletypes** directory provides a list of possible values for the corresponding `typeIID` attribute. The values are also listed in this document in the **Comments** column for each `typeIID` attribute.

### 3.1.26.5.4 Object Names vs. Database Table Names

The names of most tables in the object model are similar to their corresponding table names in the database. Also, the names of most attributes in the object model are similar to their corresponding column names in the database. The name of a table or column in the database is usually the name of the table (or attribute) in the object model, except that it is in caps, with underscores between words. For example:

LContRelationType = L\_CONT\_RELATION\_TYPE

shortDescription = SHORT\_DESCRIPTION

TAccount = T\_ACCOUNT

This convention is used for most, but not all, tables and attributes. To verify the name of a table or column in the database, refer to the corresponding XCT file, and check the External Name of the attribute or table, as described in [Viewing XCT Files](#).

### 3.1.26.5.5 List Attributes for Sub-Objects and Related Objects

Many bridge attributes have the suffix **List**. In Object Navigator, bridge attributes appear with an arrow after the suffix, like this: **List-->** This suffix is added to attributes that link to sub-objects that are added to records (for example, JContAddress, JContPhone, and JProjAssignee) and related objects that are added to records (for example, THistory, TInvolved, and TTask).

The following are important points about attributes with the suffix **List**:

- When you want to access sub-objects that have been added to a record, you should use the corresponding J-table. This includes contact addresses, contact phone numbers, appointment attendees, project assignees, and other items that are added in batch screens.

The one exception to this rule is categories, which are accessed through **detailList-->** attributes. Although they are added in a batch screen in a record and are handled as sub-objects, they are stored in an E-table rather than a J-table.

- When you want to access related objects that have been added to a record, such as the involved records added to a project, you should use the List attribute that corresponds to it (in this case, **involvedList-->**) to access the corresponding T-table (TInvolved).
- In Object Navigator, most attributes with the suffix **List** link first to an intermediary table that allows you to filter the related or sub-objects.

### 3.1.27 Object Model: Projects

The tables in this appendix provide information about the object model as it relates to projects.

#### 3.1.27.1 TProject

TProject contains the general information for a project record. This information includes the project's name, creation date, closing date, parent project, main assignee, and other information. This table also links to the list of a project's assignees, which are sub-objects of a project, categories added to the project, and related objects of the project (such as child projects and involved records).

Object: TProject (T\_PROJECT)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wiz
<b>accountList--&gt;</b>	n/a	<b>Accounts tab</b>	x	x	<a href="#">TAccount</a>		The list of accounts that have this project selected in their posting criteria.  Links to account information for	x	



							the related accounts.  In Object Navigator, links to an intermediary table named <b>Category</b> that allows you to filter the accounts based on a category, if desired.		
application -->	APPLICATION_ID  (NUMBER )		x	x	<a href="#">WObjdProjectInfo</a>	object	The Custom Object definition to which this project belongs.		
appointmentList-->	n/a	Appointments tab	x	x	<a href="#">TAppointment</a>		8 The list of appointments that are related to the project.  8 Links to specific appointment information.  8 In Object Navigator, links to a table named <b>Category</b> that allows you to filter the appointments based on a category, if desired.		
assignees-->  Before TeamConnect 3.3	n/a	Assignees tab	x	x	<a href="#">JProjAssignee</a>		8 The list of assignees added to the project.	x	

<p><b>SP2:</b> <b>assigneeList--&gt;</b></p>							<p>--&gt; Links to the join table that contains specific information for assignees added to the project.</p> <p>In Object Navigator, links to a table named <b>Assignee Role</b> that allows you to filter the assignees based on a role, if desired.</p>		
<p><b>embedded Projects--&gt;</b></p> <p><b>Before TeamConnect 3.3</b> <b>SP2:</b> <b>childList--&gt;</b></p>	n/a	<p>Tabs for child or embedded object records</p>	x	x	<a href="#">TProject</a>		<p>8 The list of all child projects of the current project, including child and embedded objects. Does not include grandchild objects.</p> <p>--&gt; In Object Navigator, links to the <b>Child Object</b> table, which allows you to select the child or embedded custom object whose records you want to identify in your path. After you select a child object, links to the <b>TProject</b> table.</p>	x	

							The <b>Child Object</b> table is not an object in the object model and appears only in Object Navigator to allow you to identify which child object's records you need.		
closedOn	<b>CLOSED_ON</b> (DATE)	<b>Closed On</b> (General block)	x			date	Date the project status was changed to Closed.	x	
contact-->	<b>CONTACT_ID</b> (NUMBER)	Field for selecting a contact, with customized label, in <b>General</b> block	x	x	<a href="#">TContact</a>	object	8 The contact selected for a contact-centric project.  --> Links to the specific information about the contact object.		x
createdBy-->	<b>CREATED_BY_ID</b> (NUMBER)	<b>Created By</b> (Security block)	x	x	<a href="#">YUser</a>	object	8 User who created the record.  --> Links to specific information about the user account.	x	x
createdOn	<b>CREATED_ON</b> (DATE)	<b>Created On</b>	x			date	Date the project record was created.	x	x

		(Security block)							
createdOn BehalfOf	CREATED_ON_BEHALF_OF_ID			x	<a href="#">TContact</a>	object	When not null, it indicates that the record was created through an e-billing system or other external application. The contact referenced will often be a vendor in TeamConnect.	<b>X</b>	
currentPhaseType-->	CURRENT_PHASE_TYPE_ID (NUMBER)	Current Phase (General block)	x	x	<a href="#">WObjdPhaseType</a>	object	8 The current phase of the project. --> Links to the definition of the phase.	<b>x</b>	
defaultCategory-->	DEFAULT_CATEGORY_ID (NUMBER)	Indicated by a blue diamond (Categories block)	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 The default category for the project. --> Links to the category's definition information.	<b>x</b>	<b>x</b>
categories-> Before TeamConnect 3.3 SP2: detailList-->	n/a		x	x	<a href="#">EProjDetail</a>  In Object Navigator, links to Category list which appears in the UI only.		8 The list of categories added to the current object. --> In the object model, the added categories link to the values that have been added for custom fields,	<b>x</b>	

							<p>according to field type.</p> <p>--&gt; In Object Navigator, this attribute is enhanced. It links to a list of all categories defined for the object definition, rather than those added to the current object. From there, you can select a category and traverse to the list of custom fields that belong to it.</p>		
documentFolder-->	DOCUMENT_FOLDER_ID (NUMBER)			x	<a href="#">TDocument</a>	object	Links to specific information about the document folder.		
expenseList-->	n/a	Expenses tab	x	x	<a href="#">TExpense</a>		<p>8 The list of expenses that are related to the project.</p> <p>--&gt; Links to specific expense information.</p> <p>In Object Navigator, links to a table named <b>Category</b> that allows you to filter the expenses based on a</p>		

							category, if desired.		
groupAccessList-->	n/a			x	<a href="#">EProjGroupAccess</a>		Links to the group security information entered through the Security block of the record.	x	
historyList-->	n/a	History tab	x	x	<a href="#">THistory</a>		8 The list of histories that are related to the project.  --> Links to specific history information for all of the project's history records.	x	
involvedFolder-->	INVOLVED_FOLDER_ID (NUMBER)			x	<a href="#">TDocument</a>	object	Links to the folder that stores documents that belong to involved parties for the project.		
involvedList-->	n/a	Involved tab	x	x	<a href="#">TInvolved</a>		8 The list of involves that are related to the project.  --> Links to specific involved party information for involved parties of the project.  In Object Navigator, links to a table named <b>Involved Role</b> that allows you	x	

							to filter the involved parties based on a role, if desired.		
isClosed	IS_CLOSED (NUMBER)		x			boolean	The setting for whether a project is in the open or closed status.  1 - is Closed 0 - is Open	x	
isElectronic	IS_ELECTRONIC		x			boolean	If TRUE, indicates that the record originated from an external application such as Collaborati.		
relations--> <b>Before TeamConnect 3.3 SP2: leftRelation List--&gt;</b>	n/a	Relations tab	x	x	<a href="#">ProjRelation</a>		8 The list of relations in which the related project record is on the left side of the relation, and the current project record is on the right side of the relation.  --> Links to specific relation information.  In Object Navigator, this attribute links to an intermediary table <b>Relation Type</b> that allows you to filter the	x	

							relations by relation type, if desired.		
							<p><b>Note:</b> The distinction between right and left project in Team Connect 3.4 is determined by a 3.4-specific attribute named <b>relationDirection</b>.</p>		
mainAssignee-->	MAIN_ASSIGNEE_ID (NUMBER)			x	<a href="#">JProjAssignee</a>	object	Links to the information about the main assignee only, such as the main assignee's user account or the assigned on date.	x	x
modifiedBy-->	MODIFIED_BY (DATE)	Modified By (Security block)	x	x	<a href="#">YUser</a>	object	8 User who last modified the record. --> Links to specific		



							information about the user account.		
modifiedOn	MODIFIED_ON (DATE)	Modified On (Security block)	x			date	Date the record was last modified.		
name	NAME (VARCHAR2) (250)	Name (General block)	x			string (250)	The name of the record.		x
nameUpper	NAME_UPPER (VARCHAR2) (250)		x			string (250)	The name of the record in uppercase characters.		
idNumber Before TeamConnect 3.3 SP2: numberString	NUMBER_STRING (VARCHAR2) (50)	Number (General block)	x			string (50)	The record number.		x
numberStringUpper	NUMBER_STRING_UPPER (VARCHAR2) (50)		x			string (50)	The record number in uppercase characters.		
openedOn	OPENED_ON (DATE)	Opened On (General block)	x			date	Date the project was opened.		
parent-->	PARENT_ID		x	x	<a href="#">IProject</a> , if a parent exists	object	8 The parent project of the current project.	x	x

	(NUMBER )						--> Links to specific information about the project record.		
phaseList-->  Before TeamConnect 3.3 SP2: phaseChanges-->	n/a			x	<a href="#">JProjPhase</a>		Links to the table that contains information for all phases for the project that have taken place.		
primaryKey	PRIMARY_KEY (NUMBER )		x			int	The unique ID of the project record.		
relations-->  Before TeamConnect 3.3 SP2: rightRelationList-->	n/a	Relations tab	x	x	<a href="#">ProjRelation</a>		8 The list of relations in which the related project record is on the right side of the relation, and the current project record is on the left side of the relation.  --> Links to specific relation information.  In Object Navigator, this attribute links to an intermediary table <b>Relation Type</b> that allows you to filter the relations by	x	

							<p>relation type, if desired.</p> <p><b>Note:</b> <i>The distinction between right and left project in Team Connect 3.4 is determined by a 3.4-specific attribute named <b>relationDirection</b>.</i></p>		
<b>securityTypeID</b>	<b>SECURITY_TYPE_ID</b>  (NUMBER)	<b>Public/Private</b> (Security block)	<b>x</b>			Enum	<p>Specifies whether the project is public or private as follows:</p> <p>0 - Public (PUBLIC)</p> <p>2 - Private (PRIVATE)</p>		
<b>taskList--&gt;</b>	n/a	<b>Tasks</b> tab	<b>x</b>	<b>x</b>	<a href="#">Task</a>		<p>8 The list of tasks that are related to the project.</p> <p>--&gt; Links to specific task</p>	<b>x</b>	

							information for tasks that are related to the project.  In Object Navigator, links to a table named <b>Category</b> that allows you to filter the tasks based on a category, if desired.		
<b>treeKey</b>	<b>TREE_KEY</b> <b>(VARCHAR2) (700)</b>		<b>x</b>			string (700)	The chain of primary keys that identifies the parent tree of the project. Used for searching and reporting purposes only.		
<b>userAccess List--&gt;</b>	n/a			<b>x</b>	<a href="#">EProjUserAccess</a>		Links to the user security information entered through the Security block of the record.	<b>x</b>	
<b>version</b>	<b>VERSION</b> <b>(NUMBER)</b>		<b>x</b>			int	Indicates how many times the project record has been updated.		

### 3.1.27.2 JProjAssignee

JProjAssignee contains information about the assignees in a project record, such as the assigned user, date assigned, project to which the assignee belongs, and assignee's role (such as adjuster). This table represents the **Assignees** block on the **Assignees** tab of a project record.

**Object:** JProjAssignee (J\_PROJ\_ASSIGNEE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
assigned On	ASSIGNED_ON (DATE)	Assigned On (Assignees block)	x			date	The date the user was assigned to the project.		
isActive	IS_ACTIVE (NUMBER)	Status (Assignees block)	x			boolean	Indicates whether the assignee is active.  1 - is Active 0 - not Active		x
owner-->	OWNER_ID (NUMBER)		x	x	<a href="#">TProject</a>	object	8 The project to which the assignee belongs.  --> Links to specific information about the project record.		x
primary Key	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the assignee.		
relationType--> Before TeamConnect 3.3 SP2: type-->	TYPE_ID (NUMBER)	Role (Assignees block)	x	x	<a href="#">LProjAssigneeType</a>	object	8 The type (or role) of the assignee.  --> Links to the available assignee types.		x
unassignedOn	UNASSIGNED_ON	Unassigned On	x			date	The date the user was un-		

	N (DATE)	(Assignees block)					assigned.		
user-->	USER_ID (NUMBER)	Assignee (Assignees block)	x	x	<a href="#">YUser</a>	object	8 The user who is an assignee. --> Links to specific information about the user account.		x
version	VERSION (NUMBER)		x			int	Indicates how many times the assignee has been updated.		

## 3.1.27.2.1 LProjAssigneeType

LProjAssigneeType contains information that defines project assignee roles. This information includes the name of the role, tree position, and custom object definition to which it belongs. This table represents the **Assignee Role Type** system lookup table, which appears on the **Assignee Roles** tab of custom object definitions.

Object: LProjAssigneeType (L\_PROJ\_ASSIGNEE\_TYPE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
embedded Projects--> Before TeamConnect 3.3 SP2: childList-->	n/a		x	x	<a href="#">LProjAssigneeType</a>		8 The list of all assignee roles that are child roles to the currently selected role.  --> Links to specific information about assignee roles that are		

							defined for the project object.		
<b>displayOrder</b>	<b>DISPLAY_ORDER</b> (NUMBER)	Order (Assignee Roles tab)	x				int	Indicates the order of this assignee role in the list of assignee roles in the GUI.	
<b>name</b>	<b>NAME</b> (VARCHAR2) (50)	Role Name (Assignee Roles tab)	x				string (50)	The name of the assignee role.	
<b>parent--&gt;</b>	<b>PARENT_ID</b> (NAME)		x	x	<a href="#">LProjAssigneeType</a>	object	8 The parent assignee role of the currently selected role.  --> Links to specific information about assignee roles that are defined for the project object.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x				int	The unique ID of the assignee role.	
<b>treePosition</b>	<b>TREE_POSITION</b> (VARCHAR2) (250)	Tree Position (Assignee Roles tab)	x				string (250)	Indicates the tree position for the assignee role.	
<b>isActive</b>	<b>IS_ACTIVE</b> (NUMBER)	Status (Assignees block)	x				boolean	Indicates whether the assignee is active.	x

							1 - is Active 0 - not Active		
version	VERSION (NUMBER)		x			int	Indicates how many times the assignee role has been updated.		

### 3.1.27.3 JProjPhase

JProjPhase contains information about the phases that have occurred in a project record. This includes the phase type (such as open or close), duration, transition date, and user who made the transition. This table represents the Phase History block on the Phases tab of a project record.

Object: JProjPhase (J\_PROJ\_PHASE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wizard
owner-->	PROJECT_ID (NUMBER)		x	x	<a href="#">TProject</a>	object	8 The project for which the phase has been selected.  --> Links to specific information about the project record.		
phaseDuration	PHASE_DURATION (NUMBER)	Duration (Phases block in project record)	x			int	The duration of the phase.		



<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	The unique ID of the phase.		
<b>transitionedBy--&gt;</b>	<b>TRANSITIONED_BY_ID</b> (NUMBER)	<b>Set By</b> (Phases block in project record)	x	x	<a href="#">YUser</a>	object	8 The user who changed the project to this phase.  --> Links to specific information about the user account.		
<b>transitionedOn</b>	<b>TRANSITIONED_ON</b> (DATE)	<b>Set On</b> (Phases block in project record)	x			date	The date on which the phase was selected for the project (or, the date that the phase was changed).		
<b>relationType--&gt;</b> <b>Before TeamConnect 3.3 SP2: type--&gt;</b>	<b>TYPE_ID</b> (NUMBER)	<b>Phase</b> (Phases block in project record)	x	x	<a href="#">WObjdPhaseType</a>	object	8 The phase type that was selected for the record.  --> Links to specific information about phases defined for the project object.		
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Indicates the number of times the phase has been updated.		

### 3.1.27.3.1 WObjdPhaseType

WObjdPhaseType contains the information that defines project phase types. This includes the name of the phase and the unique code. This table also links to lists of phase transitions for which the phase type is the beginning phase or resulting phase. This table represents the Phases block on the Phases tab of custom object definitions.

**Object:** WObjdPhaseType (W\_OBJD\_PHASE\_TYPE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
displayOrder	DISPLAY_ORDER (NUMBER)	Order (Phases tab in Object Definition)	x			int	The order in which the phase appears in the list of phases in the <b>Phase Transitions</b> tab.		
fromPhaseTransitionList-->	n/a		x	x	<a href="#">WObjdPhaseTransition</a>		8 The list of all phase transitions in which this phase is the phase from which the transition begins.  --> Links to the specific information for phase transitions in which this phase is the phase from which the transition takes place.		
name	NAME (VARCHAR2) (250)	Phase Name (Phases tab in	x			string (250)	The name of the phase.		

		Object Definition)							
<b>primaryKey</b>	<b>PRIMARY_KEY (NUMBER)</b>		x			int	The unique ID of the phase.		
<b>toPhaseTransitionList--&gt;</b>	n/a		x	x	<a href="#">WObjdPhaseTransition</a>		8 The list of all phase transitions in which this phase is the phase to which the transition goes.  --> Links to the specific information for phase transitions in which this phase is the phase to which the transition goes.		
<b>uniqueCode</b>	<b>UNIQUE_CODE (VARCHAR2) (4)</b>	<b>Unique Code (Phases tab in Object Definition)</b>	x			string (4)	The unique four-character alphanumeric code that is created when the phase is defined.		
<b>version</b>	<b>VERSION (NUMBER)</b>		x			int	Indicates the number of times the phase has been updated.		

## 3.1.27.3.2 WObjdPhaseTransition

WObjdPhaseTransition contains information about phase transitions, such as the name and the starting and ending phase. This table represents the Phase Transitions block on the Phase Transitions tab of custom object definitions.

Object: WObjdPhaseTransition (W\_OBJD\_PHASE\_TRANSITION)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
createdBy->	CREATE_D_BY_ID (NUMBER)		x	x	<a href="#">YUser</a>	object	8 The user who created the phase transition.  --> Links to specific information about the user account.		
createdOn	CREATE_D_ON (DATE)		x			date	The date the phase transition was defined.		
leftProject Phase-->	LEFT_PROJECT_PHASE_ID (NUMBER)	From Phase (Phase Transitions tab)	x	x	<a href="#">WObjdPhaseType</a>	object	8 The phase from which the phase transition begins.  --> Links to specific information about the phase definition.		
modifiedBy-->	MODIFIED_BY_ID (NUMBER)		x	x	<a href="#">YUser</a>	object	8 The user who last modified the phase transition.  --> Links to specific information about the user account.		

<b>modifiedOn</b>	<b>MODIFIED_ON</b> (DATE)		x			date	The date the phase transition was last modified.		
<b>name</b>	<b>NAME</b> (VARCHAR2) (250)	<b>Name</b>	x			string (250)	The name of the phase transition.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	The unique ID of the phase transition.		
<b>rightProjectPhase--&gt;</b>	<b>RIGHT_PROJECT_PHASE_ID</b> (NUMBER)	<b>To Phase</b> (Phase Transitions tab)	x	x	<a href="#">WObjdPhaseType</a>	object	8 The phase at which the phase transition ends.  --> Links to specific information about the phase definition.		
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Indicates the number of times the phase transition has been updated.		

### 3.1.27.4 JProjRelation

JProjRelation contains information about the project relations in a project record.

This information includes the project on the left and right sides of the relationship and the type of relationship between the two projects (such as related). This table represents the Relations block on the Relations tab of a project record.

Object: ProjRelation (J\_PROJ\_RELATION)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz

leftProject-->	LEFT_PROJECT_ID (NUMBER)	Project (left side, Relations block)		x	<a href="#">TProject</a>	object	The project on the left side of the relationship.		x
primary Key	PRIMARY_KEY (NUMBER)		x			int	The unique ID of this item.		
rightProject-->	RIGHT_PROJECT_ID (NUMBER)	Project (right side, Relations block)		x	<a href="#">TProject</a>	object	The project on the right side of the relationship.		x
relation Type--> Before TeamConnect 3.3 SP2: type-->	TYPE_ID (NUMBER)	Relation (Relations block)	x	x	<a href="#">LProjRelationType</a>	object	8 The type of relationship between the two projects.  --> Links to specific definition information for the relation.		x
version	VERSION (NUMBER)		x			int	Indicates the number of times the item has been updated.		

## 3.1.27.4.1 LProjRelationType

LProjRelationType contains information about the project relation types defined for custom objects, such as the name of the relation and the tree position. This table represents the **Project Relation Type** system lookup table.

Object: LProjRelationType (L\_PROJ\_RELATION\_TYPE)

Attribute	Database column name	Field in UI	End of	Bridge	Links to	Data	Comments	Commonly used in:

			path		object table:	type		Rules	
childList-->	n/a							Do not apply. These lookup items cannot have a hierarchical structure.	
parent->	PARENT_ID (NUMBER)								
display Order	DISPLAY_ORDER (NUMBER)	Order (System Lookup Tables, Project Relation Type)	x			int	Indicates the order in which this relation appears in the <b>Type of relationship</b> list in the <b>Relations</b> block of a project.		
name	NAME (VARCHAR2) (50)	Item Name (System Lookup Tables, Project Relation Type)	x			string (50)	The name of the relationship type.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of this item.		
treePosition	TREE_POSITION (VARCHAR2) (250)	Tree Position (System Lookup Tables, Project Relation Type)	x			string (250)	The unique four-character alphanumeric code that is created when the relation is defined.		
version	VERSION		x			int	Indicates the number of times		

	(NUMBER)						the item has been updated.		
--	----------	--	--	--	--	--	----------------------------	--	--

### 3.1.27.5 EProjDetail

EProjDetail contains all categories added to project records. Each time a category is added to a project record, an entry is made in this table in the database.

In Object Navigator, a list of all categories that have been defined for the current object definition appears. This list is labeled Category. From each category, you can traverse to a table called Detail Fields, which displays all custom fields belonging to the category you selected.

For more information about how custom field values are stored in the database, see Object Model: Custom Fields.

Object: EProjDetail (E\_PROJ\_DETAIL)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
category-->	CATEGORY_ID (NUMBER)	Category	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 The category definition to which this category, which has been added to a record, belongs.  --> Links to specific information about the category.		
isManual	IS_MANUAL (NUMBER)		x			boolean	Specifies whether the category is added manually or automatically to the project. Categories are added automatically when they are the parent of the category that is added manually.		



							0 - Added Automatically 1 - Added Manually		
<b>owner</b> -->	<b>ENTERPRISE_OBJECT_ID</b>  (NUMBER)		x	x	<a href="#">TProject</a>	object	8 The project to which the category belongs.  --> Links to the information about the project.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b>  (NUMBER)		x			int	The unique ID of the project category.		
<b>version</b>	<b>VERSION</b>  (NUMBER)		x			int	Indicates how many times the category was updated.		

### 3.1.27.6 EProjUserAccess

EProjUserAccess contains the user access information that is set in the Security block of each project record. For example, it includes whether each user that is listed in the Security block is allowed or denied the Read, Update, Delete, and Perm(ission) rights.

Object: EProjUserAccess (E\_PROJ\_USER\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
<b>isRead</b>	<b>IS_READ</b> (NUMBER)	Read check box	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).		

							This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isUpdate</b>	<b>IS_UPDATE</b> (NUMBER)	<b>Update</b> checkbox	<b>x</b>			int	Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isDelete</b>	<b>IS_DELETE</b> (NUMBER)	<b>Delete</b> checkbox	<b>x</b>			int	Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isPerm</b>	<b>IS_PERM</b> (NUMBER)	<b>Perm</b> checkbox	<b>x</b>			int	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		

<b>allowDenyID</b>	<b>ALLOW_DENY_ID</b> (CHAR) (1)	<b>Option</b>	<b>x</b>			<b>Enum</b>	Indicates whether the user is allowed or denied access to the record.  a - Allow (ALLOW) d - Deny (DENY)		
<b>isManual</b>	<b>IS_MANUAL</b> (NUMBER)		<b>x</b>			<b>boolean</b>	Indicates whether the user rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
<b>owner--&gt;</b>	<b>ENTERPRISE_OBJECT_ID</b> (NUMBER)		<b>x</b>	<b>x</b>	<a href="#">TProject</a>	<b>object</b>	8 The project to which this user access setting belongs.  --> Links to specific information about the project record.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		<b>x</b>			<b>int</b>	Unique ID for the user access rights.		
<b>user--&gt;</b>	<b>USER</b> (VARCHAR2) (250)	<b>List</b>	<b>x</b>	<b>x</b>	<a href="#">YUser</a>	<b>object</b>	8 The user the access rights pertain to.  --> Links to specific information about the user account.		
<b>version</b>	<b>VERSION</b> (NUMBER)		<b>x</b>			<b>int</b>	Indicates how many times the user access rights have been updated.		

### 3.1.27.7 EProjGroupAccess

EProjGroupAccess contains the group access information that is set in the **Security** block of each project record. For example, it includes whether each group that is listed in the **Group** field of the **Group Rights** block on the **Security** tab is allowed or denied the **Read**, **Update**, **Delete**, and **Perm(ission)** rights.

Object: EProjGroupAccess (E\_PROJ\_GROUP\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table :	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
isRead	IS_READ (NUMBER)	Read check box	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isUpdate	IS_UPDATE (NUMBER)	Update check box	x			int	Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isDelete	IS_DELETE (NUMBER)	Delete check box	x			int	Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed		

							or denied. (See allowDenyIID.)		
<b>isPerm</b>	<b>IS_PERM</b> <b>(NUMBER)</b>	<b>Perm</b> checkbox	<b>x</b>			int	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>allowDenyIID</b>	<b>ALLOW_DENY_IID</b> <b>(CHAR)</b> <b>(1)</b>		<b>x</b>			Enum	Indicates whether the group is allowed or denied access to the record.  a - Allow (ALLOW) d - Deny (DENY)		
<b>group--&gt;</b>	<b>GROUP_ID</b> <b>(NUMBER)</b>		<b>x</b>	<b>x</b>	<a href="#">YGro up</a>	object	8 The group that the access rights pertain to.  --> Links to specific information about the group account.		
<b>isManual</b>	<b>IS_MANUAL</b> <b>(NUMBER)</b>		<b>x</b>			boolean	Indicates whether the group rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
<b>owner--&gt;</b>	<b>ENTERPRISE_OBJECT_ID</b>		<b>x</b>	<b>x</b>	<a href="#">TProject</a>	object	8 The project to which this group access setting belongs.		

	(NUMBER)						--> Links to specific information about the project record.		
primary Key	PRIMARY_KEY (NUMBER)		x			int	The unique ID of this item.		
version	VERSION (NUMBER)		x			int	Indicates the number of times the item has been updated.		

### 3.1.27.8 TInvolved

TInvolved contains the general information of all involved party records. These records include the contact who is the involved party, project the involved record belongs to, default role, and users who created and modified each record.

Object: TInvolved (T\_INVOLVED)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wiz
activeOn	ACTIVE_ON (DATE)		x			date	The date on which the involved party became active.		
application-->	APPLICATION_ID (NUMBER)		x	x	<a href="#">WObjdProjectInfo</a>	int	8 The Custom Object definition to which this involved belongs.  --> Links to the table that contains information that		

							is specific to project object definitions (such as phase definitions, object autonumbering, and object naming pattern) and indirectly relates the involved to the involved object definition.		
contact-->	<b>CONTACT_ID</b> (NUMBER)	<b>Contact</b> (General block)	x	x	<a href="#">TContact</a>	object	8 The contact who is selected as the involved party for this involved record.  --> Links to specific information about the contact record.	x	x
createdBy-->	<b>CREATED_BY_ID</b> (NUMBER)	<b>Created By</b> (Security block)	x	x	<a href="#">YUser</a>	object	8 User who created the involved record.  --> Links to specific information about the user account.	x	x
createdOn	<b>CREATED_ON</b> (DATE)	<b>Created On</b> (Security block)	x			date	Date the involved record was created.	x	x
createdOnBehalfOf	<b>CREATED_ON_BEHALF_OF_ID</b>			x	<a href="#">TContact</a>	object	When not null, it indicates that the record was created through an e-billing system or other	X	

							external application. The contact referenced will often be a vendor in TeamConnect.		
defaultCategory-->	DEFAULT_CATEGORY_ID (NUMBER)	Indicated by a blue diamond (Categories block)	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 The default role for the involved.  --> Links to specific information about the role.  Note: Involved roles are the equivalent of categories for other main objects. This is why their definition information is contained in WObjdCategory .	x	
categories-->  Before TeamConnect 3.3 SP2: detailList->	n/a		x	x	<a href="#">EInvDetail</a>  In Object Navigator, links to Category list which appears in the UI only.		8 The list of categories added to the current object.  --> In the object model, the added categories link to the values that have been added for custom fields, according to field type.  --> In Object Navigator, this attribute is enhanced. It links to a list of	x	x



							all categories defined for the object definition, rather than those added to the current object. From there, you can select a category and traverse to the list of custom fields that belong to it.		
document Folder-->	DOCUMENT_FOLDER_ID (NUMBER)			x	<a href="#">TDocument</a>	object	Links to specific information about the document folder.		
groupAccessList-->	n/a			x	<a href="#">EInvGroupAccess</a>		Links to the group security information entered through the Security block of the record.		
historyList-->	n/a	History tab	x	x	<a href="#">THistory</a>		8 The list of histories that are related to the involved record.  --> Links to specific history information for all of the involved's history records.	x	
inactiveOn	INACTIVE_ON (DATE)		x			date	The date on which the involved party became inactive.		

isActive	IS_ACTIVE (DATE)		x			int	1 = Involved party is active. 0 = Involved party is inactive.		
relations-->  Before TeamConnect 3.3 SP2: leftRelationList-->	n/a	Relations tab	x	x	<a href="#">InvRelationAttrMap</a>		8 The list of relations in which the related involved record is on the left side of the relation, and the current involved record is on the right side of the relation.  --> Links to specific relation information.  In Object Navigator, this attribute links to an intermediary table <b>RelationType</b> that allows you to filter the relations by relation type, if desired.  <b>Note:</b> The <i>distinction between right and left involved in TeamConnect 3.4 is determined by a 3.4-</i>		

								<i>specific attribute named relationDirect ion.</i>		
modified By-->	MODIFIED_BY_ID (NUMBER)		x	x	<a href="#">YUser</a>	object	8	The user who last modified the record.  --> Links to specific information about the user account.		
modified On	MODIFIED_ON (DATE)		x			date		The date on which the record was last modified.		
relations-->  Before TeamConnect 3.3 SP2: rightRelationList-->	n/a	Relations tab	x	x	<a href="#">InvRelationAttrMap</a>		8	The list of relations in which the related involved record is on the right side of the relation, and the current involved record is on the left side of the relation.  --> Links to specific relation information.  In Object Navigator, this attribute links to an intermediary table <b>RelationType</b> that allows you to		

							filter the relations by relation type, if desired.  <b>Note:</b> <i>The distinction between right and left involved in TeamConnect 3.4 is determined by a 3.4-specific attribute named <b>relationDirection</b>.</i>		
note-->	NOTE_ID (NUMBER)	Notes (General block)		x	<a href="#">JNote</a>	object	8 The notes entered for the record.  --> Links to the table that contains the text entered into the <b>Notes</b> field of the record.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the involved record.		
project-->	PROJECT_ID		x	x	<a href="#">TProject</a>	object	8 The project to which this	x	x

	(NUMBER)						involved record belongs. --> Links to specific information for the project record.		
securityTypeID	SECURITY_TYPE_ID (NUMBER)	Public/Private (Security block)	x			Enum	Specifies whether the involved record is public or private.  0 - Public (PUBLIC)  2 - Private (PRIVATE)		
userAccessList-->	n/a			x	<a href="#">EInvUserAccess</a>		Links to the user security information entered through the Security block of the record.		
version	VERSION (NUMBER)		x			int	Indicates how many times the involved record has been updated.		

3.1.27.8.1 JInvRelation

JInvRelation contains the information of all relations between involved parties, such as the type of relationship and the involved contacts on the left and right sides of the relation.

Object: InvRelationAttrMap (J\_INVL\_RELATION)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp /

									Wi z
leftInvolved-->	LEFT_INVOLVED_ID (NUMBER)	Involved Contact (left side, Relations block)		x	<a href="#">TInvolved</a>	object	The involved that is on the left side of the relationship.		x
rightInvolved-->	RIGHT_INVOLVED_ID (NUMBER)	Involved Contact (right side, Relations block)		x	<a href="#">TInvolved</a>	object	The involved that is on the right side of the relationship.		x
relationType--> Before TeamConnect 3.3 SP2: type-->	TYPE_ID (NUMBER)	Relationship (Relations block)	x	x	<a href="#">LContRelationType</a>	object	8 The type of the relationship. --> Links to definition information for the relation type.		x
version	VERSION (NUMBER)		x			int	Indicates how many times the involved relation item was updated.		

### 3.1.27.8.2 EInvDetail

EInvDetail contains all categories added to involved records. Each time a category is added to an involved record, an entry is made in this table in the database.

In Object Navigator, a list of all categories that have been defined for the current object definition appears. This list is labeled **Category**. From each category, you can traverse to a table called **Detail Fields**, which displays all custom fields belonging to the category you selected.

For more information about how custom field values are stored in the database, see Object Model: Custom Fields.

Object: EInvDetail (E\_INVL\_DETAIL)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wiz
category-->	CATEGORY_ID (NUMBER)	Category	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 The category definition to which this category, which has been added to a record, belongs.  --> Links to specific information about the category.		
owner-->	ENTERPRISE_OBJECT_ID (NUMBER)			x	<a href="#">Involved</a>	object	Links to the involved record to which the category has been added.		
isManual	IS_MANUAL (NUMBER)		x			boolean	Specifies whether the category is added manually or automatically to the involved. Categories are added automatically when they are the parent of the category that is added manually.  0 - Added Automatically  1 - Added Manually		

<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	The unique ID of the involved category record.		
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Indicates how many times the added category was updated.		

## 3.1.27.8.3 EInvUserAccess

EInvUserAccess contains the user access information that is set in the **Security** block of each involved record. For example, it includes whether each user that is listed in the **Security** block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm(ission)** rights.

Object: EInvUserAccess (E\_INVL\_USER\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
isRead	IS_READ (NUMBER)	Read checkbox	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isUpdate	IS_UPDATE (NUMBER)	Update checkbox	x			int	Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).		



							This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isDelete</b>	<b>IS_DELETE</b> (NUMBER)	<b>Delete</b> checkbox	<b>x</b>			int	Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isPerm</b>	<b>IS_PERM</b> (NUMBER)	<b>Perm</b> checkbox	<b>x</b>			int	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>allowDenyIID</b>	<b>ALLOW_DENY_IID</b> (CHAR) (1)	<b>Option</b>	<b>x</b>			Enum	Indicates whether the user is allowed or denied access to the record.  a - Allow (ALLOW) d - Deny (DENY)		
<b>isManual</b>	<b>IS_MANUAL</b> (NUMBER)		<b>x</b>			boolean	Indicates whether the user rights were assigned manually		

							through the Security block of the record or automatically by the system.  0 - Assigned manually  1 - Assigned automatically		
owner->	ENTERPRISE_OBJECT_ID (NUMBER)		x	x	<a href="#">TInvolved</a>	object	8 The involved record to which this user access setting belongs.  --> Links to specific information about the involved record.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	Unique ID for the user access rights.		
user-->	USER_ID (NUMBER)	List	x	x	<a href="#">YUser</a>	object	8 The user the access rights pertain to.  --> Links to specific information about the user account.		
version	VERSION (NUMBER)		x			int	Indicates how many times the user access rights have been updated.		

## 3.1.27.8.4 EnvGroupAccess

EnvGroupAccess contains the group access information that is set in the **Security** block of each involved record. For example, it includes whether each group that is listed in the **Security** block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm**(ission) rights.

Object: EnvGroupAccess (E\_INVL\_GROUP\_ACCESS)

Attribute	Database column name	Field in UI	End of	Bridge	Links to object	Data type	Comments	Commonly used in:

			pat h		ct tabl e:			Rul es	
isRead	IS_READ (NUMBER)	Read check box	x			int	<p>Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
isUpdate	IS_UPDATE (NUMBER)	Update check box	x			int	<p>Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
isDelete	IS_DELETE (NUMBER)	Delete check box	x			int	<p>Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are</p>		

							allowed or denied. (See allowDenyIID.)		
isPerm	IS_PERM (NUMBER)	Per m chec k box	x			int	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
allowDenyIID	ALLOW_DENY_IID (CHAR) (1)	Opti on	x			Enum	Indicates whether the group is allowed or denied access to the record.  a - Allow (ALLOW) d - Deny (DENY)		
group-->	GROUP_ID (NUMBER)	List	x	x	<a href="#">YGroup</a>	object	8 The group that the access rights pertain to.  --> Links to specific information about the group account.		
isManual	IS_MANUAL (NUMBER)		x			boolean	Indicates whether the group access rights were assigned manually through the Security block of the record or automatically by the system.		

							0 - Assigned manually 1 - Assigned automatically		
owner-->	ENTERPRISE_OBJECT_ID (NUMBER)		x	x	<a href="#">Involved</a>	object	8 The involved record to which this group access setting belongs.  --> Links to specific information about the involved record.		
primary Key	PRIMARY_KEY (NUMBER)		x			int	Unique ID for the group access rights.		
version	VERSION (NUMBER)		x			int	Indicates how many times the group access rights have been updated.		

### 3.1.27.9 TMilestone

TMilestone contains the general information for all milestone records.

This information includes the projected, extension, and completion dates, milestone description, the project the milestone belongs to, as well as the users who created and modified the milestone record. This table also links to the list of custom fields created for the milestone categories and other related information contained in other tables.

Object: TMilestone (T\_MILESTONE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
application-->	APPLICATION_ID		x	x	<a href="#">WObjdProjectInfo</a>	object	8 The Custom Object		

	(NUMBER)						definition to which this milestone belongs.  --> Links to the table that contains information that is not included in ZObjectDefinition because it is only necessary for Custom Object Definitions (such as phase definitions, object autonumbering, and object naming pattern).		
completionDate	COMPLETION_DATE (DATE)	Completion Date (General block)	x			date	The date the milestone was completed.		x
completionEventID	COMPLETION_EVENT_ID (NUMBER)	phase is Started/Completed (General block)	x			Enum	Indicates whether the milestone should be automatically marked as completed when the related project phase (selected in the phase--> attribute) is		

							started or completed. 0 - Started (STARTED) 1 - Completed (COMPLETE D)		
createdBy-->	CREATED_BY_ID (NUMBER)	Create By (Security block)	x	x	<a href="#">YUser</a>	object	8 User who created the milestone record. --> Links to specific information about the user account.	x	x
createdOn	CREATED_ON (DATE)	Create On (Security block)	x			date	Date the milestone record was created.	x	x
createdOnBehalfOf	CREATED_ON_BEHALF_OF_ID			x	<a href="#">TContact</a>	object	When not null, it indicates that the record was created through an e-billing system or other external application. The contact referenced will often be a vendor in TeamConnect.	X	

defaultCategory-->	DEFAULT_CATEGORY_ID (NUMBER)	Indicated by a blue diamond (Categories block)	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 The default category for the milestone.  --> Links to specific information about the category.		x
detailList->	n/a		x	x	<a href="#">EMileDetail</a>  In Object Navigator, links to Category list which is displayed in the UI only.		8 The list of categories added to the current object.  --> In the object model, the added categories link to the values that have been added for custom fields, according to field type.  --> In Object Navigator, this attribute is enhanced. It links to a list of all categories defined for the object definition, rather than those added to the current object. From there, you can select a category and traverse to the list of custom fields		



							that belong to it.		
documentFolder-->	DOCUMENT_FOLDER_ID (NUMBER)			x	<a href="#">TDocument</a>	object	Links to specific information about the document folder.		
extensionDate	EXTENSION_DATE (DATE)	Extension (General block)	x			date	The extended date of the milestone.	x	
groupAccessList-->	n/a			x	<a href="#">EInvGroupAccess</a>		Links to the group security information entered through the Security block of the record.		
historyList-->	n/a	History tab	x	x	<a href="#">THistory</a>		8 The list of histories that are related to the milestone record.  --> Links to specific history information for all of the milestone's history records.		
isPhaseUsed	IS_PHASE_USED (NUMBER)	Mark this milestone complete	x			boolean	Indicates whether the milestone record is set to be marked as complete		

		when (checkbox - General block)					<p>automatically when the related project record reaches a certain phase.</p> <p>0 - Do not use phase to mark milestone as complete</p> <p>1 - Use phase to mark milestone as complete</p>		
<b>milestone Date</b>	<b>MILESTONE_DATE (DATE)</b>		<b>x</b>			<b>date</b>	<p>This contains one of the last of the three dates for the milestone record in the following order.</p> <p>Projected Date</p> <p>Extended Date</p> <p>Completion Date</p> <p>For example, if the Projected Date and Extended Date have values, then the Milestone Date will be populated with the</p>		

							same date as the Extended Date.		
modified By-->	MODIFIED_BY_ID (NUMBER)		x	x	<a href="#">YUser</a>	object	8 The user who last modified the record.  --> Links to specific information about the user account.		
modified On	MODIFIED_ON (DATE)		x			date	The date on which the record was last modified.		
note-->	NOTE_ID (NUMBER)			x	<a href="#">JNote</a>	object	Links to the table that contains the text entered into the <b>Notes</b> field of the record.		
phase-->	PHASE_ID (NUMBER)		x	x	<a href="#">WObjdPhaseType</a>	object	8 The phase of the related project record at which the milestone will be automatically marked as complete.  --> Links to definition information for the phase.	x	
primaryKey	PRIMARY_KEY		x			int	The unique ID of the		

	(NUMBER)						milestone record.		
project-->	PROJECT_ID (NUMBER)		x	x	<a href="#">TProject</a>	object	8 The project to which this milestone belongs.  --> Links to specific information about the project record.		x
projected Date	PROJECTED_DATE (DATE)	Projected Date (General block)	x			date	The projected date of completion of the milestone.	x	x
securityTypeID	SECURITY_TYPE_ID (NUMBER)	Public / Private (Security block)	x			Enum	Specifies whether the milestone record is public or private.  0 - Public (PUBLIC)  2 - Private (PRIVATE)		
shortDescription	SHORT_DESCRIPTION (VARCHAR2) (250)	Subject (General block)	x			string (250)	The subject (or description) of the milestone record.		x
userAccessList-->	n/a			x	<a href="#">EInvUserAccess</a>		Links to the user security information entered through the Security		

							block of the record.		
version	VERSION (NUMBER)		x			int	Indicates how many times the milestone record has been updated.		

### 3.1.27.10 EMileDetail

EMileDetail contains all categories added to milestone records. Each time a category is added to a milestone record, an entry is made in this table in the database.

In Object Navigator, EMileDetail is not displayed when you traverse using the **detailList-->** bridge. Instead, a list of all categories that have been defined for the object definition is displayed. This list is labeled **Category**. From each category, you can traverse to a table called **Detail Fields**, which displays all custom fields belonging to the category you selected.

For more information about how custom field values are stored in the database, see Object Model: Custom Fields.

Object: EMileDetail (E\_MILE\_DETAIL)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
category-->	Category_ID (NUMBER)	Category	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 The category of the milestone with which the details are associated.  --> Links to specific information about the category.		
owner-->	ENTERPRISE_OBJECT_ID (NUMBER)			x	<a href="#">TMilestone</a>	object	The milestone with which the custom fields are associated.		

							Links to the information about the milestone record.		
isManual	IS_MANUAL (NUMBER)		x			boolean	Specifies whether the category is added manually or automatically to the milestone. Categories are added automatically when they are the parent of the category that is added manually.  0 - Added Automatically 1 - Added Manually		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the milestone category record.		
version	VERSION (NUMBER)		x			int	Indicates how many times the added category was updated.		

### 3.1.27.11 EMileUserAccess

EMileUserAccess contains the user access information that is set in the **Security** block of each milestone record.

For example, it includes whether each user that is listed in the **Security** block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm**(ission) rights.

Object: **EMileUserAccess (E\_MILE\_USER\_ACCESS)**

Attribute	Database column	Field in	End	Bridge	Links to	Data type	Comments	Commonly used in:
-----------	-----------------	----------	-----	--------	----------	-----------	----------	-------------------

	name	UI	of path	e	object table :			Rules	
isRead	IS_READ (NUMBER)	Read checkbox	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isUpdate	IS_UPDATE (NUMBER)	Update checkbox	x			int	Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isDelete	IS_DELETE (NUMBER)	Delete checkbox	x			int	Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		

<b>isPerm</b>	<b>IS_PERM (NUMBER)</b>	<b>Per m che ck box</b>	<b>x</b>			<b>int</b>	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>allowDenyIID</b>	<b>ALLOW_DENY_IID (CHAR) (1)</b>	<b>Opti on</b>	<b>x</b>			<b>Enum</b>	Indicates whether the user is allowed or denied access to the record.  a - Allow (ALLOW) d - Deny (DENY)		
<b>isManual</b>	<b>IS_MANUAL (NUMBER)</b>		<b>x</b>			<b>boolean</b>	Indicates whether the user rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
<b>owner--&gt;</b>	<b>ENTERPRISE_OBJECT_ID (NUMBER)</b>		<b>x</b>	<b>x</b>	<a href="#">TMilestone</a>	<b>object</b>	8 The milestone record to which this user access setting belongs.  --> Links to specific information about the milestone.		
<b>primaryKey</b>	<b>PRIMARY_KEY (NUMBER)</b>		<b>x</b>			<b>int</b>	Unique ID for the user access rights.		



user-->	<b>USER_ID</b> (NUMBER)	List	x	x	<a href="#">YUse</a> <a href="#">r</a>	object	8 The user the access rights pertain to.  --> Links to specific information about the user account.		
version	<b>VERSION</b> (NUMBER)		x			int	Indicates how many times the user access rights have been updated.		

### 3.1.27.12 EMileGroupAccess

EMileGroupAccess contains the group access information that is set in the Security block of each milestone record. For example, it includes whether each group that is listed in the Security block is allowed or denied the Read, Update, Delete, and Perm(ission) rights.

Object: EMileGroupAccess (E\_MILE\_GROUP\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table :	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
isRead	<b>IS_READ</b> (NUMBER)	Read checkbox	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isUpdate	<b>IS_UPDATE</b> (NUMBER)	Update checkbox	x			int	Specifies whether the Update operation has been selected for granting or denial. Value is 1		

							(selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isDelete</b>	<b>IS_DELETE</b> <b>(NUMBER)</b>	<b>Delete</b> <b>checkbox</b>	<b>x</b>			<b>int</b>	Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isPerm</b>	<b>IS_PERM</b> <b>(NUMBER)</b>	<b>Perm</b> <b>checkbox</b>	<b>x</b>			<b>int</b>	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>allowDenyIID</b>	<b>ALLOW_DENY_IID</b> <b>(CHAR) (1)</b>	<b>Option</b>	<b>x</b>			<b>Enum</b>	Indicates whether the group is allowed or denied access to the record.  a - Allow (ALLOW) d - Deny (DENY)		

<b>group--&gt;</b>	<b>GROUP_ID</b> (NUMBER)	List	x	x	<a href="#">YGroup</a>	object	8 The group that the access rights pertain to.  --> Links to specific information about the group account.		
<b>isManual</b>	<b>IS_MANUAL</b> (NUMBER)		x			boolean	Indicates whether the group access rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
<b>owner-&gt;</b>	<b>ENTERPRISE_OBJECT_ID</b> (NUMBER)		x	x	<a href="#">TMilestone</a>	object	8 The milestone record with which this group access setting is associated.  --> Links to specific information about the milestone record.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	Unique ID for the group access rights.		
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Indicates how many times the group access rights have been updated.		

### 3.1.28 Object Model: Contacts

The tables in this appendix provide information about the object model as it relates to contacts.

**Note:** Designer contact groups are referred to as Address Books in the end-user interface.

#### 3.1.28.1 TContact

TContact contains general information for contact records. This information includes the contact's name, birth date, default category, driver license number, title, Social Security number or Tax ID, the company the person works for, whether the contact is a person or a company, and so on. It also links to the lists of addresses, fax and phone numbers, rates, and skills.

However, the default address, default phone, default email, and so on are identified by specific attributes of TContact, rather than through the listing attributes (for example, **addressList-->** and **faxList-->**).

Object: TContact (T\_CONTACT)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wiz
<b>addresses--&gt;</b>  Before TeamConnect 3.3 SP2: <b>addressList--&gt;</b>	n/a		x	x	<a href="#">JContAddress</a>	object	8 The list of all addresses that are added to the contact.  --> Links to the join table that contains information for all addresses for the contact.  Select this attribute when you want to access items belonging to	x	

							<p>an address, such as city or state, or when you need to access address types.</p> <p>In Object Navigator, this attribute links to a table named <b>Address Type</b> that allows you to filter the addresses based on a type, if desired.</p>		
alias	ALIAS (VARCHAR2)(250)	Alias / Nickname (General block)	x			string	The contact's nickname (for individual contact and company contact).	x	x
birthdate Before TeamConnect 3.3 SP2: birthDate	BIRTH_DATE (DATE)	Date of Birth (General block)	x			date	The birth date of the contact. Time-zone-independent.	x	
company-->	COMPANY_ID (NUMBER)	Company (General block)	x	x	<a href="#">TContact</a>	int	8 The company associated with the contact, if the contact is a person.	x	x

							--> Links to specific information about the company contact record.		
createdBy-->	CREATED_BY_ID (NUMBER)	Created By (Security block)	x	x	<a href="#">YUser</a>	object	8 The user who created the contact record.  --> Links to specific information for the user account.	x	x
createdOn	CREATED_ON (DATE)	Created On (Security block)	x			date	Date the contact record was created.	x	x
createdOnBehalfOf	CREATED_ON_BEHALF_OF_ID			x	<a href="#">TContact</a>	object	When not null, it indicates that the record was created through an e-billing system or other external application. The contact referenced will often be a vendor in TeamConnect.	X	
currency-->	CURRENCY_ID	Currency		x	<a href="#">YCurrencyInfo</a>	object	8 The selected currency for		

	(NUMBER )	(Rates block )					the contact, which applies to the contact's rates.  --> Links to the definition information of the currency.		
<b>primaryAddress--&gt;</b>  <b>Before TeamConnect 3.3 SP2: defaultAddresses--&gt;</b>	<b>DEFAULT_ADDRESSES_ID</b>  (NUMBER )	Indicated by a blue diamond	x	x	<a href="#">JContAddress</a>	object	8 The selected default address for the contact.  --> Links to specific information entered for the default address.	x	x
<b>defaultCategory--&gt;</b>	<b>DEFAULT_CATEGORY_ID</b>  (NUMBER )	Indicated by a blue diamond (Categories block )	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 The selected default category for the contact.  --> Links to specific information for the default category.	x	x
<b>primaryEmailAddress--&gt;</b>  <b>Before TeamConnect 3.3 SP2: defaultEmail--&gt;</b>	<b>DEFAULT_EMAIL_ID</b>  (NUMBER )	Indicated by a blue diamond	x	x	<a href="#">JContEmail</a>	object	8 The selected default email address for the contact.  --> Links to specific information entered for the default email address.	x	x

<p><b>primaryFaxNumber--&gt;</b></p> <p><b>Before TeamConnect 3.3 SP2: defaultFax--&gt;</b></p>	<p><b>DEFAULT_FAX_ID</b></p> <p><b>(NUMBER)</b></p>	Indicated by a blue diamond	x	x	<a href="#">JContFax</a>	object	8 The selected default fax number for the contact.	x	x
<p><b>primaryInternetAddress--&gt;</b></p> <p><b>Before TeamConnect 3.3 SP2: defaultInternetAddress--&gt;</b></p>	<p><b>DEFAULT_ADDRESS_ID</b></p> <p><b>(NUMBER)</b></p>	Indicated by a blue diamond	x	x	<a href="#">JContInternetAddress</a>	object	8 The selected default internet address for the contact.	x	x
<p><b>primaryPhoneNumber--&gt;</b></p> <p><b>Before TeamConnect 3.3 SP2: defaultPhoneNumber--&gt;</b></p>	<p><b>DEFAULT_PHONE_ID</b></p> <p><b>(NUMBER)</b></p>	Indicated by a blue diamond	x	x	<a href="#">JContPhone</a>	object	8 The selected default phone number for the contact.	x	x
<p><b>defaultRate--&gt;</b></p> <p><b>Before TeamConnect 3.3 SP2: defaultRateValue</b></p>	<p><b>DEFAULT_RATE_VALUE</b></p> <p><b>(NUMBER)</b></p>	Default Rate Value (Rates)	x			decimal	The default rate value for the contact.	x	x



		block )							
<b>detailList--&gt;</b>	n/a		x	x	<a href="#">EContDetail</a>  In Object Navigator, links to Category list which appears in the UI only.		8 The list of categories added to the current object.  --> In the object model, the added categories link to the values that have been added for custom fields, according to field type.  --> In Object Navigator, this attribute is enhanced. It links to a list of all categories defined for the object definition, rather than those added to the current object. From there, you can select a category and traverse to the list of custom fields that belong to it.	x	x
<b>distributionList--&gt;</b>	n/a			x	<a href="#">JDistMember</a>		Links to the list of all members in a contact group		

							(Address Book). From there, you can access the information about a contact group, such as whether it is public or private.		
documentFolder-->	DOCUMENT_FOLDER_ID (NUMBER)	Documents block		x	<a href="#">TDocument</a>	object	Links to the document folder that belongs to the contact.		
driverLicense	DRIVER_LICENSE (VARCHAR2) (50)	State Driver's License (General block)	x			string (50)	The contact's driver license number.	x	
emailAddresses-->  Before TeamConnect 3.3 SP2: emailList-->	n/a		x	x	<a href="#">JContEmail</a>		8 The list of all email addresses that are added to the contact.  --> Links to the join table that contains information for all email addresses of the contact.  Select this attribute when you want to access email	x	

							<p>address types.</p> <p>In Object Navigator, this attribute links to a table named <b>Email Type</b> that allows you to filter the email addresses based on a type, if desired.</p>		
<p><b>faxNumbers--&gt;</b></p> <p><b>Before TeamConnect 3.3 SP2: faxList--&gt;</b></p>	n/a		x	x	<a href="#">JContFax</a>		<p>8 The list of all fax numbers that are added to the contact.</p> <p>--&gt; Links to the join table that contains information for all fax numbers of the contact.</p> <p>Select this attribute when you want to access fax number types.</p> <p>In Object Navigator, this attribute links to a table named <b>Fax Type</b> that allows you to filter the fax numbers based on a type, if desired.</p>	x	

<b>firstName</b>	<b>FIRST_NAME</b> (VARCHAR2) (50)	First (General block)	x			string (50)	The first name of the contact.		
<b>firstNameUpper</b>	<b>FIRST_NAME_UPPER</b> (VARCHAR2) (50)		x			string (50)	The contact's first name, in uppercase letters.		
<b>groupAccessList--&gt;</b>	n/a			x	<a href="#">EContGroupAccess</a>		Links to the group security information entered through the Security block of the record.		
<b>historyList--&gt;</b>	n/a	History tab	x	x	<a href="#">THistory</a>		8 The list of all histories that are related to the contact record.  --> Links to specific history information for all of the contact's history records.	x	
<b>internetAddresses--&gt;</b>  <b>Before TeamConnect 3.3 SP2: inetAddressList--&gt;</b>	n/a		x	x	<a href="#">JContInetAddress</a>		8 The list of all internet addresses that are added to the contact record.  --> Links to the join table	x	

							<p>that contains information for all internet addresses for the contact.</p> <p>Select when you want to access internet address types.</p> <p>In Object Navigator, this attribute links to a table named <b>Internet Address Type</b> that allows you to filter the internet addresses based on a type, if desired.</p>		
<p><b>relations--&gt;</b></p> <p><b>Before TeamConnect 3.3 SP2: leftRelationList--&gt;</b></p>	n/a	Relations tab	x	x	<a href="#">JContRelation</a>		<p>8 The list of relations in which the related contact record is on the left side of the relation, and the current contact record is on the right side of the relation.</p> <p>--&gt; Links to specific relation information.</p>	x	

							<p>In Object Navigator, this attribute links to an intermediary table</p> <p><b>Relation Type</b> that allows you to filter the relations based on the relation type, if desired.</p> <p><b>Note:</b> <i>The distinction between right and left relation in TeamConnect 3.4 is determined by a 3.4-specific attribute named <b>relationDirect</b>.</i></p>		
<b>middleName</b>	<b>MIDDLE_NAME</b>	Middle	<b>x</b>			string	The middle name of the		

	<b>(VARCHAR2) (50)</b>	(General block)				(50)	contact.		
<b>modifiedBy--&gt;</b>	<b>MODIFIED_BY</b> <b>(NUMBER)</b>	Modified By (Security block)	x	x	<a href="#">YUser</a>	object	8 The user who last modified the contact record.  --> Links to specific information for the user account.		
<b>modifiedOn</b>	<b>MODIFIED_ON</b> <b>(DATE)</b>	Modified On (Security block)	x			date	Date the contact was last modified.		
<b>lastName</b> <b>Before TeamConnect 3.3 SP2: name</b>	<b>NAME</b> <b>(VARCHAR2) (250)</b>	Last or Company Name (General block)	x			string (250)	The last name of the contact, if the contact is a person.  The name of the contact, if the contact is a company.		
<b>nameUpper</b>	<b>NAME_UPPER</b> <b>(VARCHAR2) (250)</b>		x			string (250)	The last name or company name of the contact, in uppercase letters.		
<b>note--&gt;</b>	<b>NOTE_ID</b> <b>(NUMBER)</b>	Notes		x	<a href="#">JNote</a>	object	Links to the table that contains the text entered		

		(General block)					into the <b>Notes</b> field of the record.		
<b>idNumber</b> Before TeamConnect 3.3 SP2: <b>numberString</b>	<b>NUMBER_STRING</b> (VARCHAR2) (50)	ID (General block)	x			string (50)	Alphanumeric text that identifies this contact as per your company's standards.		
<b>numberStringUpper</b>	<b>NUMBER_STRING_UPPER</b> (VARCHAR2) (50)		x			string (50)	The employee number in uppercase characters.		
<b>phoneNumbers--&gt;</b> Before TeamConnect 3.3 SP2: <b>phoneList--&gt;</b>	n/a		x	x	<a href="#">JContPhone</a>		<p>8 The list of phone numbers that are added to the contact record.</p> <p>--&gt; Links to the join table that contains information for all phone numbers of the contact.</p> <p>Select this attribute when you want to access phone number types.</p> <p>In Object Navigator, this attribute links to a table named <b>Phone Type</b> that allows</p>	x	



							you to filter the phone numbers based on a type, if desired.		
<b>prefix</b>	<b>PREFIX (VARCHAR2) (50)</b>	Prefix (General block)	x			string (50)	The prefix of the contact's name (such as Mr. or Mrs.)	x	
<b>primaryKey</b>	<b>PRIMARY_KEY (NUMBER)</b>		x			int	The unique ID of the contact record.		
<b>rates--&gt;</b> <b>Before TeamConnect 3.3 SP2:</b> <b>rateList--&gt;</b>	n/a	Rates tab	x	x	<a href="#">JContRate</a>		8 The list of rates that are added to the contact record.  --> Links to the join table that contains information for all rates of the contact.  Select this attribute when you want to access specific items related to contact rates, such as rate amount, effective from date, and effective to date.	x	

							In Object Navigator, this attribute links to a table named <b>Task Category</b> that allows you to filter the rates based on a task category, if desired.		
<b>relations--&gt;</b> <b>Before TeamConnect 3.3 SP2: rightRelation List--&gt;</b>	n/a	Relations tab	x	x	<a href="#">JContRelation</a>		8 The list of relations in which the related contact record is on the right side of the relation, and the current contact record is on the left side of the relation.  --> Links to specific relation information.  In Object Navigator, this attribute links to an intermediary table <b>Relation Type</b> that allows you to filter the relations based on the relation type, if desired.	x	

							<p><i>Note: The distinction between right and left relation in TeamConnect 3.4 is determined by a 3.4-specific attribute named relationDirection.</i></p>		
salutation*	<p><b>SALUTATION</b> <b>(VARCHAR2) (50)</b></p>	<p>Salutation (General block)</p>	x			string (50)	<p>Salutation will <b>not</b> appear in TeamConnect; this field remains for users upgrading from 2.x versions of TeamConnect and should be ignored by all 4.x users.</p>		

<b>securityTypeID</b>	<b>SECURITY_TYPE_ID</b> (NUMBER)	Security Type (Security block)	x			Enum	The security type of the contact record (either Public or Private).  0 - Public (PUBLIC)  2 - Private (PRIVATE)	x	
<b>skill--&gt;</b> <b>Before TeamConnect 3.3 SP2: skillList--&gt;</b>	n/a	Skills tab	x	x	<a href="#">JContSkill</a>		8 The list of skills that are added to the contact record.  --> Links to the join table that contains information for all skills of the contact.  Select this attribute when you want to access specific items related to skills, such as skill level and skill type.  In Object Navigator, this attribute links to a table named <b>Skill Type</b> that allows you to filter the skills based on a type, if desired.	x	

<b>socialSecurityNumber</b> <b>Before TeamConnect 3.3 SP2: ssOrTaxNumberString</b>	<b>SS_OR_TAX_NUMBER_STRING</b> <b>(VARCHAR2) (50)</b>	Social Security or Tax ID (General block)	<b>x</b>			string (50)	The Social Security number of the contact, if the contact is a person.  The Tax ID of the contact, if the contact is a company.	<b>x</b>	
<b>suffix</b>	<b>SUFFIX</b> <b>(VARCHAR2) (50)</b>	Suffix (General block)	<b>x</b>			string (50)	The suffix of the contact's name (such as Jr., Sr., or II)		
<b>territories--&gt;</b> <b>Before TeamConnect 3.3 SP2: territoryList--&gt;</b>	n/a	Territories tab	<b>x</b>	<b>x</b>	<a href="#">JContTerritory</a>		8 The list of territories that are added to the contact record.  --> Links to the join table that contains information for all territories for the contact.  Select when you want to access territory types.	<b>x</b>	
<b>jobTitle</b> <b>Before TeamConnect 3.3 SP2: title</b>	<b>TITLE</b> <b>(VARCHAR2) (250)</b>	Job Title (General block)	<b>x</b>			string (250)	The job title of the contact.	<b>x</b>	
<b>typeIID</b>	<b>TYPE_IID</b>	New Pers	<b>x</b>			Enum	The type of the contact.	<b>x</b>	

	(CHAR) (1)	on or New Com pany  (Pop- up Menu )					P - Person (PERSON)  C - Company (COMPANY)		
userAccessList-->	n/a			x	<a href="#">EContUser Access</a>		Links to the user security information entered through the Security block of the record.		
version	VERSION (NUMBER )		x			int	Indicates how many times the contact record was updated.		

### 3.1.28.2 JContAddress

JContAddress contains information about the addresses in a contact record. This information includes the city, street, zip code, county, state, address type (such as Home or Business), and the contact record to which the address belongs. This table represents the **Addresses** block on the **General** tab of a contact record.

Object: JContAddress (J\_CONT\_ADDRESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/ Wiz
city	CITY (VARCHAR 2) (50)	City	x			string (50)	The city in the contact address.	x	x

<b>countryItem</b> -->	<b>COUNTRY_</b> <b>ITEM_ID</b>  (NUMBER)	Cou ntry	x	x	<a href="#">LCountry</a> <a href="#">tem</a>	obje ct	8 The country in the contact address.  --> Links to definition information for the country item.	x	x
<b>county</b>	<b>COUNTY</b>  (VARCHAR 2) (50)	Cou ntry	x			strin g (50)	The county of the contact address.	x	x
<b>currentOn</b>	<b>CURRENT_</b> <b>ON</b>  (DATE)	Curr ent as of	x			date	The last known date the address was current. Time- zone- independent.		x
<b>isCurrent</b>	<b>IS_CURRE</b> <b>NT</b>  (NUMBER)		x			bool ean	Indicates whether the address is current.  0 - Not Current 1 - Current	x	x
<b>owner--&gt;</b>	<b>CONTACT_</b> <b>ID</b>  (NUMBER)		x	x	<a href="#">TContact</a>	obje ct	8 The contact to which the address belongs.  --> Links to specific information about the contact record.		x
<b>postalCode</b>	<b>POSTAL_C</b> <b>ODE</b>  (VARCHAR 2) (50)	Zip	x			strin g (50)	The zip code in the contact address.	x	x
<b>primaryKey</b>	<b>PRIMARY_</b> <b>KEY</b>		x			int	The unique ID of the contact		

	(NUMBER)						address.		
referenceNumberString	REFERENCE_NUMBER_STRING (VARCHAR 2) (50)	Guide No.	x			string (50)	The reference number for the contact address.		x
state	STATE (VARCHAR 2) (50)	State/Province	x			string (50)	The U.S. state in the contact address.	x	x
street	STREET (VARCHAR 20) (250)	Street	x			string (250)	The street address in the contact address.	x	x
type-->	TYPE_ID (NUMBER)	Type	x	x	<a href="#">LContAddressType</a>	object	8 The type of the contact address (such as Home or Business).  --> Links to definition information for the address type.	x	x
version	VERSION (NUMBER)		x			int	Indicates how many times the contact address was updated.		

## 3.1.28.2.1 LCountryItem

LCountryItem contains information about the country items defined for contact addresses. This includes the name of the country item, tree position, and display order. This table represents the **Country Item** system lookup table.

Object: LCountryItem (L\_COUNTRY\_ITEM)

Attribute	Database column name	Field in UI	End of pat	Bridge	Links to object	Data tvp	Comments	Commonly used in:
-----------	----------------------	-------------	------------	--------	-----------------	----------	----------	-------------------



								Rules	Temp/Wiz
childList-->	n/a							Do not apply. These lookup items cannot have a hierarchical structure.	
parent-->	PARENT_ID (NUMBER)								
displayOrder	DISPLAY_ORDER (NUMBER)	Order	x			int	The order in which the table items will be listed in the GUI, lowest number first.		
name	NAME (VARCHAR2) (50)	Item Name	x			string (50)	The name of the country.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the country list item.		
treePosition	TREE_POSITION (VARCHAR2) (250)	Tree Position	x			string (250)	The unique four-character alphanumeric code that is created when the country list item is defined.		
version	VERSION (NUMBER)		x			int	Indicates how many times the country list item was updated.		

### 3.1.28.2.2 LContAddressType

LContAddressType contains information about the address types defined for contact addresses. This information includes the name of the address type, tree position, and display order. This table represents the **Address Type** system lookup table.

**Object:** LContAddressType (L\_CONT\_ADDRESS\_TYPE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
childList-->	n/a						Do not apply. These lookup items cannot have a hierarchical structure.		
parent->	PARENT_ID (NUMBER)								
display Order	DISPLAY_ORDER (NUMBER)	Order	x			int	The order in which the table items will be listed in the GUI, lowest number first.		
name	NAME (VARCHAR2) (50)	Item Name	x			string (50)	The name of the address type.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the address type list item.		
treePosition	TREE_POSITION (VARCHAR2) (250)	Tree Position	x			string (250)	The unique four-character alphanumeric code that is created when the address type is defined.		
version	VERSION (NUMBER)		x			int	Indicates how many times the address type list item was updated.		

### 3.1.28.3 JContPhone

JContPhone contains information about the phone numbers in a contact record. This information includes the number itself, phone number type (such as Home or Business), and contact record to

which each number belongs. This table represents the **Phone Numbers** block on the **General** tab of a contact record.

Object: JContPhone (J\_CONT\_PHONE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
<b>number</b> Before TeamConnect 3.3 SP2: phoneString	PHONE_STRING (VARCHAR2) (50)	Number	x			string (50)	The actual phone number.		x
<b>owner--&gt;</b>	CONTACT_ID (NUMBER)		x	x	<a href="#">TContact</a>	object	8 The contact to which the phone number belongs.  --> Links to specific information for the contact record.		
<b>primaryKey</b>	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the phone number record.		
<b>type--&gt;</b>	TYPE_ID (NUMBER)	Type	x	x	<a href="#">LContPhoneType</a>	object	8 The type of this phone number (such as Home or Business).  --> Links to the table that contains all phone number types.	x	x
<b>version</b>	VERSION		x			int	Indicates how many times		

	(NUMBER)							the phone number was updated.		
--	----------	--	--	--	--	--	--	-------------------------------	--	--

## 3.1.28.3.1 LContPhoneType

LContPhoneType contains information about the phone number types defined for contact phone numbers. This information includes the name of the phone type, tree position, and display order. This table represents the **Phone Type** system lookup table.

Object: LContPhoneType (L\_CONT\_PHONE\_TYPE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
childList-->	n/a						Do not apply. These lookup items cannot have a hierarchical structure.		
parent->	PARENT_ID (NUMBER)								
display Order	DISPLAY_ORDER (NUMBER)	Order	x			int	The order in which the table items will be listed in the GUI, lowest number first.		
name	NAME (VARCHAR2) (50)	Item Name	x			string (50)	The name of the phone type.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the phone type list item.		
treePosition	TREE_POSITION (VARCHAR2) (250)	Tree Position	x			string (250)	The unique four-character alphanumeric code that is created when the phone type is defined.		

version	VERSION (NUMBER)		x			int	Indicates how many times the phone type list item was updated.		
---------	---------------------	--	---	--	--	-----	--	--	--

### 3.1.28.4 JContFax

JContFax contains information about the fax numbers in a contact record. This information includes the number itself, fax type (such as Home or Business), and contact record to which each number belongs. This table represents the Fax Numbers block on the General tab of a contact record.

Object: JContFax (J\_CONT\_FAX)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
owner-->	CONTACT_ID (NUMBER)		x	x	<a href="#">IContact</a>	object	8 The contact to which the fax number belongs.  --> Links to specific information about the contact record.		
number  Before TeamConnect 3.3 SP2: faxString	FAX_STRING (VARCHAR2) (50)	Number	x			string (50)	The actual fax number.		x
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the fax number record.		
type-->	TYPE (NUMBER)	Type	x	x	<a href="#">LContFaxType</a>	object	8 The type of the fax number.  --> Links to definition information for the fax type.	x	x

version	VERSION (NUMBER)		x			int	Indicates how many times the fax number was updated.		
---------	---------------------	--	---	--	--	-----	--	--	--

## 3.1.28.4.1 LContFaxType

LContFaxType contains information about the fax types defined for contact fax numbers including the name of the fax type, tree position, and display order. This table represents the **Fax Type** system lookup table.

Object: LContFaxType (L\_CONT\_FAX\_TYPE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
childList-->	n/a						Do not apply. These lookup items cannot have a hierarchical structure.		
parent->	PARENT_ID (NUMBER)								
display Order	DISPLAY_ORDER (NUMBER)	Order	x			int	The order in which the table items will be listed in the GUI, lowest number first.		
name	NAME (VARCHAR2) (50)	Item Name	x			string (50)	The name of the fax type.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the fax type list item.		

<b>treePosition</b>	<b>TREE_POSITION</b> (VARCHAR2) (250)	<b>Tree Position</b>	x			string (250)	The unique four-character alphanumeric code that is created when the fax type is defined.		
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Indicates how many times the fax type list item was updated.		

### 3.1.28.5 JContEmail

JContEmail contains information about the email addresses in a contact record. This information includes the address itself, email address type (such as Personal or Business), and contact record to which each email address belongs. This table represents the **Email Addresses** block on the **General** tab of a contact record.

Object: JContEmail (J\_CONT\_EMAIL)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
<b>address</b> Before TeamConnect 3.3 SP2: emailString	<b>EMAIL_STRING</b> (VARCHAR20) (250)	<b>Email</b>	x			string (250)	The actual email address.		x
<b>owner--&gt;</b>	<b>CONTACT_ID</b> (NUMBER)		x	x	<a href="#">TContact</a>	object	8 The contact to which the email address belongs.  --> Links to specific information for the contact record.		

<b>primary Key</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	The unique ID of the email address record.		
<b>type--&gt;</b>	<b>TYPE_ID</b> (NUMBER)	<b>Type</b>	x	x	<a href="#">LContEmailType</a>	object	8 The type of the email address. --> Links to definition information for the email address type.	x	x
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Indicates how many times the email address was updated.		

## 3.1.28.5.1 LContEmailType

LContEmailType contains information about the email types defined for contact email addresses. This information includes the name of the email type, tree position, and display order. This table represents the **Email Type** system lookup table.

Object: LContEmailType (L\_CONT\_EMAIL\_TYPE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
<b>childList--&gt;</b>	n/a						Do not apply. These lookup items cannot have a hierarchical structure.		
<b>parent--&gt;</b>	<b>PARENT_ID</b> (NUMBER)								
<b>display Order</b>	<b>DISPLAY_ORDER</b>	<b>Order</b>	x			int	The order in which the table items will be listed in the GUI, lowest number first.		



	(NUMBER)								
name	NAME (VARCHAR2) (50)	Item Name	x			string (50)	The name of the email type.		
primary Key	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the email type list item.		
treePosition	TREE_POSITION (VARCHAR2) (250)	Tree Position	x			string (250)	The unique four-character alphanumeric code that is created when the email type is defined.		
version	VERSION (NUMBER)		x			int	Indicates how many times the email type list item was updated.		

### 3.1.28.6 JContlnetAddress

JContlnetAddress contains information about the internet addresses in a contact record. This information includes the address itself, internet address type (such as Personal or Business), and contact record to which each address belongs. This table represents the **Internet Addresses** block on the **General** tab of a contact record.

Object: JContlnetAddress (J\_CONT\_INET\_ADDRESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
url Before TeamConnect 3.3 SP2:	INET_ADDRESS_STRING (VARCHAR2) (250)	Address	x			string (250)	The actual internet address.		x

inetAddressString									
owner-->	CONTACT_ID (NUMBER)		x	x	<a href="#">TContact</a>	object	8 The contact to which the internet address belongs.  --> Links to specific information for the contact record.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the internet address record.		
type-->	TYPE_ID (NUMBER)	Type	x	x	<a href="#">LContlnetAddressType</a>	object	8 The type of the internet address.  --> Links to definition information for the internet address type.	x	x
version	VERSION (NUMBER)		x			int	Indicates how many times the internet address was updated.		

## 3.1.28.6.1 LContlnetAddressType

LContlnetAddressType contains information about the internet address types defined for contact internet addresses. This information includes the name of the internet address type, tree position, and display order. This table represents the **Internet Address Type** system lookup table.

Object: LContlnetAddressType (L\_CONT\_INET\_ADDRESS\_TYPE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/

									Wiz
childList-->	n/a								
parent->	PARENT_ID (NUMBER)								
display Order	DISPLAY_ORDER (NUMBER)	Order	x			int	The order in which the table items will be listed in the GUI, lowest number first.		
name	NAME (VARCHAR2) (50)	Item Name	x			string (50)	The name of the internet address type.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the internet address type list item.		
treePosition	TREE_POSITION (VARCHAR2) (250)	Tree Position	x			string (250)	The unique four-character alphanumeric code that is created when the internet address type is defined.		
version	VERSION (NUMBER)		x			int	Indicates how many times the internet address type list item was updated.		

### 3.1.28.7 JContSkill

JContSkill contains information about the skills in a contact record including the skill level, skill type (such as underwriting or litigation), and contact record to which each skill belongs. This table represents the Skills block on the Skills tab of a contact record.

Object: JContSkill (J\_CONT\_SKILL)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
owner-->	CONTACT_ID (NUMBER)		x	x	<a href="#">LContact</a>	object	8 The contact to which the skill belongs.  --> Links to specific information for the contact record.		
skillLevel	SKILL_LEVEL (NUMBER)	Level of Expertise	x			int	The skill level assigned to the contact for the skill type.		x
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the skill.		
skillType--> Before TeamConnect 3.3 SP2: type-->	TYPE_ID (NUMBER)	Type	x	x	<a href="#">LContSkillType</a>	object	8 The type of the skill.  --> Links to definition information for the skill type.	x	x
version	VERSION (NUMBER)		x			int	Indicates how many times the skill record was updated.		

## 3.1.28.7.1 LContSkillType

LContSkillType contains information about the skill types defined for contact skills. This information includes the name of the skill, tree position, and display order. This table represents the Skill Type system lookup table.

Object: LContSkillType (L\_CONT\_SKILL\_TYPE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
childList-->	n/a		x	x	<a href="#">LContSkillType</a>		8 The child skill types for the current skill type.  --> Links to definition information for a child skill type.		
display Order	DISPLAY_ORDER (NUMBER)	Order	x			int	The order in which the table items will be listed in the GUI, lowest number first.		
name	NAME (VARCHAR2) (50)	Item Name	x			string (50)	The name of the skill type.		
parent->	PARENT_ID (NUMBER)	Show items in node	x	x	<a href="#">LContSkillType</a>	object	8 The parent item of the current item.  --> Links to definition information for the parent skill type.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the skill type list item.		
treePosition	TREE_POSITION (VARCHAR2) (250)	Tree Position	x			string (250)	The unique four-character alphanumeric code that is created when the skill type is defined.		

version	VERSION (NUMBER)		x			int	Indicates how many times the skill type list item was updated.		
---------	---------------------	--	---	--	--	-----	--	--	--

### 3.1.28.8 JContDefaultRate

JContDefaultRate contains information about the default rate in a contact record. (The default rate is used when there is no specific rate for a task's category.) This information includes the rate amount, effective from and to date, and contact record to which each rate belongs.

Rates that are specific to categories are described in [JContRate](#).

Object: JContDefaultRate (J\_CONT\_DEFAULT\_RATE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table :	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
firstEffectiveDate Before TeamConnect 3.3 SP2: effective From	EFFECTIVE_FROM (DATE)	From	x			date	The date and time when the rate will begin to apply to the associated task. Time-zone-independent.	x	x
lastEffectiveDate Before TeamConnect 3.3 SP2: effective To	EFFECTIVE_TO (DATE)	To	x			date	The date and time when the rate will cease applying to the associated task. Time-zone-independent.	x	x
owner-->	CONTACT_ID (NUMBER)		x	x	<a href="#">JContact</a>	object	8 The contact to which the rate belongs.  --> Links to specific information for the contact record.		

<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	The unique ID of the rate record.		
<b>rate</b> Before TeamConnect 3.3 SP2: <b>rateAmount</b>	<b>RATE_AMOUNT</b> (NUMBER)	<b>Rate</b>	x			decimal	The actual rate for the associated task.	x	x
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Indicates how many times the rate record was updated.		

**3.1.28.9 JContRate**

JContRate contains information about the rates in a contact record. This information includes the rate amount, effective from and to date, task category, and contact record to which each rate belongs.

This table represents the **Rates** block on the **Rates** page of a contact record. It also represents the **Invoice Task Rates** block on the **Rates** page of a contact record. What determines which block will display a given rate record is the value in the TASK\_CATEGORY attribute. If this value links to a category assigned to the **Task** object type, then the rate appears in the Rates block. If instead TASK\_CATEGORY links to a category from the **Task Categories** lookup table in the **Invoice Line Item** object type, then the rate appears in the Invoice Task Rates block.

Object: JContRate (J\_CONT\_RATE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
<b>firstEffectiveDate</b> Before TeamConnect 3.3 SP2: <b>effectiveFrom</b>	<b>EFFECTIVE_FROM</b> (DATE)	<b>From</b>	x			date	The date and time when the rate will begin to apply to the associated task. Time-zone-independent.	x	x

<b>lastEffectiveDate</b> Before TeamConnect 3.3 SP2: effective To	<b>EFFECTIVE_TO</b> (DATE)	To	x			date	The date and time when the rate will cease applying to the associated task. Time-zone-independent.	x	x
<b>owner--&gt;</b>	<b>CONTACT_ID</b> (NUMBER)		x	x	<a href="#">TContact</a>	object	8 The contact to which the rate belongs.  --> Links to specific information for the contact record.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	The unique ID of the rate record.		
<b>rate</b> Before TeamConnect 3.3 SP2: rateAmount	<b>RATE_AMOUNT</b> (NUMBER)	Rate	x			decimal	The actual rate for the associated task.	x	x
<b>taskCategory--&gt;</b>	<b>TASK_CATEGORY</b> (NUMBER)	Task	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 The task category that belongs to the rate.  --> Links to definition information for the task category.	x	x
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Indicates how many times the rate record was updated.		



### 3.1.28.10 JContRelation

JContRelation contains information about a contact record's relations with other contacts. This information includes the contact on the left and right sides of the relationship and the type of relationship between the two contacts (such as "Attorney for" or "Employee of"). This table represents the **Relations** block on the **Relations** tab of a contact record.

Object: JContRelation (J\_CONT\_RELATION)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wiz
contact--> <b>Before TeamConnect 3.3 SP2: leftContact--&gt;</b>	LEFT_CONTACT_ID (NUMBER)	Contact (left-side)	x	x	<a href="#">TContact</a>	object	8 The contact that is on the left side of the relationship.  --> Links to specific information for the contact record.  <b>Note: The distinction between right and left contact in Team Connect 3.4 is determined by a 3.4-specific attribute</b>		x

							<i>named relatio nDire ction.</i>		
primaryKey	PRIMARY_ KEY (NUMBER)		x			int	The unique ID of the relation record.		
contact--> Before TeamConne ct 3.3 SP2: rightContact -->	RIGHT_CO NTACT_ID (NUMBER)	Cont act (right - side)	x	x	<a href="#">TContact</a>	obj ect	8 The contact that is on the right side of the relationship.  --> Links to specific information for the contact record.  <b>Note: The distinc tion betwe en right and left  Contac t in Team Conne ct 3.4 is deter mined by a 3.4- specifi c attribu te named relatio nDire ction.</b>		x

<b>relationType</b>  <b>Before TeamConnect 3.3 SP2: typeId</b>	<b>TYPE_ID</b> <b>(NUMBER)</b>	<b>Relation</b>	<b>x</b>	<b>x</b>	<a href="#">LContRelationType</a>	object	8 The type of the relationship.  --> Links to definition information for the relation type.	<b>x</b>	<b>x</b>
<b>version</b>	<b>VERSION</b> <b>(NUMBER)</b>		<b>x</b>			int	Indicates how many times the relation record was updated.		

## 3.1.28.10.1 LContRelationType

LContRelationType contains information about the relationship types defined for contact records. This information includes the name of the relation type, tree position, and display order. This table represents the **Contact Relation Type** system lookup table.

Object: LContRelationType (L\_CONT\_RELATION\_TYPE)

Attribute	Database column name	Field in UI	End of path	Bridge	Data type	Comments	Commonly used in:	
							Rules	Temp/Wiz
childList-->	n/a					Do not apply. These lookup items cannot have a hierarchical structure.		
parent-->	<b>PARENT_ID</b> <b>(NUMBER)</b>							
display Order	<b>DISPLAY_ORDER</b> <b>(NUMBER)</b>	<b>Order</b>	<b>x</b>		int	The order in which the table items will be listed in the GUI, lowest number first.		
name	<b>NAME</b> <b>(VARCHAR2) (50)</b>	<b>Item Name</b>	<b>x</b>		string (50)	The name of the relation type.		

<b>primary Key</b>	<b>PRIMARY_KEY</b> (NUMBER)		x		int	The unique ID of the relation type list item.		
<b>treePosition</b>	<b>TREE_POSITION</b> (VARCHAR2) (250)	<b>Tree Position</b>	x		string (250)	The unique four-character alphanumeric code that is created when the relation type is defined.		
<b>version</b>	<b>VERSION</b> (NUMBER)		x		int	Indicates how many times the relation type list item was updated.		

### 3.1.28.11 JContTerritory

JContTerritory contains information about the territories in a contact record. This information includes the territory type (such as Northwest or Southeast), and contact record to which each territory belongs. This table represents the **Territories** block on the **Territories** tab of a contact record.

Object: JContTerritory (J\_CONT\_TERRITORY)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
<b>owner--&gt;</b>	<b>CONTACT_ID</b> (NUMBER)		x	x	<a href="#">TContact</a>	object	8 The contact to which the territory belongs.  --> Links to specific information for the contact record.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	The unique ID of the territory record.		
<b>territoryItem--&gt;</b>	<b>TYPE_ID</b>	<b>Territory</b>	x	x	<a href="#">JContTerritoryType</a>	object	8 The type of the territory record.	x	x

Before TeamConnect 3.3 SP2: type-->	(NUMBER)						--> Links to definition information for the territory type.		
version	VERSION (NUMBER)		x			int	Indicates how many times the territory record was updated.		

## 3.1.28.11.1 LContTerritoryType

LContTerritoryType contains information about the territory types defined for contact records. This information includes the name of the territory type, tree position, and display order. This table represents the **Territory Type** system lookup table.

Object: LContTerritoryType (L\_CONT\_TERRITORY\_TYPE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
childList-->	n/a		x	x	<a href="#">LContTerritoryType</a>		8 The child territories for this territory.  --> Links to definition information for all child territories of the current territory.		
displayOrder	DISPLAY_ORDER (NUMBER)	Order	x			int	The order in which the table items will be listed in the GUI, lowest number first.		
name	NAME (VARCHAR2) (50)	Item Name	x			string (50)	The name of the territory type.		

parent -->	PARENT_ID (NUMBER)	Show items in node	x	x	<a href="#">LContTerritoryType</a>	object	8 The parent item of the current item.  --> Links to definition information for the parent item.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the territory type list item.		
treePosition	TREE_POSITION (VARCHAR2) (250)	Tree Position	x			string (250)	The unique four-character alphanumeric code that is created when the territory type is defined.		
version	VERSION (NUMBER)		x			int	Indicates how many times the territory type list item was updated.		

### 3.1.28.12 Y\_Contact\_Group

Y\_Contact\_Group represents contact groups (Address Books) and contains information such as the user who owns the contact group, whether the contact group is public, and name of the contact group. This table represents the **Contact Group Information** block on the **General** tab of a contact group record. The list of members is available through the **memberList** attribute.

Object: Y\_Contact\_Group

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
createdBy -->	CREATE_BY_ID		x	x	<a href="#">YUser</a>	object	8 User who created the contact group (Address Book).		

	(NUMBER )						--> Links to specific information for the user account.		
createdOn	CREATE_D_ON (DATE)		x			date	Date when the contact group (Address Book) was created.		
members-> Before TeamConnect 3.3 SP2: memberList->	n/a			x	<a href="#">JDistMember</a>		Links to the list of members in the contact group (Address Book).		
modifiedBy->	MODIFIED_BY_ID (NUMBER )		x	x	<a href="#">YUser</a>	object	8 User who last modified the contact group (Address Book).  --> Links to specific information for the user account.		
modifiedOn	MODIFIED_ON (NUMBER )		x			date	Date when the contact group (Address Book) was last modified.		
name	NAME (VARCHAR2) (250)	Group name (General block of Contact Group)	x			string (250 )	The name of the contact group (Address Book).		
owner->	USER_ID		x	x	<a href="#">YUser</a>	object	8 The user to whom the		

	(NUMBER )						contact group (Address Book) belongs.  --> Links to specific information about the user account.		
primaryKey	PRIMARY_KEY (NUMBER )		x			int	The unique ID of the contact group (Address Book).		
shared Before TeamConnect 3.3 SP2: isPublic	IS_PUBLIC (NUMBER )	Is contact group (Address Book) public  (General block of Contact Group)	x			boolean	The setting that indicates whether the contact group is public.		
version	VERSION (NUMBER )		x			int	Indicates how many times the item was updated.		

## 3.1.28.12.1 JDistMember

JDistMember contains information about the members of a contact group (Address Book). This information includes the contact who is a member of a contact group and the contact group to which the member belongs. This table represents the **Contact Group Information** block on the **Members** tab of a contact group record.

Object: JDistMember (J\_DIST\_MEMBER)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz



contact -->	CONTACT_ID (NUMBER)	Members (Members block of Contact Group)	x	x	<a href="#">TContact</a>	object	8 The contact who is a member of a contact group (Address Book).  --> Links to specific information for the contact record.		
created By-->	CREATED_BY_ID (NUMBER)		x	x	<a href="#">YUser</a>	object	8 User who created the contact group (Address Book) member.  --> Links to specific information for the user account.		
created On	CREATED_ON (DATE)		x			date	Date when the contact group (Address Book) member was created.		
distribution-->	CONTACT_GROUP_ID (NUMBER)		x	x	<a href="#">RDistribution</a>	object	8 The contact group (Address Book) to which this contact group member belongs.  --> Links to definition information for the contact group.		
modified By-->	MODIFIED_BY_ID (NUMBER)		x	x	<a href="#">YUser</a>	object	8 User who last modified the contact group (Address Book) member.  --> Links to specific information for the user account.		
modified On	MODIFIED_ON		x			date	Date when the contact group		

	(DATE)						(Address Book) member was last modified.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the contact group (Address Book) member.		
version	VERSION (NUMBER)		x			int	Indicates how many times the item was updated.		

### 3.1.28.13 EContDetail

EContDetail contains all categories added to contact records. Each time a category is added to a contact record, an entry is made in this table in the database.

In Object Navigator, **EContDetail** is not displayed when you traverse using the **detailList-->** bridge. Instead, a list of all categories that have been defined for the object definition appears. This list is labeled **Category**. From each category, you can traverse to a table called **Detail Fields**, which displays all custom fields belonging to the category you selected.

For more information about how custom field values are stored in the database, see Object Model: Custom Fields.

Object: EContDetail (E\_CONT\_DETAIL)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wizard
category-->	CATEGORY_ID (NUMBER)	Category	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 The category definition to which the added category belongs.  --> Links to definition information for the category.		

isManual	IS_MANUAL (NUMBER)		x			boolean	Specifies whether the category is added manually or automatically to the contact. Categories are added automatically when they are the parent of the category that is added manually.  0 - Added Automatically  1 - Added Manually		
owner-->	ENTERPRISE_OBJECT_ID (NUMBER)			x	<a href="#">TContact</a>	object	The contact with which the custom fields are associated.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the contact category record.		
version	VERSION (NUMBER)		x			int	Indicates how many times the added category was updated.		

### 3.1.28.14 EContUserAccess

EContUserAccess contains the user access information that is set in the **Security** block of each contact record. For example, it includes whether each user that is listed in the Security block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm**(ission) rights.

Object: EContUserAccess (E\_CONT\_USER\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table :	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz

<b>isRead</b>	<b>IS_READ (NUMBER)</b>	<b>Re ad che ck box</b>	<b>x</b>			int	<p>Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
<b>isUpdate</b>	<b>IS_UPDATE (NUMBER)</b>	<b>Up dat e che ck box</b>	<b>x</b>			int	<p>Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
<b>isDelete</b>	<b>IS_DELETE (NUMBER)</b>	<b>Del ete che ck box</b>	<b>x</b>			int	<p>Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
<b>isPerm</b>	<b>IS_PERM (NUMBER)</b>	<b>Per m che ck box</b>	<b>x</b>			int	<p>Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1</p>		

							(selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
allowDenyIID	ALLOW_DENY_IID (NUMBER)	Option	x			Enum	Indicates whether the user is allowed or denied access to the record.  a - Allow (ALLOW) d - Deny (DENY)		
isManual	IS_MANUAL (NUMBER)		x			boolean	Indicates whether the user rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
owner->	ENTERPRISE_OBJECT_ID (NUMBER)		x	x	<a href="#">TContact</a>	object	8 The contact record to which this user access setting belongs.  -> Links to specific information for the contact record.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	Unique ID for the user access rights.		
user->	USER_ID (NUMBER)	List	x	x	<a href="#">YUser</a>	object	8 The user the access rights pertain to.		

							--> Links to specific information about the user account.		
version	VERSION (NUMBER)		x			int	Indicates how many times the user access rights have been updated.		

### 3.1.28.15 EContGroupAccess

EContGroupAccess contains the user access information that is set in the Security block of each contact record. For example, it includes whether each user that is listed in the Security block is allowed or denied the Read, Update, Delete, and Perm(ission) rights.

Object: EContGroupAccess (E\_CONT\_GROUP\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
isRead	IS_READ (NUMBER)	Read checkbox	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isUpdate	IS_UPDATE (NUMBER)	Update checkbox	x			int	Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the		

							operations are allowed or denied. (See allowDenyIID.)		
<b>isDelete</b>	<b>IS_DELETE (NUMBER)</b>	<b>Delete</b> checkbox	<b>x</b>			int	Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isPerm</b>	<b>IS_PERM (NUMBER)</b>	<b>Perm</b> checkbox	<b>x</b>			int	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>allowDenyIID</b>	<b>ALLOW_DENY_IID (NUMBER)</b>	<b>Option</b>	<b>x</b>			Enum	Indicates whether the group is allowed or denied access to the record.  a - Allow (ALLOW ) d - Deny (DENY)		
<b>group--&gt;</b>	<b>GROUP_ID (NUMBER)</b>	<b>List</b>	<b>x</b>	<b>x</b>	<a href="#">YGroup</a>	object	8 The group that the access rights pertain to.  --> Links to specific information for the group account.		

<b>isManual</b>	<b>IS_MANUAL (NUMBER)</b>		<b>x</b>			boolean	Indicates whether the group access rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually  1 - Assigned automatically		
<b>owner--&gt;</b>	<b>ENTERPRISE_OBJECT_ID (NUMBER)</b>		<b>x</b>	<b>x</b>	<a href="#">TContact</a>	object	8 The contact record to which this group access setting belongs.  --> Links to specific information for the contact record.		
<b>primaryKey</b>	<b>PRIMARY_KEY (NUMBER)</b>		<b>x</b>			int	Unique ID for the group access rights.		
<b>version</b>	<b>VERSION (NUMBER)</b>		<b>x</b>			int	Indicates how many times the group access rights have been updated.		

### 3.1.29 Object Model: Appointments

The tables in this appendix provide information about the object model as it relates to appointments.

#### 3.1.29.1 TAppointment

TAppointment contains general information for an appointment record. This information includes the appointment's subject, start date and time, end date and time, default category, geographical area, location, and so on. It also links to the lists of attendees, notes, used resources, and so on.

Object: TAppointment (T\_APPOINTMENT)

Attribute	Database column name	Field in UI	End of	Bridge	Links to object table:	Data	Comments	Commonly used in:



			pat h			typ e		Ru les	
arealitem- ->	AREA_ITEM_ ID  (NUMBER)	Area (Gen eral block )	x	x	<a href="#">LApptArealitem</a>	obje ct	8 Geographical area where the appointment will take place (for example, Orange County, and Chicago).  --> Links to the definition information for the appointment area item.		x
attendees ->  Before TeamCon nect 3.3 SP2: attendee List->	n/a	Atte ndee s tab	x	x	<a href="#">JApptAttendee</a>		8 The list of all attendees that are added to the appointment record.  --> Links to the join table that contains information for all attendees who have been added to the appointment.		
categories->  Before TeamCon nect 3.3 SP2: detailList- ->	n/a		x	x	<a href="#">EApptDetail</a>  In Object Navigator, links to Category list which appears in the UI only.		8 The list of categories added to the current object.  --> In the object model, the added categories		

							link to the values that have been added for custom fields, according to field type.  --> In Object Navigator, this attribute is enhanced. It links to a list of all categories defined for the object definition, rather than those added to the current object. From there, you can select a category and traverse to the list of custom fields that belong to it.		
createdBy-->	CREATED_BY_ID (NUMBER)	Created By (Security block)	x	x	<a href="#">YUser</a>	object	8 User who created the appointment.  --> Links to specific information about the user account.	x	x
createdOn	CREATED_ON (DATE)	Created On (Security block)	x			date	Date and time the appointment was created.	x	x

createdOnBehalfOf	CREATED_ON_BEHALF_OF_ID			x	<a href="#">TContact</a>	object	When not null, it indicates that the record was created through an e-billing system or other external application. The contact referenced will often be a vendor in TeamConnect.	X	
defaultCategory-->	DEFAULT_CATEGORY_ID (NUMBER)	Default Category (General block)	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 Default category for the appointment. --> Links to the definition information of the category.		x
documentFolder-->	DOCUMENT_FOLDER_ID (NUMBER)	Documents block		x	<a href="#">TDocument</a>	object	Links to specific information about the document folder.		
endOn	END_ON (DATE)	Ends On	x			date time	Date and time when the appointment ends.	x	x
groupAccessList-->	n/a			x	<a href="#">EApptGroupAccess</a>		Links to the group security information entered through the Security block of the record.		

historyList-->	n/a	History tab	x	x	<a href="#">THistory</a>		8 The list of all histories that are added to the appointment record.  --> Links to history information for all of the appointment's history records.		
isAllDay Before TeamConnect 3.3 SP2: isAllDay	IS_ALL_DAY (NUMBER)		x			boolean	Indicates whether the appointment is scheduled for the entire day.  0 - not all day 1 - all day	x	x
location	LOCATION (VARCHAR2) (250)	Location	x			string (250)	Location where the appointment takes place (such as Corporate Headquarters or Conference Room 3rd Floor).		x
modified By-->	MODIFIED_BY_ID (NUMBER)	Modified By (Security block)	x	x	<a href="#">YUser</a>	object	8 User who most recently modified the account.  --> Links to specific information about the user account.		

modified On	MODIFIED_ON (DATE)	Modified On (Security block)	x			date	Date and time the appointment was most recently modified.		
note-->	NOTE_ID (NUMBER)	Notes (General block)		x	<a href="#">JNote</a>	object	Links to the table that contains the text entered into the <b>Notes</b> field of the record.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	Unique ID for the appointment record.		
project-->	PROJECT_ID (NUMBER)	Project (General block)	x	x	<a href="#">TProject</a>	object	8 Project for which the appointment is created.  --> Links to specific information about the project record.	x	x
resources -->  Before TeamConnect 3.3 SP2: resource List-->	n/a	Resources tab	x	x	<a href="#">JApptResource</a>		8 The list of all resources that are added to the appointment record.  --> Links to the join table that contains information for all resources added to the appointment.		x

							In Object Navigator, this attribute links to a table named <b>Resource Type</b> that allows you to filter the resources based on a type, if desired.		
securityTypeID	SECURITY_TYPE_IID (NUMBER)	Security (Security block)	x			Enum	Specifies whether the appointment is public or private.  0 - Public (PUBLIC)  2 - Private (PRIVATE)		
startOn	START_ON (DATE)	Begins On (General block)	x			date time	Date and time when the appointment begins.	x	x
subject	SUBJECT	Subject (General block)	x			string (250)	Description of the appointment (e.g. Meeting with client).		x
userAccessList-->	n/a			x	<a href="#">EApptUserAccess</a>		Links to the user security information entered through the Security block of the record.		

version	VERSION (NUMBER)		x			int	Indicates how many times the appointment record was updated.		
---------	---------------------	--	---	--	--	-----	--	--	--

### 3.1.29.2 LApptArealtem

LApptArealtem contains information about the area items defined for appointment records. This information includes the name of the area item, tree position, and display order. This table represents the **Area Item** system lookup table.

Object: LApptArealtem (L\_APPT\_\_AREA\_ITEM)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
childList-->	n/a			x	<a href="#">LApptArealtem</a>		Links to the child items in the Area Item system lookup table that belong to this area list item.		
display Order	DISPLAY_ORDER (NUMBER)	Order (Look up Tables, System tab)	x			int	The order in which the table items will be listed in the GUI, lowest number first.		
name	NAME (VARCHAR2) (250)	Item Name (Look up Tables, System tab)	x			string (50)	The name of the area item (for example, Orange County, Los Angeles).		

parent->	PARENT_ID (NUMBER)	Show items in node (Look up Tables, System tab)	x	x	<a href="#">LApptAreaItem</a>	object	Links to the parent item of this area item in the Area Item system lookup table.		
primaryKey	PRIMARY_KEY(NUMBER)		x			int	The unique ID of the area list item.		
treePosition	TREE_POSITION (VARCHAR2) (250)	Tree Position (Look up Tables, System tab)	x			string (250)	The unique four-character alphanumeric code that is created when the area list item is defined.		
version	(VERSION) (NUMBER)		x			int	Indicates how many times the area list item was updated.		

### 3.1.29.3 JApptAttendee

JApptAttendee contains information about the attendees in an appointment record. This information includes the user who has been added as an attendee, attendance status, and starting and ending date and time. This table represents the **Attendees** block on the **Attendees** tab of an appointment.

Object: JApptAttendee (J\_APPT\_ATTENDEE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz



<b>attendanceType</b> Before TeamConnect 3.3 SP2: attendanceTypeID	<b>ATTENDANCE_TYPE_ID</b> (NUMBER)	<b>Attendance</b> (Attendees block)	x			int	Indicates the attendance status of the attendee:  0 - Will Attend (WILL_ATTEND)  1 - Tentative (TENTATIVE)  2 - Will Not Attend (WILL_NOT_ATTEND)  3 - Unknown		x
<b>endOn</b>	<b>END_ON</b> (DATE)	<b>End Time</b> (Attendees block)	x			date Time	End date and time until which the user is scheduled to attend.		x
<b>owner--&gt;</b>	<b>APPOINTMENT_ID</b> (NUMBER)			x	<a href="#">TAppointment</a>	object	The appointment which the user is scheduled to attend.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	Unique ID given to each attendee added to an appointment record.		
<b>startOn</b>	<b>START_ON</b> (DATE)	<b>Start Time</b> (Attendees block)	x			date Time	Start date and time when the user is scheduled to attend.		x
<b>user--&gt;</b>	<b>USER_ID</b> (NUMBER)	<b>Attendee</b>	x	x	<a href="#">YUser</a>	object	8 User who is added as an attendee to the appointment record.	x	x

		(Attendees block)					--> Links to specific information about the user account.		
version	VERSION (NUMBER)		x			int	Indicates how many times the appointment attendee was updated.		

### 3.1.29.4 JApptResource

JApptResource contains information about the resources used in an appointment record. This information includes resource type (such as Projector or Conference Room) and the starting and ending date and time the resource is needed. This table represents the **Resources** block on the **Resources** tab of an appointment record.

Object: JApptResource (J\_APPT\_RESOURCE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
endOn	END_ON (DATE)	To (Resources block)	x			date Time	Date and time until which the resource is needed.		x
owner -->	APPOINTMENT_ID (NUMBER)			x	<a href="#">TAppointment</a>	object	The appointment for which the resource is needed.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	Unique ID given to each resource added to an appointment record.		
startOn	START_ON	From	x			date Time	Date and time when the		x

	(NUMBER )	(Resources block)					resource is needed.		
type-->	TYPE_ID (NUMBER )	Resource (Resources block)	x	x	<a href="#">LApptResourceType</a>	object	8 The type of resource (such as Projector or Conference Room).  --> Links to the definition information for the appointment resource type.		x
version	VERSION (NUMBER )		x			int	Indicates how many times the appointment resource was updated.		

### 3.1.29.5 LApptResourceType

LApptResourceType contains information about the resource types defined for appointment records. This information includes the resource name, tree position, and display order. This table represents the **Resource Type** system lookup table.

Object: LApptResourceType (L\_APPT\_RESOURCE\_TYPE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
childList-->	n/a			x	<a href="#">LApptResourceType</a>		Links to the list of child resource types that belong to this resource type.		
display Order	DISPLAY_ORDER (NUMBER )	Order	x			int	The order in which the table items will be listed in the GUI, lowest number first.		

<b>name</b>	<b>NAME</b> (VARCHAR2) (50)	<b>Item Name</b>	x			string (50)	The name of the resource type (e.g. Projector).		
<b>parent-&gt;</b>	<b>PARENT_ID</b> (NUMBER)	<b>Show items in node</b>	x	x	<a href="#">LApptResourceType</a>	object	8 The parent of the current resource type.  --> Links to the definition information for the appointment resource type.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	The unique ID of the resource list item.		
<b>treePosition</b>	<b>TREE_POSITION</b> (VARCHAR2) (250)	<b>Tree Position</b>	x			string (250)	The unique four-character alphanumeric code that is created when the resource list item is defined.		
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Indicates how many times the resource type was updated.		

### 3.1.29.6 EApptDetail

EApptDetail contains all categories added to appointment records. Each time a category is added to an appointment record, an entry is made in this table in the database.

In Object Navigator, **EApptDetail** is not displayed when you traverse using the **detailList-->** bridge. Instead, a list of all categories that have been defined for the object definition appears. This list is labeled **Category**. From each category, you can traverse to a table called **Detail Fields**, which displays all custom fields belonging to the category you selected.

For more information about how custom field values are stored in the database, see Object Model: Custom Fields.

Object: EApptDetail (E\_APPT\_DETAIL)

Attribute	Database column	Field in UI	End	Bridge	Links to object	Data	Comments	Commonly used
-----------	-----------------	-------------	-----	--------	-----------------	------	----------	---------------

	name		of path	e	table:	type		in:	
								Rules	Temp / Wizard
category-->	CATEGORY_ID (NUMBER)	Category	x	x	<a href="#">YRecentlyViewedRecord</a>	object	The category of the appointment with which the details are associated.		
owner-->	ENTERPRISE_OBJECT_ID (NUMBER)			x	<a href="#">TAppointment</a>	object	The appointment with which the custom fields are associated.		
isManual	IS_MANUAL (NUMBER)		x			boolean	Specifies whether the category is added manually or automatically to the appointment. Categories are added automatically when they are the parent of the category that is added manually.  0 - Added Automatically 1 - Added Manually		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the appointment category record.		
version	VERSION (NUMBER)		x			int	Indicates how many times the added category was updated.		

### 3.1.29.7 EAptUserAccess

EAptUserAccess contains the user access information that is set in the **Security** block of each appointment record. For example, this information includes whether each user that is listed in the Security block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm**(ission) rights.

Object: EAptUserAccess (E\_APPT\_USER\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
isRead	IS_READ (NUMBER)	Read checkbox	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isUpdate	IS_UPDATE (NUMBER)	Update checkbox	x			int	Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isDelete	IS_DELETE (NUMBER)	Delete checkbox	x			int	Specifies whether the Delete operation has been selected for granting or denial. Value is 1		

							(selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isPerm</b>	<b>IS_PERM</b> (NUMBER)	<b>Per m che ck box</b>	<b>x</b>			<b>int</b>	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>allowDenyIID</b>	<b>ALLOW_DENY_IID</b> (NUMBER)	<b>Opt ion</b>	<b>x</b>			<b>Enum</b>	Indicates whether the user is allowed or denied access to the record.  a - Allow (ALLOW) d - Deny (DENY)		
<b>isManual</b>	<b>IS_MANUAL</b> (NUMBER)		<b>x</b>			<b>boolean</b>	Indicates whether the user rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		

owner->	ENTERPRISE_OBJECT_ID (NUMBER)		x	x	<a href="#">TAppointment</a>	object	8 The appointment record with which this user access setting is associated.  --> Links to the specific information about the appointment record.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	Unique ID for the user access rights.		
user-->	USER_ID (NUMBER)	List	x	x	<a href="#">YUser</a>	object	8 Specifies the user the access rights pertain to.  --> Links to specific information about the user account.		
version	VERSION (NUMBER)		x			int	Indicates how many times the user access rights have been updated.		

### 3.1.29.8 EAptGroupAccess

EAptGroupAccess contains the group access information that is set in the **Security** block of each appointment record. For example, it includes whether each group that is listed in the Security block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm(ission)** rights.

Object: EAptGroupAccess (E\_APPT\_GROUP\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
isRead	IS_READ (NUMBER)	Read che	x			int	Specifies whether the Read operation has been selected for granting or		



		ck box					denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isUpdate</b>	<b>IS_UPDATE (NUMBER)</b>	<b>Update checkbox</b>	<b>x</b>			int	Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isDelete</b>	<b>IS_DELETE (NUMBER)</b>	<b>Delete checkbox</b>	<b>x</b>			int	Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isPerm</b>	<b>IS_PERM (NUMBER)</b>	<b>Perm checkbox</b>	<b>x</b>			int	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine		

							whether the operations are allowed or denied. (See allowDenyIID.)		
allowDenyIID	ALLOW_DENY_IID (CHAR) (1)	Option	x			Enum	Indicates whether the group is allowed or denied access to the record.  a - Allow (ALLOW) d - Deny (DENY)		
group->	GROUP_ID (NUMBER)	List	x	x	<a href="#">YGroup</a>	object	8 Specifies the group that the access rights pertain to.  --> Links to specific information about the group account.		
isManual	IS_MANUAL (NUMBER)		x			boolean	Indicates whether the group access rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
owner->	ENTERPRISE_OBJECT_ID (NUMBER)		x	x	<a href="#">TAppointment</a>	object	8 The appointment record with which this group access setting is associated.  --> Links to the specific information about the appointment record.		
primaryKey	PRIMARY_KEY		x			int	Unique ID for the group access rights.		

	(NUMBER)								
version	VERSION (NUMBER)		x			int	Indicates how many times the group access rights have been updated.		

### 3.1.30 Object Model: Accounts

The tables in this appendix provide information about the object model as it relates to accounts.

#### 3.1.30.1 TAccount

TAccount contains the general information for an account record including the account's allocation amount and limit, used amount, and activity status. This information also includes expense information, default category, posting criteria, and so on. This table also links to the list of custom fields created for the account categories and other related information contained in other tables.

Object: TAccount (T\_ACCOUNT)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
accountInvolvedType  Before TeamConnect 3.3 SP2: applTypeInvolvedIID	INVOLVED_APPLIED (NUMBER)	Post Account to Involved  (Posting Criteria block)	x			Enum	Specifies whether the involved is associated to a specific project or to any project.  1 - Any (ANY)  2 - One (ONE)	x	x
accountOverdraftType	OVERDRAFT_TYP	Overdraft Type	x			Enum	Specifies what	x	x

<p><b>Before TeamConnect 3.3 SP2: overdraftTypeID</b></p>	<p><b>E_IID (NUMBER)</b></p>	<p>(General block)</p>					<p>happens when there are not enough funds to cover the manual withdrawals , transfers, or automatically posted transactions.</p> <p>1 - Do not allow negative account (NONNEGATIVE)</p> <p>2 - Allow negative account (ALLOW_NEGATIVE)</p> <p>3 - Automatic overdraft protection (OVERDRAFT)</p>		
<p><b>accountProjectType</b> <b>Before TeamConnect 3.3 SP2: applTypeProjectID</b></p>	<p><b>PROJECT_APPL_TYPE_IID (NUMBER)</b></p>	<p>Post Project to this Account (Posting Criteria block)</p>	<p><b>x</b></p>			<p>Enum</p>	<p>Indicates whether to post transactions associated to a specific project, any project, or a project of a specific category.</p>	<p><b>x</b></p>	<p><b>x</b></p>

							1 - Any (ANY_PROJECT) 2 - One (ONE_PROJECT) 3 - By Type (BY_TYPE)		
<b>accountVendorType</b>  Before TeamConnect 3.3 SP2: <b>applTypeVendorID</b>	<b>VENDOR_APPL_TYPE_ID</b>  (NUMBER)	Post Account to Vendor  (Posting Criteria block)	<b>x</b>			Enum	Indicates whether to post transactions associated to a specific vendor or any vendor.  1 - Any (ANY_VENDOR) 2 - One (ONE_VENDOR)	<b>x</b>	<b>x</b>
<b>allocated</b>	<b>TOTAL_ALLOCATED</b>  (NUMBER)	Total Allocated  (General block)	<b>x</b>			decimal	Total amount of money allocated to the account through deposits or automatic transfers.	<b>x</b>	<b>x</b>
<b>allocationLimit</b>	<b>ALLOC_LIMIT</b>  (NUMBER)	Allocation Limit  (General block)	<b>x</b>			decimal	Total amount of funds allowed to be allocated to the account records as specified	<b>x</b>	<b>x</b>

							by the user.		
<b>accrual</b>	<b>TOTAL_ACCRUAL</b> <b>(NUMBER)</b>	Total Accruals (General block)	<b>x</b>			decimal	Total amount of accrual invoices that have been posted to this account, to date. Takes into account voids of accrual invoices.	<b>x</b>	<b>x</b>
<b>applProjDetailField--&gt;</b>	<b>APPL_PROJECT_DETAIL_FIELD_ID</b> <b>(NUMBER)</b>						Currently not used.		
<b>autoPost</b> <b>Before</b> <b>TeamConnect</b> <b>3.3 SP2:</b> <b>isAutoPost</b>	<b>AUTO_POST</b> <b>(NUMBER)</b>	Allow Posting (General block)	<b>x</b>			boolean	Indicates whether expense, task, and invoice line items can be posted to the account.  0 - Do not post 1 - Post	<b>x</b>	<b>x</b>
<b>available</b>	<b>TOTAL_BG_AVAILABLE</b>	Balance (General block)	<b>x</b>			decimal	Account balance, which results from the		

	(NUMBER)						following equation:  allocated - used = available		
<p>categories--&gt;  Before TeamConnect 3.3 SP2: detailList--&gt;</p>	n/a		x	x	<p><a href="#">EAcctDetail</a></p> <p>In Object Navigator, links to Category list which appears in the UI only.</p>		<p>8 The list of categories added to the current object.</p> <p>--&gt; In the object model, the added categories link to the values that have been added for custom fields, according to field type.</p> <p>--&gt; In Object Navigator, this attribute is enhanced. It links to a list of all categories defined for the object definition, rather than those added to the current object. From there, you can select a category and traverse to</p>		

							the list of custom fields that belong to it.		
<b>createdBy--&gt;</b>	<b>CREATE D_BY_ID (NUMBER)</b>	Created By (Security block)	<b>x</b>	<b>x</b>	<a href="#">YUser</a>	object	8 User who created the account.  --> Links to specific information about the user account.		
<b>createdOn</b>	<b>CREATE D_ON (DATE)</b>	Created On (Security block)	<b>x</b>			date	Date and time the account was created.		<b>x</b>
<b>createdOnBehalfOf</b>	<b>CREATE D_ON_ BEHALF_ OF_ID</b>			<b>x</b>	<a href="#">TContact</a>	object	When not null, it indicates that the record was created through an e-billing system or other external application. The contact referenced will often be a vendor in TeamConnect.	<b>X</b>	
<b>defaultCategory--&gt;</b>	<b>DEFAULT_CATE GORY_ ID</b>	Default Category  (indicated by a	<b>x</b>	<b>x</b>	<a href="#">YRecentlyViewedRecord</a>	object	8 Default category for the account.		



	(NUMBER)	blue diamond - Categories block)					--> Links to the definition information of the category.		
documentFolder-->	DOCUMENT_FOLDER_ID (NUMBER)	Documents block		x	<a href="#">TDocument</a>	object	Links to specific information about the document folder.		
EndOn Before TeamConnect 3.3 SP2: applEndOn	APPL_END_ON (DATE)	Period ends on: (General block)	x			date	Specifies the date after which transactions cannot be posted to the account.	x	x
expenseCategory--> Before TeamConnect 3.3 SP2: applExpenseCategory-->	APPL_EXPENSE_CATEGORY_ID (NUMBER)	Post Expenses of Category (Posting Criteria block)	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 Expense category specified as the posting criterion for the account.  --> Links to the definition information of the category.	x	x
expensePercent Before TeamConnect 3.3 SP2: applExpensePercent	APPL_EXPENSE_PERCENT (NUMBER)	Expense Percent (Posting Criteria block)	x			decimal	Percentage of the total expense amount allowed to be posted to the account.	x	x

groupAccessList-->	n/a			x	<a href="#">EAcctGroup Access</a>		Links to the group security information entered through the Security block of the record.		
historyList-->	n/a	History tab	x	x	<a href="#">THistory</a>		8 The list of all histories that are added to the account record.  --> Links to specific history information for all of the account's history records.	x	
active Before TeamConnect 3.3 SP2: iisActive	STATUS_IID (NUMBER)	Status (General block)	x			boolean	Indicates whether the account is currently active or inactive.  0 - Inactive 1 - Active	x	x
allowExpense Before TeamConnect 3.3 SP2: iisAppExpense	APPL_EXPENSE (NUMBER)	Post Expenses to this Account (Posting Criteria block)	x			boolean	Indicates whether expense transactions can be posted to the account.  0 - Do not post 1 - Post	x	x

<b>allowInvoiceExpense</b>  <b>Before TeamConnect 3.3 SP2: iisApplLineExpense</b>	<b>APPL_INVD_EXPENSE</b>  <b>(NUMBER)</b>	Post Invoice Expenses to this Account  (Posting Criteria block)	<b>x</b>			boolean	Indicates whether invoice line item transactions of type Expense can be posted to the account.  0 - Do not post 1 - Post	<b>x</b>	<b>x</b>
<b>allowInvoiceTask</b>  <b>Before TeamConnect 3.3 SP2: iisApplLineTask</b>	<b>APPL_INVD_TASK</b>  <b>(NUMBER)</b>	Post Invoice Tasks to this Account  (Posting Criteria block)	<b>x</b>			boolean	Indicates whether invoice line item transactions of type Task can be posted to the account.  0 - Do not post 1 - Post	<b>x</b>	<b>x</b>
<b>allowTask</b>  <b>Before TeamConnect 3.3 SP2: iisApplTask</b>	<b>APPL_TASK</b>  <b>(NUMBER)</b>	Post Tasks to this Account  (Posting Criteria block)	<b>x</b>			boolean	Indicates whether task transactions can be posted to the account.  0 - Do not post 1 - Post	<b>x</b>	<b>x</b>
<b>invoiceExpenseCategory--&gt;</b>  <b>Before TeamConnect</b>	<b>APPL_INVD_EXP_CATEG_ID</b>	Post Invoice Expense of	<b>x</b>	<b>x</b>	<a href="#">YRecentlyViewedRecord</a>	object	8 Expense category for an invoice line item that is	<b>x</b>	<b>x</b>

3.3 SP2: applLineExpense Category-->	(NUMBER)	Category (Posting Criteria block)					specified as the posting criterion for the account.  --> Links to the definition information of the category.		
invoiceExpensePercent  Before TeamConnect 3.3 SP2: applLineExpense Percent	APPL_IN VD_EXP ENSE _PERCE NT (NUMBER)	Invoice Expense Percent (Posting Criteria block)	x			decimal	Percentage of the total expense line item amount allowed to be posted to the account.	x	x
invoiceTaskCategory  Before TeamConnect 3.3 SP2: applLineTask Category-->	APPL_IN VD_TAS K_CA TEGORY _ID (NUMBER)	Post Invoice Tasks of Type (Posting Criteria block)	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 Task category for an invoice line item that is specified as the posting criterion for the account.  --> Links to the definition information of the category.	x	x
invoiceTaskPercent  Before TeamConnect 3.3 SP2:	APPL_IN VD_TAS K_P RCENT (NUMBER)	Invoice Task Percent (Posting Criteria block)	x			decimal	Percent of the task invoice line item allowed to be posted	x	x

<b>applLineTask Percent</b>							to the account.  Note: This column is populated only when <b>allowInvoiceTask</b> is set to 1.		
<b>applInvcNonU STaxType--&gt;</b>	<b>APPL_N ON_US_ TAX_T YPE  (NUMBE R)</b>	Non-US Tax Type		<b>x</b>	<a href="#">LInvcNonUS TaxType</a>	obje ct	8 The tax type category associated with the tax amount.	<b>x</b>	<b>x</b>
<b>applInvcNonU STaxPercent</b>	<b>APPL_IN VC_NON US_ TAX_PE RCENT  (NUMBE R)</b>	Non-US Tax Percent  (Posting Criteria block)	<b>x</b>			deci mal	Percentage of the total non-US tax amount allowed to be posted to the account.	<b>x</b>	<b>x</b>
<b>involved--&gt;  Before TeamConnect 3.3 SP2: applInvolved- -&gt;</b>	<b>APPL_IN VOLVED _ID  (NUMBE R)</b>	This Involved  (Posting Criteria block)	<b>x</b>	<b>x</b>	<a href="#">TContact</a>	obje ct	8 The contact involved in the project who is specified as the posting criterion for the account.  --> Links to the specific information about the contact object.	<b>x</b>	<b>x</b>
<b>modifiedBy--&gt;</b>	<b>MODIFIE D_BY_ID</b>	Modified By	<b>x</b>	<b>x</b>	<a href="#">YUser</a>	obje ct	8 User who most recently		

	<b>(NUMBER)</b>	(Security block)					modified the account. --> Links to specific information about the user account.		
<b>modifiedOn</b>	<b>MODIFIED_ON</b> <b>(DATE)</b>	Modified On (Security block)	<b>x</b>			date	Date and time the account was most recently modified.		
<b>name</b>	<b>NAME</b> <b>(VARCHAR2)</b> <b>(250)</b>	Name (General block)	<b>x</b>			string (250)	Name used to describe the account.		<b>x</b>
<b>note--&gt;</b>	<b>NOTE_ID</b> <b>(NUMBER)</b>	Notes (General block)		<b>x</b>	<a href="#">JNote</a>	object	Links to the table that contains the text entered into the <b>Notes</b> field of the record.		
<b>parentAccount--&gt;</b>	<b>PARENT_ACCOUNT_ID</b> <b>(NUMBER)</b>	Parent Account (General block)	<b>x</b>	<b>x</b>	<a href="#">TAccount</a>	object	8 The account that is specified as the parent of this account. --> Links to specific information about the account record.	<b>x</b>	<b>x</b>

<b>primaryKey</b>	<b>PRIMARY_KEY (NUMBER)</b>		x			int	The unique ID of the account record.		
<b>project</b> <b>Before TeamConnect 3.3 SP2: applProject</b>	<b>PROJECT_ID (NUMBER)</b>	This Project  (Posting Criteria tab)	x	x	<a href="#">TProject</a>	object	8 Project that is specified as the posting criterion for the account.  --> Links to specific information about the project record.	x	x
<b>projectCategory--&gt;</b> <b>Before TeamConnect 3.3 SP2: applProjCategory--&gt;</b>	<b>APPL_PROJECT_CATEGORY_ID (NUMBER)</b>	Post Projects of Type  (Posting Criteria block)	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 Project category specified as the posting criterion for the account.  --> Links to the definition information of the category.	x	x
<b>securityTypeID</b>	<b>SECURITY_TYPE_ID (NUMBER)</b>	Security Type  (Security block)	x			int	Specifies whether the account is public or private.  0 - Public (PUBLIC)  2 - Private (PRIVATE)		

<b>startOn</b> Before TeamConnect 3.3 SP2: applStartOn	<b>APP_ST ART_ON</b>  (DATE)	Account period begins:  (General block)	<b>x</b>			date	Specifies the start date from which transac- tions can be posted to the account.	<b>x</b>	<b>x</b>
<b>taskCategory- -&gt;</b> Before TeamConnect 3.3 SP2: applTaskCate- gory-->	<b>APPL_T ASK_CA TEGOR Y_ID</b>  (NUMBE R)	Post Tasks of Type  (Posting Criteria block)	<b>x</b>	<b>x</b>	<a href="#">YRecentlyV iewedRecor d</a>	obje ct	8 Task category specified as the posting criterion for the account.  --> Links to the definition information of the category.	<b>x</b>	<b>x</b>
<b>taskPercent</b> Before TeamConnect 3.3 SP2: applTaskPerc ent	<b>APPL_T ASK_PE RCENT</b>  (NUMBE R)	Task Percent  (Posting Criteria block)	<b>x</b>			deci mal	Percentage of the total task amount allowed to be posted to the account.	<b>x</b>	<b>x</b>
<b>iisApplInvcSt andard</b>	<b>APPL_IN VC_STA NDAR D</b>  (NUMBE R)	Post Invoice Type	<b>x</b>			bool ean	Indicates whether standard invoices can be posted to the account.  0 - No  1 - Yes	<b>x</b>	<b>x</b>
<b>iisApplInvcAc crual</b>	<b>APPL_IN VC_ACC</b>	Post Invoice	<b>x</b>			bool ean	Indicates whether	<b>x</b>	<b>x</b>



	<b>RUAL</b> <b>(NUMBER)</b>	Type					accrual invoices can be posted to the account. 0 - No 1 - Yes		
<b>iisApplInvcCreditNote</b>	<b>APPL_INVC_CREDIT_NOTE</b> <b>(NUMBER)</b>	Post Invoice Type	<b>x</b>			boolean	Indicates whether credit notes can be posted to the account. 0 - No 1 - Yes	<b>x</b>	<b>x</b>
<b>iisApplInvcShadow</b>	<b>APPL_INVC_SHADOW</b> <b>(NUMBER)</b>	Post Invoice Type	<b>x</b>			boolean	Indicates whether shadow invoices can be posted to the account. 0 - No 1 - Yes	<b>x</b>	<b>x</b>
<b>iisApplInvcNonUSTax</b>	<b>APPL_NON_US_TAX</b> <b>(NUMBER)</b>	Post Non-US Tax	<b>x</b>			boolean	Indicates whether non-US tax amounts can be posted to the account. 0 - No 1 - Yes	<b>x</b>	<b>x</b>
<b>treeKey</b>	<b>TREE_KEY</b>		<b>x</b>			string	The chain of primary keys that		

	(VARCHAR2) (700)					(700)	identifies the parent tree of the account. Used for searching and reporting purposes only.		
<b>type</b> <b>Before TeamConnect 3.3 SP2: typeIID</b>	<b>TYPE_IID</b> <b>(NUMBER)</b>	Account Type (General block)	<b>x</b>			Enum	Indicates the account type.  1 - Budget (BUDGET)  2 - Reserve (RESERVE)	<b>x</b>	<b>x</b>
<b>used</b>	<b>TOTAL_USED</b> <b>(NUMBER)</b>	Total Used (General block)	<b>x</b>			decimal	Total amount of money used through transfers, withdrawals, and posted transactions.	<b>x</b>	
<b>userAccessList--&gt;</b>	n/a			<b>x</b>	<a href="#">EAcctUser Access</a>		Links to the user security information entered through the Security block of the record.		
<b>vendor</b> <b>Before TeamConnect</b>	<b>APPL_VENDOR_ID</b>	This Vendor	<b>x</b>	<b>x</b>	<a href="#">IContact</a>	object	8 Contact specified as the posting criterion for	<b>x</b>	<b>x</b>

<p><b>3.3 SP2: applVendor</b></p>	<p><b>(NUMBER)</b></p>	<p>(Posting Criteria block)</p>					<p>the account.  --&gt; Links to the specific information about the contact object.</p>		
<p><b>version</b></p>	<p><b>VERSION (NUMBER)</b></p>		<p><b>x</b></p>			<p>int</p>	<p>Indicates how many times the account record was updated.</p>		
<p><b>postedType</b></p>	<p><b>POSTED_TYPE_ID (NUMBER)</b></p>		<p><b>x</b></p>			<p>int</p>	<p>Indicates which types of transactions have previously posted to this account.  0 - None (NONE_POSTED)  1 - Non-shadow (NON_SHADOW_POSTED)  2 - Shadow invoices (SHADOW_POSTED)</p>	<p><b>x</b></p>	<p><b>x</b></p>

**3.1.30.2 JTranDetail**

JTranDetail contains information about the transactions that have been posted against an account record. This information includes the account to which the transaction belongs, a short description, total amount, and whether the amount was deposited or withdrawn. It also links to [RTransaction](#), which contains the transaction to which the transaction detail belongs. This table represents the Transactions block on the Transactions tab of an account record.

**Object: JTranDetail (J\_TRAN\_DETAIL)**

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wizard
account->	ACCOUNT_ID (NUMBER)	Account (Transaction block of Expense, Invoice, or Task)	x	x	<a href="#">TAccount</a>	object	8 The account to which the transaction detail belongs.  --> Links to specific information about the account record.		
objectID	OBJECT_ID (NUMBER)	Used to display the link to the associated record in the <b>Transactions</b> block.	x			int	The primary key of the record (such as an account, expense, or task record) with which the transaction detail is associated.		
owner-->	ACCOUNT_TRANS_ID (NUMBER)			x	<a href="#">RTransaction</a>	object	The transaction to which this transaction detail belongs.		
primary Key	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the transaction detail.		
shortDescription	SHORT_DESCRIPTION (VARCHAR2) (250)	Name (Transactions block, Account)	x			string (250)	Specifies the name of the financial transaction. Depending on the object, the		

							<p>short description is equal to one of the following:</p> <ul style="list-style-type: none"> <li>• Description of the transfer, deposit, withdrawal, or expense</li> <li>• Task subject</li> <li>• Invoice number and line item type (task or expense) if it is posted</li> </ul>		
<b>statusTypeID</b>	<b>STATUS_TYPE_ID</b> (NUMBER)		x			int	<p>Indicates whether the amount was deposited to the account or withdrawn from the account.</p> <p>1 - Deposit 2 - Withdraw</p>		
<b>totalAmount</b>	<b>TOTAL_AMOUNT</b> (NUMBER)	<p>One of the following:</p> <ul style="list-style-type: none"> <li>• <b>Amount</b> (Transactions block of Expense, Invoice, or Task)</li> </ul>	x			decimal	<p>The total amount that was deposited, withdrawn, transferred, posted or voided from/to the account.</p>		

		<ul style="list-style-type: none"> <li>Debit / Credit (Account Transactions block)</li> </ul>							
uniqueCode	TRANS_UNIQUE_CODE (VARCHAR2) (4)	Used in conjunction with objectID to display the link to the associated record in the Transactions block.	x			string (4)	The 4-character unique code of the object to which the record associated with the transaction detail belongs.		
version	VERSION (NUMBER)		x			int	Indicates how many times the transaction detail was updated.		

### 3.1.30.3 RTransaction

RTransaction contains information about instances of complete financial transactions, usually comprised of smaller transactions across multiple accounts. This information includes the user who created the transaction and the transaction date. It also links to [JTranDetail](#), which contains transaction details for particular accounts. This table represents the **Transaction** tab of an invoice record, which displays the entire transaction.

Object: RTransaction (R\_TRANSACTION)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp /

									Wi z
createdBy-->	CREATED_BY_ID (NUMBER)		x	x	<a href="#">YUser</a>	obj ect	8 The user who created the financial transaction.  --> Links to specific information about the user account.		
createdOn	CREATED_ON (DATE)	Created On (Transaction block of Expense, Task or Invoice)	x			dat e	The date the transaction was created.		
modifiedBy-->	MODIFIED_BY (NUMBER)		x	x	<a href="#">YUser</a>	obj ect	8 The user who voided the expense, task, or invoice.  --> Links to specific information about the user account.		
modifiedOn	MODIFIED_ON (DATE)		x			dat e	The date the expense, task, or invoice was voided.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the financial transaction.		
shortDescription	SHORT_DESCRIPTION (VARCHAR 2) (250)						Currently not used.		

<b>transaction Date</b>	<b>TRANSACTION_DATE (DATE)</b>	<b>Date (Transaction block in Account)</b>	<b>x</b>			<b>date</b>	The date the transaction took place.		
<b>transaction DetailList--&gt;</b>	n/a		<b>x</b>	<b>x</b>	<a href="#">JTran Detail</a>		Links to the join table that contains information for all transaction details of this financial transaction.		
<b>version</b>	<b>VERSION (NUMBER)</b>		<b>x</b>			<b>int</b>	Indicates how many times the transaction has been updated.  Note: The version is updated only when financial transactions are voided or re-posted.		

### 3.1.30.4 EAcctDetail

EAcctDetail contains all categories added to account records. Each time a category is added to an account record, an entry is made in this table in the database.

In Object Navigator, **EAcctDetail** is not displayed when you traverse using the **detailList-->** bridge. Instead, a list of all categories that have been defined for the object definition appears. This list is labeled **Category**. From each category, you can traverse to a table called **Detail Fields**, which displays all custom fields belonging to the category you selected.

For more information about how custom field values are stored in the database, see Object Model: Custom Fields.

Object: EAcctDetail (E\_ACCT\_DETAIL)

<b>Attribute</b>	<b>Database column name</b>	<b>Field in UI</b>	<b>End of path</b>	<b>Bridge</b>	<b>Links to object table:</b>	<b>Data type</b>	<b>Comments</b>
------------------	-----------------------------	--------------------	--------------------	---------------	-------------------------------	------------------	-----------------



category-->	CATEGORY_ID	Category	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 The category of the account with which the details are associated.  --> Links to the definition information of the category.
isManual	IS_MANUAL (NUMBER)		x			boolean	Specifies whether the category is added manually or automatically to the account. Categories are added automatically when they are the parent of the category that is added manually.  0 - Added Automatically 1 - Added Manually
owner-->	ENTERPRISE_OBJECT_ID (NUMBER)			x	<a href="#">TAccount</a>	object	The account with which the custom fields are associated.
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the account category record.
version	VERSION (NUMBER)		x			int	Indicates how many times the added category was updated.

### 3.1.30.5 EAcctUserAccess

EAcctUserAccess contains the user access information that is set in the **Security** block of each account record. For example, it includes whether each user that is listed in the Security block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm**(ission) rights.

Object: EAcctUserAccess (E\_ACCT\_USER\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table :	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz

<b>isRead</b>	<b>IS_READ (NUMBER)</b>	<b>Read checkbox</b>	<b>x</b>			int	<p>Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
<b>isUpdate</b>	<b>IS_UPDATE (NUMBER)</b>	<b>Update checkbox</b>	<b>x</b>			int	<p>Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
<b>isDelete</b>	<b>IS_DELETE (NUMBER)</b>	<b>Delete checkbox</b>	<b>x</b>			int	<p>Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
<b>isPerm</b>	<b>IS_PERM (NUMBER)</b>	<b>Perm checkbox</b>	<b>x</b>			int	<p>Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p>		

							This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>allowDenyIID</b>	<b>ALLOW_DENY_IID</b> (CHAR) (1)	<b>Option</b>	<b>x</b>			char	Indicates whether the user is allowed or denied access to the record for purposes of the operations described in isRead, isUpdate, isDelete, and isPerm.  a - Allow (ALLOW) d - Deny (DENY)		
<b>isManual</b>	<b>IS_MANUAL</b> (NUMBER)		<b>x</b>			boolean	Indicates whether the user rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
<b>owner-&gt;</b>	<b>ENTERPRISE_OBJECT_ID</b> (NUMBER)		<b>x</b>	<b>x</b>	<a href="#">TAccount</a>	object	8 The account record with which this user access setting is associated.  -> Links to the specific information about the account record.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		<b>x</b>			int	Unique ID for the user access rights.		
<b>user-&gt;</b>	<b>USER_ID</b>	<b>List</b>	<b>x</b>	<b>x</b>	<a href="#">YUser</a>	object	8 Specifies the user the access rights		

	(NUMBER)						pertain to.  --> Links to the specific information about the user account.		
version	VERSION (NUMBER)		x			int	Indicates how many times the user access rights have been updated.		

### 3.1.30.6 EAcctGroupAccess

EAcctGroupAccess contains the group access information that is set in the **Security** block of each account record. For example, it includes whether each group that is listed in the Security block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm(ission)** rights.

Object: EAcctGroupAccess (E\_ACCT\_GROUP\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
isRead	IS_READ (NUMBER)	Read checkbox	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isUpdate	IS_UPDATE (NUMBER)	Update checkbox	x			int	Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).		

							This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isDelete	IS_DELETE (NUMBER)	Delete checkbox	x			int	Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isPerm	IS_PERM (NUMBER)	Perm checkbox	x			int	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
allowDenyIID	ALLOW_DENY_IID (CHAR) (1)	Option	x			char	Indicates whether the group is allowed or denied access to the record.  a - Allow (ALLOW) d - Deny (DENY)		
group-->	GROUP_ID (NUMBER)	List	x	x	<a href="#">YGroup</a>	object	8 Specifies the group that the access rights pertain to.		

							--> Links to the specific information about the group account.		
<b>isManual</b>	<b>IS_MANUAL</b> (NUMBER)		<b>x</b>			boolean	Indicates whether the group access rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
<b>owner-&gt;</b>	<b>ENTERPRISE_OBJECT_ID</b> (NUMBER)		<b>x</b>	<b>x</b>	<a href="#">TAccount</a>	object	8 The account record with which this group access setting is associated.  --> Links to the specific information about the account record.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		<b>x</b>			int	Unique ID for the group access rights.		
<b>version</b>	<b>VERSION</b> (NUMBER)		<b>x</b>			int	Indicates how many times the group access rights have been updated.		

### 3.1.31 Object Model: Expenses

The tables in this appendix provide information about the object model as it relates to expenses.

### 3.1.31.1 TExpense

TExpense contains the general information for an expense record. This information includes the expense's date, the user who made the expense, unit price of the item or service, quantity, total amount, posting status, description, and so on. This table links to the list of custom fields created for the expense categories and other related information contained in other tables.

Object: Expense (T\_EXPENSE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wiz
<b>categories--&gt;</b>  <b>Before TeamConnect 3.3 SP2: detailList --&gt;</b>	n/a		x	x	<a href="#">EExpeDetail</a>  In Object Navigator, links to <b>Category</b> list which appears in the UI only.		8 The list of categories added to the current object.  --> In the object model, the added categories link to the values that have been added for custom fields, according to field type.  --> In Object Navigator, this attribute is enhanced. It links to a list of all categories defined for the object definition, rather than those added to the current object. From there, you can select a category and traverse to the list of custom		

							fields that belong to it.		
contact-->	CONTACT_ID (NUMBER)	Contact (General block)	x	x	<a href="#">TContact</a>	object	8 Vendor who is associated with the expense.  --> Links to the specific information about the contact object.	x	x
createdBy-->	CREATED_BY_ID (NUMBER)	Created By (Security block)	x	x	<a href="#">YUser</a>	object	8 User who created the expense record.  --> Links to specific information about the user account.	x	x
createdOn	CREATED_ON (DATE)	Created On (Security block)	x			date	Date and time the expense record was created.		x
createdOnBehalfOf	CREATED_ON_BEHALF_OF_ID			x	<a href="#">TContact</a>	object	When not null, it indicates that the record was created through an e-billing system or other external application. The contact referenced will often be a vendor in TeamConnect.	x	
defaultCategory-->	DEFAULT_CATEGORY_ID	Default Category	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 Default expense category.		



	(NUMBER)	(General block)					--> Links to the definition information of the category.		
documentFolder-->	DOCUMENT_FOLDER_ID (NUMBER)	Documents block (General block)		x	<a href="#">TDocument</a>	object	Links to specific information about the document folder.		
expense Date	EXPENSE_DATE (DATE)	Expensed On (General block)	x			date	Date the expense was incurred. Time-zone-independent.	x	x
expenseBy-->	EXPENSED_BY_ID (NUMBER)	Expensed By (General block)	x	x	<a href="#">YUser</a>	object	8 User who is responsible for the expenditure. This is not necessarily the user who created the expense record.  --> Links to specific information about the user account.	x	x
groupAccessList-->	n/a			x	<a href="#">EExpenseGroupAccess</a>		Links to the group security information entered through the Security block of the record.		
historyList-->	n/a	History tab	x	x	<a href="#">THistory</a>		8 The list of all histories that are related to		

							the expense record. --> Links to specific history information for all of the expense's history records.		
modified By-->	MODIFIED_BY_ID (NUMBER)	Modified By Security block)	x	x	<a href="#">YUser</a>	object	8 User who most recently modified the expense.  --> Links to specific information about the user account.		
modified On	MODIFIED_ON (DATE)	Modified On (Security block)	x			date	Date and time the expense record was most recently modified.		
note-->	NOTE_ID (NUMBER)	Notes (General block)		x	<a href="#">JNote</a>	object	Links to the table that contains the text entered into the <b>Notes</b> field of the record.		
postingStatus  Before TeamConnect 3.3 SP2: postingStatusID	STATUS_IID (NUMBER)	Posting Status (General block)	x			int	Specifies whether the expense was posted or not:  1 - Not Posted 2 - Posted 3 - Failed 4 - Posted For Approval	x	
primaryKey	PRIMARY_KEY		x			int	The unique ID of the expense		

	(NUMBER)						record.		
project-->	PROJECT_ID (NUMBER)	Project (General block)	x	x	<a href="#">Project</a>	object	8 The project for which the expense was incurred.  --> Links to specific information about the project record.	x	x
quantity	QUANTITY (NUMBER)	Quantity (General block)	x			decimal	Either the total number of units bought or the total length of time spent to complete the service (Hours.Minutes).	x	x
securityTypeID	SECURITY_TYPE_ID (NUMBER)	Security Type (Security block)	x			int	Specifies whether the expense is public or private.  0 - Public 2 - Private		
shortDescription	SHORT_DESCRIPTION (VARCHAR2) (250)	Description (General block)	x			string (250)	Description of the expense.		x
totalAmount	TOTAL_AMOUNT (NUMBER)	Total (General block)	x			decimal	Total amount of money spent for the expense that is the product of the price per unit/hour ( <b>unitPrice</b> ) and the number of	x	x

							units/hours (quantity).		
transaction-->	TRANSACTION_ID (NUMBER)	Transactions block		x	<a href="#">RTransaction</a>	object	Links to the transactions associated with the expense record.		
unitPrice	UNIT_PRICE (NUMBER)	Unit Price (General block)	x			decimal	Either the price per unit or the price per hour of service.	x	x
userAccessList-->	n/a			x	<a href="#">EExpenseUserAccess</a>		Links to the user security information entered through the Security block of the record.		
version	VERSION (NUMBER)		x			int	Indicates how many times the expense record was updated.		

### 3.1.31.2 EExpeDetail

EExpeDetail contains all categories added to expense records. Each time a category is added to an expense record, an entry is made in this table.

In Object Navigator, each table **Category** displays only categories belonging to the selected object. From each category, you can traverse to a table called **Detail Fields**, which displays only custom fields belonging to the category you selected.

For more information about how custom field values are stored in the database, see Object Model: Custom Fields.

Object: EExpeDetail (E\_EXPE\_DETAIL)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp

									/
									Wi
									Z
category-->	CATEGORY_ID (NUMBER)	Category	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 The category of the expense with which the details are associated.  --> Links to the definition information of the category.		
owner-->	ENTERPRISE_OBJECT_ID (NUMBER)			x	<a href="#">Expense</a>	object	The expense with which the custom fields are associated.		
isManual	IS_MANUAL (NUMBER)		x			boolean	Specifies whether the category is added manually or automatically to the expense. Categories are added automatically when they are the parent of the category that is added manually.  0 - Added Automatically  1 - Added Manually		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the expense category record.		
version	VERSION (NUMBER)		x			int	Indicates how many times the added category was updated.		

### 3.1.31.3 EExpeUserAccess

EExpeUserAccess contains the user access information that is set in the **Security** block of each expense record. For example, it includes whether each user that is listed in the Security block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm(ission)** rights.

Object: EExpeUserAccess (E\_EXPE\_USER\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table :	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
isRead	IS_READ (NUMBER)	Read checkbox	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isUpdate	IS_UPDATE (NUMBER)	Update checkbox	x			int	Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isDelete	IS_DELETE (NUMBER)	Delete checkbox	x			int	Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed		

							or denied. (See allowDenyIID.)		
<b>isPerm</b>	<b>IS_PERM (NUMBER)</b>	<b>Per m che ck box</b>	<b>x</b>			int	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>allowDenyIID</b>	<b>ALLOW_DENY_IID (NUMBER)</b>	<b>Opt ion</b>	<b>x</b>			char	Indicates whether the user is allowed or denied access to the record.  a - Allow d - Deny		
<b>isManual</b>	<b>IS_MANUAL (NUMBER)</b>		<b>x</b>			boolean	Indicates whether the user rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
<b>owner-&gt;</b>	<b>ENTERPRISE_OBJECT_ID (NUMBER)</b>		<b>x</b>	<b>x</b>	<a href="#">Expense</a>	object	8 The expense record with which this user access setting is associated.  --> Links to the specific information about the expense record.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b>		<b>x</b>			int	Unique ID for the user access rights.		

	(NUMBER)								
user-->	USER_ID (NUMBER)	List	x	x	<a href="#">YUse</a> r	obje ct	8 Specifies the user the access rights pertain to.  --> Links to the specific information about the user account.		
versio n	VERSION (NUMBER)		x			int	Indicates how many times the user access rights have been updated.		

### 3.1.31.4 EExpeGroupAccess

EExpeGroupAccess contains the group access information that is set in the **Security** block of each expense record. For example, it includes whether each group that is listed in the Security block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm**(ission) rights.

Object: EExpeGroupAccess (E\_EXPE\_GROUP\_ACCESS)

Attribut e	Database column name	Fiel d in UI	En d of pat h	Bri dge	Link s to obje ct tabl e:	Dat a typ e	Comments	Commonl y used in:	
								Rul es	Te mp/ Wiz
isRead	IS_READ (NUMBER)	Rea d chec k box	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isUpda te	IS_UPDATE (NUMBER)	Upd ate chec k box	x			int	Specifies whether the Update operation has been selected for granting or denial.		



							Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isDelete</b>	<b>IS_DELETE (NUMBER)</b>	<b>Delete checkbox</b>	<b>x</b>			int	Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isPerm</b>	<b>IS_PERM (NUMBER)</b>	<b>Perm checkbox</b>	<b>x</b>			int	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>allowDenyIID</b>	<b>ALLOW_DENY_IID (CHAR) (1)</b>	<b>Option</b>	<b>x</b>			char	Indicates whether the group is allowed or denied access to the record.  a - Allow d - Deny		
<b>group--&gt;</b>	<b>GROUP_ID (NUMBER)</b>	<b>List</b>	<b>x</b>	<b>x</b>	<a href="#">YGroup</a>	object	8 Specifies the group that the access rights pertain to.		

							--> Links to the specific information about the group account.		
<b>isManual</b>	<b>IS_MANUAL</b> (NUMBER)		<b>x</b>			boolean	Indicates whether the group access rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
<b>owner-&gt;</b>	<b>ENTERPRISE_OBJECT_ID</b> (NUMBER)		<b>x</b>	<b>x</b>	<a href="#">Expense</a>	object	8 The record with which this group access setting is associated.  --> Links to the specific information about the expense record.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		<b>x</b>			int	Unique ID for the group access rights.		
<b>version</b>	<b>VERSION</b> (NUMBER)		<b>x</b>			int	Indicates how many times the group access rights have been updated.		

### 3.1.32 Object Model: Tasks

The tables in this appendix provide information about the object model as it relates to tasks.

#### 3.1.32.1 TTask

TTask contains the general information for a task record including the task's due date, estimated hours, rate amount, start-on date, and completed date. This table links to the list of custom fields created for the task categories and other related information contained in other tables.

Object: TTask (T\_TASK)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wiz
acceptanceIID	ACCEPTANCE_IID (NUMBER)						Currently not used.		
activityItem-->	ACTIVITY_ITEM_ID (NUMBER)	Activity (General block)	x	x	<a href="#">LTaskActivityItem</a>	object	8 Activity item specified for the task.  --> Links to the definition information for the task activity item.	x	x
actualHours	ACTUAL_HOURS (NUMBER)	Actual Work (General block)	x			decimal	Actual length of time spent to complete the task (Hours.Minutes).	x	
assigneeHistory-->  Before TeamConnect 3.3 SP2: assigneeList-->	n/a			x	<a href="#">JTaskAssignee</a>		Links to the join table that contains information for all assignees added to the task.	x	x
completedDate  Before TeamConnect 3.3 SP2:	COMPLETE_DATE (DATE)	Completed On	x			date	Date and time the task was completed. Time-zone-independent.	x	

completedOn		(General block)							
completedPercent	COMPLETE_PERCENT (NUMBER)	% Complete (General block)	x			decimal	Percentage of the task that is currently complete.	x	
contact-->	CONTACT_ID (NUMBER)	Contact (General block)	x	x	<a href="#">TContact</a>	object	8 Contact who is associated to the task, e.g. for whom the task is performed.  --> Links to the specific information about the contact object.		x
createdBy-->	CREATED_BY_ID (NUMBER)	Created By (Security block)	x	x	<a href="#">YUser</a>	object	8 User who created the task record.  --> Links to specific information about the user account.	x	x
createdOn	CREATED_ON (DATE)	Created On (Security block)	x			date	Date and time the task record was created.		x
createdOnBehalfOf	CREATED_ON_BEHALF_OF_ID			x	<a href="#">TContact</a>	object	When not null, it indicates that the record was created through an e-	X	

							billing system or other external application. The contact referenced will often be a vendor in TeamConnect.		
currentAssignee-->	CURRENT_ASSIGNEE_ID (NUMBER)	Assigned To (General block)		x	<a href="#">JTaskAssignee</a>	object	Links to the definition information of the user who is currently assigned to the task.		x
currentAssigneeUser-->	CURRENT_ASSIGNEE_USER_ID (NUMBER)			x	<a href="#">YUser</a>	object	The user who is currently assigned to the task. It is more efficient to use this field in searches and queries than to use the field "currentAssignee" to access user info.	x	x
defaultCategory-->	DEFAULT_CATEGORY_ID (NUMBER)	Default Category (General block)	x	x	<a href="#">YRecently ViewedRecord</a>	object	8 Default task category. --> Links to the definition information of the category.		x
detailList-->	n/a		x	x	<a href="#">ETaskDetail</a>  In Object Navigator, links to		8 The list of categories added to the current object.		

					<b>Category</b> list which appears in the UI only.		<p>--&gt; In the object model, the added categories link to the values that have been added for custom fields, according to field type.</p> <p>--&gt; In Object Navigator, this attribute is enhanced. It links to a list of all categories defined for the object definition, rather than those added to the current object. From there, you can select a category and traverse to the list of custom fields that belong to it.</p>		
documentFolder-->	DOCUMENT_FOLDER_ID (NUMBER)	Documents block		x	<a href="#">TDocument</a>	object	Links to specific information about the document folder.		
dueDate Before TeamConnect 3.3 SP2: dueOn	DUE_ON (DATE)	Due On (General block)	x			date	The date when the task is due. Time-zone-independent.	x	x

<b>estimatedHours</b>	<b>ESTIMATE_D_HOURS</b> (NUMBER)	<b>Estimated Duration</b> (General block)	x			decimal	Amount of time (Hours.Minutes) needed to complete the task.		x
<b>forwardedByAssignee--&gt;</b>	<b>FORWARDED_BY_ASSIGNEE_ID</b> (NUMBER)			x	<a href="#">JTaskAssignee</a>	object	Links to the definition information of the user who forwarded the task to the current assignee.		
<b>groupAccessList--&gt;</b>	n/a			x	<a href="#">ETaskGroupAccess</a>		Links to the group security information entered through the Security block of the record.		
<b>historyList--&gt;</b>	n/a	<b>History tab</b>	x	x	<a href="#">THistory</a>		8 The list of all histories that are related to the task record.  --> Links to specific history information for all of the task's history records.		
<b>modifiedBy--&gt;</b>	<b>MODIFIED_BY_ID</b> (NUMBER)	<b>Modified By</b> (Security block)	x	x	<a href="#">YUser</a>	object	8 User who most recently modified the task.  --> Links to specific information		

							about the user account.		
modifiedOn	MODIFIED_ON (DATE)	Modified On (Security block)	x			date	Date and time the task was most recently modified.		
note-->	NOTE_ID (NUMBER)	Notes (General block)		x	<a href="#">JNote</a>	object	Links to the table that contains the text entered into the <b>Notes</b> field of the record.		
postingStatus  Before TeamConnect 3.3 SP2: postingStatusID	STATUS_ID (NUMBER)	Posting Status (General block)	x			Enum	This enumeration is based on TAccount.  Specifies whether the task was posted or not:  1 - Not Posted  2 - Posted  3 - Failed  4 - Posted For Approval	x	
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the task record.		
priority  Before TeamConnect 3.3 SP2: priorityID	PRIORITY_ID (NUMBER)	Priority (General block)	x			int	Task priority.  1 - Highest  2 - High  3 - Normal  4 - Low	x	x



							5 - Lowest		
project-->	PROJECT_ID (NUMBER)	Project (General block)	x	x	<a href="#">TProject</a>	object	8 Project record for which this task record is created.  --> Links to specific information about the project record.	x	x
rateAmount	RATE_AMOUNT (NUMBER)	Rate (General block)	x			decimal	Rate the user charges (per hour) to perform the task.	x	x
securityTypeIID	SECURITY_TYPE_IID (NUMBER)	Security Type (Security block)	x			int	Specifies whether the task is public or private.  0 - Public 2 - Private		
shortDescription	SHORT_DESCRIPTION (VARCHAR2) (250)	Subject (General block)	x			string (250)	Description of the task.		x
startDate Before TeamConnect 3.3 SP2: startOn	START_ON (DATE)	Start On (General block)	x			date	Date when the task was started. Time-zone-independent.	x	x
totalAmount	TOTAL_AMOUNT (NUMBER)	Total (General block)	x			decimal	Total amount of money spent to completing the task as the product of	x	x

							the rate ( <b>rateAmount</b> ) and amount of time spent ( <b>actualHours</b> ).		
transaction->	TRANSACTION_ID (NUMBER)	Transactions block		x	<a href="#">RTransaction</a>	object	Links to the transactions associated with the task record.		
userAccess List-->	n/a			x	<a href="#">ETaskUser Access</a>		Links to the user security information entered through the Security block of the record.		
version	VERSION (NUMBER)		x			int	Numeric instance that tells you how many times each row was modified.		
workStatus Before TeamConnect 3.3 SP2: workStatusID	WORK_STATUS_ID (NUMBER)	Status (General block)				int	The current status of the task.  1 - Not Started  2 - Started  3 - Completed		

### 3.1.32.2 LTaskActivityItem

LTaskActivityItem contains information about the activity items defined for task records. This information includes the name of the activity item, tree position, and display order. This table represents the **Activity Item** system lookup table.

Object: LTaskActivityItem (L\_TASK\_ACTIVITY\_ITEM)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
childList-->	n/a						Do not apply. These lookup items cannot have a hierarchical structure.		
parent-->	PARENT_ID (NUMBER)								
displayOrder	DISPLAY_ORDER (NUMBER)	Order	x			int	The order in which the table items will be listed in the GUI, lowest number first.		
isActive	IS_ACTIVE	activate and inactivate buttons	x			int	Indicates whether the table entry is active (value 1) and can be used when editing records containing that entry. If it is inactive (value 0), records may contain that entry for reporting and historical purposes, but cannot be saved when editing, unless that entry is changed to some other, active table entry.		
name	NAME (VARCHAR2) (50)	Item Name	x			string (50)	Name of activity item.		
primaryKey	PRIMARY_KEY		x			int	The unique ID of the activity item.		

	(NUMBER)								
treePosition	TREE_POSITION (VARCHAR2) (250)	Tree Position	x			string (250)	The unique four-character alphanumeric code that is created when the activity item is defined.		
version	VERSION (NUMBER)		x			int	Indicates how many times the activity item was updated.		

### 3.1.32.3 JTaskAssignee

JTaskAssignee contains information about the assignees of a task record. This information includes all past assignees as well as the current assignee and the date assigned. This table represents the **Task Assignees** block on the **Assignees** tab of a task record.

Object: JTaskAssignee (J\_TASK\_ASSIGNEE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
assignedOn	ASSIGNED_ON (DATE)	Assigned On (Assignees block)	x			date	Date and time when the user was assigned to the task.		x
isInformed	IS_ASSIGNEE_INFORMED (NUMBER)						Currently not used.		x
owner -->	TASK_ID (NUMBER)			x	<a href="#">Task</a>	object	The task the user is assigned to.		x

primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the task assignee.		
user-->	USER_ID (NUMBER)		x	x	<a href="#">YUser</a>	object	8 The user assigned to the task. --> Links to specific information about the user account.		
version	VERSION (NUMBER)		x			int	Indicates how many times the activity item was updated.		

### 3.1.32.4 ETaskDetail

ETaskDetail contains all categories added to task records. Each time a category is added to a task record, an entry is made in this table in the database.

In Object Navigator, a list of all categories that have been defined for the current object definition appears. This list is labeled **Category**. From each category, you can traverse to a table called **Detail Fields**, which displays all custom fields belonging to the category you selected.

For more information about how custom field values are stored in the database, see Object Model: Custom Fields.

Object: ETaskDetail (E\_TASK\_DETAIL)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wizard
category-->	CATEGORY_ID (NUMBER)	Category	x	x	<a href="#">YRecentlyViewedRecord</a>	object	The category of the task with which the details are associated.		
isManual	IS_MANUAL (NUMBER)		x			boolean	Specifies whether the category is added manually or		

							automatically to the task. Categories are added automatically when they are the parent of the category that is added manually.  0 - Added Automatically  1 - Added Manually		
owner-->	ENTERPRISE_OBJECT_ID (NUMBER)			x	<a href="#">ITask</a>	object	The task with which the custom fields are associated.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the task category record.		
version	VERSION (NUMBER)		x			int	Indicates how many times the added category was updated.		

### 3.1.32.5 ETaskUserAccess

ETaskUserAccess contains the user access information that is set in the **Security** block of each task record. For example, it includes whether each user that is listed in the Security block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm**(ission) rights.

Object: ETaskUserAccess (E\_TASK\_USER\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table :	Data type	Comments	Commonly used in:	
								Rules	Temp / Wizard

<b>isRead</b>	<b>IS_READ (NUMBER)</b>	<b>Read check box</b>	<b>x</b>			int	<p>Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
<b>isUpdate</b>	<b>IS_UPDATE (NUMBER)</b>	<b>Update check box</b>	<b>x</b>			int	<p>Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
<b>isDelete</b>	<b>IS_DELETE (NUMBER)</b>	<b>Delete check box</b>	<b>x</b>			int	<p>Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
<b>isPerm</b>	<b>IS_PERM (NUMBER)</b>	<b>Perm check box</b>	<b>x</b>			int	<p>Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed</p>		

							or denied. (See allowDenyIID.)		
allowDenyIID	ALLOW_DENY_IID (CHAR)	Option	x			char	Indicates whether the user is allowed or denied access to the record.  a - Allow d - Deny		
isManual	IS_MANUAL (NUMBER)		x			boolean	Indicates whether the user rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
owner->	ENTERPRISE_OBJECT_ID (NUMBER)		x	x	<a href="#">TTask</a>	object	8 The task record with which this user access setting is associated.  --> Links to the specific information about the task record.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	Unique ID for the user access rights.		
user-->	USER_ID (NUMBER)	List	x	x	<a href="#">YUser</a>	object	8 Specifies the user the access rights pertain to.  --> Links to the specific information about the user account.		
version	VERSION (NUMBER)		x			int	Indicates how many times the user access		



								rights have been updated.		
--	--	--	--	--	--	--	--	---------------------------	--	--

### 3.1.32.6 ETaskGroupAccess

ETaskGroupAccess contains the group access information that is set in the **Security** block of each task record. For example, it includes whether each group that is listed in the Security block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm(ission)** rights.

Object: ETaskGroupAccess (E\_TASK\_GROUP\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
isRead	IS_READ (NUMBER)	Read checkbox	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isUpdate	IS_UPDATE (NUMBER)	Update checkbox	x			int	Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isDelete	IS_DELETE (NUMBER)	Delete checkbox	x			int	Specifies whether the Delete operation has been selected for granting or denial.		

							Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isPerm</b>	<b>IS_PERM (NUMBER)</b>	<b>Per m che ck box</b>	<b>x</b>			int	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>allowD enyIID</b>	<b>ALLOW_DEN Y_IID (CHAR) (1)</b>	<b>Opt ion</b>	<b>x</b>			char	Indicates whether the group is allowed or denied access to the record.  a - Allow d - Deny		
<b>group-- &gt;</b>	<b>GROUP_ID (NUMBER)</b>	<b>List</b>	<b>x</b>	<b>x</b>	<a href="#">YGroup</a>	obje ct	8 Specifies the group that the access rights pertain to.  --> Links to the specific information about the group account.		
<b>isManu al</b>	<b>IS_MANUAL (NUMBER)</b>		<b>x</b>			bool ean	Indicates whether the group access rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually		

							1 - Assigned automatically		
owner->	ENTERPRISE_OBJECT_ID (NUMBER)		x	x	<a href="#">Task</a>	object	8 The task record with which this group access setting is associated.  --> Links to the specific information about the task record.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	Unique ID for the group access rights.		
version	VERSION (NUMBER)		x			int	Indicates how many times the group access rights have been updated.		

### 3.1.33 Object Model: Invoices

The tables in this appendix provide information about the object model as it relates to invoices.

#### 3.1.33.1 TInvoice

TInvoice contains the general information for an invoice record. This information includes the invoice's creation date, received date, posting status, vendor, currency, exchange rate, and so on. This table also links to the list of custom fields created for the invoice categories, as well as the list of line items that have been added to the invoice.

Object: TInvoice (T\_INVOICE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wiz
categories-->	n/a		x	x	<a href="#">EInvcDetail</a> In Object Navigator,		8 The list of categories added to the		

<p><b>Before TeamConnect 3.3 SP2: detailList--&gt;</b></p>					<p>links to <b>Category</b> list which appears in the UI only.</p>		<p>current object.</p> <p>--&gt; In the object model, the added categories link to the values that have been added for custom fields, according to field type.</p> <p>--&gt; In Object Navigator, this attribute is enhanced. It links to a list of all categories defined for the object definition, rather than those added to the current object. From there, you can select a category and traverse to the list of custom fields that belong to it.</p>		
<p><b>comment</b></p>	<p><b>COMMENTS</b> (VARCHAR2) (2053)</p>	<p><b>Comments</b> (General block)</p>	<p><b>x</b></p>			<p>string (2053)</p>	<p>The comments intended to appear on the invoice for the vendor.</p>		

<b>createdBy--&gt;</b>	<b>CREATE_D_BY_ID</b>  (NUMBER)	<b>Created By</b>  (Security block)	x	x	<a href="#">YUser</a>	object	8 The user who created the invoice record.  --> Links to specific information about the user account.	x	x
<b>createdOn</b>	<b>CREATE_D_ON</b>  (DATE)	<b>Created On</b>  (Security block)	x			date	Date the invoice was created.	x	x
<b>createdOnBehalfOf</b>	<b>CREATE_D_ON_BEHALF_OF_ID</b>			x	<a href="#">TContact</a>	object	When not null, it indicates that the record was created through an e-billing system or other external application. The contact referenced will often be a vendor in TeamConnect.	X	
<b>currency--&gt;</b>	<b>CURRENCY_ID</b>  (NUMBER)	<b>Currency</b>  (General block)		x	<a href="#">YCurrencyInfo</a>	object	8 The currency selected for the invoice.  --> Links to the definition information of the currency.		

defaultCategory-->	DEFAULT_CATEGORY_ID (NUMBER)	Default Category (Categories block)			<a href="#">YRecentlyViewedRecord</a>	object	8 The default category for the invoice. --> Links to the definition information of the category.		x
documentFolder-->	DOCUMENT_FOLDER_ID (NUMBER)	Documents block		x	<a href="#">TDocument</a>	object	Links to specific information about the document folder.		
exchangeRate	EXCHANGE_RATE (NUMBER)		x			decimal	The exchange rate of the invoice.		
expenseAdjustmentTotal	EXPENSE_ADJUSTMENT_TOTAL (NUMBER)	Invoice Header - Total Fees & Expenses section, Expenses row Adjustments column intersection	x			decimal	Sum of line item expense adjustments	x	x
expenseDiscountTotal	EXPENSE_DISCOUNT_TOTAL (NUMBER)	Invoice Header - Total Fees & Expenses section, Expenses row Discount	x			decimal	Sum of line item expense original discounts	x	x

		ts column intersect ion							
expenseTaxTotal	EXPENSE_TAX_TOTAL (NUMBER)	Invoice Header - Total Fees & Expenses section, Expenses row Taxes column intersection	x			decimal	(Taxable line item expense net total) *(Invoice tax rate)	x	x
feeAdjustmentTotal	FEE_ADJUSTMENT_TOTAL (NUMBER)	Invoice Header - Total Fees & Expenses section, Fees row Adjustments column intersection	x			decimal	Line item fee adjustments total.	x	x
feeDiscountTotal	FEE_DISCOUNT_TOTAL (NUMBER)	Invoice Header - Total Fees & Expenses section, Fees row Discounts column intersection	x			decimal	Line item fee original discounts total.	x	x

<b>feeTaxTotal</b>	<b>FEE_TAX_TOTAL (NUMBER)</b>	<b>Invoice Header - Total Fees &amp; Expenses</b> section, <b>Fees</b> row <b>Taxes</b> column intersection	<b>x</b>			dec ima l	(Taxable line item fee net total) * (Invoice tax rate)	<b>x</b>	<b>x</b>
<b>groupAccessList--&gt;</b>	n/a			<b>x</b>	<a href="#">EInvGroup Access</a>		Links to the group security information entered through the Security block of the record.		
<b>hasNonUSTax</b>	<b>HAS_NO_N_US_TAX</b>		<b>x</b>			boo lea n	If any line items have non-US tax codes entered in them, this value is TRUE, else it is FALSE.		
<b>historyList--&gt;</b>	n/a	<b>History</b> tab	<b>x</b>	<b>x</b>	<a href="#">THistory</a>	obj ect	8 The list of all histories that are related to the invoice record.  --> Links to specific history information for all of the invoice's history records.	<b>x</b>	



<b>invoiceAdjustmentTotal</b>	<b>INVOICE_ADJUSTMENT_TOTAL (NUMBER)</b>	<b>Invoice Header - Total Fees &amp; Expenses</b> section, <b>Line Items Total</b> row <b>Adjustments</b> column intersection	x			decimal	(Fee adjustments total) + (Expense adjustments total)	x	x
<b>invoiceDate</b>	<b>INVOICE_DATE (DATE)</b>	<b>Invoice Date</b> (General block)	x			date	The date when the invoice was issued. Time-zone-independent.	x	
<b>invoiceDiscountTotal</b>	<b>INVOICE_DISCOUNT_TOTAL (NUMBER)</b>	<b>Invoice Header - Total Fees &amp; Expenses</b> section, <b>Line Items Total</b> row <b>Discounts</b> column intersection	x			decimal	(Fee discounts total) + (Expense discounts total)	x	x
<b>invoiceTaxTotal</b>	<b>INVOICE_TAX_TOTAL (NUMBER)</b>	<b>Invoice Header - Total Fees &amp; Expenses</b> section,	x			decimal	(Fee taxes total) + (Expense taxes total)	x	x

		Line Items Total row Taxes column intersect ion							
isElectronic	IS_ELEC TRONIC		x			boo lea n	If TRUE, indicates that the invoice originated from an e- billing application such as Collaborati.		
isPercentageDiscount	IS_PERC ENTAGE_ DISCOUN T		x			boo lea n	An invoice discount in percent form.		
lineItems--> Before TeamConne ct 3.3 SP2: lineItemList -->	n/a		x	x	<a href="#">JInvLineItem</a>		8 The list of all line items that are added to the invoice record.  --> Links to the join table that contains information for all line items added to the invoice.	x	x
modifiedBy- ->	MODIFIED_BY_ID  (NUMBER )	Modified By  (Security block)	x	x	<a href="#">YUser</a>	obj ect	8 The user who last modified the invoice record.  --> Links to specific		

							information about the user account.		
<b>modifiedOn</b>	<b>MODIFIED_ON</b> <b>(DATE)</b>	<b>Modified On</b> <b>(Security block)</b>	<b>x</b>			<b>date</b>	The date the invoice was last modified.		
<b>netExpenseTotal</b>	<b>NET_EXPENSE_TOTAL</b> <b>(NUMBER)</b>	<b>Invoice Header - Total Fees &amp; Expenses</b> section, <b>Expenses row Total</b> column intersection	<b>x</b>			<b>decimal</b>	(Line item expenses total) + (Expense taxes total)	<b>x</b>	<b>x</b>
<b>netFeeTotal</b>	<b>NET_FEE_TOTAL</b> <b>(NUMBER)</b>	<b>Invoice Header - Total Fees &amp; Expenses</b> section, <b>Fees row Total</b> column intersection	<b>x</b>			<b>decimal</b>	(Line item fees net total) + (Fees taxes total)	<b>x</b>	<b>x</b>
<b>netInvoiceTotal</b>	<b>NET_INVOICE_TOTAL</b> <b>(NUMBER)</b>	<b>Invoice Header - Total Fees &amp; Expenses</b> section, <b>Line Items</b>	<b>x</b>			<b>decimal</b>	(Net fees total) + (Net expenses total)	<b>x</b>	<b>x</b>

		<b>Total</b> row, <b>Total</b> column intersect ion							
nextLineItemNumber	NEXT_LINE_ITEM _NUMBER		x			int	Displays the next available line item number for this invoice.		
note-->	NOTE_ID (NUMBER)	Notes (General block)		x	<a href="#">JNote</a>	object	Links to the table that contains the text entered into the <b>Notes</b> field of the record.		
numberString	NUMBER_STRING (VARCHAR2) (50)	Number (General block)	x			string (50)	Any alphanumeric text that describes the invoice record.		
originalExpenseTotal	ORIGINAL_EXPENSE_TOTAL (NUMBER)	Invoice Header - Total Fees & Expenses section, <b>Expenses</b> row <b>Original</b> column intersection	x			decimal	(Line item expenses Original total) + (Line item expenses Original discounts total)	x	x
originalFeeTotal	ORIGINAL_FEE_TOTAL	Invoice Header - Total Fees &	x			decimal	(Line item fees Original total) + (Line item	x	x

	(NUMBER )	Expenses section, Fees row Original column intersection					fees Original discounts total)		
originalInvoiceTotal	ORIGINAL_INVOICE_TOTAL (NUMBER )	Invoice Header - Total Fees & Expenses section, Line Items Total row Original column intersection	x			decimal	(Original fees total) + (Original expenses total)	x	x
periodEndDate Before TeamConnect 3.3 SP2: periodEndOn	PERIOD_END_ON (DATE)	Period To (General block)	x			date	The end date of the invoice period. Time-zone-independent.		
periodStartDate Before TeamConnect 3.3 SP2: periodStartOn	PERIOD_START_ON (DATE)	Period From (General block)	x			date	The start date of the invoice period. Time-zone-independent.		
statusType Before TeamConnect 3.3 SP2:	STATUS_TYPE_IID (NUMBER )	Status (General block)	x			int	The posting status of the invoice.	x	

<b>postingStat usID</b>							1 - Not Submitted 2 - Posted 3 - Not Posted 5- Rejected*		
<b>primaryKey</b>	<b>PRIMARY_KEY (NUMBER)</b>		x			int	The unique ID of the invoice record.		
<b>receivedDate</b>	<b>RECEIVED_DATE (DATE)</b>	<b>Date Received (General block)</b>	x			date	The date when the invoice was received. Time-zone-independent.	x	
<b>referenceNumber</b>	<b>REFERENCE_NUMBER (VARCHAR 50)</b>	<b>Reference Number</b>				string	If this invoice is of type Credit Note, then referenceNumber matches the numberString value of the original standard invoice for which this credit is intended.		
<b>securityTypeID</b>	<b>SECURITY_TYPE_ID (NUMBER)</b>	<b>Security Type (Security block)</b>	x			int	The security type of the invoice record.  0 - Public 2 - Private		

submittedElectronically  Before TeamConnect 3.3 SP2: isElectronic	IS_ELECTRONIC		x			boolean		x	
submittedTotal  Before TeamConnect 3.3 SP2: manualTotal	MANUAL_AMOUNT  (NUMBER)	Manual Amount  (General block)	x			decimal	The total amount of the invoice as entered manually by the user.	x	
taxRate	TAX_RATE  (NUMBER)						Currently not used in TeamConnect.		
transaction->	TRANSACTION_ID  (NUMBER)	Transaction block		x	<a href="#">RTransaction</a>	object	Contains the last transaction associated with the invoice that was either posted to or voided from an account.		
typeID	TYPE_ID  (NUMBER)	Invoice Type  (General block)	x			Enum	The general type of the invoice.  0 - Standard (STANDARD)  1 - Accrual (ACCRUAL)  2 -Credit Note (CREDIT_NOTE)	x	

							3 - Shadow Invoice (SHADOW)		
<b>userAccess List--&gt;</b>	n/a			x	<a href="#">EInvcUserAccess</a>		Links to the user security information entered through the Security block of the record.		
<b>vendor--&gt;</b>	<b>VENDOR_ID</b> (NUMBER)	<b>Vendor</b> (General block)	x	x	<a href="#">TContact</a>	object	8 The contact record of the vendor associated with the invoice.  --> Links to the specific information about the contact object.		
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Indicates how many times the invoice record was updated.		
<b>warnings</b>	<b>WARNINGS</b>	<b>Warnings</b>				string (2000)	Warning message for the invoice captured by TeamConnect that is generated by another application that processes invoices.	X	



### 3.1.33.2 JInvcCategoryAdjustment

JInvcCategoryAdjustment contains information about the adjustment of line items in an invoice record. This information includes the nature of the adjustment calculation, the reason for the adjustment, and comments about the adjustment.

Object: JInvcCategoryAdjustment (J\_INV\_CATEGORY\_ADJUSTMENT)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
	ADJUSTED_BY_ID (NUMBER)		x	x	<a href="#">YUser</a>	Number	8 The user who made the adjustment.  --> Links to specific information about the user account.	x	x
adjustmentDate  Before TeamConnect 3.3 SP2: adjustedOn	ADJUSTED_ON (DATE)					date	Defaults to current date/time.		
adjustmentAmount	ADJUSTMENT_AMOUNT (NUMBER)		x			decimal	Adjustment amount.		
adjustmentMethod  Before TeamConnect 3.3 SP2: adjustmentTypeID	IS_TAXABLE (NUMBER)		x			int	Adjustment type. One of these choices:  1 = By Amount 2 = By Percentage 3 = To Amount		
adjustmentReason-->	ADJUSTMENT_REASON		x	x		object	8 The selected reason for the adjustment.		

	(NUMBER)								
dcategory->	CATEGORY_ID (NUMBER)	Category				object	8 The category fto which this adjustment applies.		x
comments ToVendor	COMMENT_TO_VENDOR (VARCHAR 250))		x			string (250)	Comments about the adjustment that will be visible to vendors outside your company.		
inHouseComments	IN_HOUSE_COMMENT< (VARCHAR 250)		x			string (250)	Comments that are private, not transmitted to vendors or other outside sources.		
iowner-->	INVOICE_ID (NUMBER)			x	<a href="#">TInvoice</a>	object	8 The invoice to which the line item belongs.  --> Links to the specific information about the invoice object.		
project-->	PROJECT_ID (NUMBER)	Project	x	x	<a href="#">TProject</a>	object	8 The project associated with the adjustment.  --> Links to the specific information about the project object.	x	x
previousNetTotal	PREVIOUS_NET_TOTAL (NUMBER)	Original Discount	x			decimal	The net total of the entire invoice immediately before the current adjustment was applied.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of this object.		

version	VERSION (NUMBER)		x			int	Indicates how many times the object was updated.		
---------	---------------------	--	---	--	--	-----	--	--	--

### 3.1.33.3 JInvcHeaderAdjustment

JInvcHeaderAdjustment contains information about the adjustment of line items in an invoice record. This information includes the nature of the adjustment calculation, the reason for the adjustment, and comments about the adjustment.

Object: JInvcHeaderAdjustment (J\_INV\_C\_ADJUSTMENT)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
	ADJUSTED_BY_ID (NUMBER)		x	x	<a href="#">YUser</a>	Number	8 The user who made the adjustment.  --> Links to specific information about the user account.	x	x
adjustmentDate  Before TeamConnect 3.3 SP2: adjustedOn	ADJUSTED_ON (DATE)					date	Defaults to current date/time.		
adjustmentMethod  Before TeamConnect 3.3 SP2: adjustmentTypeID	IS_TAXABLE (NUMBER)		x			int	Adjustment type. One of these choices:  1 = By Amount  2 = By Percentage  3 = To Amount		

<b>adjustmentReason--&gt;</b>	<b>ADJUSTMENT_REASON</b> (NUMBER)		x	x		object	8 The selected reason for the adjustment.		
<b>adjustmentTargetID</b>  <b>Before TeamConnect 3.3 SP2: adjustmentTarget</b>	<b>ADJUSTMENT_TARGET_ID</b> (NUMBER)		x			int	The target area to be impacted by the adjustment. One of these choices:  1 = Total Fees  2 = Total Expenses  3 = Total Invoice		
<b>adjustmentTotalExpenses</b>	<b>ADJUSTMENT_TOTAL_EXPENSES</b> (NUMBER)		x			decimal	Adjustment amount to be applied only to the expense portion of the invoice. Calculation is done as specified by attribute adjustmentTypeID.		
<b>adjustmentTotalFees</b>	<b>ADJUSTMENT_TOTAL_FEES</b> (NUMBER)		x			decimal	Adjustment to be applied only to the fee portion of the invoice. Calculation is done as specified by attribute adjustmentTypeID.		
<b>adjustmentTotalInvoice</b>	<b>ADJUSTMENT_TOTAL_INVOICES</b> (NUMBER)		x			decimal	Adjustment to be applied to the entire amount of the invoice.		

							Calculation is done as specified by attribute adjustmentTypeID.		
commentsTo Vendor	COMMENT_TO_VENDOR (VARCHAR 250))		x			string (250)	Comments about the adjustment that will be visible to vendors outside your company.		
inHouseComments	IN_HOUSE_COMMENT< (VARCHAR 250)		x			string (250)	Comments that are private, not transmitted to vendors or other outside sources.		
owner--> Before TeamConnect 3.3 SP2: invoice-->	INVOICE_ID (NUMBER)			x	<a href="#">Invoice</a>	object	8 The invoice to which the line item belongs.  --> Links to the specific information about the invoice object.		
previousNetTotal	PREVIOUS_NET_TOTAL (NUMBER)	Original Discount	x			decimal	The net total of the entire invoice immediately before the current adjustment was applied.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of this object.		
version	VERSION (NUMBER)		x			int	Indicates how many times the object was updated.		

## 3.1.33.4 JInvcLineItem

JInvcLineItem contains information about the line items in an invoice record. This information includes the project to which the line item is associated, amount, line item type (such as task or expense), activity, task category, timekeeper, and service date. This table represents the **Line Items** block on the **Line Items** tab of an invoice record.

Object: JInvcLineItem (J\_INVNC\_LINE\_ITEM)

Attribute	Database column name	Field in user interface	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
<b>activity</b> Before TeamConnect 3.3 SP2: activityItem	ACTIVITY_ITEM_ID (NUMBER)	Activity Item	x	x	<a href="#">L_TaskActivityItem</a>	object	8 The activity item associated with the line item of type task.  --> Links to the definition information for the task activity item.	x	x
<b>adjustedDiscount</b> Before TeamConnect 3.3 SP2: netDiscount	ADJUSTED_DISCOUNT (NUMBER)	Net Discount	x			decimal	The net discount of the line item.		
<b>adjustedQuantity</b> Before TeamConnect 3.3 SP2: netQuantity	ADJUSTED_QUANTITY (NUMBER)	Net Quantity	x			decimal	The net quantity of the line item.		
<b>adjustedRate</b>	ADJUSTED_RATE	Net Rate	x			decimal	The net rate of the line		

<b>Before TeamConnect 3.3 SP2: netRate</b>	(NUMBER)						I	item.		
<b>adjustedTotal Before TeamConnect 3.3 SP2: netTotal</b>	<b>TOTAL_AMOUNT</b> (NUMBER)	<b>Net Total</b>	x				decimal	The net total of the line item.		
<b>adjustments Before TeamConnect 3.3 SP2: adjustmentList</b>			x	x			list	8 Any invoice adjustments that have been entered against this line item.  --> Links to any adjustments that have been entered against this line item.		
<b>categories--&gt; Before TeamConnect 3.3 SP2: detailList--&gt;</b>			x	x	<a href="#">ELitmDetail</a>		list	8 The custom field detail values that have been entered against this line item.  --> Links to the collection of custom field detail values that have been entered against this line item.		

createdBy-->	CREATED_BY_ID (NUMBER)	Created By (Security block)	x	x	<a href="#">YUser</a>	object	8 The user who created the invoice record.  --> Links to specific information about the user account.	x	x
createdOn	CREATED_ON (DATE)	Created On (Security block)	x			date	Date the invoice was created.	x	x
currencyItem->	CURRENCY_ITEM_ID (NUMBER)						Currently not used.		
defaultCategory	DEFAULT_CATEGORY_ID	Default Category (Categories block)	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 The default category for the line item.  --> Links to the definition information of the category.		
defaultCurrencyNetDiscount	DEFAULT_CURR_DISCOUNT (NUMBER)		x			decimal	The net discount of the line item in the system's default currency.		
defaultCurrencyNetRate	DEFAULT_CURR_RATE		x			decimal	The net rate of the line item in the system's		



	(NUMBER)						default currency.		
defaultCurrencyNetTotal	DEFAULT_CURR_TOTAL (NUMBER)		x			decimal	The net total of the line item in the system's default currency.		
expenseCategory-->	EXPENSE_TYPE_ID (NUMBER)	Category (when Expense is selected in the Task/Expense field)	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 The expense category of the line item, if the line item is an expense.  --> Links to the definition information of the category.	x	x
groupAccessList-->				x	<a href="#">ELitmGroupAccess</a>	list	8 The non-default group access rights that have been granted against this line item.  --> Links to the collection of non-default group access rights that have been granted against this line item.		
hasWarnings	HAS_WARNINGS		x			boolean	Indicates whether		

	(NUMBER)					n	warning conditions have been noted for this line item.		
involved-->	INVOLVE D_ID (NUMBER)		x	x			Currently not used.		
itemNumber Before TeamConnect 3.3 SP2: itemNum	ITEM_NUM MBER		x			int	The number assigned to the line item at the time it was created. Invoice item numbers begin at 1 and increment as high as necessary to accommodate the invoice's line item. If a line item is deleted, its item number is not reused for any existing or future line item.		
modifiedBy-->	MODIFIED _BY_ID (NUMBER)		x	x	<a href="#">YUser</a>	object	8 The user who last modified the line item.  --> Links to specific information about the		

							user account.		
modifiedOn	MODIFIED_ON (DATE)	Last Modified On	x			date	The date that the line item was modified.		
note Before TeamConnect 3.3 SP2: shortDescription	ITEM_DESCRIPTION (VARCHAR2) (250)	Description	x			string (2000)	The description of the line item.		x
originalDiscount	DISCOUNT_PERCENT (NUMBER)	Original Discount	x			decimal	The original discount of the line item.		
originalQuantity	QUANTITY (NUMBER)	Original Quantity	x			decimal	The original quantity of the line item.		
originalRate	UNIT_PRICE (NUMBER)	Original Rate	x			decimal	The original rate of the line item.		
originalTotal	GROSS_AMOUNT (NUMBER)	Original Total	x			decimal	The original total of the line item.		
owner-->	INVOICE_ID (NUMBER)			x	<a href="#">Invoice</a>	object	8 The invoice to which the line item belongs.  --> Links to the specific information about the invoice object.		

<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	The unique ID of the line item.		
<b>project--&gt;</b>	<b>PROJECT_ID</b> (NUMBER)	<b>Project</b>	x	x	<a href="#">TProject</a>	object	8 The project associated with the line item.  --> Links to the specific information about the project object.	x	x
<b>securityTypeID</b>	<b>SECURITY_TYPE_ID</b> (NUMBER)		x			int	Security type. One of these values:  0= Public  2 = Private		
<b>serviceDate</b>	<b>SERVICE_DATE</b> (DATE)	<b>Service Date</b>	x			date	The date when the service to which the line item refers was rendered. Time-zone-independent.		
<b>taskCategory--&gt;</b>	<b>TASK_CATEGORY_ID</b> (NUMBER)	<b>Category</b> (when <b>Task</b> is selected in the <b>Task/Expense</b> field)	x	x	<a href="#">YRecentlyViewedRecord</a>	object	8 The task category of the line item, if the line item is a task.  --> Links to the definition information of the category.	x	x

<b>taxable</b> Before TeamConnect 3.3 SP2: isTaxable	<b>IS_TAXABLE</b>  (NUMBER)		x				Currently not used.		
<b>timekeeper</b> Before TeamConnect 3.3 SP2: vendorRep-->	<b>VENDOR_ID</b>  (NUMBER)	<b>Timekeeper</b>	x	x	<a href="#">TContact</a>	object	8 The contact who performed the service or provided the goods.  --> Links to the specific information about the contact object.	x	x
<b>totalAdjustment</b>	<b>TOTAL_ADJUSTMENT</b>	<b>Total Adjustment</b>	x			decimal	The total amount of all adjustments made to the line item.	x	
<b>type</b> Before TeamConnect 3.3 SP2: typeIID	<b>TYPE_IID</b>  (NUMBER)	<b>Task/Expense</b>	x			int	The type of the line item.  E - Expense T - Task	x	x
<b>userAccessList--&gt;</b>				x	<a href="#">ELitmUser Access</a>	list	8 The non-default user access rights that have been granted against this line item.  --> Links to the collection of non-default user access		

							rights that have been granted against this line item.		
version	VERSION (NUMBER)		x			int	Indicates how many times the invoice record was updated.		
warnings	LINEITEM_WARNINGS (NUMBER)		x			string (2000)	Collected text of warning messages related to this line item.		

### 3.1.33.5 JInvcNonUSTax

JInvcNonUSTax contains information about the non-U.S. tax amounts in line items in an invoice record.

Object: JInvcAdjustment (J\_INVC\_NON\_US\_TAX)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of this object.		
version	VERSION (NUMBER)		x			int	Indicates how many times the object was updated.		
owner-->	INVOICE_ID (NUMBER)			x	<a href="#">Invoice</a>	object	8 The invoice to which the tax		

Before TeamConnect 3.3 SP2: invoice-->							amount belongs. --> Links to the specific information about the invoice object.		
project-->	PROJECT_ID (NUMBER)	Project	x	x	<a href="#">TProject</a>	object	8 The project associated with the tax amount. --> Links to the specific information about the project object.	x	x
nonUSTaxType-->	NON_US_TAX_TYPE (NUMBER)	Non-US Tax Type		x	<a href="#">LInvcNonUSTaxType</a>	object	8 The tax type category associated with the tax amount.	x	x
taxAmount	TAX_AMOUNT (NUMBER)	Tax Amount	x			decimal	The tax amount in original, local currency.		
defaultCurrencyTaxAmount	DEFAULT_CURR_TAX_AMOUNT (NUMBER)	Tax Amount	x			decimal	The tax amount in the invoice's default currency.		

### 3.1.33.5.1 LInvcNonUSTaxType

LInvcNonUSTaxType contains information about the tax types available when creating invoice line items containing non-U.S. taxes. This information includes the description of the tax type, tree position, and display order. This table represents the **Non-US Tax Type** tab in the **Invoice** object definition in Designer.

Object: LInvcNonUSTaxType (L\_INVC\_NON\_US\_TAX\_TYPE)

Attribute	Database column name	Field in UI	End of	Bridge	Links to object table:	Data type	Comments	Commonly used in:

			path					Rules	
displayOrder	DISPLAY_ORDER (NUMBER)	Order	x			int	The order in which the items will be listed in the GUI, lowest number first.		
name	NAME (VARCHAR2) (50)	Item Name	x			string (50)	The description of the tax type.		
parent -->	PARENT_ID (NUMBER)	Show items in node	x	x	<a href="#">LInvcNonU STaxType</a>	object	8 The parent item of the current item.  --> Links to definition information for the parent tax type.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the tax type list item.		
treePosition	TREE_POSITION (VARCHAR2) (250)	Tree Position	x			string (250)	The unique four-character alphanumeric code that is created when the tax type is defined.		
version	VERSION (NUMBER)		x			int	Indicates how many times the skill type list item was updated.		
uniqueCode	UNIQUE_CODE (VARCHAR2) (250)	Tax Code	x			string (20)	The unique 20-character alphanumeric code that the solution developer specifies when entering a new tax code.		



isActive	IS_ACTIVE (BOOLEAN)		x			boolean	If TRUE, the tax type is active and can be selected by the end user.		
----------	------------------------	--	---	--	--	---------	--	--	--

### 3.1.33.6 JInvcTimekeeperAdjustment

JInvcTimekeeperAdjustment contains information about the adjustment of line items in an invoice record. This information includes the nature of the adjustment calculation, the reason for the adjustment, and comments about the adjustment.

Object: JInvcTimekeeperAdjustment (J\_INV\_C\_TIMEKEEPER\_ADJUSTMENT)

Attribute	Database column name	Field in user interface	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
	ADJUSTED_BY_ID (NUMBER)		x	x	<a href="#">YUser</a>	Number	8 The user who made the adjustment.  --> Links to specific information about the user account.	x	x
adjustmentDate  Before TeamConnect 3.3 SP2: adjustedOn	ADJUSTED_ON (DATE)					date	Defaults to current date/time.		
adjustmentMethod  Before TeamConnect 3.3 SP2: adjustmentTypeID	IS_TAXABLE (NUMBER)		x			int	Adjustment type. One of these choices:  1 = By Amount 2 = By Percentage 3 = To Amount		

adjustmentReason-->	ADJUSTMENT_REASON (NUMBER)		x	x		object	8 The selected reason for the adjustment.		
commentsToVendor	COMMENT_TO_VENDOR (VARCHAR 250))		x			string (250)	Comments about the adjustment that will be visible to vendors outside your company.		
inHouseComments	IN_HOUSE_COMMENT< (VARCHAR 250)		x			string (250)	Comments that are private, not transmitted to vendors or other outside sources.		
owner-->	INVOICE_ID (NUMBER)			x	<a href="#">Invoice</a>	object	8 The invoice to which the line item belongs.  --> Links to the specific information about the invoice object.		
previousNetTotal	PREVIOUS_NET_TOTAL (NUMBER)	Original Discount	x			decimal	The net total of the entire invoice immediately before the current adjustment was applied.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of this object.		
version	VERSION (NUMBER)		x			int	Indicates how many times the object was updated.		

### 3.1.33.7 JLitmAdjustment

JLitmAdjustment contains information about the adjustment of line items in an invoice record. This information includes the nature of the adjustment calculation, the reason for the adjustment, and comments about the adjustment.

**Object:** JLitmAdjustment (J\_LITM\_ADJUSTMENT)

Attribute	Database column name	Field in user interface	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wiz
	ADJUSTED_BY_ID (NUMBER)		x	x	<a href="#">YUser</a>	Number	8 The user who made the adjustment.  --> Links to specific information about the user account.	x	x
adjustmentDate  Before TeamConnect 3.3 SP2: adjustedOn	ADJUSTED_ON (DATE)					date	Defaults to current date/time.		
adjustmentMethod  Before TeamConnect 3.3 SP2: adjustmentTypeID	ADJUSTMENT_TYPE_ID (NUMBER)		x			int	Adjustment type. One of these choices:  1 = By Amount  2 = By Percentage  3 = To Amount		
adjustmentReason-->	ADJUSTMENT_REASON (NUMBER)		x	x		object	8 The selected reason for the adjustment.		
adjustmentTarget  Before TeamConnect 3.3	ADJUSTMENT_TARGET_ID (NUMBER)		x			int	The target area to be impacted by the adjustment. One of these choices:		

<b>SP2: adjustmentTargetID</b>							1 = Total Fees 2 = Total Expenses 3 = Total Invoice		
<b>adjustmentQuantity</b>	<b>ADJUSTMENT_QUANTITY (NUMBER)</b>		x			decimal	Adjustment quantity to be applied to the existing line item quantity.		
<b>adjustmentAmount</b>	<b>ADJUSTMENT_AMOUNTS (NUMBER)</b>		x			decimal	Adjustment amount to be applied to the line item cost. Calculation is done as specified by attribute adjustmentTypeIID.		
<b>commentsToVendor</b>	<b>COMMENT_TO_VENDOR (VARCHAR 2000)</b>		x			string (2000)	Comments about the adjustment that will be visible to vendors outside your company.		
<b>inHouseComments</b>	<b>IN_HOUSE_COMMENT&lt; (VARCHAR 250)</b>		x			string (250)	Comments that are private, not transmitted to vendors or other outside sources.		
<b>LineItemID</b>	<b>LINEITEM_ID (NUMBER)</b>			x	<a href="#">JInvcLineItem</a>	object	8 The invoice line item to which the adjustment belongs.  --> Links to the specific information about the line item.		
<b>previousNetTotal</b>	<b>PREVIOUS_NET_TOTAL (NUMBER)</b>	<b>Original Discount</b>	x			decimal	The net total of the entire invoice immediately before the current		

							adjustment was applied.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	The unique ID of this object.		
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Indicates how many times the object was updated.		

### 3.1.33.8 EInvcDetail

EInvcDetail contains all categories added to invoice records. Each time a category is added to an invoice record, an entry is made in this table in the database.

In Object Navigator, **EInvcDetail** is not displayed when you traverse using the **detailList-->** bridge. Instead, a list of all categories that have been defined for the object definition appears. This list is labeled **Category**. From each category, you can traverse to a table called **Detail Fields**, which displays all custom fields belonging to the category you selected.

For more information about how custom field values are stored in the database, see Object Model: Custom Fields.

Object: EInvcDetail (E\_INVC\_DETAIL)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wiz
<b>category--&gt;</b>	<b>CATEGORY_ID</b> (NUMBER)	<b>Category</b>	x	x	<a href="#">YRecentlyViewedRecord</a>	object	The category of the invoice with which the details are associated.		
<b>owner--&gt;</b>	<b>ENTERPRISE_OBJECT_ID</b> (NUMBER)			x	<a href="#">TInvoice</a>	object	The invoice with which the custom fields are associated.		

<b>isManual</b>	<b>IS_MANUAL</b> (NUMBER)		<b>x</b>			boolean	Specifies whether the category is added manually or automatically to the invoice. Categories are added automatically when they are the parent of the category that is added manually.  0 - Added Automatically  1 - Added Manually		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		<b>x</b>			int	The unique ID of the invoice category record.		
<b>version</b>	<b>VERSION</b> (NUMBER)		<b>x</b>			int	Indicates how many times the added category was updated.		

## 3.1.33.8.1 ELitmDetail

ELitmDetail organizes the detail values for an invoice line item (JInvcLineItem object). Detail values originate from custom fields associated with the category assigned to the line item.

There can be multiple ELitmDetail objects for each JInvcLineItem. There is one ELitmDetail for each category used in JInvcLineItem, including any parent categories, plus the Root category. For example, if the category assigned to the JInvcLineItem object were **Root : Expense : Parking**, there would be three associated ELitmDetail objects: one for Root, one for Expense, and one for Parking.

Note that in ELitmDetail there are attributes for multiple lists of detail values. However, in any given ELitmDetail object, it is possible that only some of those attributes will really have values. Since an ELitmDetail object has a specific category, only the custom fields that match the category will be used. Thus there will only be values for the field types that those custom fields represent.

Object: **ELitmDetail (E\_LITM\_DETAIL)**

Attribute	Database column name	End of	Bridge	Links to object table:	Data type	Comments	Commonly used in:
-----------	----------------------	--------	--------	------------------------	-----------	----------	-------------------

							Rules	Temp/Wiz
isManual	IS_MANUAL (NUMBER)	x			boolean	Specifies whether the category is added manually or automatically to the line item. Categories are added automatically when they are the parent of the category that is added manually.  0 - Added Automatically 1 - Added Manually		
primaryKey	PRIMARY_KEY (NUMBER)	x			int	Unique ID automatically assigned to this custom field value.		
version	VERSION (NUMBER)	x			int	Indicates how many times the line item detail was updated.		
category-->	CATEGORY_ID (NUMBER)		x	<a href="#">YRecentlyViewedRecord</a>	object	8 The category to which this detail belongs.  Links to a category to which the line item belongs. Note that most line items belong to multiple categories - a specific category, plus any parent category, grandparent category, etc.		
owner-->	ENTERPRISE_OBJECT_ID (NUMBER)	x	x	<a href="#">JInvcLineItem</a>	object	8 The invoice line item record to which this detail belongs.		





<b>allowDenyID</b>	<b>ALLOW_DENY_ID</b> (NUMBER)	<b>Option</b>	<b>x</b>			char	Indicates whether the group is allowed or denied access to the record.  a - Allow d - Deny		
<b>group--&gt;</b>	<b>GROUP_ID</b> (NUMBER)	<b>List</b>	<b>x</b>	<b>x</b>	<a href="#">YGroup</a>	object	8 Specifies the group that the access rights pertain to.  --> Links to the specific information about the group account.		
<b>isManual</b>	<b>IS_MANUAL</b> (NUMBER)		<b>x</b>			boolean	Indicates whether the group access rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
<b>owner--&gt;</b>	<b>ENTERPRISE_OBJECT_ID</b> (NUMBER)		<b>x</b>	<b>x</b>	<a href="#">JInvcLineItem</a>	object	8 The invoice line item record with which this group access setting is associated.  --> Links to the specific information about the invoice line item record.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		<b>x</b>			int	Unique ID for the group access rights.		

version	VERSION (NUMBER)		x			int	Indicates how many times the group access rights have been updated.		
---------	---------------------	--	---	--	--	-----	---	--	--

### 3.1.33.10 ELitmUserAccess

ELitmUserAccess controls the privileges that a user has for a specific invoice line item.

Object: ELitmUserAccess (E\_LITM\_USER\_ACCESS)

Attribute	Database column name	Field in user interface	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
isRead	IS_READ (NUMBER)	Read checkbox	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isUpdate	IS_UPDATE (NUMBER)	Update checkbox	x			int	Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		

<b>isDelete</b>	<b>IS_DELETE (NUMBER)</b>	<b>Delete checkbox</b>	<b>x</b>			int	<p>Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
<b>isPerm</b>	<b>IS_PERM (NUMBER)</b>	<b>Perm checkbox</b>	<b>x</b>			int	<p>Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
<b>allowDenyIID</b>	<b>ALLOW_DENY_IID (NUMBER)</b>	<b>Option</b>	<b>x</b>			char	<p>Indicates whether the group is allowed or denied access to the record.</p> <p>a - Allow d - Deny</p>		
<b>isManual</b>	<b>IS_MANUAL (NUMBER)</b>		<b>x</b>			boolean	<p>Indicates whether the group access rights were assigned manually through the Security block of the record or automatically by the system.</p> <p>0 - Assigned manually</p>		

							1 - Assigned automatically		
owner->	ENTERPRISE_OBJECT_ID (NUMBER)		x	x	<a href="#">JInvcLinItem</a>	object	8 The invoice line item record with which this group access setting is associated.  --> Links to the specific information about the invoice line item record.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	Unique ID for the group access rights.		
user	USER_ID (NUMBER)	List	x	x	<a href="#">YUser</a>	object	8 Specifies the user that the access rights pertain to.  --> Links to the specific information about the user account.		
version	VERSION (NUMBER)		x			int	Indicates how many times the group access rights have been updated.		

### 3.1.33.11 EInvcUserAccess

EInvcUserAccess contains the user access information that is set in the **Security** block of each invoice record. For example, it includes whether each user that is listed in the Security block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm**(ission) rights.

Object: EInvcUserAccess (E\_INVC\_USER\_ACCESS)

Attribute	Database column name	Field in user interface	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz

<b>isRead</b>	<b>IS_READ (NUMBER)</b>	<b>Read checkbox</b>	<b>x</b>			int	<p>Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
<b>isUpdate</b>	<b>IS_UPDATE (NUMBER)</b>	<b>Update checkbox</b>	<b>x</b>			int	<p>Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
<b>isDelete</b>	<b>IS_DELETE (NUMBER)</b>	<b>Delete checkbox</b>	<b>x</b>			int	<p>Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)</p>		
<b>isPerm</b>	<b>IS_PERM (NUMBER)</b>	<b>Perm checkbox</b>	<b>x</b>			int	<p>Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).</p> <p>This attribute does not determine whether the operations are</p>		

							allowed or denied. (See allowDenyIID.)		
allowDenyIID	ALLOW_DENY_IID (NUMBER)	Option	x			char	Indicates whether the user is allowed or denied access to the record.  a - Allow d - Deny		
isManual	IS_MANUAL (NUMBER)		x			boolean	Indicates whether the user rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
owner-->	ENTERPRISE_OBJECT_ID (NUMBER)		x	x	<a href="#">Invoice</a>	object	8 The invoice record with which this user access setting is associated.  --> Links to the specific information about the invoice record.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	Unique ID for the user access rights.		
user-->	USER_ID (NUMBER)	List	x	x	<a href="#">User</a>	object	8 Specifies the user the access rights pertain to.  --> Links to the specific information about the user account.		

version	VERSION (NUMBER)		x			int	Indicates how many times the user access rights have been updated.		
---------	---------------------	--	---	--	--	-----	--	--	--

### 3.1.33.12 EInvcGroupAccess

EInvcGroupAccess contains the group access information that is set in the **Security** block of each invoice record.

For example, it includes whether each group that is listed in the **Security** block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm**(ission) rights.

Object: EInvcGroupAccess (E\_INVC\_GROUP\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
isRead	IS_READ (NUMBER)	Read checkbox	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isManual	IS_MANUAL (NUMBER)		x			boolean	Indicates whether the group access rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
owner -->	ENTERPRISE_OBJECT_ID		x	x	<a href="#">Invoice</a>	object	8 The invoice record with which this group access		

	(NUMBER)						setting is associated. --> Links to the specific information about the invoice record.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	Unique ID for the group access rights.		
version	VERSION (NUMBER)		x			int	Indicates how many times the group access rights have been updated.		

### 3.1.33.13 YCurrencyInfo

YCurrencyInfo contains information about currencies defined in the multi-currency lookup table, which are used to display currency in financial fields, including the currency name, symbol, code, and exchange rate.

Object: YCurrencyInfo (Y\_CURRENCY\_INFO)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
code	CODE (VARCHAR) (3)	Currency code	x			string (3)	The universal 3-digit code of the currency item.		
created By-->	CREATE_BY_ID (NUMBER)		x	x	<a href="#">YUser</a>	object	8 The user who created the currency item. --> Links to specific information about the user account.		
created On	CREATE_ON (DATE)		x			date	The date that the currency item was created.		



modifiedBy-->	MODIFIED_BY_ID (NUMBER)		x	x	<a href="#">YUser</a>	object	8 The user who last modified the currency item.  --> Links to specific information about the user account.		
modifiedOn	MODIFIED_ON (DATE)	Last Modified On	x			date	The date that the currency item was modified.		
name	NAME (VARCHAR2) (50)	Name	x			string (50)	The name of the currency item.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the currency item.		
rate	RATE (NUMBER)	Exchange Rate	x			decimal	The exchange rate of the currency item.		
symbol	SYMBOL (VARCHAR) (3)	Symbol	x			string (3)	The ISO symbol of the currency item.		
version	VERSION (NUMBER)		x			int	Indicates how many times the currency item was updated.		

JInvcHeaderAdjustment contains information about the adjustment of line items in an invoice record. This information includes the nature of the adjustment calculation, the reason for the adjustment, and comments about the adjustment.

### 3.1.34 Object Model: Histories

The tables in this appendix provide information about the object model as it relates to histories.

### 3.1.34.1 THistory

THistory contains the general information for a history record. This information includes the user who created the history, creation date, short description, history text, default category and so on. This table links to the list of custom fields created for the history categories and related information contained in other tables.

Object: THistory (T\_HISTORY)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wizard
archived On	ARCHIVED_ON (NUMBER)	Date Time (General block)	x			date Time	Date and time when the history was created, as entered by the user.	x	
categories--> Before TeamConnect 3.3 SP2: detailList -->	n/a		x	x	<a href="#">EHistDetail</a> In Object Navigator, links to <b>Category</b> list which appears in the UI only.		8 The list of categories added to the current object.  --> In the object model, the added categories link to the values that have been added for custom fields, according to field type.  --> In Object Navigator, this attribute is enhanced. It links to a list of all categories		

							defined for the object definition, rather than those added to the current object. From there, you can select a category and traverse to the list of custom fields that belong to it.		
createdBy-->	CREATED_BY_ID (NUMBER)	Created By (Security block)	x	x	<a href="#">YUser</a>	object	8 User who created the history record.  --> Links to specific information about the user account.	x	x
createdOn	CREATED_ON (DATE)	Created On (Security block)	x			date	Date and time the history record was created.	x	x
createdOnBehalfOf	CREATED_ON_BEHALF_OF_ID			x	<a href="#">TContact</a>	object	When not null, it indicates that the record was created through an e-billing system or other external application. The contact referenced will often be a vendor in	X	

							TeamConnect		
defaultCategory-->	DEFAULT_CATEGORY_ID (NUMBER)	Default category (General block)	x	x	<a href="#">YRecentlyViewedRecord</a>	object	The default category of the history record.	x	x
documentFolder-->	DOCUMENT_FOLDER_ID (NUMBER)			x	<a href="#">TDocument</a>	object	Links to specific information about the document folder.		
enteredBy-->	ENTERED_BY_ID (NUMBER)	Entered by (General block)	x	x	<a href="#">YUser</a>	object	8 User who created the history record.  <b>Note:</b> <i>This is the same user as in the createdBy</i>  --> Links to specific information about the user account.	x	x
groupAccessList-->	n/a			x	<a href="#">EHistGroupAccess</a>		Links to the group security information entered through the Security block of the record.		

<b>modified By--&gt;</b>	<b>MODIFIED_BY_ID</b> (NUMBER)	<b>Modified By</b> (Security block)	x	x	<a href="#">YUser</a>	object	8 User who last modified the record.  --> Links to specific information about the user account.		
<b>modified On</b>	<b>MODIFIED_ON</b> (DATE)	<b>Modified On</b> (Security block)	x			date	Date and time the record was last modified.		
<b>objectID</b>	<b>OBJECT_ID</b> (NUMBER)	displayed as the Parent Object (General block)	x			int	The object record to which the history record belongs.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	The unique ID of the history record.		
<b>securityTypeID</b>	<b>SECURITY_TYPE_ID</b> (NUMBER)	<b>Security Type</b> (Security block)	x			int	Specifies whether the history record is public or private.  0 - Public 2 - Private		
<b>shortDescription</b>	<b>SHORT_DESCRIPTION</b> (VARCHAR2 ) (250)		x			string (250)	The first 250 characters in the Description field.		x

text	TEXT (VARCHAR)	Description (General block)	x			long String	The contents of the Description field.		x
uniqueCode	PARENT_UNIQUE_CODE (VARCHAR2 ) (4)		x			string (4)	The four- character alphanumeric code that uniquely identifies the object to which the history record belongs.		
userAccessList-->	n/a			x	<a href="#">EHistUserAccess</a>		Links to the user security information entered through the Security block of the record.		
version	VERSION (NUMBER)		x			int	Indicates the number of times the history record has been updated.		

### 3.1.34.2 EHistDetail

EHistDetail contains all categories added to history records. Each time a category is added to a history record, an entry is made in this table in the database.

In Object Navigator, **EHistDetail** is not displayed when you traverse using the **detailList-->** bridge. Instead, a list of all categories that have been defined for the object definition appears. This list is labeled **Category**. From each category, you can traverse to a table called **Detail Fields**, which displays all custom fields belonging to the category you selected.

For more information about how custom field values are stored in the database, see Object Model: Custom Fields.

**Object:** EHistDetail (E\_HIST\_DETAIL)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
category-->	CATEGORY_ID (NUMBER)	Category	x	x	<a href="#">YRecently ViewedRecord</a>	object	The category of the history with which the details are associated.		
detailDateValueList-->	n/a			x			Links to the information for history custom fields of type Date.		
detailMemoValueList-->	n/a			x			Links to the information for history custom fields of type Memo Text.		
detailNumberValueList-->	n/a			x			Links to the information for history custom fields of type Number and Check Box.		
detailObjectValueList-->	n/a			x			Links to the information for history custom fields of type List (drop-down or radio button lists).		
detailObjectValueList-->	n/a			x			Links to the information for history custom fields of type		

							Custom Object.		
<b>detailText ValueList--&gt;</b>	n/a			x			Links to the information for history custom fields of type Text.		
<b>owner--&gt;</b>	<b>ENTERPRISE_OBJECT_ID</b> (NUMBER)			x	<a href="#">THistory</a>	object	The history with which the custom fields are associated.		
<b>isManual</b>	<b>IS_MANUAL</b> (NUMBER)		x			boolean	Specifies whether the category is added manually or automatically to the history. Categories are added automatically when they are the parent of the category that is added manually.  0 - Added Automatically 1 - Added Manually		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	The unique ID of the history category record.		
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Indicates how many times the added category was updated.		



### 3.1.34.3 EHistUserAccess

EHistUserAccess contains the user access information that is set in the **Security** block of each history record. For example, it includes whether each user that is listed in the Security block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm(ission)** rights.

Object: EHistUserAccess (E\_HIST\_USER\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
isRead	IS_READ (NUMBER)	Read checkbox	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isUpdate	IS_UPDATE (NUMBER)	Update checkbox	x			int	Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isDelete	IS_DELETE (NUMBER)	Delete checkbox	x			int	Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed		

							or denied. (See allowDenyIID.)		
isPerm	IS_PERM (NUMBER)	Per m che ck box	x			int	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
allowDenyIID	ALLOW_DENY_IID (NUMBER)	Opt ion	x			char	Indicates whether the user is allowed or denied access to the record.  a - Allow d - Deny		
isManual	IS_MANUAL (NUMBER)		x			boolean	Indicates whether the user rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
owner -->	ENTERPRISE_OBJECT_ID (NUMBER)		x	x	<a href="#">This tory</a>	object	8 The history record with which this user access setting is associated.  --> Links to the specific information about the history record.		
primaryKey	PRIMARY_KEY		x			int	Unique ID for the user access rights.		

	(NUMBER)								
user-->	USER_ID (NUMBER)	List	x	x	<a href="#">YUser</a>	object	8 Specifies the user the access rights pertain to.  --> Links to the specific information about the user account.		
version	VERSION (NUMBER)		x			int	Indicates how many times the user access rights have been updated.		

### 3.1.34.4 EHistGroupAccess

The EHistGroupAccess table contains the group access information that is set in the **Security** block of each history record. For example, it includes whether each group that is listed in the Security block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm(ission)** rights.

Object: EHistGroupAccess (E\_HIST\_GROUP\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
isRead	IS_READ (NUMBER)	Read checkbox	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isUpdate	IS_UPDATE (NUMBER)	Update checkbox	x			int	Specifies whether the Update operation has been selected for granting or denial. Value		

		ck box					is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isDelete	IS_DELETE (NUMBER)	Delete checkbox	x			int	Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isPerm	IS_PERM (NUMBER)	Perm checkbox	x			int	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
allowDenyIID	ALLOW_DENY_IID (NUMBER)	Option	x			char	Indicates whether the group is allowed or denied access to the record.  a - Allow d - Deny		
group->	GROUP_ID (NUMBER)	List	x	x	<a href="#">YGroup</a>	object	8 Specifies the group that the access rights pertain to.		

							--> Links to the specific information about the group account.		
<b>isManual</b>	<b>IS_MANUAL</b> (NUMBER)		<b>x</b>			boolean	Indicates whether the group access rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
<b>owner-&gt;</b>	<b>ENTERPRISE_OBJECT_ID</b> (NUMBER)		<b>x</b>	<b>x</b>	<a href="#">History</a>	object	8 The history record with which this group access setting is associated.  --> Links to the specific information about the history record.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		<b>x</b>			int	Unique ID for the group access rights.		
<b>version</b>	<b>VERSION</b> (NUMBER)		<b>x</b>			int	Indicates how many times the group access rights have been updated.		

### 3.1.35 Object Model: Documents

The tables in this appendix provide information about the object model as it relates to documents.

#### 3.1.35.1 TDocument

TDocument contains general information for a document record, such as the document's author, date, content type, name, file size, version, and so on. This table links to the list of custom fields created for the document categories and related information contained in other tables.

Object: TDocument (T\_DOCUMENT)

Attribute	Database	Field in UI	End of	Bridge	Links to object table:	Data	Comments	Commonly used in:
-----------	----------	-------------	--------	--------	------------------------	------	----------	-------------------

	column name		path			type		Rules	
author-->	AUTHOR_ID (NUMBER)	Author (General block)	x	x	<a href="#">TContact</a>	object	8 The contact who created or uploaded the document file.  --> Links to specific information about the user account.		
authorDate	AUTHOR_DATE (DATE)	Date Author ed (General block)	x			date	The date when the file was uploaded. This field can be edited by the user. Time-zone-independent.		
categories->  Before TeamConnect 3.3 SP2: detailList-->	n/a		x	x	<a href="#">EDocuDetail</a>  In Object Navigator, links to <b>Category</b> list which appears in the UI only.		8 The list of categories added to the current object.  --> In the object model, the added categories link to the values that have been added for custom fields, according to field type.  --> In Object Navigator, this attribute is enhanced. It links to a list of all categories defined for the object definition, rather than those added to the current object. From		

							there, you can select a category and traverse to the list of custom fields that belong to it.		
checkedIn By-->	CHECKED_IN_BY_ID (NUMBER)		x	x	<a href="#">YUser</a>	object	8 The user who last checked in the document.  --> Links to specific information about the user account.		
checkedIn On	CHECKED_IN_ON (DATE)		x			date	Date when the document was last checked in.		
checkedOut By-->	CHECKED_OUT_BY_ID (NUMBER)	Checked-Out By (General block)	x	x	<a href="#">YUser</a>	object	8 The user who has checked out the document.  --> Links to specific information about the user account.		
checkedOut On	CHECKED_OUT_ON (DATE)	Checked-Out On (General block)	x			date	The date when the document was checked out.		
checkOutLocation	CHECK_OUT_LOCATION		x			string (250)	The location where the file was downloaded when it was checked out.		

	(VARCH AR2) (250)								
contentObj ect-->  Before TeamConn ect 3.3 SP2: content-->	CONTE NT_ID  (NUMBE R)			x	<a href="#">JDocuConte nt</a>	obj ect	Links to the content of the uploaded file.		
contentType e-->	CONTE NT_TYP E_ID  (NUMBE R)	File Type  (Gener al block)	x	x	<a href="#">YDocuCont entType</a>	obj ect	8 The file type of the document (For example, Microsoft Excel or WordPerfect).  --> Links to the definition information of the document type.		
createdBy-- >	CREATE D_BY_I D  (NUMBE R)	Create d By  (Securi ty block)	x	x	<a href="#">YUser</a>	obj ect	8 The user who created the document record.  --> Links to specific information about the user account.		
createdOn	CREATE D_ON  (DATE)	Create d On  (Securi ty block)	x			dat e	Date and time when the document record was created.		
createdOn BehalfOf	CREATE D_ON_B EHALF_ OF_ID			x	<a href="#">TContact</a>	obj ect	When not null, it indicates that the record was created through an e-billing system or other external	X	



							application. The contact referenced will often be a vendor in TeamConnect.		
defaultCategory-->	DEFAULT_CATEGORY_ID (NUMBER)	<ul style="list-style-type: none"> <li>Default Category (General block)</li> <li>Indicated by a blue diamond in Categories block</li> </ul>	x	x	<a href="#">YRecentlyViewedRecord</a>	object	The default category of the document record.		
groupAccessList-->	n/a			x	<a href="#">EDocGroupAccess</a>		Links to the group security information entered through the Security block of the record.		
historyList-->	n/a	History tab	x	x	<a href="#">THistory</a>		8 The list of all histories that are related to the document.		

							--> Links to specific history information for all of the document's history records.		
modifiedBy -->	MODIFIED_BY_ID (NUMBER)	Modified By (Security block)	x	x	<a href="#">YUser</a>	object	8 The user who last modified the document record.  --> Links to specific information about the user account.		
modifiedOn	MODIFIED_ON (DATE)	Modified On (Security block)	x			date	The date and time when the document record was last modified.		
name	NAME (VARCHAR2) (250)	Name (General block)	x			string (250)	The file name of the document. This name can be edited by the user.		
note-->	NOTE_ID (NUMBER)	Notes (General block)		x	<a href="#">JNote</a>	object	Links to the table that contains the text entered into the <b>Notes</b> field of the record.		
parentFolder-->	PARENT_FOLDER_ID (NUMBER)			x	<a href="#">TDocument</a>	object	Links to the information about the parent document folder of this document folder.		
primaryKey	PRIMARY_KEY		x			int	The unique ID of the document record.		

	(NUMBER)								
<b>securityTypeIID</b>	<b>SECURITY_TYPE_IID</b> (NUMBER)	<b>Public/Private</b> (Security block)	<b>x</b>				int	Security Type of Document Record:  0 - Public 1 - Currently Not Used 2 - Private	
<b>shortcutToDocument--&gt;</b>	<b>SHORTCUT_TO_DOCUMENT_ID</b> (NUMBER)			<b>x</b>	<a href="#">TDocument</a>		object	When this document is a Shortcut (specifically, if typeID = S), links to the document to which the shortcut is pointing.	
<b>size</b>	<b>DOCUMENT_SIZE</b> (NUMBER)	<b>Size</b> (Documents screen: Java Applet version only)	<b>x</b>				int	File size, in kilobytes (KB).	
<b>treeKey</b>	<b>TREE_KEY</b> (VARCHAR2) (700)		<b>x</b>				string (700)	The chain of primary keys that identifies the parent tree of the file or folder. Used for searching and reporting purposes only.	
<b>type Before TeamConnect 3.3</b>	<b>TYPE_ID</b> (NUMBER)		<b>x</b>				int	Specifies whether the document is a:  D - Folder	

SP2: typeIID							L - File P - Hyperlink S - Shortcut T - History		
url  Before TeamConn ect 3.3 SP2: urlString	URL_ST RING  (VARCH AR2) (2000)	Create New Hyperli nk Button	x			stri ng (20 00)	The URL for a hyperlink.		
userAccess List-->	n/a			x	<a href="#">EDocuUser Access</a>		Links to the user security information entered through the <b>Security</b> block of the record.		
version	VERSIO N  (NUMBE R)		x			int	Indicates how many times the record has been updated.		
versionText	VERSIO N_TEXT  (VARCH AR2) (250)	Comm ents  (Check In dialog box)	x			stri ng (25 0)	The text that is entered in the <b>Comments</b> field when a user checks in the document.  <b>Note:</b> <i>This is availabl e to users only when Docum ent Version Control is enabled in</i>		

										<b>System Settings</b>		
--	--	--	--	--	--	--	--	--	--	------------------------	--	--

### 3.1.35.2 JDocuContent

JDocuContent represents the contents of a document uploaded to TeamConnect, including the actual file and its file extension.

Object: JDocuContent (T\_DOCU\_CONTENT)

Attribute	Database column name	End of path	Bridge	Data type	Comments	Commonly used in:	
						Rules	Temp/Wiz
<b>data</b>	<b>DATA</b> (VARCHAR)	x		data	The actual content of the corresponding uploaded file.		
<b>documentType</b>	<b>DOCUMENT_TYPE</b> (VARCHAR2) (50)	x		string (50)	The file extension of the document.  <i>Note: The preceding period (.) character is added to the file extension, for example, .txt.</i>		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)	x		int	The unique ID of the content of a file uploaded to TeamConnect.		
<b>version</b>	<b>VERSION</b> (NUMBER)	x		int	Indicates how many times the file content has been updated.		

#### 3.1.35.2.1 YDocuContentType

YDocuContentType contains information about content types defined for document records. This information includes the name of the document type (such as Text or XML), file extension, and mime type. This table represents the **Document Types** block on the **Document Types** tab of the Document object definition.

Object: YDocuContentType (Y\_DOCU\_CONTENT\_TYPE)

Attribute	Database column name	Field in UI	End of path	Bridge	Data type	Comments	Commonly used in:	
							Rules	Temp/Wiz
fileExtension	FILE_EXTENSION (VARCHAR2) (50)	File Extension (Document Types tab of Object Definition)	x		string (50)	The file extension of this document type.		
iconFileName	ICON_FILE_NAME (VARCHAR2) (50)	Icon File (Document Types tab of Object Definition)	x		string (50)	The name of the image file that should be used in TeamConnect to represent this document type.		
mimeType  Before TeamConnect 3.3 SP2: mimeTypeString	MIME_TYPE_STRING (VARCHAR2) (50)	Mime Type (Document Types tab of Object Definition)	x		string (50)	The mime type that should be used for documents of this type.		
name	NAME (VARCHAR2) (50)	Document Type Name (Document Types tab of Object	x		string (50)	The name of the document type, as it should appear in the Documents screen.		

		Definiton)						
<b>primaryKey</b>	<b>PRIMARY_KEY</b> <b>(NUMBER)</b>		<b>x</b>		int	The unique ID of the document type.		
<b>version</b>	<b>VERSION</b> <b>(NUMBER)</b>		<b>x</b>		int	Indicates how many times the document type has been updated.		

### 3.1.35.3 EDocuDetail

EDocuDetail contains all categories added to document records. Each time a category is added to a document record, an entry is made in this table in the database.

In Object Navigator, when you list categories, a list of all categories that have been defined for the current object definition appears. This list is labeled **Category**. From each category, you can traverse to a table called **Detail Fields**, which displays all custom fields belonging to the category you selected.

For more information about how custom field values are stored in the database, see Object Model: Custom Fields.

Object: EDocuDetail (E\_DOCU\_DETAIL)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wiz
<b>category--&gt;</b>	<b>CATEGORY_ID</b> <b>(NUMBER)</b>	<b>Category</b>	<b>x</b>	<b>x</b>	<a href="#">YRecentlyViewedRecord</a>	object	The category of the document with which the details are associated.		
<b>owner--&gt;</b>	<b>ENTERPRISE_OBJECT_ID</b> <b>(NUMBER)</b>			<b>x</b>	<a href="#">TDocument</a>	object	The document with which the custom fields are associated.		
<b>isManual</b>	<b>IS_MANUAL</b>		<b>x</b>			boolean	Specifies whether the category is		

	(NUMBER)						added manually or automatically to the document. Categories are added automatically when they are the parent of the category that is added manually.  0 - Added Automatically 1 - Added Manually		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the document category record.		
version	VERSION (NUMBER)		x			int	Indicates how many times the added category was updated.		

#### 3.1.35.4 EDocuUserAccess

EDocuUserAccess contains the user access information that is set in the Security block of each document record. For example, it includes whether each user that is listed in the **Security** block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm**(ission) rights.

Object: EDocuUserAccess (E\_DOCU\_USER\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
isRead	IS_READ (NUMBER)	Read checkbox	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).		



							This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isUpdate</b>	<b>IS_UPDATE (NUMBER)</b>	<b>Update checkbox</b>	<b>x</b>			int	Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isDelete</b>	<b>IS_DELETE (NUMBER)</b>	<b>Delete checkbox</b>	<b>x</b>			int	Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
<b>isPerm</b>	<b>IS_PERM (NUMBER)</b>	<b>Perm checkbox</b>	<b>x</b>			int	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		

<b>allowDenyID</b>	<b>ALLOW_DENY_ID</b> (CHAR) (1)	<b>Option</b>	x			char	Indicates whether the user is allowed or denied access to the record.  a - Allow d - Deny		
<b>isManual</b>	<b>IS_MANUAL</b> (NUMBER)		x			boolean	Indicates whether the user rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		
<b>owner-&gt;</b>	<b>ENTERPRISE_OBJECT_ID</b> (NUMBER)		x	x	<a href="#">TDocument</a>	object	8 The document record with which this user access setting is associated.  -> Links to the specific information about the document record.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	Unique ID for the user access rights.		
<b>user-&gt;</b>	<b>USER_ID</b> (NUMBER)	<b>List</b>	x	x	<a href="#">YUser</a>	object	8 Specifies the user the access rights pertain to.  -> Links to the specific information about the user account.		
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Indicates how many times the user access rights have been updated.		

### 3.1.35.5 EDocuGroupAccess

EDocuGroupAccess contains the group access information that is set in the **Security** block of each document record. For example, it includes whether each group that is listed in the **Security** block is allowed or denied the **Read**, **Update**, **Delete**, and **Perm**(ission) rights.

Object: EDocuGroupAccess (E\_DOCU\_GROUP\_ACCESS)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
isRead	IS_READ (NUMBER)	Read checkbox	x			int	Specifies whether the Read operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isUpdate	IS_UPDATE (NUMBER)	Update checkbox	x			int	Specifies whether the Update operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
isDelete	IS_DELETE (NUMBER)	Delete checkbox	x			int	Specifies whether the Delete operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the		

							operations are allowed or denied. (See allowDenyIID.)		
isPerm	IS_PERM (NUMBER)	Per m che ck box	x			int	Specifies whether the Set Permission operation has been selected for granting or denial. Value is 1 (selected) or 0 (not selected).  This attribute does not determine whether the operations are allowed or denied. (See allowDenyIID.)		
allowDenyIID	ALLOW_DENY_IID (CHAR) (1)	Opti on	x			char	Indicates whether the group is allowed or denied access to the record.  a - Allow d - Deny		
group-->	GROUP_ID (NUMBER)	List	x	x	<a href="#">YGroup</a>	obje ct	8 Specifies the group that the access rights pertain to.  --> Links to the specific information about the group account.		
isManual	IS_MANUAL (NUMBER)		x			bool ean	Indicates whether the group access rights were assigned manually through the Security block of the record or automatically by the system.  0 - Assigned manually 1 - Assigned automatically		

owner->	ENTERPRISE_OBJECT_ID (NUMBER)		x	x	<a href="#">TDocument</a>	object	8 The document record with which this group access setting is associated.  --> Links to the specific information about the document record.		
primaryKey	PRIMARY_KEY (NUMBER)		x			int	Unique ID for the group access rights.		
version	VERSION (NUMBER)		x			int	Indicates how many times the group access rights have been updated.		

### 3.1.36 Object Model: Common

The tables in this appendix are not related specifically to one of TeamConnect's main T-tables. They have relationships to other objects in the object model and are referred to as common or general objects.

#### 3.1.36.1 JNote

JNote contains the text of notes entered for a record. It is related to all the T-tables except for **TProject** and **THistory**.

Object: JNote (J\_NOTE)

Attribute	Database column name	Field in UI	End of path	Bridge	Data type	Comments	Commonly used in:	
							Rules	Temp/Wiz
noteText	NOTE_TEXT (VARCHAR2)	Notes (General block)	x		longstring	Contains the text of the notes entered for the record. The TeamConnect user interface enforces a limitation of 20,000 characters for this field.		
primaryKey	PRIMARY_KEY		x		int	Unique ID for the notes text.		

	(NUMBER)							
version	VERSION (NUMBER)		x		int	Indicates how many times the note record has been updated.		

### 3.1.36.2 UGrupMember

UGrupMember contains information about a group account's members. This includes the user who is a member of a group and the group to which the user who is a member—essentially, joining the **YGroup** and **YUser** tables. This table represents the **Group Members** block on the **Members** tab of a group account.

Object: UGrupMember (U\_GRP\_MEMBER)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
group->	GROUP_ID (NUMBER)		x	x	<a href="#">YGroup</a>	object	8 The group to which this group member belongs. --> Links to specific information about the user group.	x	
primaryKey	PRIMARY_KEY (NUMBER)		x			int	The unique ID of the group member.		
user->	USER_ID (NUMBER)		x	x	<a href="#">YUser</a>	object	8 The user who is a group member. --> Links to specific information about the user account.		
version	VERSION (NUMBER)		x			int	Indicates how many times the group member has been updated.		

### 3.1.36.3 WObjdDetailField

WObjdDetailField contains information about custom fields defined for a category, which includes the field name, default value, field type (such as Text or Number), label, corresponding database name, whether or not it is included in Data Warehouse, whether or not it is required, and category to which the custom field belongs. This table represents the **Custom Fields** block on the **Custom Fields** tab of an object definition.

Object: WObjdDetailField (Y\_OBJ\_DETAIL\_FIELD)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
defaultValue	DEFAULT_VALUE (VARCHAR 2)	Default Value (Custom Fields tab, Object Definition)	x			long String	The default value of the custom field.		
detailFieldTypeId	DETAIL_FIELD_TYPE_ID (NUMBER)	Field Type (Custom Fields tab, Object Definition)	x			int	Specifies the custom field type, which is either:  1 - Text 2 - Number 3 - Date Field 4 - Scroll Text 5 - Check Box 6 - List 8 - Involved 9 - Memo Text 10 - Label Only 11 - Custom Object		

extraInfo	EXTRA_INFO (VARCHAR2) (50)	Label (Custom Fields tab, Object Definition)	x			string (50)	The label of the custom field.		
invlCategory-->	INVOLVED_CATEGORY_ID (NUMBER)	The list for selecting involved role, if field is of type Involved (Custom Fields tab, Object Definition)	x	x	<a href="#">YRecently ViewedRecord</a>	object	The involved role selection for the custom field, if it is of type Involved.		
isDataMart	IS_DATA_MART (NUMBER)	Include in Data Warehouse (Custom Fields tab, Object Definition)	x			boolean	Indicates whether the field is to be included in the Data Warehouse tables.		
isExcludedFromCustomSearch	IS_EXCLUDED_FROM_CUSTOM_SEARCH (NUMBER)	Is Excluded from Custom Search?	x			boolean	Indicates whether the custom field is hidden when constructing custom searches, or not.		



		(Custom Fields tab, Object Definition)							
<b>isRequired</b>	<b>IS_REQUIRED</b> (NUMBER)	<b>Is Required?</b> (Custom Fields tab, Object Definition)	x			boolean	Indicates whether the custom field is required or not.		
<b>isTimeZoneIndependent</b>	<b>IS_TIME_ZONE_INDEPENDENT</b> (NUMBER)	<b>Is Time Zone Independent</b> (Custom Fields tab, Object Definition)	x			boolean	Indicates whether the custom field of date datatype is adjusted in the UI for the end user's timezone (FALSE), or not (TRUE).		
<b>name</b>	<b>NAME</b> (VARCHAR2) (30)	<b>Field Name</b> (Custom Fields tab, Object Definition)	x			string (30)	The database name for the custom field.		
<b>objectQualifier</b>	<b>OBJECT_QUALIFIER</b> (VARCHAR2) (2000)	<b>Qualifier</b> (Custom)	x			string (2000)	The qualifier path that is used to filter the available records for the		

		<b>Fields</b> tab, Object Definiti on)					custom field, if the field is of type Custom Object.		
<b>owner--&gt;</b>	<b>CATEGORY_ID</b> <b>(NUMBER)</b>	<b>For</b> <b>Category</b> <b>(Custo</b> <b>m</b> <b>Fields</b> tab, Object Definiti on)	<b>x</b>	<b>x</b>		obje ct	8 The category to which this custom field belongs.  --> Links to the definition information of the category.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> <b>(NUMBER)</b>		<b>x</b>			int	Unique ID for the custom field.		
<b>searchView</b>	<b>SEARCH_VIEW_ID</b> <b>(NUMBER)</b>	<b>Search</b> <b>View</b> <b>(Custo</b> <b>m</b> <b>Fields</b> tab, Object Definiti on)		<b>x</b>		obje ct	Available only for custom fields of type Involved or Custom Object.		
<b>table--&gt;</b>	<b>TABLE_ID</b> <b>(NUMBER)</b>	The list for selectin g a lookup table, if the field is of type List  <b>(Custo</b> <b>m</b> <b>Fields</b> tab, Object	<b>x</b>	<b>x</b>	<a href="#">YDetailLookupTable</a>	obje ct	Specifies the custom lookup table linked to the custom field, if the field is of type List.		

		Definiton)							
version	<b>VERSION (NUMBER)</b>		x			int	Number that indicates how many times the custom field has been updated.		

### 3.1.36.4 WObjdProjectInfo

WObjdProjectInfo represents a custom project object definition, such as whether the project is contact-centric, embedded, requires a parent project, and various numbering pattern properties. This table links to lists of phases and phase transitions that have been defined for this object.

Object: WObjdProjectInfo (Y\_APPLICATION)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
autoNumPattern	<b>AUTO_NUM_PATTERN (VARCHAR2) (250)</b>	Pattern  (Unique ID tab)	x			string (250)	Indicates the pattern used for auto-numbering.		
autoNumSequence	<b>AUTO_NUM_SEQUENCE (NUMBER)</b>	Next Sequence Number  (Unique ID tab)	x			int	Indicates the next number in the sequence for auto-numbering.		
childList-->	n/a			x	<a href="#">WObjdProjectInfo</a>		Links to the child and embedded object definitions for this custom		

							object definition.		
<b>conditionTypeIID</b>	<b>CONDITION_TYPE_IID</b> <b>(NUMBER)</b>	<b>Unique ID</b> tab	<b>x</b>			int	Indicates how instances of this object definition will be uniquely identified:  0 - Auto numbering  1 - Pattern of object attributes  2 - Entered manually by user  3 - Default numbering (will be hidden from the user)		
<b>displayFormatID</b>	<b>DISPLAY_FORMAT_IID</b> <b>(NUMBER)</b>	<b>Name</b> tab	<b>x</b>			int	Indicates whether the object definition has been set to use a pattern of attributes for automatic naming.  5 - Allow users to enter name manually  6 - Name constructed from a pattern of attributes		
<b>displayOrder</b>	<b>DISPLAY_ORDER</b>	<b>Order</b> <b>(General tab)</b>	<b>x</b>			int	Indicates the order in which the custom object should appear in the		

	(NUMBER)						user interface drop-down lists.		
formatIID	FORMAT_IID (NUMBER)	Format information in Unique ID tab	x			int	Indicates how the instances of this object definition should be displayed in the UI:  1 - Name 2 - Number 3 - Name - Number 4 - Number - Name		
isAutoNum Editable	IS_AUTO_NUM_EDITABLE (NUMBER)		x			boolean	Indicates whether users are able to edit the number		
isAutoNum Enabled	IS_AUTO_NUM_ENABLED (NUMBER)		x			boolean	Indicates whether the custom object definition is set to be auto-numbered.		
isContactCentric	IS_CONTACT_CENTRIC (NUMBER)	Contact-centric (General tab)	x			boolean	Indicates whether the custom object definition is contact-centric.		
isEmbedded	IS_EMBEDDED		x			boolean	Indicates whether the custom		

	(NUMBER)						object definition is embedded.		
isNotDisplayedInMenu	IS_NOT_DISPLAYED_MENU (NUMBER)	Do not show this child object definition in the All Services menu  (General tab)	x				boolean  Indicates whether the child custom object definition appears in the <b>All Services</b> menu drop-down lists or is available only through parent records.		
isParentRequired	IS_PARENT_REQUIRED (NUMBER)	Parent Custom Object Required  (General tab)	x				boolean  Indicates whether instances of the custom object definition require a parent.		
listDisplayTypeID	LIST_DISPLAY_TYPE_ID (NUMBER)	Options in List Display tab of embedded object definition	x				int  For an embedded custom object definition, indicates how the embedded object records appear in the UI:  0 - Search view  1 - Batch view		

parent-->	<b>PARENT_ID</b> (NUMBER)	Parent Custom Object (General tab)	x	x	<a href="#">WObjdProjectInfo</a>	object	8 Indicates the custom object definition that is the parent of this custom object definition.  --> Links to information about the parent custom object definition.		
phaseList-->	n/a	Phase s tab		x	<a href="#">WObjdPhaseType</a>		Links to the list of phases that have been defined for this object definition.		
phaseTransitionList-->	n/a	Phase Transitions tab		x	<a href="#">WObjdPhaseTransition</a>		Links to the list of phase transitions that have been defined for this object definition.		
primaryKey	<b>PRIMARY_KEY</b> (NUMBER)		x			int	The primary key of the object definition.		
role	<b>ROLE</b> (VARCHAR2) (250)	Role (General tab)	x				If the object definition is contact-centric, indicates the label for the contact field.		
rootAssigneeType-->	<b>ROOT_ASSIGNEE</b>		x	x	<a href="#">LProjAssigneeType</a>	object	8 Indicates the root node		

	<b>E_TYPE_ID</b> (NUMBER)						of the assignee role lookup table.  --> Links to information about the assignee role.		
<b>selectionDisplayTypeID</b>	<b>SELECTION_DISPLAY_TYPE_ID</b> (NUMBER)	<b>Default selection mechanism</b>  (General tab)	x			int	Indicates which selection mechanism is the default for this object definition:  1 - Drop-down list  2 - Search module		
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Number that indicates how many times the object definition has been updated.		

### 3.1.36.5 YDetailLookupItem

YDetailLookupItem contains information about custom lookup table items, such as the item's name, tree position, and display order. The lookup item hierarchy is handled by the attributes **parent-->** and **childList-->**. This table represents the item list displayed in the **Custom Lookup Tables** block on the **Custom** tab of the **Lookup Tables** screen, which are defined to be used for custom fields of type List.

Object: YDetailLookupItem (Y\_DETAIL\_LOOKUP\_ITEM)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz



<b>childList--&gt;</b>	n/a			x	<a href="#">YDetailLookupItem</a>		Links to the list of all child items that belong to this lookup table item.		
<b>displayOrder</b>	<b>DISPLAY_ORDER</b> (NUMBER)	<b>Order</b> (Lookup Tables, Custom tab)	x			int	The order in which the item appears in the list.		
<b>name</b>	<b>NAME</b> (VARCHAR2) (50)	<b>Item Name</b> (Lookup Tables, Custom tab)	x			string (50)	The name of the lookup item.		
<b>parent--&gt;</b>	<b>PARENT_ID</b> (NUMBER)	<b>In node:</b> (drop-down list, Lookup Tables, Custom tab)		x	<a href="#">YDetailLookupItem</a>	object	Links to the parent item of this lookup table item.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	The unique ID of the custom lookup item.		
<b>table--&gt;</b>	<b>TABLE_ID</b> (NUMBER)	<b>Show items belonging to</b> (drop-down list, Lookup Tables, Custom tab)		x	<a href="#">YDetailLookupTable</a>	object	Links to the lookup table to which this lookup item belongs.		

<b>treePosition</b>	<b>TREE_POSITION</b> (VARCHAR2) (250)	<b>Tree Position</b> (Lookup Tables, Custom tab)	x			string (250)	The unique four-character alphanumeric code that is created when the lookup item is defined.		
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Indicates how many times the lookup item has been updated.		

### 3.1.36.6 YDetailLookupTable

YDetailLookupTable contains information about custom lookup tables, such as the name and unique code. This table links to **YDetailLookupItem** for the root lookup item. This table represents the **Custom Lookup Tables** block on the **Custom** tab of the **Lookup Tables** screen.

Object: YDetailLookupTable (Y\_DETAIL\_LOOKUP\_TABLE)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wiz
<b>isActive</b>	<b>IS_ACTIVE</b>	<b>activate</b> and <b>inactivate</b> buttons	x			int	Indicates whether the table entry is active (value 1) and can be used when editing records containing that entry. If it is inactive (value 0), records may contain that entry for reporting and historical purposes, but cannot be saved when editing, unless that entry is changed to some other, active table entry.		

<b>name</b>	<b>NAME</b> (VARCHAR2) (50)	<b>Show items belonging to</b>  (Lookup Tables, Custom tab)	x			string (50)	The name of the custom lookup table.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> (NUMBER)		x			int	Unique ID for the custom lookup table.		
<b>rootItem→</b>	<b>ROOT_ITEM_ID</b> (NUMBER)			x	<a href="#">YDetailLookupItem</a>	object	Links to the information about the root node in the lookup table, such as its name and its child items.		
<b>uniqueCode</b>	<b>UNIQUE_CODE</b> (VARCHAR) (4)	<b>Unique Code</b>  (Lookup Tables, Custom tab)	x			string (4)	Four-character alphanumeric code that uniquely identifies the lookup table.		
<b>version</b>	<b>VERSION</b> (NUMBER)		x			int	Indicates how many times the lookup table has been updated.		

### 3.1.36.7 YGroup

YGroup contains information about the group accounts in TeamConnect including the name of the group account and its creation date. This table also links to lists that describe a group's access rights settings, the group's members, tools rights settings, and default object view settings.

Object: YGroup (Y\_GROUP)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:
-----------	----------------------	-------------	-------------	--------	------------------------	-----------	----------	-------------------

								Rules	Temp / Wiz
byPassSearchLimits	BYPASS_SEARCH_LIMITS		x			boolean	Indicates whether the user account is set to allow users to see the <b>Return all results</b> hyperlink in the search results heading.  Clicking the hyperlink allows users to bypass the system settings for the maximum number of records retrieved or maximum number of seconds for a search.		
createdBy->	CREATED_BY_ID (NUMBER)		x	x	<a href="#">User</a>	object	8 The user who created the group account.  --> Links to specific information about the user account.		
createdOn	CREATED_ON (DATE)		x			date	The date when the group account was created.		
functional AccessList->	n/a			x			Links to the group's rights settings.		
users--> Before TeamConnect 3.3	n/a			x	<a href="#">UGroupMember</a>		Links to the join table that contains information for all		

<b>SP2: groupMemberList--&gt;</b>							members of the group.		
<b>modifiedBy--&gt;</b>	<b>MODIFIED_BY_ID</b> <b>(NUMBER)</b>		x	x	<a href="#">YUser</a>	object	8 The user who last modified the group account.  --> Links to specific information about the user account.		
<b>modifiedOn</b>	<b>MODIFIED_ON</b> <b>(DATE)</b>		x			date	The date when the group account was last modified.		
<b>displayName</b>  <b>Before TeamConnect 3.3</b> <b>SP2: name</b>	<b>NAME</b> <b>(VARCHAR2)</b> <b>(50)</b>	<b>Group Account</b>  <b>(General tab)</b>	x			string (50)	The name of the group account.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b> <b>(NUMBER)</b>		x			int	The unique ID of the group account.		
<b>shortDescription</b>	<b>SHORT_DESCRIPTION</b> <b>(VARCHAR2)</b> <b>(250)</b>	<b>Description</b>  <b>(General tab)</b>	x			string (250)	A description of the group account.		
<b>toolList--&gt;</b>	n/a			x			Links to the tools rights settings of the group.		
<b>uniqueKey</b>	<b>UNIQUE_KEY</b> <b>(VARCHAR2)</b> <b>(50)</b>	<b>Unique Key</b>  <b>(General tab)</b>	x			string (50)	The alphanumeric code that uniquely identifies the group.	x	

<b>version</b>	<b>VERSION (NUMBER)</b>		<b>x</b>			int	Indicates how many times the group account has been updated.		
<b>viewDefaultList--&gt;</b>	n/a			<b>x</b>			Links to the default object view settings for the group.		

### 3.1.36.8 YRecentlyViewedRecord

YRecentlyViewedRecord allows you to define what records appear on the page when the user selects the Recently Viewed Records collection in the left pane.

Object: YRecentlyViewedRecord (Y\_RECENTLY\_VIEWED\_RECORD)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
<b>primaryKey</b>	<b>PRIMARY_KEY (NUMBER)</b>					int	The ID number of the current Recently Viewed record entry.		
<b>recordPrimaryKey</b>	<b>RECORD_PRIMARY_KEY (NUMBER)</b>					int	The primary key number of a recently viewed record.		
<b>uniqueCode</b>	<b>UNIQUE_CODE (VARCHAR2) (4)</b>					string (4)	The unique code of a recently viewed record.		
<b>version</b>	<b>VERSION (NUMBER)</b>					int	The version number of the current Recently Viewed record entry.		
<b>viewedByID--&gt;</b>	<b>VIEWED_BY_ID (NUMBER)</b>				<a href="#">YUser</a>	int	The ID number of the user viewing the current Recently Viewed collection.		

viewedOn	VIEWED_ON (DATE)					date	The date and time a record was last viewed by the current user whose ID is <b>viewedByID</b> .		
----------	---------------------	--	--	--	--	------	--	--	--

### 3.1.36.9 YUser

YUser contains information about the user accounts in TeamConnect. This information includes the username, user type (such as normal or super), expiration date, password, authentication mechanism, default group, and the associated contact record. It also links to lists that describe a user's access rights settings, group accounts to which a user is a member, tools rights settings, and preference settings.

Object: YUser (Y\_USER)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp / Wiz
accountExpiresOn	ACCOUNT_EXPIRES_ON (DATE)	Account expires on (General tab)	x			date	Date when the user account expires.		
authenticationType	AUTH_TYPE (VARCHAR2) (10)	Authentication (General tab)	x			string (10)	The mechanism by which the user is authenticated.		
byPassSearchLimits	BYPASS_SEARCH_LIMITS		x			boolean	Indicates whether the user account is set to allow users to see the <b>Return all results</b> hyperlink in		

							<p>the search results heading.</p> <p>Clicking the hyperlink allows users to bypass the system settings for the maximum number of records retrieved or maximum number of seconds for a search.</p>		
<b>calendarAppUserID</b>	<b>CALENDAR_APP_USER_ID (VARCHAR2) (250)</b>	<b>Calendaring Application User ID</b>				string (250)	<p>Allows you to map a calendaring server user to a TeamConnect user. User ID of the appropriate calendaring application must be entered. Then, for TeamConnect, you must set: whether the account is active, the expiration date, whether the user must change password at next log-in, and whether to enable assistive technology.</p>		



<b>contact--&gt;</b>	<b>CONTACT_ID</b> <b>(NUMBER)</b>	Account belongs to <b>(General tab)</b>	x	x	<a href="#">TContact</a>	object	8 The contact record associated with this user account.  --> Links to specific information about the contact record.		
<b>createdBy--&gt;</b>	<b>CREATE_BY_ID</b> <b>(NUMBER)</b>		x	x	<a href="#">YUser</a>	object	8 The user who created the user account.  --> Links to specific information about the user account.		
<b>createdOn</b>	<b>CREATE_ON</b> <b>(DATE)</b>		x			date	The date when the user account was created.		
<b>defaultGroup--&gt;</b>	<b>DEFAULT_GROUP_ID</b> <b>(NUMBER)</b>	Indicated by a blue diamond on <b>Groups</b> tab	x	x	<a href="#">YGroup</a>	object	8 The group that is selected as the default group for this user account.  --> Links to specific information about the user group.	x	
<b>documentFolder--&gt;</b>	<b>DOCUMENT_FOLDER_ID</b> <b>(NUMBER)</b>			x	<a href="#">TDocument</a>	object	Links to the information about the personal document folder of the user.		

<b>functionalAccessList--&gt;</b>	n/a	<b>Rights</b> tab  <b>Category</b> <b>Rights</b> tab		x	UUser Function alAcc ess		Links to the user's rights settings.		
<b>groupMemberList--&gt;</b>	n/a			x	<a href="#">UGrup Membe r</a>		Links to the join table that contains information for all groups in which this user is a member.	x	
<b>isAccountActive</b>	<b>ACCOU NT_TYP E</b>  <b>(NUMBE R)</b>	<b>Accoun t expires</b> on (check box, <b>Genera l</b> tab)	x			bool ean	Indicates whether the user account is set to expire on a certain date.  0 - false 1 - true		
<b>isActive</b>	<b>IS_ACTI VE</b>  <b>(NUMBE R)</b>	<b>Accoun t is active</b>  <b>(Gener al</b> tab)	x			bool ean	Indicates whether the user account is active.  0 - false 1 - true		
<b>isChangePasswordNextLogin</b>	<b>IS_CHA NGE_P WD_NE XT_LOG IN</b>  <b>(NUMBE R)</b>	<b>Must change passwo rd on next login</b>  <b>(Gener al</b> tab)	x			bool ean	Indicates whether the user is required to change the password the next time he/ she logs in to TeamConnect.  0 - false 1 - true		

<b>isLocked</b>	<b>IS_LOCKED</b> <b>(NUMBER)</b>	<b>Account is locked</b> <b>(General tab)</b>	<b>x</b>			boolean	Indicates whether the user account is locked after a number of failed login attempts.  0 - false 1 - true		
<b>lastFailedAttemptOn</b>	<b>LAST_FAILED_ATTEMPT_ON</b> <b>(DATE)</b>		<b>x</b>			date	Date and time of the last failed login.		
<b>lastLoginOn</b>	<b>LAST_LOGIN_ON</b> <b>(DATE)</b>		<b>x</b>			date	Indicates the last time the user logged on to TeamConnect.		
<b>ldapStatusID</b>	<b>LDAP_STATUS_ID</b> <b>(DATE)</b>		<b>x</b>			int	Currently not used.		
<b>longSettingList--&gt;</b>	n/a						Currently not used.		
<b>modifiedBy--&gt;</b>	<b>MODIFIED_BY_ID</b> <b>(NUMBER)</b>		<b>x</b>	<b>x</b>	<a href="#">YUser</a>	object	8 The user who last modified this user account.  --> Links to specific information about the user account.		
<b>modifiedOn</b>	<b>MODIFIED_ON</b> <b>(DATE)</b>		<b>x</b>			date	The date when the user account was last modified.		

<b>numberOfFailedAttempts</b>	<b>NUMBER_OF_FAILED_ATTEMPTS</b>  (NUMBER)		x			int	Indicates the number of failed login attempts.		
<b>parent--&gt;</b>	<b>PARENT_ID</b>  (NUMBER)						Currently not used.		
<b>password</b>	<b>PASSWORD</b>  (VARCHAR2) (250)	<b>Password</b>  (General tab)	x			string (250)	The password required for the user to log in using this user account.		
<b>passwordSavedDate</b>	<b>PASSWORD_SAVED_DATE</b>  (DATE)		x			date	The date when the password was last saved. This value is used to determine whether the password has expired.		
<b>primaryKey</b>	<b>PRIMARY_KEY</b>  (NUMBER)		x			int	The unique ID of the user account.		
<b>settingList--&gt;</b>	n/a			x	UUser Setting		Links to the user's preferences settings.		
<b>shortDescription</b>	<b>SHORT_DESCRIPTION</b>  (VARCHAR2) (250)	<b>Description</b>  (General tab)	x			string (250)	Description of the user account.		

<b>toolList--&gt;</b>	n/a			x			Links to the tools rights settings of the user.		
<b>userId</b>	<b>USER_T TYPE_ID</b> <b>(NUMBER)</b>	<b>User Type</b> <b>(General tab)</b>	x			int	Indicates the type of this user account, which may be:  1 - Super 2 - Normal 3 - Limited-Privilege		
<b>username</b>	<b>USERNA ME</b> <b>(VARCHAR2)</b> <b>(100)</b>	<b>Username</b> <b>(General tab)</b>	x			string (100)	The username assigned to this user account.		
<b>version</b>	<b>VERSIO N</b> <b>(NUMBER)</b>		x			int	Indicates how many times the user account has been updated.		

3.1.36.10 YUserSubscribedCollection

YUserSubscribedCollection allows you to define, configure, and set up the display of the types of collections groups of users may subscribe to.

Object: YUserSubscribedCollection (Y\_USER\_SUBSCRIBED\_COLLECTION)

Attribute	Database column name	Field in UI	End of path	Bridge	Links to object table:	Data type	Comments	Commonly used in:	
								Rules	Temp/Wiz
<b>contactGroupID</b>	<b>CONTACT_GROUP_ID</b> <b>(NUMBER)</b>				YContactGroup	int	The primary key number of the static Contact collection that the user has subscribed to.		

display Order	DISPLAY_ORDER (NUMBER)					int	The display order of the subscribed collections, expressed as a number, for example, 5th in order = 5.		
primary Key	PRIMARY_KEY (NUMBER)					int	The ID number of the subscribed collection.		
searchID->	SEARCH_ID (NUMBER)					int	The primary key number of the search view that the user has subscribed to.		
userID-->	USER_ID (NUMBER)				<a href="#">YUser</a>	int	The primary key number of the subscribing user.		
version	VERSION (NUMBER)					int	The version number of the collection subscription.		

### 3.1.37 Glossary

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) |

#### A

##### **Account**

This term can refer to the system object definition called Account, that can be used for tracking various amounts of money allocated for specific purposes.

It can also refer to the TeamConnect records that store the [User](#) and [Group](#) information, such as name, password, rights and so on. User and Group Accounts can be accessed only by the designated users, such as [TeamConnect administrators](#), on the **Admin** tab.

##### **Action**

There are two types of actions in TeamConnect—wizard actions and rule actions.

A wizard page action is an action that is triggered by a page transition in a [Wizard](#). For example, when a user is creating a [Record](#) using a wizard, the wizard can do one of the following actions:

- Populate a field in the record with a specified value.
- Execute a Java class file, which can create another record, do a calculation, or do another task automatically.

For more details about wizard actions, see [Creating Custom Pages](#).

A rule action is an action that TeamConnect performs when a [Rule](#) is triggered in TeamConnect. Depending on the type of rule, the action that the rule executes could be the following:

- Deny the user the ability to do an operation, such as delete, post, void, update, or create a [Record](#), and display a message to the user.
- Allow the user to do one of the operations above (in other words, do nothing).
- Send the record into a [Route](#) of users who can approve or reject the operation attempted by the user.
- Do [Custom Actions](#) specified in a JavaScript or Java class file, such as create a record or interact with an outside application (such as Microsoft Exchange).

For more details about rule actions, see the [Rule Actions](#).

### **Admin Rights**

These are rights that [control access to all features](#) in TeamConnect that can be accessed by the **Admin** tab in the user interface. These include object definition, lookup tables, routes, home pages, system settings, and so on.

### **Administrator (TeamConnectAdmin User)**

A user whose job responsibilities include creating or maintaining user and group accounts, specifying the appropriate system settings for the organization, creating and/or maintaining system-wide Home Pages and Portal Panes, and making minor changes to the system design if necessary. [These responsibilities](#) are reflected by the [Admin Rights](#) assigned to this user accordingly.

### **API**

Application Program Interface, a set of routines, protocols, and tools for building software applications.

TeamConnect's API can be used for customizing TeamConnect. You can use the API for purposes such as creating Java rules, integration with other systems, and enhancing the capabilities of various areas of TeamConnect, such as Home Pages.

### **API Rule**

See [Custom Action](#).

### **Appender**

A definition of a system or audit log. Appenders can be of various types, including SMTP, socket and file. File appenders are the most commonly used, as the logging statements are simply captured in a file. A log can be a general TeamConnect log, or it can be associated with a specific [logger](#), so that it captures messages only from a certain TeamConnect area, such as portal panes or custom blocks.

### **Applet**

A small utility written in Java. In TeamConnect, applets can be used for the document screen interface. If applets are enabled on the system, users can enable them for their own use on the **Preferences** tab of the **Preferences** screen.

### **Application**

A term used in the previous versions of TeamConnect to refer to [Custom Objects](#), namely, the objects that are created for your organization.

For more details on custom objects, see [Creating Custom Pages](#).

### ***Appointment***

A [System Object](#) definition that is intended to allow end [Users](#) to track their scheduled events, engagements, or meetings. Appointments are automatically available as a system [Block](#) in [Object Views](#) for [Custom Objects](#).

### ***Approval***

The approval process is the major component of TeamConnect [Workflow](#). Authorization from one or more designated users (known as [approvers](#)), which is required for certain actions that are attempted by other users. For example, deleting a matter, posting an [invoice](#), or changing the [phase](#) of a matter could require approval and therefore trigger an approval process. When users trigger an approval process, they are considered [requestors](#) in the approval process. Users' attempted actions are completed upon their final approval.

### ***Approval Rule***

A type of TeamConnect rule, that allows you to define [Conditions](#) that when an action requiring an [Approval](#) is attempted by a user, it is automatically routed to the designated [Approvers](#). The approval process is defined with the use of a [Route](#).

### ***Approvers***

Designated [Users](#) who have the authority to approve or reject certain actions attempted by other users who do not have the permission to complete these actions. There can be several approvers at each level of the [Approval](#) process. This approval level is also known as an approval [Stop](#), which is a component of a [Route](#). During an approval process, approvers may be able to send the approval request to [reviewers](#) for review before they make their decision. Approvers can view their approval requests on the **My Approvals** screen.

### ***Assignees***

TeamConnect [Users](#) who are assigned to one or more [Projects](#) or [Tasks](#). There can be one or several users assigned to a project. In projects each assignee can have a different role, such as attorney, agent, appraiser, paralegal, and so on. Typically, the main assignee is in charge of the whole project, whereas the rest of the assigned users are responsible for certain parts of the project. Tasks can have only one current assignee.

### ***Attendees***

TeamConnect [Users](#) and members of [Contact Groups \(Address Books\)](#) who are expected to take part in an event, such as a meeting or conference call, for which there is an [Appointment](#) scheduled. Designer contact groups are called Address Books in the end-user interface.

### ***Attributes***

Building blocks of [Objects](#), which represent data entered by end users. Attributes correspond to columns within tables in the database.

For example, every [Custom Object](#) consists of over a dozen attributes, all of which are listed in the main database table for that object: **mainAssignee**, **createdOn**, **CreatedBy**, **ClosedOn**, **defaultCategory**, and so on. The corresponding collection of data constitutes an object record.



You use attributes to identify their corresponding fields when working in various areas of TeamConnect, such as [Search Views](#), [Rules](#), and [Wizards](#). To navigate through selecting the necessary attributes, you can use a tool called [Object Navigator](#). For details, see [Using Object Navigator](#).

### **Automated Action**

An action performed when a [rule](#) is triggered that is driven by a Javascript or Java class file. For example, a related [Record](#) can be automatically created when the rule is triggered. For more details, see [Writing Automated Actions](#).

### **Audit Settings**

Settings for tracking the actions of users—which screens they access, which selections they make in the screens they visit, when they log in and log out, and other actions.

## **B**

### **Batch Screen**

A type of tab or section in a tab of a record screen where certain multiple data items of the [Record](#) can be entered and viewed as a single list, all in one screen. In the object model, items added to a record with [batch screens](#) are considered [Sub-objects](#), and are usually represented by J-tables.

An example of a batch screen in the administrative user interface can be the **Custom Fields** tab in object definitions.

### **Block**

A group of system and/or custom fields included into a single file, XML (for custom blocks) or JSP (for system blocks). In the end-user interface, each block is typically displayed as a separate tab or a section on the record screen.

TeamConnect [Object Views](#) provided by Mitrastech consist of system blocks, such as **General**, **Categories**, **Workflow**, **Assignees**, **Attendees**, and so on. You can also create your own Custom Blocks using [Custom Fields](#).

### **Business Objects**

Representations of the nature and behavior of real world things or concepts in terms that are meaningful to the business. Customers, products, orders, employees, trades, invoices, payments, appointments, accounts, shipping containers and vehicles are all examples of real-world concepts or things that could be represented by Business Objects.

There are [System Objects](#) and [Custom Objects](#) in TeamConnect.

## **C**

### **Cache**

A special high-speed storage mechanism. The TeamConnect application server has its own user interface caching mechanism to store the screen designs and other user interface data accessed by the application. You can clear the TeamConnect user interface Cache by either re-starting the application server or by clicking **Clear Cache** in the **User Interface Settings** screen of the **System Settings** on the **Admin** tab.

Browsers have their own memory caching mechanisms that can be cleared by closing the browser.

### **Calendar**

A special search results display type available in for search views only in the Appointment object definition. It displays scheduled appointments in a day planner layout.

### **Cascading Security**

The transferring of parent projects' record-level security, automatically in a waterfall fashion, to their related records. This is done on the **Security** tab of only custom object definitions.

### **Category**

Lookup items that allow you to organize custom fields into blocks or sections on the screen for the end-user interface. By adding or deleting categories in a record, the end users display or hide respectively the associated blocks of details in that record. Categories also help users to organize their records by certain types, for example, employee contact records vs. non-employee contact records, domestic vs. international accounts, and so on.

You must specify categories when defining the following:

- [Custom Fields](#)
- [Blocks](#)
- Access to object records at the category level

### **Category Rights**

See [Category Security](#).

### **Category Security**

The [Rights](#) to the object's [categories and their custom fields](#). This is the second level of security, after the object level. All rights assigned at this level take effect only if the access to the object itself is granted.

Category-level rights add more granularity to the object-level rights. A user might have access to the object, but you can still control access to the organization-specific information stored in the custom fields and displayed in custom blocks.

### **Check In/Out**

A way to manage files that are uploaded to the TeamConnect Documents area so that only one person at a time can edit them. If a file is checked out by a user, then no other users can edit the file. When the user is finished editing, he or she then checks in the file, making it available to other users.

### **Child**

Hierarchical relations where one (parent) item can have multiple (child) items, whereas each child item can have only one parent. Typically, parent-child relations can be established between [Custom Objects](#) in their object definitions. For example, one Claim can have multiple Features, one Feature can have multiple Sub-features, and so on.

The term "parent" can be also used with reference to the custom object that has [Embedded Objects](#) defined for it.

[Categories](#) can also have parent-child relations.

### **Condition**

A value that you select for a qualifier in a [Rule](#). For example, a condition could be an assignee role, a phase, a number of days from when the [Record](#) was opened or closed, and so on. See also [Attribute](#) and [Qualifier](#).

### **Contact**

A [System Object](#) definition that is intended to allow end [Users](#) to create record for individuals, organizations or separate entities within a organization (such as departments) that deal with their organization. Contacts can be organized into various [Contact Groups \(Address Books\)](#).

### **Contact Group (Address Book)**

A [System Object](#) definition that is intended to allow end [Users](#) to organize their [Contact](#) records together. Users can create groups of contacts with whom they frequently interact, so that their phone numbers are readily accessible when needed. The members of contact groups who are also [Users](#) can be added as attendees to appointments.

Contact Groups are called Address Books in the end-user interface.

### **Custom Action**

An action performed by a rule in TeamConnect that is defined in a JavaScript or Java class file. Custom Actions can do a wide range of functions - anything that is possible with Java or Javascript.

### **Custom Fields**

More custom fields that can be created for various objects. These fields allow you to meet the individual needs of your organization. Custom fields differ in each client installation of TeamConnect. When you create a custom field, you specify the label that appears in the user interface and what type of value the field holds (text, number, check box, and so on). You can create custom fields for type of object, system or custom.

Typically, you organize custom fields into [Custom Blocks](#) so that you can specify which groups of users can view them.

### **Custom Object**

[Business Objects](#) created for your organization. They do not come with the factory default TeamConnect, but are created by a TeamConnect Solution Developer using the object definition area of TeamConnect. Custom Objects can have names used within your industry for files, such as matters, claims, litigation, or policies.

The individual instances of Custom Objects are sometimes referred to as [Projects](#) within this documentation.

In TeamConnect 1.6.x, Custom Objects were referred to as "Applications."

For more details on custom objects, see [Creating Custom Pages](#).

### **Custom View**

See [Object View](#).

### **Custom vs. System**

In TeamConnect, various components are referred to as either system or custom. System means that the component is a default component of TeamConnect. Custom means that it is created for your organization. For example, objects, [Blocks](#), fields can be system or custom.

## D

### **DataMart**

An older TeamConnect feature providing a repository of data for reporting and analysis. This feature has been superseded by [Data Warehouse](#).

### **Data Warehouse**

An optional TeamConnect product. A repository of data gathered from operational data and other sources that is designed to serve a particular community of knowledge workers. In scope, the data derives from an enterprise-wide database, such as TeamConnect database, and meets your specific demands in terms of analysis, content, presentation, and ease-of-use. Typically, Data Warehouse is used for general reporting purposes.

If you are using Business Objects™ or other software to run reports and you want to include the information entered in a [Custom Field](#) for your reports, then you must make this selection when creating the custom field in an object definition.

### **Debug Settings**

A tool that allows you to select whether you want TeamConnect to [log various program operations](#).

### **Default**

A value or setting that TeamConnect is automatically selects if the user does not specify a substitute. For example, TeamConnect might select certain [Categories](#) as default categories for [Records](#), or set one of a contact's addresses as default.

Likewise, the default directory (for example in documents) is the directory the operating system searches unless you specify a different directory.

The default can also be an action that the program takes. For example, a user who creates an appointment is automatically (or by default) added as an attendee for this appointment.

### **Document**

A [System Object](#) definition that is intended to allow end [Users](#) to manage files uploaded to the Documents area of TeamConnect. These can be plain text, Microsoft Word, Excel, PDF, image (JPEG, GIF, BMP), HTML, XML, and so on. They are usually attached to TeamConnect [Records](#).

Documents are automatically available as a system [Block](#) in all [Object Views](#).

Many files necessary for customization of TeamConnect are also uploaded as Documents. These include XML files for [Custom Blocks](#), [Custom Actions](#) for Java [Rules](#), files included in [Templates](#), and other files related to your customized system design.

### **Document Type**

File type. For example, Microsoft Word document (DOC), text file (TXT), or JPEG image (JPG). When uploading files into the Documents area of TeamConnect, you specify the document type. TeamConnect then recognizes that file type and displays the appropriate icon.

As a solution developer, you can specify which file types TeamConnect is able to identify in the Documents area by adding them to the list of Document Types in the [Document](#) object definition.

### **Dynamic Value**

A value that is automatically generated based on a value that is entered or selected in another field, as opposed to a [Static Value](#). Dynamic values are often used in TeamConnect [Templates](#), [Wizards](#), Unique IDs and names for custom object [Records](#).

## E

### **Entity**

This term is sometimes used to refer to [Objects](#) in TeamConnect.

### **End User**

The final or ultimate [User](#) of the system. The end user is the individual who uses TeamConnect after it has been fully developed and designed, by way of the user interface. This term is typically used when the distinction from the users with administrative rights needs to be emphasized. It usually implies an individual with a relatively low level of computer expertise. Unless you are a programmer, engineer, or solution developer, you are likely an end user.

### **Embedded Objects**

Simplified Custom Objects created within other (parent/main) [Custom Objects](#) that can only exist within the context of the parent Custom Object. Within [Wizards](#) and [Templates](#), embedded objects are considered to be [Related Objects](#).

### **Expense**

A [System Object](#) definition that is intended to allow end [Users](#) to track internal costs of doing business in their organization. Expenses are automatically available as a system [Block](#) in [Object Views](#) for [Custom Objects](#).

## F

### **Friendly Name**

A label that you give to a TeamConnect component, such as a [Custom Block](#), in the form of a text name instead of a code. Friendly names allow you to identify the component among the other components with which it is listed in the screen, so that you do not have to remember a unique code or a key that is automatically assigned by TeamConnect.

### **Full-Text Search**

Allows you to search for items in the database according to their content. TeamConnect provides the feature of content searching in uploaded documents, and certain fields in [Records](#), if Microsoft® SQL Server Full-Text Search or Oracle® Text (interMedia) is available on your database server. For example, using content searching, you could search the content of uploaded documents to find a file in which a certain person's name is mentioned. This feature is available in search screens if the appropriate fields are included in the corresponding [Search View](#).

### **Functional Level Security**

Discontinued term. Replaced by the concept of object-level [Rights](#).

## G

### **Group**

Refers to either [Contact Groups \(Address Books\)](#) or [User](#) Groups.

User groups are certain groups of users who have the same or similar responsibilities within the organization. Each User Group has its own set of access rights assigned to it. Groups can also have different [Object Views](#).

### **Group Rights**

The [Rights](#) to all features that can be accessed in the end user interface. These primarily include the rights to all [System Object](#) and [Custom Object](#) records, and certain [Tools](#), such as Scheduler, Batch Task and Expense Entry. Rights are assigned by user group.

## **H**

### **History**

A [System Object](#) definition that is intended to allow end [Users](#) to make chronological entries always associated with a specific [Record](#). History is a [Related Object](#) only; that is, History records can only exist when related to other records. History is automatically available as a system [Block](#) in [Object Views](#).

### **Home Page**

The first screen that users see after logging in to TeamConnect. Home Pages function as a starting point for all TeamConnect users and can be personalized to fit their individual needs. Users can access various types of information or do different actions from [Portal Panes](#) that are available on the Home Page.

## **I**

### **Invoice**

A [System Object](#) definition that is intended to allow end [Users](#) to track invoices or bills sent by [Vendors](#) who provide their organization with goods or services. Invoices are automatically available as a system [Block](#) in [Object Views](#) for [Custom Objects](#).

### **Involved**

A [System Object](#) definition that is always related to a [Custom Object](#). It is intended to allow end [Users](#) to keep track of various involved parties associated with a [Project](#). Involved is automatically available as a system [Block](#) in [Object Views](#) for [Custom Objects](#).

## **L**

### **Line Item**

A [System Object](#) definition that is intended to allow end [Users](#) to list the goods or services specified in a vendor [Invoice](#). The details of each line item, such as the type (fee or expense), price, quantity, and dates must be entered in the Line Items screen of the corresponding invoice. Line Items are automatically available as a system [Block](#) in [Object Views](#) for [Custom Objects](#).

### **Logger**

A component for a specific area of TeamConnect for which a system or audit log can be defined. For example, TeamConnect has loggers for portal panes, custom blocks, the XML layer, and user login/sign off. When you define a log (or [appender](#)), you select which logger's messages it captures.

### **Lookup Table**

Tables that organize, store and quickly access multiple items, such as contacts' roles, types of addresses, phone, fax and other contact information, and so on. In addition to adding, renaming, inactivating, and deleting items in system lookup tables, you can define new custom lookup tables and their contents. In the end-user interface, lookup tables are represented by either drop-down lists (most commonly) or radio buttons, where the user can look up the necessary information and make the appropriate selection.

## **M**

### **Member**

This term can refer to the following:

- [Group](#) members are [Users](#) who are included in a group, from which they can obtain rights to various objects and tools in TeamConnect.
- [Route](#) members are users who are included in the Approval process in an [Approval Rule](#).
- [Contact Group \(Address Book\)](#) members are contacts that are included in a user's personal Contact Groups.

### **Memo Text**

A type of text field in TeamConnect in which you can type free-form text of a length up to what your database permits.

## **N**

### **Node**

Different levels in the [Lookup Table](#) tree structures that have their own sub-levels.

### **Null Value**

An empty or blank value in a field. For example, a field has a null value if the user has not entered anything into the field. It is important to understand this term when writing rules because the Rules screen sometimes indicates that there is a null value selected in a qualifier by displaying **(null)**. The `allowNullValue` tag can also be used for custom fields of type Drop-down List.

## **O**

### **Object**

Within this documentation, it is the same as [Business Object](#).

This term also overlaps the object-oriented programming terminology, where an object is defined as a self-contained entity that consists of both data and procedures to manipulate the data.

### **Object Attribute**

See [Attribute](#).

### **Object Definition**

A specification of all properties of a [System Object](#) or [Custom Object](#). After fully defined, it can contain all details regarding the object, from its name, icon, categories, and custom fields, to the way its record and search screens look, wizards that can be used for creating its [Records](#), naming and numbering patterns for (custom object) records, and rules that are triggered when users are working with the object records.

### **Object-Level Security**

See [System Record Rights](#) and [Custom Record Rights](#).

### **Object Model**

The architecture in which data exists in TeamConnect. In the user interface, it is represented in the form of tables that hold information about objects and [Records](#) in the form of [Attributes](#). [Object Navigator](#) allows you to navigate through these tables to select attributes when you are creating [Rules](#), [Wizards](#) or [Templates](#), or possibly when you are defining the naming convention of an object. The object model is fully documented in [Object Model: Read This First](#) and the additional reference tables it points the user to.

### **Object Navigator**

A tool that allows you to traverse through the tables of the [Object Model](#) to select [Attributes](#). This is necessary when creating rules, templates, wizards, and possibly when defining the naming convention of an object. Using Object Navigator, you create a [Path](#) that identifies an [Attribute](#). For details, see [Using Object Navigator](#).

### **Object Table**

A table in the [Object Model](#) that contains a list of [Attributes](#) belonging to a specific object.

### **Object View**

A specific layout of tabs, sections, blocks, and other user interface elements on the screen that can be assigned to users or groups, which you can customize by adding/removing system and custom [Blocks](#) in the tabs as desired. In each object definition, several Object Views can be created so that different groups of users see the information that is relevant to them.

### **Operator**

A specific action that allows you to manipulate and assign values to fields in templates and wizards. For example, **set to value of path** sets the value of a field to the value specified by the [Path](#).

## **P**

### **Parameter**

A wizard page component that allows you to add certain data items that you might need to use as qualifiers in rules when creating conditions for page transitions. You can also use parameter values to do certain [Actions](#) (or operations) on the object attributes. These data items are not stored in the TeamConnect database and can be added and used only within [Wizards](#).

### **Parent**

See [Child](#).

### **Path**



A series of [Attributes](#) that is created using [Object Navigator](#). A path is used to identify how to get to a specific attribute in a table of the [Object Model](#). Paths can be used in [Rules](#), [Templates](#), [Wizards](#), and in the naming patterns for object records.

### **Phases**

Certain states of a [Project](#). A specific sequence of phases and their transitions define the life cycle of the project. For example, a Litigation record might go through the following phases Suit Filed > Discovery > Depositions > Settlement > Arbitration > Trial > Closed. All Phases and their transitions are defined in the [Custom Object](#) definitions.

### **Portal Pane**

A component of a [Home Page](#), which represents a specialized menu that allows the end users to access different information or do specific tasks directly from the Home Page. includes one or more items of content.

### **Preferences**

System options that you can use to customize your TeamConnect to fit your individual needs and personal preferences. They are listed when you click the **Preferences** link in the user interface.

### **Primary Key**

An automatically generated number that uniquely identifies the newly created item in the database.

Primary keys are rarely displayed in the record screens. However, they can be used when building paths with Object Navigator, for example in rule qualifiers, in custom object record names and unique identifiers, or when working with the XML layer.

### **Process Manager**

A TeamConnect user who is responsible for monitoring [approval](#) processes that are sent to approvers by requestors. Process Managers receive notifications of approval requests when there is an error. They may also be notified when a request is rejected or expired. They can restart an approval process to resolve errors, rejections, or expirations, or reassign the task of approving a request to other users when necessary. They can also reject approval requests.

### **Project**

Is a generic term used to refer to an individual [Record](#) or instance of a [Custom Object](#).

## **Q**

### **Qualifier**

[Rule](#) qualifiers are criteria used to determine whether a user's attempted operation, such as deleting a [Record](#), should be permitted. Qualifiers can include who the user is, the phase of the record, the number of days from when the record was closed, and countless other possibilities.

[Search View](#) qualifiers are [Attributes](#) used as criteria that a user can use to search for records, such as the name of the record, the record number, the phase of the record, the record's assignees, and countless other possibilities.

### **Quick Search**

Search performed directly from a Search Module field by typing search criteria in the field and then clicking the **Find** icon. The formatting requirements for typing your criteria vary depending on whether the Search Module field is for a [Project](#), [Contact](#), [User](#), or [Account](#).

## R

### ***Read-only Access***

Record-level security state in which the record cannot be modified as a result of the rights assigned to it or for workflow reasons, when the record is pending approval, and the approval rule prevents anyone from modifying the record. This relates to Record Security and not to the read-only display state.

### ***Record***

An individual instance of a [System Object](#) or [Custom Object](#) in TeamConnect, created and accessed in the user interface. Note that Custom Object records are sometimes generically referred to as [Projects](#).

### ***Record Security***

Access protection set at the level of individual object records. For example, you can make a [Record](#) Private or Public or you can grant or deny the rights to read, update, and delete a record to other [Users](#) or [Groups](#). This level of security can be set only on the **Security** tabs of [individual records](#) in the end-user interface. This is the lowest level of access rights. All rights assigned at this level take effect only if the appropriate rights are assigned at [the object level](#).

### ***Rejection***

Denial of an [Approval](#) request to do a user-attempted operation by members of a [Route](#). If the operation is rejected, this is noted on the **Workflow** tab of the [Record](#).

### ***Related Objects***

Objects that are in one way or another are associated with other objects in TeamConnect, either using hierarchical relations (child-parent, [Involved](#), [Embedded Objects](#)), or non-hierarchical relations established in [Custom Fields](#) of type Custom Object.

### ***Request***

An attempted operation by a user that is not completed until it has been approved by the approvers in a specified approval route. A request can be tracked by the requestor, the approvers who receive the request, and possibly a Process Manager, if one is specified for that approval process.

### ***Requestor***

A user who performs an action that triggers an [approval](#) process, such as posting an [invoice](#), voiding an invoice, or changing the [phase](#) of a [project](#) (such as a matter). The [request](#) is sent to [approvers](#) according to the approval route that is defined for that workflow process. The requestor's request is approved or rejected by the approvers. Requestors can view the progress of their requests on the **My Requests** screen. They can also cancel a request if necessary.

### ***Required Fields***

Fields that must have a value entered in them because a [Record](#) cannot be saved. Depending on the end-user TeamConnect configuration, required field labels might be identified in the user interface in red (or other color different from the default text color), or other special formatting applied to the label text, such as **bold** or *italics*.

You can make [Custom Fields](#) required by specifying them as such when you create them. You can also make fields required only under certain conditions, with the use of [Rules](#).

### **Resources**

Resources are means and/or facilities necessary for conducting an event, for which an [Appointment](#) has been scheduled. For example, you might need to book a particular conference room or a projector.

### **Reviewer**

A user who is asked by an [approver](#) to review an [approval](#) request. Reviewers are not approvers, so they do not approve or reject the request. They can provide feedback to the approver who makes a decision in the process. Reviewers can view their requests for review on the **My Approvals** screen.

### **Rights**

Privileges or permissions to use information that taken together constitute [security at different levels](#), object, category, and individual [Record](#). In TeamConnect, access rights ensure that no record information can be read and/or compromised by unauthorized individuals. Typically, rights are assigned by the TeamConnect system [Administrator](#) to individual [Users](#) and/or [Groups](#).

### **Root**

The top [Category](#) in the categories hierarchy tree. It is always automatically added to every record in TeamConnect. In [Projects](#), this category has the same name as the object itself.

### **Route**

A defined hierarchy of users whose [Approval](#) is required to authorize a requested operation. It is an approval path that can consist of one or more nodes, called [Stops](#), that can contain one or more Members who can approve or reject the operation. Routes are used for the [Workflow](#) purposes, to control the actions that take place in your business processes.

### **Rules**

Business [Workflow](#) requirements that allow you to depend on TeamConnect to ensure that the proper processes, which are specific to your industry and organization, are maintained by users.

TeamConnect allows you to create rules that control whether the operations attempted by users are in accordance with your business processes. A rule is triggered when the operation is attempted, and the [Qualifiers](#) are checked to determine what should occur. For example, rules can ensure that certain [Custom Fields](#) are filled out if a particular [Category](#) is selected.

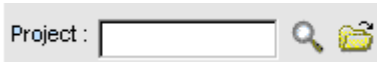
[Approval rules](#) postpone the operation until all members in the approval process, or [Route](#), have approved the operation.

Rules written using Java class files or JavaScript can also perform [Automated Actions](#) when they are triggered.


## **S**

### **Search Module**

A special utility that allows the user to quickly search for a specific record and link it to the record with which you are currently working without leaving its screen. The [Search Module](#) is represented by a field with two icons, **Find** and **Open**, next to it.



### **Search Screen**

A type of screen that appears whenever you are looking for a specific record. Search screens consist of two tabs: **Filter** and **Results**. In the user interface search screens can be displayed as separate screens, tabs, or sections of a tab. You can access these screens if you select a provided search option or if you click the **Find**  icon in record screens. You can define what types of search screens are available to users by creating [Search Views](#).

### **Search Views**

Specific sets of search [Qualifiers](#) and search results display settings that you can define in each object definition to allow end users to quickly find the necessary object [Records](#) and/or create quick reports. Typically, there are two default Search Views each object definition: **Default** and **Advanced**. However, you can create customized search views (such as **My Tasks**, **Recent Invoices**, and so on).

### **Security**

See [Rights](#).

### **Security Rule**

A type of TeamConnect rule that you can create to check the specified [Qualifiers](#) before allowing or denying the user's attempted operation. A security rule is based on the conditions in the [Record](#) when it was first opened by the user, so that the user cannot change the values of fields if he or she should not have permission. For more details on security rules, see [Security Rules](#). See also [Validation Rule](#).

### **Smart Search**

A search utility located in the user interface. Smart Search allows users to search for specific record types or across **All Records** or **All Projects**. Users can enter a few consecutive characters from any record's name or number, select the type of record, and the system will figure out what to search by, name or number, and whether the characters come from the beginning, middle, or end.

### **SQL**

Structured Query Language. Used for performing operations in a database, such as searching, adding values, and altering values.

### **Static Value**

A value that the user manually enters or selects. Unlike [Dynamic Values](#), static values are not based on the values in other fields. It is important to understand the difference when working with TeamConnect [Templates](#), [Wizards](#), Unique IDs and names for custom object [Records](#).

### **Stop**

An [Approval](#) node, or check-point in a [Route](#) within the TeamConnect [Workflow](#). Each stop consists of a certain number of [Approvers](#), whose authorization is required for a pending action. Each approval route might have several stops.

### **Sub-object**

In end-user interface, it is an item that is typically added to an object [Record](#) using a [Batch Screen](#). Different objects have different sub-objects. For example, the following items are all sub-objects:

categories, task assignees, project assignees, project relations, contact relations, and the addresses, emails, skills, phone numbers and other items added to contact records.

In the TeamConnect object model, sub-objects are always represented by J-tables, such as JProjAssignee, JContRate, JApptAttendee, and so on. The only exception is categories that are always represented by the WObjdCategory object table.

### **Super User**

A user who can access private and public [Records](#) regardless of the settings on the **Security** tab of these records, provided this user has the object-level [Rights](#) to these records.

### **System vs. Custom**

In TeamConnect, various components are referred to as either system or custom. System means that the component is a predefined component that is installed with TeamConnect. Custom means that it is created for your organization. For example, objects, [Blocks](#), and fields can be system or custom.

Each object definition in TeamConnect has its own system fields, organized into system blocks that are added to its system views. All system views are automatically set as default system-wide object views.

### **System Object**

Default [Business Objects](#) provided by Mitrtech to your organization. They represent business objects that are common to most business models regardless of the specific industry.

There are 13 system objects within TeamConnect: [Account](#), [Contact](#), [Contact Groups \(Address Books\)](#), [Appointment](#), [Expense](#), [Invoice](#), [Task](#), [History](#), [Document](#), [Involved](#), [User](#) account, and [Group](#) account.

### **System Settings**

Settings, such as password policy, user interface settings, and so on that affect the whole system. Typically available to TeamConnect [Administrators](#).

## **T**

### **Tab**

A user interface element that appears at the top of all user and administrative record screens. When clicked each tab opens a new set of information from the record.

In the end-user interface, all tabs in object [Records](#) are defined using the [Search Views](#) in the object definitions. When you define tabs, you must add at least one [Block](#) in order for the tab to be displayed on the screen.

### **Table**

See [Lookup Table](#), [Object Table](#).

### **Tag**

An instruction that is included in an HTML or XML file. The instruction is enclosed between the greater than (>) and less than symbols (<), also known as angled brackets. You use tags when creating [Blocks](#), or [Document Templates](#).

### **Tag Attribute**

A value that modifies an HTML or XML tag. For example, if you want to specify that a number field is formatted to display as a dollar amount in a form, you would use the attribute `format="DOLLAR"`.

For a list of available tag attributes for TeamConnect [Blocks](#), see [Using Tag Attributes](#) section.

### **Task**

A [System Object](#) definition that is intended to allow end [Users](#) to track internal assignments that they have to do, usually while working on [Projects](#), for example preparing reports, following up on cases, reviewing documentation, and so on. Tasks are automatically available as a system [Block](#) in [Object Views](#) for [Custom Objects](#).

### **Template**

This term can refer to the following:

- Document templates
- Templates used with [Wizards](#)
- Templates used with rules
- Templates used with wizards and rules

A document template is a template designed for use with the Document Generator. You can use this capability to generate documents from [Records](#) using certain fields, such as the name of the contact associated with the record.

Templates used with wizards or rules define default values that should be automatically filled in for certain fields, [Sub-objects](#), and [Related Objects](#). For more details on templates, see [Creating Custom Pages](#).

### **Tool**

A utility that helps to automate an operation to decrease the time and effort that would be expended on a task, such as reassigning work in bulk to various [Users](#). There are system tools that come by default with TeamConnect; however, it is also possible to create custom tools. For details about creating custom tools, see [Creating Custom Tools](#).

### **Transaction**

Events or happenings in your business that change its financial position, such as buying supplies, paying bills, withdrawing money, or buying equipment. For example [Expenses](#), [Tasks](#), and [Invoices](#) and regular money withdrawals from and transfers between [Accounts](#) incur transactions against accounts.

### **Traversing**

The act of jumping from one table to another in [Object Navigator](#). This is accomplished by selecting one of the [Attributes](#) that appears with an arrow next to it. Each time you traverse to another table, you add the selected attribute to the [Path](#) which you are creating to identify a field in TeamConnect.

### **Tree Position**

The last branch in a lookup table hierarchy tree, represented by 4-character alphanumeric combination assigned to an item listed in a [Lookup Table](#), a [Category](#) or an [Assignee](#) Role. Tree positions uniquely identify these items in TeamConnect database. They can include letters and numbers, but letters used in a tree position must be capitalized.

The full tree position path of each child lookup item includes the tree positions of all of its parents.

## U

### **UBB code**

Ultimate bulletin board code--a variation of HTML formatting tags that allows you to add functionality and styles to your messages.

### **Unique Code**

A unique code that identifies a system or custom [Object](#) in the TeamConnect database. It consists of a 4-character, alphanumeric combination, and its letters must be capitalized. System objects have predefined unique codes, whereas custom objects must have unique codes defined when first created.

[Phases](#) have 4-character, alphanumeric combination assigned to them too, which are also known as unique codes.

### **Unique ID**

Identifies an individual custom object record. You must set unique IDs in the object definition of each custom object. For example, you can set the records to be automatically numbered by the system, or to be given a unique id based on a pattern of [Attributes](#) that are selected in the record.

Do not confuse the Unique ID for Custom Object records with the [Unique Code](#) of the Custom Object. Also, do not confuse it with naming patterns for Custom Object records.

### **User**

An account created for a specific person who uses TeamConnect. Users can be members of [Groups](#), and can have [Rights](#) assigned to them or from a group in which they are a member. Each user has their own password to log in to TeamConnect, which is first specified in their user account (accessible by the [Administrator](#) only) together with the username and rights.

### **User Interface**

The graphical representation of TeamConnect on the computer screen, where the user can click buttons and hyperlinks, make selections from drop-down lists, enter text into fields, and so on. It is also referred to as the graphic user interface or GUI. The end-user interface in TeamConnect includes all available TeamConnect features, because end users do not typically have access to items in the **Admin** tab.

### **User-Invoked Actions**

See [Custom Actions](#).

## V

### **Value Set**

A group of attributes in a [Sub-object](#) which end users must always populate to make a valid sub-object entry. When creating [Templates](#), you must make sure that you that all attributes from the sub-object's value set are defined. For more details on defining sub-objects in templates, see the [Creating Custom Pages](#).

### **Validation Rule**

A type of [Rule](#) in TeamConnect that you can create to check the specified [Qualifiers](#) before allowing or denying a user's attempted operation, such as creating or updating a [Record](#). For more details on validation rules, see the [Validation Rules](#). See also [Security Rule](#).

### **Vendor**

Outside [Contacts](#) with whom your organization conducts business. These can be suppliers of goods or services, outside counsel, and so on.

### **Version Control**

A method of tracking the edits made to a document that is uploaded to TeamConnect. Version control allows users to access previous versions of a document, and even revert the current version back to a previous version if needed. Version control must be activated by the TeamConnect [Administrator](#).

## **W**

### **Wizard**

A utility that you can create allows users to create object [Records](#). A wizard guides the user through pages of questions and instructions, and, if you have a [Template](#) associated with the wizard, automatically enters values into the correct fields. A wizard can even do actions such as automatically creating related records, performing calculations, and other automated actions.

### **Workflow**

Workflow is a sequence of activities within an organization to produce a final outcome.

For example, in a financial setting, a request to delete an account might be automatically routed from a financial officer to accounting manager to accounting director for approval and back to the officer for processing.

At each stage in the workflow, one [User](#) or [Group](#) is responsible for a specific operation. After the operation is complete, TeamConnect ensures that the users responsible for the next operation are notified and receive the data they need to execute their stage of the process.

## **3.2 Legal Customization Help**

Welcome to the *TeamConnect® Legal Matter Management Customization Help*.

<a href="#">File Naming Conventions</a>	<a href="#">Time Entry Tool</a>
<a href="#">CSM Configuration for Legal</a>	<a href="#">SOP Configuration for Legal</a>

### **3.2.1 Legal Customization Guidelines**

**Important:** *Mitratech recommends that the shipped design files remain untouched.*



When customizing TeamConnect® Legal Matter Management, you should follow these general guidelines:

- If you modify any files that are delivered with TeamConnect® Legal Matter Management, you must not use the same file names. Instead, copy and rename the files. This will help distinguish between problems with the modifications and problems in the delivered product.
- A Java developer should make any necessary modifications to copies of TeamConnect® Legal Matter Management's Java class files.
- Renaming TeamConnect® Legal Matter Management components usually will not cause problems, but you should not change their tree positions because many rules and wizards depend on them.
- Hiding or deactivating a component is preferable to deleting it. For example, rather than delete unneeded blocks, remove them from the object view.
- Before deleting a component, determine whether any class files, XML files, rules, or wizards depend on it so that you can make corresponding modifications to the dependent components. You can search the source code for the unique key of the item to determine whether any class file rules or wizards depend on the item.

You must have access to the Java source code to be able to search for such text strings.

## 3.2.2 File Naming Conventions

This section explains the file naming conventions for the Java class, XML, and JavaScript files that are shipped with TeamConnect® Legal Matter Management. These conventions provide clues as to the function of the files. This section explains the file naming conventions so that you can identify files that you may want to copy, rename, and then modify to suit your business needs.

The files delivered with TeamConnect® Legal Matter Management all use the **SYS** suffix. In order to distinguish the files that were shipped with the product from files that have been customized, solution developers must not simply modify the delivered files, but should at least change the **SYS** suffix to another string—for example, **IMP**.

**Caution:** *If you modify any files that are delivered with TeamConnect® Legal Matter Management, you must not use same file names. Instead, copy and rename the files.*

Solution developers can create their own naming patterns. However, Mitratesh recommends that they follow the general file naming conventions so that the function of the corresponding Java class, XML, and JavaScript files will be implied by their file names.

### 3.2.2.1 Filename Parts

TeamConnect® Legal Matter Management file names can include some, but not necessarily all, of the following components in the following order:

**PrefixComponentMatter\_ObjectDescriptionTrigger/QualifierSuffix.FileExtension**

The parts of the filename are delimited with capital letters. For example, the file **CommonWzMatterPopulateCategoriesQSYS.class** has the following parts:

- **Common**—The Common prefix indicates that this file contains a super class that is inherited by more than one object.
- **Wz**—The **Wz** component indicates that this file is used by a wizard.
- **MatterPopulateCategories**—This descriptive string indicates that the file is used to populate categories for matter records.
- **Q**—The **Q** indicates that this file acts as a qualifier.
- **SYS**—The **SYS** indicates that the file was delivered with TeamConnect® Legal Matter Management and was not developed during or after implementation.
- **.class**—Indicates a Java file.

## Prefixes

The following filename prefixes are used with the **SYS** files:

- **Common**—Indicates that the file contains a super class that is inherited by more than one object.
- **TCL**—Indicates that the file contains utilities used by various class files.

For example, **TCLActionHandlerSYS.class** contains an action handler utility used by other class files.

## Components

The following filename components are used with the **SYS** files:

- **Cjb**—Indicates that the file contains a custom Java block (CJB).
- **Rule**—Indicates that the file is used by a rule. This also indicates that the file is not associated with a wizard.
- **Wz**—Indicates that the file is used by a wizard.
- **WzCjb**—Indicates that the file is a CJB for a wizard page.

For example, **RuleTCreateBudgetCASYS.class** is used by a rule for creating budgets upon the creation of a Transaction record, but the file is not directly associated with a wizard.

## Matter Object

The following letters indicate the matter object to which the **SYS** files apply:

- **AC**—Advice & Counsel
- **D**—Dispute
- **T**—Transaction

For example, **RuleTUpdateBudgetUASYS.class** contains a rule for the Transaction object that updates a budget upon the Update action.

## Description

The description is usually an intuitive string that indicates the file's functionality. For example, the string may be as simple as **Budget** or include a description of a qualifier, such as **IsNoFirmsSelectedAndFirmsListEmpty**. If the string is prefixed by **Matter**, the qualifier or action affects more than one matter object.

The following acronyms appear in the descriptive string of some filenames:

- **AC**—Advice & Counsel
- **IP**—Intellectual Property
- **OC**—Outside Counsel
- **OP**—Opposing Party

For example, **WzTValidateFieldsQSYS.class** contains qualifiers used by the Transaction object's New Transaction wizard to validate fields.

### Trigger/Qualifier

The following acronyms indicate triggers or qualifiers in the **SYS** files:

- **A**—Action (most likely for a wizard action)
- **CA**—Performs an action on Create
- **CUA**—Performs an action on Create or Update
- **DA**—Performs an action on Delete
- **Q**—Qualifier
- **AQ**—A qualifier for an Approval rule
- **UA**—Performs an action on Update

For example, **CommonWzOutsideCounselBudgetASYS.class** contains a common wizard action for the creation of outside counsel budgets.

**Note:** A file that does not have a prefix in its filename or a trigger/qualifier before the **SYS** suffix is probably a user interface design XML file that does not use Java. For example, the **LawsuitDetailsSYS.xml** file is not prefixed by **Common** or **TCL** and does not have an **A** or **Q** before **SYS**.

#### 3.2.2.2 Utility Class Files

The following utility class files perform the corresponding function:

- **ConstantsSYS.class**—Holds the system constants in TeamConnect® Legal Matter Management. Stores tree positions, custom field names, wizard parameter names, etc.
- **TCLAccountUtilitySYS.class**—Provides alternatives to the standard TeamConnect Enterprise account-processing methods. Holds transaction, deposit, and withdrawal methods that provide alternatives to the standard TeamConnect Enterprise account-processing API.

**Caution:** To avoid account-processing errors, you should carefully plan any modifications to the functionality contained in the **TCLAccountUtilitySYS.class** file.

- **TCLBudgetUtilitySYS.class**—Holds the budget constants and utilities in TeamConnect® Legal Matter Management.
- **TCLFacadeSYS.class**—Stores many utility methods used by TeamConnect® Legal Matter Management.
- **TCLParameterTokenizerSYS.class**—Holds lists of parameters that are created during execution. This class provides methods to generate and parse one long string that holds all the parameter data. This class file uses a corresponding JavaScript file called **TCLParameterTokenizerSYS.js**.
- **TCLSecurityUtilitySYS.class**—Stores security- and access-management-related utility methods.

### 3.2.2.3 Base Classes

The following base class files provide enhanced functionality in CJBs:

- **TCLActionHandlerSYS.class**—An abstract super class that stores essential utilities shared by all action classes.
- **TCLConditionHandlerSYS.class**—An abstract super class that is used in qualifiers and stores essential utilities shared by all condition classes.
- **TCLCustomBlockSYS.class**—An abstract super class used by all custom block classes and stores essential utilities shared by all CJBs.

### 3.2.2.4 Batch Display Files for Custom Java Blocks

Most of the CJBs in TeamConnect® Legal Matter Management 3.4 use one XML file and one class file. However, CJBs with batch displays require two additional files to define the new and regular rows of the batch display. For example, the **Cost Center** block for the Transaction object uses the following four files:

- **CjbTCostCenterSYS.class**—Java class file
- **CjbTCostCenterSYS.xml**—Primary XML CJB design file
- **CjbTCostCenter\_NewRowSYS.xml**—XML design file that represents the New Row
- **CjbTCostCenter\_RegularRowSYS.xml**—XML design file that represents the Regular Row

The same convention is used for the **Cost Center** blocks for the Dispute object:

- **CjbDCostCenterSYS.class**—Java class file
- **CjbDCostCenterSYS.xml**—Primary XML CJB design file
- **CjbDCostCenter\_NewRowSYS.xml**—XML design file that represents the New Row
- **CjbDCostCenter\_RegularRowSYS.xml**—XML design file that represents the Regular Row

The batch display convention is also used for the **Environmental Corrective Action** block, which is displayed on the **General** tab of Dispute records that have the Environmental category:

- **CjbDCorrectiveActionSYS.class**—Java class file
- **CjbDCorrectiveActionSYS.xml**—Primary XML CJB design file
- **CjbDCorrectiveAction\_NewRowSYS.xml**—XML design file that represents the New Row
- **CjbDCorrectiveAction\_RegularRowSYS.xml**—XML design file that represents the Regular Row

The batch display convention is also used for the **Real Estate Lessee** block of the Transaction object, which is displayed on the **General** tab if the Real Estate category is selected. The **Real Estate Lessee** block uses the following files:

- **CjbTLesseeSYS.class**—Java class file
- **CjbTLesseeSYS.xml**—Primary XML CJB design file
- **CjbTLessee\_NewRowSYS.xml**—XML design file that represents the New Row
- **CjbTLessee\_RegularRowSYS.xml**—XML design file that represents the Regular Row

### 3.2.3 Object Definitions

This section provides TeamConnect® Legal Matter Management-specific guidelines regarding the following activities:

- Changing the name of an object definition
- Modifying object definition attributes
- Configuring object definition naming patterns

Performing these procedures is covered by the TeamConnect Enterprise Solution Developer documentation.

#### Changing the Name of an Object Definition

You can change the name of a custom object definition without causing any problems because no rules or wizards depend on object definition names. However, you must not change the tree position of a custom object because many rules and wizards depend on the tree position.

#### Modifying Object Definition Attributes

You can change the attributes of a custom object definition without causing any problems because no rules or wizards depend on object definition attributes.

#### Configuring Object Definition Naming Patterns

You can change the naming pattern of a custom object definition without causing any problems because no rules or wizards depend on object definition naming patterns.

### 3.2.4 Embedded Objects

An embedded object is a simplified custom object created within another custom object. You cannot define phases, assignees, or rules in embedded objects, so they have few dependencies.

This section provides TeamConnect® Legal Matter Management-specific guidelines regarding adding, deleting, and modifying embedded objects. Performing these procedures is covered by the TeamConnect Enterprise Solution Developer documentation.

#### Adding Embedded Objects

Adding embedded objects should not cause any unanticipated problems, but you will have to determine how to integrate them into the business solution and modify the object view of the parent object.

#### Deleting Embedded Objects

If custom blocks depend on the embedded object, deleting the embedded object can cause problems unless you also delete the block. To determine which files use an embedded object, look at the **Blocks** tab of the parent object and analyze the filenames to determine whether any files correspond to the embedded object. If there are any corresponding files, you can delete them accordingly.

#### Modifying Embedded Objects

You can edit the name of an embedded object without causing problems because CJBs reference the embedded object's unique code, not the object's name.

If custom blocks depend on the embedded object, modifying the embedded object can cause problems. Look at the **Blocks** tab of the parent object and analyze the filenames to determine whether any files correspond to the embedded object. If there are any corresponding files, you can modify them accordingly.

Adding, deleting, and modifying embedded objects' categories and custom fields is addressed in [Categories](#) and [Custom Fields](#).

#### Special Embedded Objects

The following embedded objects in TeamConnect® Legal Matter Management use CJBs to display information and will break custom blocks when the object is deleted or their custom fields are modified:

- Corrective Action (of Dispute)
- Lessee (of Transaction)
- Dispute Cost Center
- Transaction Cost Center
- Related Advice & Counsel Matter (of Dispute)
- Related Advice Matter (of Transaction)

### 3.2.5 Categories

All categories can be added, modified, and deleted through the **Categories** tab of the respective object definition. Many categories are used by other components, so you should proceed with caution when modifying them.

All custom screens are dependent on categories. For more information, see [Custom Screens](#).

This section provides TeamConnect® Legal Matter Management-specific guidelines regarding adding, deleting, renaming, and reordering categories. Performing these procedures is covered by the TeamConnect Enterprise Solution Developer documentation. Additionally, this section provides tips for searching for category dependencies and highlights categories requiring special attention.

#### Adding Categories

Adding new object categories has no impact on the application and requires nothing more than the work necessary to add new custom fields, object blocks, search views, and rules that may be associated with the new categories.

#### Deleting Categories

Deleting a category can seriously impact the application if fields, blocks, rules, or wizards depend on it. Dependent fields are essentially deleted. Dependent blocks might display errors unless they are also removed. Dependent rules will not work properly and may cause java error messages to be displayed.

If a category has no dependent components, its deletion will not cause any issues. However, it is best to proceed with extreme caution by determining whether the category is used by other components and make any necessary modifications.

#### Renaming Categories

You can edit category names without causing problems because rules and CJBs reference the category's tree position, not the category's name. You may have to modify the UI to be consistent with the category name change.

#### Reordering Categories

Reordering the display order of categories does not cause any issues because the order of the categories is not referenced by XML files, CJBs, rules, or wizards.

**Note:** In TeamConnect® Legal Matter Management 3.4, the default order of all categories is zero.

#### Searching for Category Dependencies

To determine dependencies on categories, you can search the source code for a variety of text strings. For example, you can search according to the category's tree position to find all the class files that might use that category. Alternatively, you can search for all categories using the following strings:

- **category='Tree\_Position'**—This is useful for finding XML blocks that use specific categories. For example, if you search the source code for **category='DISP\_EMPL'**, the search results will

point to XML files such as **WzEmploymentDetailsSYS.xml** (the **Employment Details** block in the **New Dispute** wizard's **Details** page). Thus, you would have a list of all the blocks dependent on the Dispute object's Employment category.

- **CAT\_**—This string is useful for determining which class files use categories. You can use it to search for all categories used by class files or append the tree position of a specific category. For example, if you search the source code for **CAT\_DISP\_EMPL**, the search results include **WzDDetailsASYS.java**. Thus, you would know that the Dispute object's Employment category is used by page actions in the **New Dispute** wizard's **Details** page.

You must have access to the source code (.java files, NOT .class files) to be able to search for such text strings.

## Special Categories

The tree positions of the following categories are used by many rules and wizards, so deleting them would result in a major impact:

- General Liability/Claim (for Dispute)
- Lawsuit (for Dispute)
- Subpoena (for Dispute)
- Outside Counsel Attorney (for Dispute and Transaction Involved Party)
- Outside Counsel Attorney : Primary (for Dispute and Transaction Involved Party)
- Outside Counsel Firm (for Dispute and Transaction Involved Party)
- Outside Counsel Firm : Primary (for Dispute and Transaction Involved Party)

### 3.2.6 Phases and Phase Transitions

TeamConnect® Legal Matter Management 3.4 matters come with the following phases that follow a basic process flow and should be acceptable to most customers:

- Intake
- Open
- Closed
- Reopen

In addition, the Advice & Counsel object has the Research Pending category.

A few rules and wizards reference phases and phase transitions; all of them are visible in the UI because the rule names contain the phase and phase transitions.

This section provides Matter Management-specific guidelines regarding adding, deleting, and modifying of phases and phase transitions. Performing these procedures is covered by the other chapters in this documentation.

## Adding Phases and Phase Transitions



Adding phases should not cause any unanticipated problems; you will have to determine how to integrate them into the business solution.

***Important:** Adding phases to a live system may negatively impact phase histories.*

## Deleting Phases and Phase Transitions

If a rule depends on a phase or phase transition, deleting the phase would cause the rule to break and errors to be displayed.

***Important:** Deleting phases on a live system may negatively impact phase histories.*

The only dependency on phases in TeamConnect® Legal Matter Management is the rule that ensures that a contact involved in an open matter cannot be deleted. The Dispute and Transaction objects use the **RuleContactInvolvedInOpenMatterQSYS** class file that depends on the **Open** and **Reopen** phases.

## Modifying Phases and Phase Transitions

Editing phases should not cause any unanticipated problems, although you would have to determine how to integrate them into the business solution.

***Important:** Editing phases on a live system may negatively impact phase histories.*

### 3.2.7 Assignee Roles

Assignee roles are defined on the **Assignee Roles** tab of the Advice & Counsel, Dispute, and Transaction object definitions. Some assignee roles are used by other components, so you should proceed with caution when modifying them.

This section provides Matter Management-specific guidelines regarding adding, deleting, renaming, and reordering assignee roles. Performing these procedures is covered by other chapters in this documentation. Additionally, this section highlights assignee roles requiring special attention.

#### Adding Assignee Roles

Adding new assignee roles has no impact other than the work necessary to add new custom fields, blocks, search views, and rules that may be associated with the new role.

#### Deleting Assignee Roles

Deleting an assignee role can cause a negative impact if rules depend on it. TeamConnect® Legal Matter Management has the following rules for the Dispute and Transaction objects that depend on the **Attorney** assignee role:

- Set Attorney As Main Assignee - Create - SYS
- Set Attorney As Main Assignee - Update - SYS

If you delete all delivered assignee roles, you should turn off the following rules:

- Require At Least One Assignee For Matter - Create - SYS
- Require At Least One Assignee For Matter - Update - SYS
- Track Changes To Main Assignee - SYS

If an assignee role has no dependent components, its deletion will not cause any issues. However, it is best to proceed with caution by determining whether the assignee roles is used by other components and make any necessary modifications.

When possible, it is preferable to change the name and display order while leaving the unique code as-is rather than deleting an assignee role.

### Renaming Assignee Roles

You can edit the names of assignee roles without causing problems because rules and CJBs reference the assignee roles' unique code, not the assignee roles' name. You may have to modify the UI to be consistent with the assignee role name change.

### Reordering Assignee Roles

Reordering the display order of assignee roles does not cause any issues because the order of the assignee roles is not referenced by XML files, CJBs, rules, or wizards.

## 3.2.8 Custom Fields

Many of TeamConnect® Legal Matter Management's fields are used by blocks, rules, wizards, and reports. To determine such dependencies, you can search the source code for the field name.

Business Objects reports depend on field names and the **Include in Data Warehouse** setting. See [Reports](#) and the Business Objects documentation for more information about report dependencies.

This section provides Matter Management-specific guidelines regarding adding, deleting, and renaming custom fields. Performing these procedures is covered by other chapters in this documentation.

### Adding Custom Fields

Adding custom fields does not cause any impact other than adding them to existing object blocks and CJBs if necessary.

### Deleting Custom Fields

Deleting custom fields can impact object blocks and CJBs, rules, wizards, reports, and search views. When deleting fields, you must determine such dependencies and fix any affected files, rules, search views and so on. Hiding fields may be preferable to deleting them.

### Renaming Custom Fields

Renaming field labels has no impact because the field name is used by other components, not the label. However, you may have to reconcile the new label with other UI terminology.

**Note:** If an object block "hard-codes" the label value, edits to the field label will not be reflected in the UI.

The field name is essentially a unique ID, so changing the name impacts any dependent object blocks and CJBs, rules, wizards, and search views throughout the system as well as in the API and reports.

### 3.2.9 Lookup Tables

All TeamConnect® Legal Matter Management lookup tables can be viewed and modified on the **Lookup Tables** screen. [Using Categories and Lookup Tables](#) provides a reference of TeamConnect® Legal Matter Management's lookup table items as well as the locations of all the fields and class files that use them.

You can find all instances of lookup tables that are used by class files by searching the source code for the `LT_` text string.

**Note:** You must have access to the source code (.java files, NOT .class files) to be able to search for such text strings.

This section provides Matter Management-specific guidelines regarding adding, modifying, reordering, and deleting items in lookup tables. Performing these procedures is covered by other chapters in this documentation. Additionally, this section highlights lookup tables requiring special attention.

#### Adding Items to Lookup Tables

Adding lookup tables should not cause any impact other than integrating them into the business solution.

#### Modifying Items in Lookup Tables

If you must modify a particular lookup table, determine which fields use the lookup table by referring to [Using Categories and Lookup Tables](#) and then determine which rules, wizards, or reports use those fields.

With the exception of the **Outside Counsel Evaluation** custom lookup table, you can rename lookup table items to suit your implementation without causing problems. However, the tree position of many lookup table items are used by rules and wizards.

#### Reordering Items in Lookup Tables

All lookup table items have an integer assigned to them that indicates the order in which they are to be displayed with respect to the other items in the same node. Lookup table items with the same integer are displayed alphabetically.

With the exception of the **Outside Counsel Evaluation** custom lookup table, you can re-order lookup table items to suit your implementation without causing problems. However, you must not change the tree position of any lookup table items because many rules and wizards depend on them.

**Important:** Do not change the order of the Outside Counsel Evaluation custom lookup table because several rules and wizards depend on the order of its items.

## Deleting Items in Lookup Tables

Deleting a lookup table can cause serious impact if fields or rules depend on it. Dependent fields are essentially deleted. Dependent rules will not work properly and may cause Java error messages to be displayed.

If a lookup table has no dependent components, its deletion will not cause any issues. However, it is best to proceed with caution by determining whether the lookup table is used by other components and make any necessary modifications.

## Lookup Tables Requiring Special Caution

The following lookup tables are essential to the corresponding functionality:

- **Budgeting Type**—Used for budgeting functionality.
- **Contact Relation Type**—This system lookup table is used in several CJBs such as the **Outside Counsel**, **Opposing Party**, and **Jurisdiction** blocks.
- **Matter Type**—Used for Invoice functionality. The tree positions must match the applicable matter objects (DISP and TRAN).
- **Matter Specific Budgeting Type**—Used for changing budgeting mode functionality.

You can safely rename the lookup items in these lookup tables, but you may have to reconcile the naming differences within the UI.

### 3.2.10 Custom Screens

Custom blocks and object views offer flexibility in screen layout and control over user access to record information. Custom fields are the core component of custom screens.

Custom fields can be created and displayed by category only. For all categories and custom fields you add to an object definition, associated access rights are automatically created. Users and user groups must be assigned the appropriate access rights to be able to view the custom screens.

**Note:** Some blocks use the **TCLegalJavascript.js** file to handle the show/hide behavior of some blocks. You can find all the XML files that use it by searching the source code for **TCLegalJavascript.js**.

This section provides Matter Management-specific guidelines regarding adding, modifying, and deleting custom screens.

## Adding Custom Screens

You can add custom screens with no impact other than the work necessary to integrate them with the rest of the product, such as UI design.

## Modifying Custom Screens

You can perform many modifications to the custom blocks and object views using the **Admin** menu, but some dependencies in class files may not be readily obvious.

## Deleting Custom Screens

If a custom field is deleted (or a category is deleted upon which the field depends) but remaining elements of the design require the field, TeamConnect® Legal Matter Management will display an error that shows which custom field is missing. For more information about determining category dependencies, see [Categories](#).

Some custom screens are used only for wizards and use wizard parameters. It is best not to change or remove wizard parameters, because many CJBs, actions, and conditions are dependent on the parameters.

## Searching for Categories Used by XML Files

You can find all categories that are used by XML files by searching the source code for the **category=** text string.

***Note:** You must have access to the source code (.java files, NOT .class files) to be able to search for such text strings.*

### 3.2.11 Rules

TeamConnect® Legal Matter Management 3.4 includes the following general sets of business rules that should meet the common needs of most clients:

- Business rules for matter objects
- Business rules for Involved party objects
- Business rules for Contact system object
- Business rules for Invoice system object
- Business rules for matter budgeting

This section provides Matter Management-specific guidelines regarding adding, modifying, activating, deactivating, and reordering rules.

#### Adding Rules

If you need to incorporate new rules, you should use the configuration tool in the UI. All rules can be created, viewed, and modified on the **Rules** tab of the corresponding object definition.

If you must use Java code to create the necessary rules, see [Utility Class Files](#) and [Base Classes](#) and review the corresponding files in the code.

#### Modifying Rules

If you must modify a rule, do not modify it directly. Instead, copy and rename it and make your modifications to the copy. You should deactivate the original rule.

You should be very careful when modifying rules for the Dispute and Transaction Involved Party objects because there are many dependent fields, categories, and other items. Most of these rules use class files.

## Activating Rules

You can activate a rule on the **General** tab of the corresponding rule screen.

Before activating rules, consider the affect it will have on users. For example, if you activate the Contact object rule **Only certain users can update contacts of type External Attorney or External Law Firm - SYS**, only users that belong to the Legal Administrator group as their default group will be allowed to update contact records with the **External : Attorney** or **External : Law Firm** categories.

## Deleting Rules

You should not delete rules; instead, deactivate them.

## Deactivating Rules

You can deactivate a rule on the **General** tab of the corresponding rule's screen.

However, you must not deactivate or delete the following rules:

- **Create Accounts For Matter Budget - SYS**
- **Set Matter Specific Budgeting Type - Update - SYS**
- **Update Dispute Budget - SYS**
- **Update Transaction Budget - SYS**
- **Create Accounts For Vendor Budget - SYS** (for the Dispute and Transaction Involved Party objects)

By default, the Invoice object requires a matter type and matter. To deactivate this feature, you must modify the **New Invoice** wizard and deactivate the following rules:

- **Require Matter When 'MatterType' Set - Create - SYS**
- **Require Matter When 'MatterType' Set - Update - SYS**
- **Set Invoice Project To Line Items - Update - SYS**
- **Set Invoice Project To Line Items - Post - SYS**
- **Set Invoice Project To Line Items - Create - SYS**

You must modify the **New Invoice** wizard. Set the **Default Value** field of the **WzSingleMatterTypePreference** wizard parameter to **No**. On the Invoice object's **Line Items** tab, set the **Show in Concise View** and **Show in Detailed View** lists to **Show** for the **Project** field.

## Reordering Rules

Rules of the same type are executed in order from lowest to highest. You can modify the order of most rules. However, the following rules are exceptions:

- **Set Matter Specific Budgeting Type - Update - SYS** Dispute and Transaction rule **MUST** have a higher order than **Update Dispute Budget - SYS or Update Transaction Budget - SYS**.
- **Set Non-Root Category As Default - Create - SYS** **MUST** have a higher order than **Matter Must Have Category Besides Root - Create - SYS**
- **Set Non-Root Category As Default - Update - SYS** **MUST** have a higher order than **Matter Must Have Category Besides Root - Update - SYS**

### 3.2.12 Reports

Multiple Business Objects universes for ad-hoc reporting and a set of standard Business Objects reports are an optional feature that can be purchased separately. For information on how to install Business Objects, see the *Installation Guide* for Business Objects.

The universes include:

- Standard Reporting universe
- Matter universe
- Finance universe
- Contact Management universe
- Events Universe
- Admin universe
- Document universe

The Business Objects Analytics option includes these reports:

- Top 25 Outside Counsel Spending
- Top 10 Matter by Spending
- Budget Summary
- Invoice Summary
- Spending by Cost Center
- Spending by Matter Category
- Settlement Payout
- Budget/Expenditure Detail
- Matter Count
- Case Progress

Contact Product Management for specific details regarding Business Objects reports.

**Caution:** The Business Objects reports depend on many fields. Changing the **Include in Data Warehouse** setting of custom Fields can impact the creation of reports. For specific field dependencies, see [Custom Fields](#).

### 3.2.13 Wizards

Wizards use categories, custom fields, actions, conditions, and CJBs. Their code is tied to the system and is difficult to modify without breaking.

TeamConnect® Legal Matter Management comes with the following wizards:

#### TeamConnect Legal Matter Management wizards

Object	Wizard
<b>Advice &amp; Counsel</b>	Create a New Advice & Counsel Matter
<b>Dispute</b>	New Dispute
	Create a Subpoena
<b>Invoice</b>	New Invoice
<b>Involved Party for Dispute</b>	New Jurisdiction
	New Opposing Party
<b>Involved Party</b>	New Outside Counsel
<b>Transaction</b>	New Transaction

The **New Outside Counsel** wizard is essentially the same for the Dispute and Transaction objects (some of their lists vary according to matter type).

When the **Create a New Advice & Counsel Matter** wizard is launched from within a Dispute or Transaction record, it creates an embedded Advice & Counsel record in order to display Advice & Counsel information within a custom block on the **Related Advice** tab. The wizard is essentially the same for the Dispute and Transaction objects.

When the **Create a Subpoena** wizard is launched from **Create a new Dispute**, it creates a Dispute record with **Subpoena** category. From the wizard, you can choose to create a subpoena of type: **Order to produce documents** or **Order to appear**. The wizard performs duplicate records checking based on these fields: **Issuing Court, Docket Number, Type of Subpoena**. If a new subpoena contains the same **Issuing Court, Docket Number, Type of Subpoena** values as another subpoena, the user will have an option to create a new record or update an existing record.

The **New Outside Counsel** wizard is particularly dependent on wizard parameters and the following Involved Party categories:



- Outside Counsel Firm
- Outside Counsel Firm : Primary
- Outside Counsel Attorney
- Outside Counsel Attorney : Primary

End users will use various wizards to do the following:

- Create a general matter records
- Create matter records with the Advice & Counsel category
- Add involved parties to matter records
- Create Invoice records

For more information about the wizards, see the corresponding Gap Analysis document and the wizard specifications in Maestro.

### Java Classes used by Wizards

The following table shows the Java classes used by specific wizard pages:

Java classes used by wizard pages

Wizard name	Page	Class	Description
<b>Create a New Advice &amp; Counsel Matter</b>	General Information	WzACValidateFieldsQSY S	Validate required fields on the <b>General Information</b> page.
		WzACValidateFieldsASY S	If there are empty required fields, display the General Information page and the error message "Required field: fieldname cannot be empty."
		WzACCheckWasRelated ASYS	Checks whether the user made a selection on the <b>Suggestion</b> page. If so, the wizard proceeds to the <b>Use Existing AC</b> page.
		WzACNoSearchResultsQ SYS	Checks whether there are any search results based on user input on the <b>General Information</b> page. If not, the wizard proceeds to the <b>Assignees</b> page.
	Assignees	CreateEmbeddedObject	When invoked from a dispute or transaction record, this page transition rule creates an embedded advice & counsel record in order to

			display advice & counsel information within a custom block on the <b>Related Advice</b> tab.
<b>Create a Subpoena</b>	General Information	WzDSetCategorySubpoenaASYS	Sets the <b>Subpoena</b> category for a dispute so that related custom fields will be shown.
		WzDValidateSubpoenaFieldsASYS	Validates required fields on the <b>General Information</b> page
	Potential Duplicate Records	WzDSetSelectedSubpoenaASYS	If a duplicate existing subpoena is detected, sets the user's selection to create a subpoena or update an existing subpoena (on the <b>Please make a Selection</b> page).
	Matter Relations	WzDCreateOrUpdateSubpoenaASYS	Sets whether to create a subpoena or update an existing subpoena after the user clicks <b>Finish</b> on the <b>Relations</b> page.
<b>New Invoice</b>	General	WzInvoiceValidateFieldsQSYS	Validate fields on the <b>General</b> page. If the values of any required fields are not valid, display an error and do not leave the <b>General</b> page.
<b>New Jurisdiction</b>	Jurisdiction Selection	WzJValidateContactsQSYS	Validate the contact fields on the <b>Jurisdiction Selection</b> page. If either value is not valid, display an error and do not leave the <b>Jurisdiction Selection</b> page.
<b>New Outside Counsel (Dispute)</b>	Outside Counsel Attorney Selection	WzDOCValidatePrimaryAttorneyQSYS	Validate the fields on the <b>Outside Counsel Attorney</b> Selection page. If either value is not valid, display an error and do not leave the <b>Outside Counsel Attorney</b> Selection page.
	Outside Counsel Firm Selection	WzDOCValidatePrimaryFirmQSYS	Validate the fields on the <b>Outside Counsel Firm Selection</b> page. If either value is not valid, display an error and do not leave the <b>Outside Counsel Firm Selection</b> page.
<b>New Outside</b>	Outside Counsel	WzTOCValidatePrimaryAttorneyQSYS	Validate the fields on the <b>Outside Counsel Attorney Selection</b> page. If

<b>Counsel (Transaction)</b>	Attorney Selection		either value is not valid, display an error and do not leave the <b>Outside Counsel Attorney Selection</b> page.
	Outside Counsel Firm Selection	WzTOCValidatePrimaryFirmQSYS	Validate the fields on the Outside Counsel Firm Selection page. If either value is not valid, display an error and do not leave the <b>Outside Counsel Firm Selection</b> page.
<b>New Dispute</b>	Details	WzDValidateDetailsQSYS	Validate fields on the <b>Details</b> page. If the values of any required fields are not valid, display an error and do not leave the <b>Details</b> page.
	General	WzDPopulateCategoriesQSYS	Before proceeding to the next page, populate the user's category selections in the dispute record.
		WzDValidateFieldsQSYS	Validate fields on the <b>General</b> page. If not valid, do not leave the <b>General</b> page.
		WzDHasNoDetailsQSYS	If users select only the <b>Contracts</b> or <b>Intellectual Property</b> categories (which have no associated <b>Details</b> page blocks), the wizard does not display the <b>Details</b> page but goes instead to the <b>Assignees</b> page.
	Outside Counsel Attorney Selection	WzDValidatePrimaryAttorneyQSYS	Validate that there is only one primary attorney. If there is more than one primary attorney, an error message is displayed and the wizard does not proceed past the <b>Attorney Selection</b> page.
	Outside Counsel Firm Selection	WzDIsNoFirmsSelectedAndFirmsListEmptyQSYS	If there are no contacts with the <b>External : Law Firm</b> category, the wizard displays the No Firms Found message on the <b>Outside Counsel Firm Selection</b> page. When users click <b>next</b> , the wizard proceeds directly to the <b>Budget</b> page.
WzDValidatePrimaryFirmQSYS		Validate that there is only one primary firm. If there is more than one	

			primary firm, an error message is displayed and the wizard does not proceed past the <b>Firm Selection</b> page.
<b>New Transaction</b>	Details	WzTValidateDetailsQSYS	Validate fields on the <b>Details</b> page. If the values of any required fields are not valid, display an error and do not leave the <b>Details</b> page.
	General	WzTPopulateCategoriesQSYS	Before proceeding to the next page, populate the user's category selections in the transaction record.
		WzTValidateFieldsQSYS	Validate fields on the <b>General</b> page. If the values of any required fields are not valid, do not leave the <b>General</b> page.
		WzTHasNoDetailsQSYS	If the user selects a transaction category that has no associated <b>Details</b> page blocks, the wizard does not display the <b>Details</b> page but goes instead to the Assignees page.  <i><b>Note:</b> All delivered transaction categories have associated <b>Details</b> page blocks, so this class is only used if there are additional custom categories.</i>
	Outside Counsel Attorney Selection	WzTValidatePrimaryAttorneyQSYS	Validate that there is only one primary attorney. If there is more than one primary attorney, an error message is displayed and the wizard does not proceed past the <b>Outside Counsel Attorney Selection</b> page.
<b>New Transaction</b>	Outside Counsel Firm Selection	WzTValidatePrimaryFirmQSYS	Validate that there is only one primary firm. If there is more than one primary firm, an error message is displayed and the wizard does not proceed past the <b>Outside Counsel Firm Selection</b> page.
		WzTIsNoFirmsSelectedAndFirmsListEmptyQSYS	If there are no contacts with the <b>External : Law Firm</b> category, the

			wizard displays the No Firms Found message on the <b>Outside Counsel Firm Selection</b> page. When users click <b>next</b> , the wizard proceeds directly to the <b>Budget</b> page.
--	--	--	--

TeamConnect® Legal Matter Management's wizards also use CJBs. See the wizard sections of the Gap Analysis documents for the names of the CJB files corresponding to each CJB.

### 3.2.14 Custom Tools

TeamConnect® Legal Matter Management has the following custom tools:

- **Invoice Validation Rule Settings**—Allows you to activate or deactivate invoice validation rules. For more details, see the *CSM Configuration Guide*.
- [Default Settings](#)—Allows you to set the budgeting mode for the entire system.
- **Map Service of Process**—Allows you to map SOP actions, agents, and targets. For more details, see the *SOP 1.0 Installation and Configuration Guide*.
- [Time Entry Tool](#)—Allows you to manage timekeepers' entries.

The **Month** field references the **Fiscal Period** custom lookup table, which includes all twelve months. The default fiscal period is **January**.

**Important:** *The month should be set before any dispute or transaction records are created. Alternatively, the month can be set just before running the Fiscal Year Tool.*

### Default Settings

By default, all the following check boxes in the **Budget for** section are selected:

- Outside Counsel Legal Fees
- Outside Counsel Legal Expenses
- Expert or Other Vendors Fees
- Expert or Other Vendor Expenses
- Internal Expenses
- Settlements or Awards
- Internal Time

Each dispute or transaction record will have corresponding accounts automatically created for each item selected in the **Budget for** section.

By default, **Do not budget by fiscal period** is selected in the **Extend budget level by** section. Corresponding fiscal period accounts are created for each dispute or transaction record when the one of the following options are selected:

- Annually

- Annual+Quarter
- Annual+Month

For example, if **Internal Time** is selected in the **Budget for** section and **Annual+Month** is selected in the **Extend budget level by** section, the following corresponding accounts are created for each dispute or transaction record:

- Internal Fees
- Internal Fees : Annual (one for each fiscal year)
- Internal Fees : Monthly (twelve for each fiscal year)

### 3.2.14.1 Time Entry Tool

The Time Entry Tool comes with default settings that allow you to use it out of the box. You will need to configure rights for Users/User Groups and Contacts to enable rights to perform Time Entry Tool Administrator and User functions.

***Important:*** You must set a default rate for each timekeeper or errors will occur when using the Time Entry Tool.

This section describes:

- Out-of-the-box default settings for the Time Entry Tool, as well as how to set up and customize the tool
- Administrator's usage of the Time Entry Tool

For more information, see information about using the Time Entry Tool from a user's point of view.

#### 3.2.14.1.1 Editing Time Entry Tool Settings

You can customize the Time Entry Tool by changing the default settings that determine:

- Billing Period definition
- Daily Billable hours
- Timekeeper required categories
- Task Category

**Details**

\* Time Period: Weekly

\* Total Billable Hours per day: 8

\* Days per week: 5

\* Timekeeper Category: External  
: Agency  
: : Federal  
: : Local

Task Category: Internal Tasks

Time Entry Settings Details Block

### To edit Time Entry Tool Settings

You can edit the default settings that affect billing Time Period organization, the Time Entry Tool **Available Hours** value, that determine who can use the Time Entry Tool, and available Task Categories from the Time Entry Tool.

1. From the **Main** menu, click the **Go to** drop-down list and select **Time Entry Settings**.
2. Click the **Time Entry Settings** link.
3. Click **Edit**. The following fields are on the **General** tab, **Details** block.
4. (optional) To change the **Time Period**, click the drop-down list and select **Daily**, **Weekly** (default), or **Monthly**. The period determines the time span for which Legal Administrators can define billing Time Periods and time spans for which users can enter time entries.
5. (optional) If you selected **Weekly** for the **Time Period** above, type the billable **Days per week**.

Days per week only displays if the **Time Period** was set to **Weekly**.

6. (optional) Type the **Total Billable Hours per day** (default - 8). On the Time Entry Tool, this value is used to calculate the total **Available Hours** for a Time Period.

For weekly time periods, the **Total Billable Hours per day** value is multiplied by the **Days per week** value (described above) to produce the **Available Hours** (displayed on the Time Entry Tool). For example, for one week's time period, the default Available Hours could be (6 Days per week \* 8 Total Billable Hours per day) = 48 hours.

For monthly time periods, Monday through Friday are considered billable (or there are 5 billable days per week). For example, **Available Hours** for the month of June (21 weekdays) would be (21 Days per month \* 8 Total Billable Hours per day) = 168 hours.

7. (optional) Select one or more **Timekeeper Categories**. These values determine the default category required for a Matter Management contact to be authorized as a timekeeper (prerequisite to using the Time Entry Tool).

8. To restrict the **Task Category** to describe a new record, click the drop-down list and select a category. In the Time Entry Tool, this value (and child categories) displays as an option from the Category drop-down list when adding time entries.

In the Time Entry Tool, only tasks (created from Disputes or Transaction Matter records) with the **Default Category** that is specified in the drop-down list as the current **Task Category** will display. For example, if a new task record with **Default Category** of Internal Tasks (default **Task Category** value in Time Entry Settings) is saved, then that task will display as a time entry in the Time Entry Tool.

9. Click **Save**.

#### 3.2.14.1.2 Editing Default Time Entry Tool Fields

The Time Entry Tool has been created with default fields and field settings. For all fields, you can change the following settings:

- Column Label
- Column Order
- Column Display Width
- Required

By default, one field, **Task Activity** is hidden. You can activate this field or later hide it again. The existing default fields cannot be hidden.

### Time Entry Tool Setting

Number of entries you would like to add:

Field Name	Column Label	Column Order	Column Display Width	Required
1	(None Selected) <input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
<a href="#">+ add more</a>				
Field Name	Column Label	Column Order	Column Display Width	Required
1 <input type="checkbox"/>	actualHours	Hours	60	6 YES
2 <input type="checkbox"/>	completedOn	Completed On	50	15 YES
3 <input type="checkbox"/>	defaultCategory	Category	40	30 YES
4 <input type="checkbox"/>	Project	Matter	10	20 YES
5 <input type="checkbox"/>	shortDescription	Description	30	30 YES

[Check All](#) - [Uncheck All](#)

Time Entry Settings, Time Entry Tool Setting Block



**Note:** This image displays default fields and related values for **Column Label**, **Column Order**, **Column Display Width**, and **Required** fields.

### To edit default Time Entry Tool fields

You can change how Time Entry Tool fields display and whether the fields are required by users.

1. From the **Main** menu, click the **Go to** drop-down list and select **Time Entry Settings**.
2. Click the **Time Entry Settings** link.
3. Click **Edit**. The following fields are on the **General** tab, **Time Entry Tool Setting** Block.
4. To change the text that displays for default fields on the Time Entry Tool:
  - a. Check the box(es) next to the existing **Field Name** and click **edit**.
  - b. Type the new text to identify the field in the **Column Label** box.
  - c. Click **ok** on that field row.
5. To change the position of a Time Entry Tool field from left to right:
  - a. Check the box(es) next to the existing **Field Name** and click **edit**.
  - b. Type a numeric value in the **Column Order** box, where the lowest value displays furthest to the left and the highest value displays furthest to the right on the screen.
  - c. Click **ok** on that field row.
6. To change the maximum length for displaying the field's **Column Label**:
  - a. Check the box(es) next to the existing **Field Name** and click **edit**.
  - b. Type a numeric value in the **Column Display Width** box. This value determines the length of space allotted to display the corresponding Time Entry Tool field value. For example, for the **actualHours** field, the default **Column Display Width** of 6 would allow a Time Entry Tool user to enter a value like 100.25
  - c. Click **ok** on that field row.
7. To set whether an existing Time Entry Tool field is required by users to complete:
  - a. Check the box(es) next to the existing **Field Name** and click **edit**.
  - b. Clear the **Required** box to display the field but allow blank values, or check the **Required** box prevent saving this field with blank values.
  - c. Click **ok** on that field row.
8. To display the **activityItem** field on the Time Entry Tool:
  - a. From the **Number of entries you would like to add**: drop-down list, select 1.
  - b. From the **Field Name** drop-down list, select **activityItem**.
  - c. Type a descriptive field label in **Column Label** box (for example, Task Activity).
  - d. Type the field's **Column Order** (number).

- e. Type the **Column Display Width**.
  - f. Set the **Required** check box. See field explanations above for more information.
9. Click **Save**.

### 3.2.14.1.3 Creating Billing Time Periods

You can define Time Periods to organize billing periods in which timekeepers enter records for the Time Entry Tool. There are two sections from which you can create a Time Period. You can use the Create Next Fiscal Budget tool (found in the end-user interface in TeamConnect) to create Billing Time Periods that match corresponding budgets (for example, yearly periods), or you can define independent Time Periods from the Time Entry Settings area (for example, on a daily, weekly, or monthly basis).

General | **Time Periods** | Designated Timekeepers |

**Generate Time Periods**

To automatically generate Time Periods, use the "Create Next Fiscal Budget" tool. Otherwise please provide the Start Date and End Date then click Generate button below.

Start Date :  End Date :

**Time Periods**

Current View:

| **Results** | Filter |

Time Periods 1-7 of 7 Time Periods per page:

Object Link
<a href="#">06/01/2007 - 06/30/2007</a>
<a href="#">06/01/2008 - 06/30/2008</a>
<a href="#">07/01/2007 - 07/31/2007</a>
<a href="#">08/01/2007 - 08/31/2007</a>
<a href="#">08/01/2009 - 08/31/2009</a>
<a href="#">09/01/2007 - 09/30/2007</a>
<a href="#">10/01/2007 - 10/31/2007</a>

Time Entry Settings, Time Periods Tab

You can create Time Periods to match corresponding budgets (using the Fiscal Year tool), or you can create autonomous time periods using the date fields in the Time Entry Settings. For more details, see [Default Settings](#).

#### To create Time Periods from Time Entry Settings

If you have already created billing Time Periods from the Create Next Fiscal Budget tool, you can skip this procedure. The **Generate Time Periods** section allows quick creation of multiple Time Periods. The **Time Periods** section allows creation of individual Time Periods and overriding the default **Available Hours** calculation.

1. From the **Main** menu, click the **Go to** drop-down list and select **Time Entry Settings**.

2. Click the **Time Entry Settings** link.
3. Click **Edit**.
4. Select the **Time Periods** tab.
5. To create multiple Time Periods quickly, from the **Generate Time Periods** section, click the Calendar icon for **Start Date** and select a date. Click the Calendar icon for **End Date** and select a date.

If the date range spans multiple time periods, then multiple time periods will be created. For example, if the time period is monthly and the start date is

06/01/2007 and the end date is 08/31/2007, then three time periods will be created.

- a. Click **Generate**.
- b. Click **Save** (for the Time Entry Setting Record).

It is recommended to create Time Periods from the Generate Time Periods section because the Time Entry Tool **Available Hours** will be automatically calculated based on the **Billable Hours Per Day** \* (billable) **Days in the Time Period**.

6. To create individual Time Periods, from the **Time Periods** section, click **New**.
  - a. Click the Calendar icon for **End Date** and select a date.
  - b. Click the Calendar icon for **Start Date** and select a date.
  - c. Type a number in **Total Billable Hours** (sets the **Available Hours** in the Time Entry Tool).
  - d. Click **Save**.

**Note:** The default naming convention for new Time Periods is "Start Date"- "End Date". For example, 06/01/2007 - 06/30/07.

#### 3.2.14.1.4 Managing Timekeepers

Members of Legal Administrator group have the authority to set up designated timekeepers. A designated timekeeper has the authority to enter another timekeeper's time entries.

**Important:** You must set a default rate in the contact record of each timekeeper or errors will occur when using the Time Entry Tool.

General | Time Periods | **Designated Timekeepers** |

### Designated Timekeeper

Number of entries you would like to add:

Timekeeper	Designated Timekeeper
1 <input type="text" value="(None Selected)"/>	<input type="text" value="Clouds, Steve"/> <input type="text" value="Finan, Steve T"/> <input type="text" value="Kirby, Joe"/> <input type="text" value="Rutherford, Madison"/>

[+ add more](#)

Timekeeper	Designated Timekeeper
1 <input type="checkbox"/> Clouds, Steve	Madison Rutherford
2 <input type="checkbox"/> The, Phok	Steve T Finan, Madison Rutherford
3 <input type="checkbox"/> zev, zevon	Steve Clouds, Madison Rutherford

[Check All](#) - [Uncheck All](#)

Time Entry Settings Designated Timekeepers Tab

### To delegate Timekeeping duties

Members of the Legal Administrator group can set up designated timekeepers. For example, if Ralph and Tim are timekeepers, Tim can be designated with the ability to enter Ralph's time entries in the Time Entry Tool.

1. From the **Main** menu, click the **Go to** drop-down list and select **Time Entry Settings**.
2. Click the **Time Entry Settings** link.
3. Click **Edit**.
4. Select the **Designated Timekeepers** tab.
5. From the **Number of entries you would like to add**: drop-down list, select the number of timekeepers for which you want to delegate time entry actions.
6. From the **Timekeeper** drop-down list, select the timekeeper to delegate time entry actions for. For example, if you would like Tim to be able to enter Ralph's time entries, select Ralph.
7. From the **Designated Timekeeper** drop-down list, select one or more timekeeper delegates. The maximum number of designated timekeepers for a single Timekeeper is 3.

For example, if you would like Susan, Tim, and George to be able to enter Ralph's time entries, select Susan. Click the CTRL key and select Tim. Click the CTRL key and select George.

8. Click **ok** for new timekeeper-delegates rows.
9. Click **Save** (for the Time Entry Setting record).

### To update the designated Timekeepers list

You can add or remove delegated timekeepers to a Designated Timekeepers list.

1. From the **Main** menu, click the **Go to** drop-down list and select **Time Entry Settings**.
2. Click the **Time Entry Settings** link.
3. Click **Edit**.
4. Select the **Designated Timekeepers** tab.
5. Check the box next to existing **Timekeeper-Delegates** rows to update.
6. Select or deselect **Timekeeper delegates**.
7. Click **ok** for updated **Timekeeper-Delegates** rows.
8. Click **Save** (for the Time Entry Setting record).

### To delete the designated Timekeeper-Delegate combinations

You can remove designated timekeeper combinations.

1. From the **Main** menu, click the **Go to** drop-down list and select **Time Entry Settings**.
2. Click the **Time Entry Settings** link.
3. Click **Edit**.
4. Select the **Designated Timekeepers** tab.
5. Check the box next to existing timekeeper-delegates rows to delete.
6. Click **delete**.
7. Click **Save** (for the Time Entry Setting record).

#### 3.2.14.1.5 Using the Time Entry Tool as Administrator

The **Legal Administrator** group can do the following:

- Review and verify the time entries for all timekeepers' entries in the Time Entry Tool.
- Change the phase of a billing time period.
- Add, change (edit, post, void), or delete time entries for other timekeepers.

For more information, see the Legal Matter Management User Help.

### Reviewing Time Entries

While reviewing timekeepers' entries in the Time Entry Tool or after the time period has ended, you can lock the corresponding time period. If the time period phase is locked, timekeepers will be prevented from [entering, changing, posting, voiding, or deleting time entries](#) from the Time Entry Tool.

#### ▣ To verify time entries for a Time Period

1. From the **Main** menu, **Tools** drop-down list, select **Time Entry Tool**.
2. For the **Task For**, select the time period to view by clicking the Calendar icon to select a date.
  - If the time period is weekly, the corresponding week's time period is displayed.
  - If the time period is monthly, the corresponding month's time period is displayed.
3. From the **Timekeepers** drop-down list, select the timekeeper whose time entries to view.

### Time Period Phases

A Time Period can have the following phases:

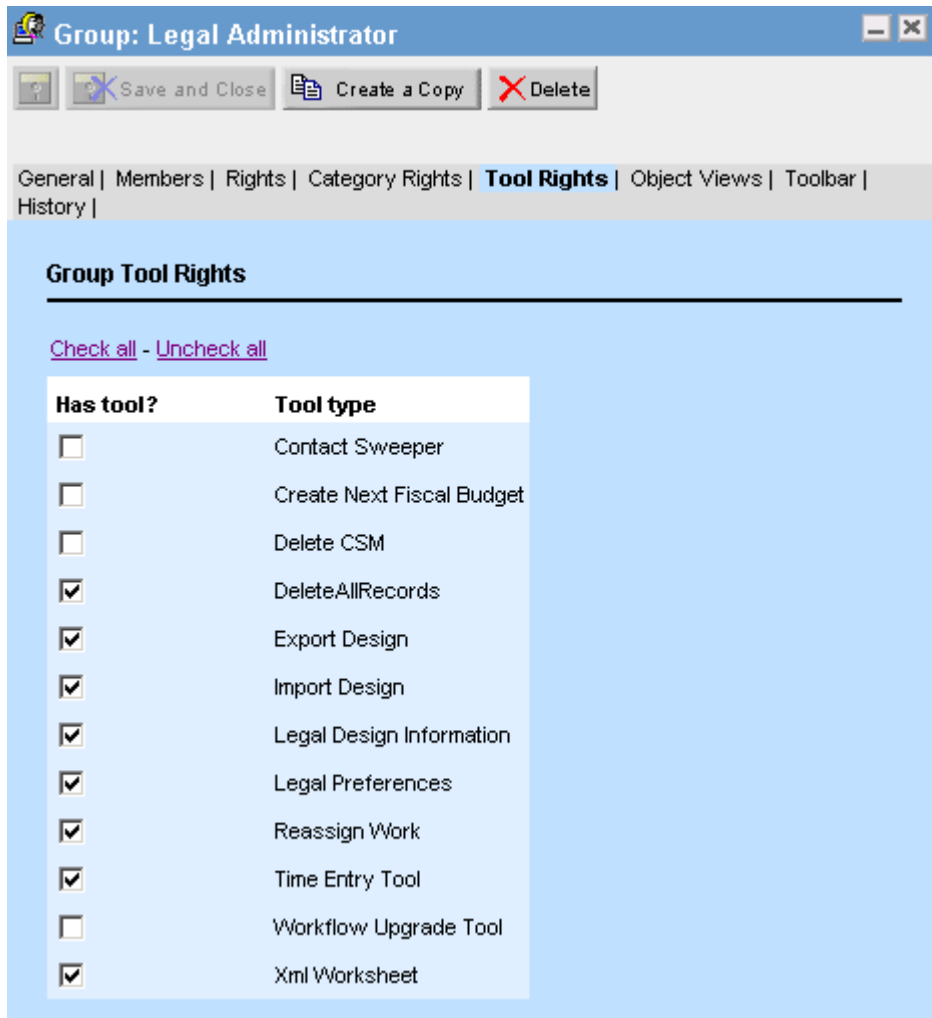
- **Un-Lock**—(default) Allows timekeepers to enter, change (edit, post, void), and delete time entries in the Time Entry Tool
- **Lock**—Prevents timekeepers from entering, changing (editing, posting, voiding), and deleting time entries in the Time Entry Tool. A Time Period in Locked phase can be changed to **Un-Lock** or **Processed** phases.
- **Processed**—Prevents timekeepers from entering, changing (editing, posting, voiding), and deleting time entries in the Time Entry Tool. After a Time Period is set to the **Processed** phase, you cannot change it to previous phases.

#### To change a Time Period's phase

1. From the **Main** menu, click the **Go to** drop-down list and select **Time Entry Settings**.
2. Click the **Time Entry Settings** link.
3. Click **Edit**.
4. Select the **Time Periods** tab.
5. From the **Time Periods** section **Filter** tab, search for a Time Period.
6. From the **Results** tab, click a Time Period name.
7. From the **(actions)** drop-down list, select one of the following:
  - **Un-Lock**—Only displays if the current phase is Lock
  - **Lock**—Only displays if the current phase is Unlocked
  - **Processed**
8. From the confirmation dialog, click **OK**.

### 3.2.14.2 Tool Rights

You can set the tool rights on the **Tool Rights** tab of the group account screen:



**Group Tool Rights**

[Check all](#) - [Uncheck all](#)

Has tool?	Tool type
<input type="checkbox"/>	Contact Sweeper
<input type="checkbox"/>	Create Next Fiscal Budget
<input type="checkbox"/>	Delete CSM
<input checked="" type="checkbox"/>	DeleteAllRecords
<input checked="" type="checkbox"/>	Export Design
<input checked="" type="checkbox"/>	Import Design
<input checked="" type="checkbox"/>	Legal Design Information
<input checked="" type="checkbox"/>	Legal Preferences
<input checked="" type="checkbox"/>	Reassign Work
<input checked="" type="checkbox"/>	Time Entry Tool
<input type="checkbox"/>	Workflow Upgrade Tool
<input checked="" type="checkbox"/>	Xml Worksheet

Alternatively, you can set the tool rights according to user on the **Tool Rights** tab of the user's account screen:

**User: Clouds, Steve (steve)**

Save and Close | Create a Copy | Delete

General | Groups | Rights | Category Rights | **Tool Rights** | History

**User Tool Rights**

[Check all](#) - [Uncheck all](#)

Has tool?	Has tool from group?	Tool type
<input type="checkbox"/>	NO	Contact Sweeper
<input checked="" type="checkbox"/>	NO	Create Next Fiscal Budget
<input type="checkbox"/>	NO	Delete CSM
<input checked="" type="checkbox"/>	YES	DeleteAllRecords
<input checked="" type="checkbox"/>	YES	Export Design
<input checked="" type="checkbox"/>	YES	Import Design
<input checked="" type="checkbox"/>	YES	Legal Design Information
<input checked="" type="checkbox"/>	YES	Legal Preferences
<input checked="" type="checkbox"/>	YES	Reassign Work
<input checked="" type="checkbox"/>	YES	Time Entry Tool
<input type="checkbox"/>	NO	Workflow Upgrade Tool
<input checked="" type="checkbox"/>	YES	Xml Worksheet

### 3.2.15 CSM Configuration for Matter Management

Three of the Collaborati Spend Management (CSM) Settings tabs are modified for Matter Management:

- E-Billing Roles
- Vendors
- Payments

Information about the default Matter Management configurations is described for reference in this section.

For more information about the default CSM Settings configuration for the **Connection** tab, **Timekeepers** tab, **Default Rates** tab, and **History** tab, see the *Collaborati Spend Management Configuration Guide*.



### 3.2.15.1 CSM e-Billing Roles Tab Configuration

By default, the CSM Settings, **E-Billing Roles** tab will be configured with the following e-Billing Roles:

#### E-Billing Roles in CSM Settings

Object Type	Object	Role
Involved	Dispute	Outside Counsel Firm
Involved	Transaction	Outside Counsel Firm

#### E-Billing Roles

Number of entries you would like to add:

Object Type	Object	Role
1	<input type="text" value="(Select)"/>	

[+ add more](#)

Object Type	Object	Role
1	<input type="checkbox"/> Involved	Dispute      Outside Counsel Firm
2	<input type="checkbox"/> Involved	Transaction      Outside Counsel Firm

[Check All](#) - [Uncheck All](#)   [edit](#)   [delete](#)

### 3.2.15.2 CSM Vendors Tab Configuration

By default, the CSM Settings, Vendors tab will be configured with the following Vendor Settings enabled (checked):

- Automatically update contact records with latest Collaborati vendor information.
- Automatically activate vendors after authorization.
- Automatically authorize new codes.

#### Vendor Settings

- Automatically update contact records with latest Collaborati vendor information.
- Automatically activate vendors after authorization.
- Automatically authorize new codes.

### 3.2.15.3 CSM Payment Tab Configuration

By default, the **CSM Settings, Payments** tab will be configured with the following setting:

**Note:** Share payment information with vendors will be disabled (not checked).

#### CSM Payment Settings

Share	Information	Invoice Field Name
Not checked	Check Number	CheckNumberIN
Not checked	Payment Amount	TotalAmountOnCheckIN
Not checked	Payment Date	CheckDateIN
Not checked	PO Number	PONumberIN

#### Payment Settings

Share payment information with vendors

Please specify which payment information you would like to share with vendors from the list below. For each field, please select the invoice field where you store this information.

Share	Information	Invoice Field Name
<input type="checkbox"/>	Check Number	CheckNumberIN
<input type="checkbox"/>	Payment Amount	TotalAmountOnCheckIN
<input type="checkbox"/>	Payment Date	CheckDateIN
<input type="checkbox"/>	PO Number	PONumberIN

Payment Currency: System Currency

### 3.2.16 Groups and Users

TeamConnect® Legal Matter Management 3.4 comes with five [user groups](#) with the following sets of pre-configured rights based on their job responsibilities within a legal department:

#### Pre-configured user group rights

User group	Rights
<b>CSM Admin</b>	<p>Members of the <b>CSM Admin</b> group have no Admin, User, or category rights. They have rights only to the following:</p> <ul style="list-style-type: none"> <li><b>CSM Settings</b> object and its embedded objects</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Activity Code Authorization</b> object</li> <li>• <b>Expense Code Authorization</b> object</li> <li>• <b>Invoice Payment</b> object</li> <li>• <b>Payment Field Mapping</b> object</li> <li>• <b>Synchronization History</b> object</li> <li>• <b>Task Code Authorization</b> object</li> <li>• <b>Timekeeper</b> object</li> <li>• <b>Vendor</b> object and its embedded objects</li> <li>• <b>Vendor Timekeeper Classification Contact Mgmt</b> object</li> <li>• <b>Invoice Validation Rule Setting</b> tool</li> <li>• <b>Xml Worksheet</b> tool</li> </ul>
<p><b>Dispute Attorney</b></p>	<p>Members of the <b>Dispute Attorney</b> group have no Admin rights. They have rights only to the following:</p> <ul style="list-style-type: none"> <li>• <b>All User Rights</b></li> </ul> <p><i>Caution: "All User Rights" includes the Account object.</i></p> <ul style="list-style-type: none"> <li>• <b>Dispute</b> object, and its embedded objects</li> <li>• <b>Advice &amp; Counsel</b> object</li> <li>• <b>Cost Center</b> object and its embedded object</li> <li>• <b>Category</b> rights to some <b>Line Item</b> categories</li> <li>• <b>Time Entry Tool</b> tool</li> </ul> <p><i>Note: The contact's default <b>Category</b> needs to be set to Internal (or the current Time Entry Setting, <b>General</b> tab, <b>Timekeeper Category</b> value).</i></p>
<p><b>Legal Administrator</b></p>	<p>Members of the <b>Legal Administrator</b> group have no Admin rights. They have rights only to the Dispute object, its related and embedded objects, and to the following:</p> <ul style="list-style-type: none"> <li>• <b>All User Rights</b></li> </ul> <p><i>Caution: "All User Rights" includes the Account object.</i></p> <ul style="list-style-type: none"> <li>• <b>Advice &amp; Counsel</b> object</li> <li>• <b>Cost Center</b> object and its embedded object</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Dispute</b> object and its embedded objects</li> <li>• <b>Time Entry Setting</b> object and its embedded objects</li> <li>• <b>Time Period</b> object</li> <li>• <b>Transaction</b> object and its embedded objects</li> <li>• <b>Category</b> rights to some <b>Line Item</b> categories</li> <li>• <b>DeleteAllRecords</b> tool</li> <li>• <b>Import Design</b> tool</li> <li>• <b>Legal Preferences</b> tool</li> <li>• <b>Reassign Work</b> tool</li> <li>• <b>Time Entry Tool</b> tool</li> </ul> <p><i>Note: The contact's default <b>Category</b> needs to be set to Internal (or the current <b>Time Entry Setting</b>, <b>General</b> tab, <b>Timekeeper Category</b> value).</i></p> <ul style="list-style-type: none"> <li>• <b>Xml Worksheet</b> tool</li> </ul>
<b>Paralegal</b>	<p>Members of the <b>Paralegal</b> group have no Admin rights. They have rights only to the following:</p> <ul style="list-style-type: none"> <li>• <b>All User Rights</b></li> </ul> <p><i>Caution: "All User Rights" includes the Account object.</i></p> <ul style="list-style-type: none"> <li>• <b>Dispute</b> object and its embedded objects</li> <li>• <b>Advice &amp; Counsel</b> object</li> <li>• <b>Cost Center</b> object and its embedded object</li> <li>• <b>Transaction</b> object, and its embedded objects</li> <li>• <b>Category</b> rights to some <b>Line Item</b> categories</li> <li>• <b>Time Entry Tool</b> tool</li> </ul> <p><i>Note: The contact's default <b>Category</b> needs to be set to Internal (or the current <b>Time Entry Setting</b>, <b>General</b> tab, <b>Timekeeper Category</b> value).</i></p>
<b>Reporting</b>	<p>Members of this group can view the <b>Reports</b> home page.</p> <p>This user group has no Admin, User, Category or Tool rights.</p>

<b>SOP Admin</b>	<p>Members of the <b>SOP Admin</b> group have no Admin, User, or Category rights. They have rights only to the following:</p> <ul style="list-style-type: none"><li>• <b>SOP Settings</b> object:<ul style="list-style-type: none"><li>○ Project, Read</li><li>○ Project, Update</li><li>○ Project, All Categories, Read</li><li>○ Project, All Categories, Create</li><li>○ Project, All Custom Fields, Read</li><li>○ Project, All Custom Fields, Update</li></ul></li><li>• <b>Retrieval Log</b> object</li><li>• <b>Map Service Of Process</b> tool</li></ul>
<b>SOP Process Manager</b>	<p>Members of the <b>SOP Process Manager</b> group have no Admin, Category, or Tool rights. They have rights only to the following:</p> <ul style="list-style-type: none"><li>• <b>SOP Settings</b> object:<ul style="list-style-type: none"><li>○ Project, Read</li><li>○ Project, Update</li><li>○ Project, All Categories, Read</li><li>○ Project, All Categories, Create</li><li>○ Project, All Custom Fields, Read</li><li>○ Project, All Custom Fields, Update</li></ul></li><li>• <b>Consolidated Case Info</b> object</li><li>• <b>Retrieval Log</b> object:<ul style="list-style-type: none"><li>○ Project, Read</li><li>○ Project, All Categories, Read</li><li>○ Project, All Categories, Create</li><li>○ Project, All Custom Fields, Read</li><li>○ Project, All Custom Fields, Update</li></ul></li><li>• <b>Service</b> object</li><li>• <b>Service Of Process - CSC</b> object</li><li>• <b>Service Of Process - CT</b> object</li><li>• <b>SOP Note</b> object</li></ul>

	<ul style="list-style-type: none"> <li>• Contact, Read</li> <li>• Document, Read</li> <li>• Document, All Categories, Read</li> </ul>
<b>Transaction Attorney</b>	<p>Members of the <b>Transaction Attorney</b> group have no Admin. They have rights only to the following:</p> <ul style="list-style-type: none"> <li>• <b>All User Rights</b></li> </ul> <p><i>Caution: "All User Rights" includes the Account object.</i></p> <ul style="list-style-type: none"> <li>• <b>Advice &amp; Counsel</b> object</li> <li>• <b>Cost Center</b> object and its embedded object</li> <li>• <b>Transaction</b> object, and its embedded objects</li> <li>• <b>Category</b> rights to some <b>Line Item</b> categories</li> <li>• <b>Time Entry Tool</b> tool</li> </ul> <p><i>Note: The contact's default <b>Category</b> needs to be set to Internal (or the current Time Entry Setting, <b>General</b> tab, <b>Timekeeper Category</b> value).</i></p>

Because users should not alter TeamConnect® Legal Matter Management's automated accounts, you should restrict user access to the Account object. Many rules and wizards depend on the standard configuration.

### 3.2.17 Service of Process Manager for Matter Management

Service of Process (SOP) Manager for Matter Management supports integration with CT and CSC Registered Agents (RA).

For each Registered Agent, you must add the mapping files to Matter Management in order to use them. Details about adding the mapping files and the contents of the files is described for reference in this section.

For more information about SOP Manager, see the *Service of Process Manager User Guide* and the *Service of Process Manager Installation and Configuration Guide*.

**Note:** If you use both RAs, install each mapping file so that all mappings are available to you.

### 3.2.17.1 Adding the CT Mapping File to Matter Management

To use SOP mapping, add the CT.xml file to Matter Management. This file is available on your installation media.

#### To add the CT Mapping file to Matter Management

1. In TeamConnect, go to the following location and create a new document folder named **Mapping Files**:  
**Root/System/Object Definitions/Service Of Process - CT**
2. On your installation media, in the data\* folder, locate the file named **Dispute for CT.xml** and copy it into the *Mapping Files* folder that you created.

#### 3.2.17.1.1 CT Mapping Files

Matter Management is mapped to CT, which enables a Dispute matter to be created based on the mapping files. Participants of the matters are created as Involved(s). The mapping file for CT is named Dispute for CT.

A created Dispute matter contains the following information:

From SOP Field (Type)	To Target Category	To Target Filed	Target Field Label	Target Field Type
Type or name of the lawsuit being served (Text)	DISP	name	Matter Name	Text
Lawsuit reason (Text)	DISP	MatterDescription DI	Matter Description	Text

#### Default Category = Lawsuit

**Note:** Date fields can not be populated when creating dispute matters because the CT source data is in a (Text) format and not a (Date) format as required by the dispute matter.

The following sections will create a contact record and add it to the client's dataset if a unique identifier is not defined. It will only create a record if a properly formatted contact record is manually created as described in the Service of Process Installation and Configuration Guide.

From SOP Field (Type)	To Target Category	To Target Filed	Target Field Label	Target Field Type
Court Contact (Contact)		.contact	System field	Contact

Case Number (Text)	INPA_COUR	DocketNumberDICO	Docket Number	Text
--------------------	-----------	------------------	---------------	------

**Involved Role = Court**

From SOP Field (Type)	To Target Category	To Target Filed	Target Field Label	Target Field Type
Plaintiff Contact (Contact)		.contact	System field	Contact

**Involved Role = Opposing Party**

From SOP Field (Type)	To Target Category	To Target Filed	Target Field Label	Target Field Type
Defendant Contact (Contact)		.contact	System field	Contact
Entity Contact (Contact)		.contact	System field	Contact

**Involved Role = Defendant**

**Note:** Only populated SOP records will create Involved(s) in the created Dispute matter. If both SOP records exist, the Dispute matter will contain two defendants

**3.2.17.2 Adding the CSC Mapping File to Matter Management**

To use SOP mapping, add the **Dispute for CSC.xml** file to Matter Management. This file is available on your installation media.

**To add the CSC Mapping file to Matter Management**

1. In TeamConnect, go to the following location and create a new document folder named Mapping Files:  
**Root/System/Object Definitions/Service Of Process - CSC**
2. On your installation media, in the data\* folder, locate the file named **Dispute for CSC.xml** and copy it into the Mapping Files folder that you created.

**3.2.17.2.1 CSC Mapping Files**

Matter Management is mapped to CSC, which enables a Dispute matter to be created based on the mapping files. Participants of the matters are created as Involved(s). The mapping file for CSC is named **Dispute for CSC**.



A created Dispute matter for CSC contains the following information:

From SOP Field (Type)	To Target Category	To Target Field	Target Field Label	Target Field Type
Name of the transaction (Text)	DISP	name	Matter Name	Text
The date in which the SOP was served (Date)	DISP_LWST	SuitServedDILW	Suit Served	Date
Answer due date (Date)	DISP_LWST	AnswerDueDILW	Answer Due	Date

#### Default Category = Lawsuit

The following sections will create a contact record and add it to the client's dataset if a unique identifier is not defined. It will only create a record if a properly formatted contact record is manually created as described in the Service of Process Installation and Configuration Guide.

From SOP Field (Type)	To Target Category	To Target Filed	Target Field Label	Target Field Type
Court Contact (Contact)		.contact	System field	Contact
Matter docket number (Text))	INPA_COU R	DocketNumberDI CO	Docket Number	Text

#### Involved Role = Court

From SOP Field (Type)	To Target Category	To Target Filed	Target Field Label	Target Field Type
Plaintiff Contact (Contact)		.contact	System field	Contact

#### Involved Role = Opposing Party

From SOP Field (Type)	To Target Category	To Target Filed	Target Field Label	Target Field Type
Sender Contact (Contact)		.contact	System field	Contact

#### Involved Role = Opposing Representative

From SOP Field (Type)	To Target Category	To Target Filed	Target Field Label	Target Field Type
Defendant Contact (Contact)		.contact	System field	Contact
Entity Contact (Contact)		.contact	System field	Contact

**Involved Role = Defendant**

*Note: Only populated SOP records create Involved(s) in the created Dispute Matter. If both SOP records exist, the Dispute matter contains two defendants.*

### 3.3 Collaborati Spend Management Customization Help

Welcome to the *TeamConnect® Collaborati Spend Management Customization Help*.

<a href="#">Invoice Validation Rules</a>	<a href="#">Rate Request Approval Rule</a>
<a href="#">AFA Activation and Rules</a>	CSM Settings

#### 3.3.1 Setup Overview

CSM is installed using the [Available Updates feature](#) of TeamConnect. Refer to the Installation Help for this version of CSM to learn more about the requirements for installing CSM.

After installation, and prior to configuring CSM, review the prerequisites and setup process to become familiar with the stages of CSM configuration.

##### 3.3.1.1 Prerequisites

The following steps include business and technical prerequisites:

1. Subscribe to CSM. For details, contact your Mitratesh account manager. CSM users may need to install Legal Matter Management prior to CSM installation, if desired (see below).
2. Provide your Mitratesh account manager with an email address at your organization to be used for notifications related to CSM setup.
3. Provide your Mitratesh account manager with a list of vendors to add to e-billing. The list must contain each vendor's country code, default currency, name, Tax Id, address, and a vendor contact person's email address so that vendors can receive email notifications related to Collaborati registration.

You can add additional vendors at any time by contacting your Mitratesh account manager.

4. Create a contact record in TeamConnect for each vendor. Verify that each vendor's contact has been added as [an involved party](#), in active status, to the appropriate matter records.
5. Before running CSM Synchronization, your TeamConnect developers must review any new or existing custom Invoice rules to avoid conflict with CSM behavior.

For example:

- If creation of an e-invoice in TeamConnect fails due to a rule triggered on Create, the invoice is rejected. For example, if a validation rule requires an (otherwise optional) invoice field to be populated, an e-invoice might not comply. In this case, the e-invoice will be auto-rejected for the Collaborati vendor with the **Rule Action** error message displayed in the **Comments to Vendor** field.
  - If creation of an e-invoice in TeamConnect fails due to a run-time error in a custom rule, the invoice is rejected. The CSM business admin will receive email notification about the error condition.
  - If an e-invoice has been created in TeamConnect but then fails to post due to a rule triggered on Update or Post, CSM will continue to attempt to auto-post the invoice in TeamConnect until the error is fixed (for example, a rule is fixed or turned off). The CSM business admin will receive email notification about the failure to post.
6. Verify that TeamConnect's task codes, expense codes, activity codes, and non-US tax categories that are used for CSM and Collaborati reflect the values you want your vendors to use in the invoices by doing the following:
    - a. Click the **Setup** link from the upper right area of TeamConnect.
    - b. Select **Object Definitions** from the **Go To** drop-down list.
    - c. Next to the **Invoice** object, click + and then click the **Line Item** object.  
The **Line Item** object definition screen is displayed.
    - d. Click the **Task Categories** tab.
    - e. Verify that the appropriate task codes are displayed in the **Task Code** column.
    - f. Click the **Expense Categories** tab.
    - g. Verify that the appropriate expense codes display in the **Expense Code** column.
    - h. From the TeamConnect **Admin** menu, select **Lookup Tables** from the **Go To** drop-down list.
    - i. On the **System** tab of the **Lookup Tables** screen, select **Activity Items** from the **Show items belonging to** drop-down list.
    - j. Verify that the appropriate activity codes are displayed in the **Activity Code** column.
    - k. For **Non-US Tax Categories**, return to the **Object Definitions** screen. Click **Invoice**.
    - l. Select the **Non-US Tax Categories** tab.
    - m. Verify that the appropriate Non-US Tax Categories are displayed in the Tax Code column. Note that **Tax Codes** will vary by the selected Non-US Tax Category node.
    - n. If you modify or add any **Task Codes**, **Expense Codes**, or **Activity Codes** use the Line Item Codes Mapping Tool.

7. Verify that CSM administrators (members of the CSM Admin user group who set up your CSM configuration) have sufficient security rights to view the necessary contact records.

For example, if a vendor's contact record has a security setting of **Private**, the CSM administrator will not have Read rights and cannot establish the necessary relationship between the contact record and the corresponding involved party record.

In this scenario, you could grant the **Super** user type to the CSM user, which includes Read permissions to private contacts. For details on granting rights, see the TeamConnect Administration Guide.

8. For new TeamConnect installations, determine the appropriate time zone to use for the **Date Received** field of invoice records. This field is time zone independent—it displays the date using TeamConnect's system time zone, not the end-user's time zone. For example, if TeamConnect's **Time Zone** field (on the **Admin Settings / Region** page) is set to **GMT** (the default), the **Date Received** field will display GMT time, even if the user's time zone preferences are set to **EST**.

For existing TeamConnect installations, it is recommended that you not change TeamConnect's **Time Zone** field on the system settings screen, because this may cause confusion regarding the time zone of original time stamp. That is, if you change the system **Time Zone** setting, the existing data is not updated.

### 3.3.1.2 Overview of the Initial Setup Process

1. Once you have signed up for CSM, a client record for your law department is created in Collaborati, and your designated contact (as identified in item 2 of [Prerequisites](#)) will receive the initial email notification containing a user ID and a password for connecting to Collaborati.
2. Your designated **CSM Admin** group members will need to fill in information on the following pages of the CSM Settings object:
  - **Connection** page
  - **General** page
  - **Default Rates** page
  - **E-billing** Roles page
3. Once the connection has been established, the vendor records for all of your vendors (step 4 of [Prerequisites](#)) will be created in Collaborati by Mitrtech, and pulled into your CSM settings. This information will enable CSM to pop up suggestions that help a member of your CSM Admin group to correctly map Collaborati vendor records to your vendor contacts.
  - For each authorized vendor, you can optionally restrict the billing codes that they would be able to use (by default, all your invoice line item categories will be available to all vendors).
  - You can deactivate any vendor at any time. Once deactivated, the vendor would not be able to continue submitting invoices electronically.
4. Once the vendor records are mapped and activated in TeamConnect, each vendor will receive a notification inviting them to register for Collaborati.

For each authorized vendor, you may be required to authorize new timekeeper records from Collaborati, mapping them to your existing contact cards (depending on how you have chosen to maintain timekeeper contact cards in TeamConnect). If this is the case, you will be notified once a vendor has registered and submitted timekeepers for your authorization.

5. You can optionally use the **Default Rates** tab to set up a structure to easily specify various levels of rates (based on task codes or timekeeper categories). You can also view expense unit prices.
  - The global-level default settings can be overridden by rates specification at the vendor level or further at the timekeeper level.
  - These rates will be used for invoice validation based on the out-of-the-box e-billing Rules (see [Viewing Invoice Validation Rule Settings](#)).
6. You can optionally use the **Payments** tab if your TeamConnect invoices are updated with payment information from your Accounts Payable system, and you would like to share the payment information with your vendors.
7. You can optionally activate any or all of the out-of-the-box e-billing rules that have been created based on most common billing guidelines. These rules provide validation against your established timekeeper rates, expense unit prices, as well as duplicate charges, unauthorized charges (for example, charges for work performed on a closed matter, as established in your guidelines), etc.
8. Budget request information is available in TeamConnect by clicking the **Budgets** link under the **Finance** tab in global navigation. You might also want to add budget request information to other parts of the user interface, such as the pages that display information about your legal matters. Doing so requires a few manual steps:
  - a. In **Documents**, navigate to **Top Level / System / Screens**.
  - b. Locate block file **BudgetRequestSearchMatter.xml** and copy this file.
  - c. Navigate to the **Screens** folder of whatever custom object definition represents your legal matters.
  - d. Paste the file into that folder.
  - e. In Designer, open that custom object definition and add the block file to one or more object views.

### 3.3.1.3 What Happens After Setup is Complete

The vendor-specific setting "Also share documents attached to the CSM Settings record with this vendor" is set to "Yes" for newly added vendors. However, you may have some existing vendors where that option is set to "No". If you wish that setting to be "Yes" for all vendors, an upgrade script is available for changing your CSM configuration. It must be run manually, with a SQL query tool, when logged on to the database with the TeamConnect username. The script can be found in the **Documents** page of TeamConnect at this path:

Top Level / System / Scripts

For MSSQL databases, run script **MSSQL\_upgrade\_CSM\_DocumentSharingOption.sql**

For Oracle databases, run script **Oracle\_upgrade\_CSM\_DocumentSharingOption.sql**

For more information about this setting, see [Sharing Documents with a Collaborati Vendor](#).

## First Sync

After completing setup, make sure you synchronize with Collaborati. See [Defining CSM Connection and Synchronization](#) for details.

## Invoices

Your vendors will immediately be able to start submitting invoices electronically. Once TeamConnect receives submitted invoices, they will be created and submitted for posting automatically, and will follow the same business processes you have established for invoices (for example, invoices will be automatically routed to the desired approval chain based on your approval rules for invoice posting).

**Note:** *You can optionally create separate workflow rules specific to electronic invoices; use the **isElectronic** field in the rule qualifier for that option. For details on establishing rule qualifiers, see the [Rules Configuration Guide](#).*

Once invoices are in the approval chain, they display the "Pending Client Approval" status for the respective vendors. Neither intermediate workflow steps (for example, multiple approval steps, ad-hoc reviews, etc.), nor internal correspondence is visible to vendors.

Rejected invoices show the "Rejected" status for the vendors. Approved and posted invoices show the "Approved" status and any adjustments and comments to vendors that were made during the approval process.

**Note:** *If you selected to share invoice payment information with your vendors, then paid invoices will show the "Paid" status for the vendors. Otherwise, invoice approval and rejection will be the final states communicated to your vendors. If an invoice is voided after posting, this will not be communicated automatically to your vendor but rather, the invoice will remain "Approved" in Collaborati. If an invoice is re-posted after rejection, this is not communicated automatically to your vendor and the invoice remains in Rejected status in Collaborati.*

## New vendors

Repeat steps 3-4 from the [initial setup process](#) to authorize new vendors.

To add new vendors to the list, contact your Mitratech account manager with this request.

### 3.3.1.4 Administer CSM Setup

Each user who requires access to CSM screens needs to be added to the **CSM Admin** group. Do this by following the instructions for maintaining [user groups](#).

### 3.3.1.5 Viewing Invoice Validation Rule Settings

Submitted invoices go through a comprehensive validation process based on your custom rules (if any) and the active invoice validation rules defined in the Invoice Validation Rule Setting screen.

**Note:** *There are four invoice types: Standard, Credit Note, Accrual, and Shadow. Invoice validation rules are enforced only against Standard invoices.*

You can deactivate any or all of the out-of-the-box invoice validation rules. Once the rules are activated, they will be active on all invoice updates and newly created invoices (the rules apply to both create and update actions).

**Caution:** Do not edit invoice validation rules from the Designer area Invoice object definition. Use the Invoice Validation Rule Setting Tool to disable or enable rules or edit conditional parameters.

For a complete description of available rules and the values to configure their settings, see [Adjusting the Invoice Validation Rule Settings](#).

**Note:** Invoice validation rules only apply to Standard type invoices by default. For further assistance, contact Mitratach Support.

#### 3.3.1.5.1 Accessing the Invoice Validation Rule Setting Screen

Members of the CSM Admin group have default access rights to the Invoice Validation Rule Setting Tool.

#### To access the Invoice Validation Rule Setting Screen

1. Click the **Setup** link.
2. From the **Tools** drop-down list, select **Invoice Validation Rule Setting**.

The **Invoice Validation Rule Setting** screen opens, allowing you to activate or deactivate the out-of-the-box invoice validation rules. For a complete description of available rules and the values to configure their settings, see [Adjusting the Invoice Validation Rule Settings](#).

#### 3.3.1.6 Activating the Rate Request Approval Rule

Rate requests are requests with proposed hourly rates that vendors submit through Collaborati. To be able to accept rates and approve requests, you must activate a rate request approval rule. You also use the approval rule to specify who can approve requests.

**Note:** Vendors can only send rates if rate collaboration is enabled. To enable rate collaboration, contact Mitratach Support.

See Applying Vendor Requested Rates for more information about receiving and approving rate requests.

#### To activate the rate request approval rule

1. In the Designer, from the **Go to** drop-down menu, select **Object Definitions**.
2. From the **Object Definition List**, navigate to **CSM Settings>Vendor>Rate Request**.
3. Click the **Rules** tab.

4. From the list of rules, click the **CSM - Rate Request Approval** link.
5. Click the **Create a Copy** button, and rename the rule if necessary.
6. Click the **Qualifier** tab. Create rule qualifiers to specify conditions for the rule. See [Creating Rule Qualifiers](#) for more information.
7. Click the **Action** tab.
8. Click the **View Route Details** link.
9. Click the **Create a Copy** button, and rename the route if necessary.
10. Create routes to specify who can approve requests. See the sections of [Creating Routes](#) for more information.
11. Return to the **Action** tab of the new rule. Select the new route from the **Approval route** field.
12. From the **Process Manager** field, select the group that approvers must be part of to approve rate requests. If you want approvers to be able to approve from **My Approvals**, select the **Workflow Process Manager** group.  
  
Make sure to add all approvers to the group you select.
13. From **Updates to record pending approval**, select one of the following fields:
  - **Allowed to approvers of current stop**—Allows rate request approvers to update rates of a request.
  - **Allowed to anyone**—Allows any user to update rates of a request.
14. Return to the **General** tab of the new rule, and place a checkmark in **This rule is Active** to activate the rule.

### 3.3.1.7 Activating Alternative Fee Arrangements and Rules

Instead of using hourly rates to determine an invoice amount, users can enter alternative fees for each vendor on a matter. From the **Alternative Fee Arrangement Settings** screen, you can activate the alternative fee arrangements that appear on the **AFA** page of a matter and determine how they work in CSM.

**Note:** Users can only enter alternative fee arrangements if the ability is enabled. To enable alternative fee arrangements, contact Mitratach Support.

#### To activate alternative fee arrangements and rules

1. Click the **Setup** link.
2. From the **Tools** drop-down list, select **Alternative Fee Arrangement Settings**. The **Alternative Fee Arrangement Settings** screen opens.

If you see a message that the AFAs are disabled, but you have enabled AFAs, sync your system with Collaborati. See Starting or Stopping CSM Synchronization with Collaborati for more information.



3. Make alternative fee arrangements active by placing a check-mark in the **This AFA is active** check-box. When you make an alternative fee arrangement active, the option for that alternative fee arrangement appears on the **AFA** page of the matter. Make rules for each arrangement active by placing check-marks in the **This Rule is Active** check-boxes. See the [Alternative Fee Arrangements and Rules table](#) for a description of rules.

***If you do not select these check-boxes for active alternative fee arrangements, even if the AFA is active check-box is selected, CSM will not enforce the AFA.***



If conditions for an active rule are met, the following results occur:

- A default warning message displays in the invoice with an explanation of the condition of the rule. Update the **Default Warning Message** to change the warning message.
  - The invoice automatically adjusts to the alternative fee if you have placed a checkmark in the **Auto-adjust...** check-box of the rule. If the rule is not set to auto-adjust, a message appears in the invoice about updating the invoice manually. See the [Alternative Fee Arrangements and Rules table](#) for an explanation of how the invoice automatically adjusts.  
  
If you use this check-box, select an **Adjustment Reason** so that, if the vendor receives a rejection, the invoice will include information about the reason for the rejection.
4. Select an option for the **Hourly Fee Tracking** field to specify whether or not the invoice tracks the hourly fee details.

***Important:*** Make sure you do not change this option after you have already saved the AFA settings. Changing the **Hourly Fee Tracking** field after TeamConnect starts receiving Collaborati invoices can cause issues with the invoice savings amount.

**Volume Discount** and **Shared Costs** do not have this option.

- **Always Disabled**—The invoice does not track hourly fee details.
  - **Always Enabled**—The **Tracking Method** field automatically appears on the matter with the tracking method option that you specify in the setup listed as the default.
  - **Selectable by Matter**—The **Hourly Fee Tracking** check-box appears with the matter so that the user can specify whether or not to track the hourly invoice fees.
5. Select an option for the **Tracking Method** field. The **Tracking Method** for some AFAs is already set.
    - **Fee Adjustment**—A standard invoice includes the original invoice fees with an adjustment to the alternative fee amount.
    - **Shadow Invoicing**—A shadow invoice includes the original invoice fees, while the alternative fee appears as the only amount on the standard invoice.
    - **Selectable by Matter**—The user can specify how to track hourly invoice fees.
  6. Click **Save AFA Settings**.

**Fixed Fee(Matter Based)**

This arrangement sets a sum of money to be paid for all fees in a matter.

This AFA is Active

**CSM(Validation) - Fee charges exceed agreed fixed fee**

This rule is Active

Default Warning Message:

Auto-adjust in-compliant charges to the agreed fee

Adjustment Reason

Hourly Fee Tracking

Tracking Method

Example of Active Alternative Fee Arrangement and Rule

Details on the available Alternative Fee Arrangements can be found below. For more information on each option, see Adding Alternative Fee Arrangements to a Matter.

**Alternative Fee Arrangements and Rules**

Alternative Fee Arrangement	Rule	Rule applies if...	When Auto-adjust has a check-mark, the invoice automatically...
<b>Fixed Fee (Matter Based)</b>	<b>Fee charges exceed agreed fixed fee</b>	The invoice fee amount exceeds the fixed fee amount.	Updates with the fixed fee amount.
<b>Fixed Fee (Milestone Based)</b>	<b>Milestone not completed</b>	No milestones have occurred.	Adjusts to an amount of zero.
	<b>Fees exceed completed milestone fee total</b>	The invoice fee amount exceeds the sum of the completed milestone fees.	Updates with the fixed fee amount.
<b>Fixed Fee (Time Based)</b>	<b>Invoice Date prior to agreed payment date</b>	The date on each invoice is earlier than the due date for that invoice. The due date for each invoice is the start date of the next interval	Adjusts to an amount of zero.

	<b>Invoice fee total does not equal Fixed Fee Payment Amount</b>	The invoice fee amount does not equal the total fixed fee amount.	Updates with the fixed fee payment amount.
<b>Capped Fee</b>	<b>Fees exceed Capped Fee Amount</b>	The invoice fee amount exceeds the capped fee amount.	Updates with the capped fee amount.
<b>Blended Hourly Rates</b>	<b>Line item rate exceeds Blended Hourly Rate</b>	The fee rate on an invoice exceeds the blended hourly rate.	Updates with the blended hourly rate.
<b>Contingency Fee</b>	<b>Contingency not achieved</b>	The contingency has not occurred.	Adjusts to an amount of zero.
	<b>Fees exceed Contingency Fee Amount</b>	The invoice fee amount exceeds the contingency fee.	Updates with the contingency fee amount.
<b>Volume Discount</b>	<b>Discount percent is less than negotiated discount percent</b>	The sum of all invoice fee amounts for a matter exceeds the threshold.	Updates with the volume discount percentage if the discount percentage on the invoice is less than the volume discount percentage.
<b>Shared Costs</b>	<b>Matter percent share for fees is incorrect</b>	The discount percentage on the invoice is higher than the discount of the shared cost (100% - percent share of matter fees).	Updates with the shared cost of fees.
	<b>Matter percent share for expenses is incorrect</b>	The discount percentage on the invoice is higher than the discount of the shared cost (100% - percent share of matter expenses).	Updates with the shared cost of expenses.

### 3.3.1.7.1 Updating Additional Rules for Alternative Fee Arrangements

After you enable alternative fee arrangements, you may need to make updates to a few additional rules.

#### **If you want TeamConnect to Reject Multi-Matter Invoices**

When you initially enable AFAs, if TeamConnect receives a multi-matter invoice and the **AFA Type** for one of the matters is set to **Volume Discount** or **Shared Costs**, TeamConnect rejects the invoice.

#### **To update TeamConnect to reject all multi-matter invoices**

1. From the **Tools** drop-down under the **Setup**, select **Invoice Validation Rule Setting**.
2. From the drop-down, select **Show All Rules**.
3. For the **CSM (Validation) - Do Not Allow Multi-Matter Invoice except Line Item Projects with Specific Project Categories** rule, select **This Rule is Active**.
4. Click **Save Rule Setting**.

#### **If You Have TeamConnect Legal**

If you have TeamConnect Legal and you enable AFAs, you must update the **AFA Invoice Posting Rule - Post - SYS** rule.

#### **To update TeamConnect Legal when you enable AFAs**

1. From the **Go to** drop-down under the **Setup**, select **Object Definitions**.
2. Click the **Invoice** object definition.
3. Click the **Rules** tab.
4. Click the **AFA Invoice Posting Rule - Post - SYS** rule.
5. Change the **Order** to a number higher than 1000.
6. Click **Save and Close**.
7. Log out of TeamConnect to update the changes.

### 3.3.1.7.2 Creating Alternative Fee Arrangement Pages

After you have activated alternative fee arrangements and their rules, you must configure the **AFA** page to appear.

The **AFA** page of a matter or project record is where users enter alternative fee information for each vendor associated with the matter or project.

**AFA Information**

**vendor3**

AFA Type: Fixed Fee (Matter Based) ▼

\*Fixed Fee Amount: 5,000.00

Total Fees (To Date): 0.00

Hourly fee Tracking

AFA Page with Fixed Fee (Matter Based) Example

**To create the AFA page**

1. From the **All** tab, click **CSM Settings**.
2. From **CSM Settings**, click the **E-Billing Roles** link in the left pane.
3. Click **Edit**.
4. Under **E-Billing Roles**, for each type of object record that you want to include the **AFA** page, specify the object if it is not currently specified. Do the following for each object:
  - Select **Involved** from the **Object Type** field.
  - Select the name of the object from the **Object** field.
  - Select the **Role** associated with the object.
  - Click **Add** to add the role to the list.

**E-Billing Roles**

Object Type: Involved ▼      Role: Third-Party Firm ▼

Object: Dispute ▼

**Add** **Clear**

Specifying that Alternative Fee Arrangements Can Appear for Disputes Example

5. Click the save drop-down, and select **Save & View**.
6. Sync your system with Collaborati. See Starting or Stopping CSM Synchronization with Collaborati for more information.
7. Click the **Setup** link.
8. From the **Go to** drop-down, select **Object Definitions**.
9. Click the object definition link for the matter or project where you want to add the **AFA** page.
10. Click the **Object Views** tab.

11. Click the link for the current object view.

To look up the current object view, select **Default Object Views** from the **System Settings** drop-down and look for the object name and the chosen object view.

12. In the **Object View** tab that opens, enter **AFA** in the **New** tab name field and click **add**.

13. From the **Add the block** drop-down, select **AFA Information** and click **add** next to the **Add the block** drop-down.

14. Click **Save & Close**.

### 3.3.2 Uninstalling CSM

***Note:** This section assumes that you installed and configured the tool that deletes/restores CSM. For more information, see [Creating Custom Tools](#).*

TeamConnect provides a tool that lets you delete your current CSM design. This is useful if you need to delete an older version of CSM before installing a newer one.

When you use this tool, the following CSM items are deleted:

- CSM settings
- Vendors
- Timekeepers
- CSM object definitions

Non-CSM records, such as Contacts and Invoices, are not deleted if they were created by CSM.

Additionally, all versions of CSM that are listed on the **About/Version** tab are removed and a new entry appears that shows that the delete tool was used to delete the last CSM design.

***Note:** While the tool is running, all rules are automatically disabled, and automatically enabled when the uninstall is complete. It is recommended that you perform this task during off-peak hours.*

You can also use this tool to restore CSM default data. See [Restoring CSM Default Data](#) for more information.

### How Uninstalling CSM Affects your Client Account in Collaborati

When you open the tool, if an active connection exists between CSM and Collaborati, a warning message appears to inform you of the following:

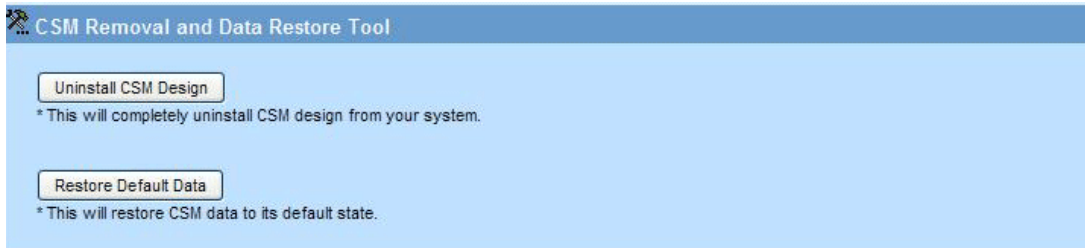
- Uninstalling CSM permanently disables the client account in Collaborati.
- After the uninstall, you must request a new client account if you wish to reinstall CSM.

Terminating the corresponding Collaborati client account prevents users from reusing the same client account on Collaborati to synchronize with a different CSM design.

**Important:** Before you uninstall CSM, you must turn off synchronization between CSM and Collaborati. See *Starting or Stopping CSM Synchronization with Collaborati* for more information.

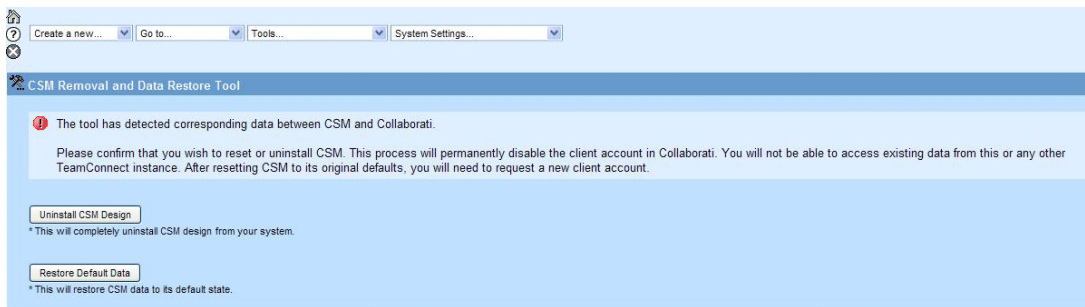
## To uninstall CSM

1. In the TeamConnect Designer, click **Tools**, and then select the CSM delete/restore tool.  
The tool's main screen opens.



**Delete/Restore Tool Main Screen**

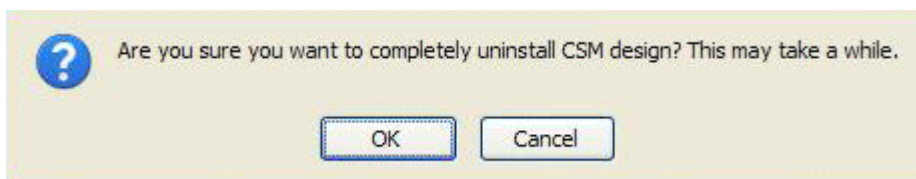
If the tool detects an active connection between CSM and Collaborati, a warning message appears. This is an informational message and will not prevent the uninstall of CSM.



**Warning Message Detecting Active Connection Between CSM and Collaborati**

2. To continue, click **Uninstall CSM Design**.

A confirmation message opens.



**Uninstall Confirmation Message**

3. Click **OK** to begin uninstalling CSM.

When the uninstall is complete, a confirmation message appears.



**Uninstall Completed Message**

4. Click **OK** to close the message.

### 3.3.2.1 Restoring CSM Default Data

**Note:** This section assumes that you installed and configured the tool that deletes/restores CSM default data. For more information, see [Creating Custom Tools](#).

If you need to drop your CSM settings, but you do not need to upgrade to a later version, use the restore tool to quickly restore default CSM data without removing the CSM design.

Once the default data is restored, you can go to CSM and re-associate vendors. See [Authorizing Vendors](#) for more information.

When you use this tool, the following CSM items are deleted:

- CSM settings
- Vendors
- Timekeepers
- CSM object definitions

Non-CSM records, such as Contacts and Invoices, are not deleted if they were created by CSM.

You can also use this tool to uninstall CSM. See [Uninstalling CSM](#) for more information.

### How Restoring CSM Affects your Client Account in Collaborati

When you open the tool, if an active connection exists between CSM and Collaborati, a warning message appears to inform you about the following:

- Restoring default CSM data permanently disables the client account in Collaborati.
- After you reinstall the default CSM data, you must request a new client account.

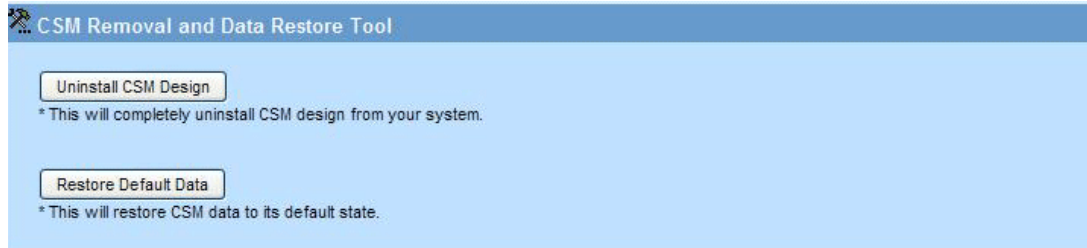
**Important:** Before you restore CSM default data, you must turn off synchronization between CSM and Collaborati. See [Starting or Stopping CSM Synchronization with Collaborati](#) for more information.

**Note:** While the tool is running, all rules are automatically disabled, and automatically enabled when the uninstall is complete. It is recommended that you perform this task during off-peak hours.

### To restore default CSM data

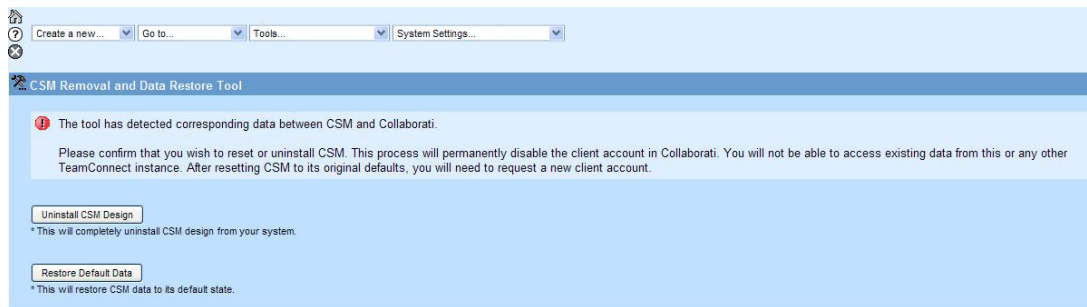


1. In the TeamConnect Designer, click **Tools**, and then select the CSM delete/restore tool.  
The tool's main screen opens.



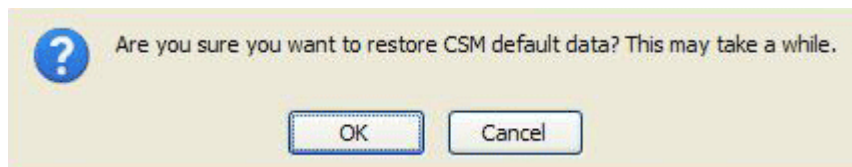
**Delete/Restore Tool Main Screen**

If the tool detects an active connection between CSM and Collaborati, a warning message appears. This is an informational message and will not prevent the tool from restoring CSM data.



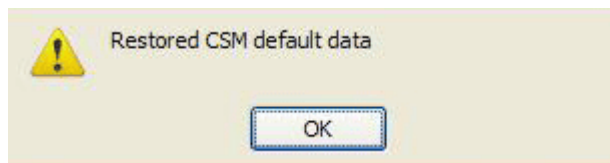
**Warning Message Detecting Active Connection Between CSM and Collaborati**

2. To continue, click **Restore Default Data**.  
A confirmation message opens.



**Restore Confirmation Message**

3. Click **OK** to begin restoring CSM's default data.  
When the data is restored, a confirmation message appears.



**Restore Completed Message**

4. Click **OK** to close the message.

### 3.3.3 Adjusting the Invoice Validation Rule Settings

See the following sections for specific rule information:

**Caution:** Do not edit invoice validation rules.

- [Fee Charges Exceed Agreed Rates](#)
- [Expense Charges Exceed Acceptable Unit Costs](#)
- [Do Not Allow Multi-Matter Invoices](#)
- [Timekeeper Billed Too Much Time](#)
- [Invoice Service/Expense Charges are Too Old](#)
- [Invoice is Too Old](#)
- [Invoice Contains Duplicate Timekeeper Charge](#)
- [Invoice Contains Duplicate Expense](#)
- [Invoice Line Item Description Contains Unauthorized Keywords](#)
- [Invoice currency doesn't match vendor currency](#)

**Note:** This section describes rules defined in the invoice object definition.

#### 3.3.3.1 Fee Charges Exceed Agreed Rates

**Fee charges exceed agreed rates**

---

This Rule is Active

Warning Message (include %s to display the specified rate per hour)

Timekeeper rate exceeds the negotiated rate of %s /hr

Auto-adjust in compliant charges to the agreed rate

#### Fee Charges Exceed Agreed Rates Rule Settings

#### Concept:

If rates for timekeepers are set at any level, this rule ensures that all fee charges on an invoice are less than or equal to a timekeeper's set rates for the appropriate time period. This rule will validate that charges do not exceed the "lowest" level rates available for each timekeeper. Rates for timekeepers can be defined at various levels (timekeeper Contact, Vendor, or Global).

Timekeeper rates can be defined at various levels in TeamConnect and CSM. This rule looks for the timekeeper rate in the following order of priority (if the rate is not defined or set to 0, the rule will use the next timekeeper rate definition):

1. Timekeeper Invoice Task Rate (from TeamConnect contact, Rates page) where the rate for a specific line item task code would be used first. If not found, the rate for the parent of a specific line item task code would be used\*.
2. Timekeeper Default Rate (from TeamConnect contact, Rates page).

3. Vendor's rate (for associated timekeepers; from CSM Vendor account, Rates page). Depending on settings, the rate for a specific line item task code would be used first. If not found, the rate for the parent of a specific line item task code would be used. Otherwise, the rate for a timekeeper category would be used.
4. CSM Settings Default rate (for timekeepers; from CSM Settings, Default Rates page). Depending on settings, the rate for a specific line item task code would be used first. If not found, the rate for the parent of a specific line item task code would be used. Otherwise, the rate for a timekeeper category would be used.

**Note:** For this rule description, the timekeeper's default currency is defined from the TeamConnect contact record, Rates page, Default Currency.

This validation rule is triggered on invoice creation, and on invoice update.

If an invoice is submitted in a currency that is different from the timekeeper's default currency, this rule will not apply to the invoice.

**Note:** This is an optional rule; do not set this rule if you do not wish to validate fee charges.

### Field Descriptions:

- **This Rule is Active**—Select this check box to activate the rule. If fee charges exceed agreed rates on invoice line item creation/update, the rule will be triggered. This means that a warning message will be inserted into each incompliant line item charge, unless the **Auto-adjust incompliant charges** option is checked.
- **Warning Message**—Type a warning message to display when this rule is triggered. If the fee charge exceeds the timekeeper's lowest level rate, this warning will be inserted into the line item. The default warning message is: "Timekeeper rate exceeds the negotiated rate of %s /hr", where "%s" is used to display the specified rate per hour.
- **Auto-adjust incompliant charges**—If this option is checked, the fee line item that exceeds your company's specified rate will be auto-adjusted to timekeepers lowest level available rate. The adjustment reason will be defined in the rule parameter "Adjustment reason", if one exists. The **Adjusted By** value will be *System*.

### 3.3.3.2 Expense Charges Exceed Acceptable Unit Costs

#### Concept:

**Expense charges exceed acceptable unit costs**

---

This Rule is Active

Warning Message (include %s to display the specified unit cost)

Expense exceeds the maximum allowed unit cost of %s /unit

Auto-adjust incompliant charges to the maximum allowed unit cost

#### Expense Charges Exceed Acceptable Unit Costs Rule Settings

This rule ensures that all expense charges on the invoice are less than or equal to your company's specified unit costs for the appropriate time period. This validation rule is triggered on invoice creation and on invoice update.

If the submitted invoice's currency is different from the system Default Currency, this rule will not perform any validation on the invoice.

**Note:** This is an optional rule; do not set this rule if you do not wish to validate expenses for all vendors.

### Field Descriptions:

- **This Rule is Active**—Select this check box to activate the rule. If expense charges exceed acceptable unit costs on invoice line item creation/update, for each expense line item of the triggered invoice, the rule will be triggered. This means that a warning message will be inserted into each in-compliant line item charge, unless the **Auto-adjust in-compliant charges to the maximum allowed unit cost** option is checked.
- **Warning Message**—Type a warning message to display when this rule is triggered. If the expense unit price of a line item exceeds your company's specified unit costs (defined on the Invoice Line Item object definition, **Expense Codes** tab), this warning will be inserted into the line item. The default warning message is: "Expense exceeds the maximum allowed unit cost of %s /unit", where "%s" is used to display the specified unit cost.
- **Auto-adjust in-compliant charges to the maximum allowed unit cost**—If this option is checked, the expense line item unit price that exceeds your company's specified unit costs will be auto-adjusted to the maximum allowed unit cost. The adjustment reason will be defined in the rule parameter "Adjustment reason", if one exists. The **Adjusted By** value will be *System*.

#### 3.3.3.3 Do Not Allow Multi-Matter Invoices

**Do Not Allow Multi-Matter Invoice except Line Item Projects with Specified Project Categories**

---

This Rule is Active

Rejection Reason

Project Categories

(None Selected)

Auto

Book

Collaborati Settings

::Notification Settings

::Connection Settings

::Synchronization Settings

::Rates

Synchronization History

Vendor

Do Not Allow Multi-Matter Invoices Rule Settings

### Concept:

This rule ensures that all invoices submitted for post are either single-matter invoices, or multi-matter invoices that belong to specific project categories for the appropriate time period. This validation rule is triggered on invoice creation (for manual invoices).

**Note:** This is an optional rule; do not set this rule if your company allows multi-matter invoices, or if you wish to allow vendors to submit multi-matter invoices under specified conditions.

### Field Descriptions:

- **This Rule is Active**—Select this check box to activate the rule. If an invoice is created with multiple matters, all projects of the invoice will be checked against project categories that accept multiple matters for all line items of the triggered invoice. Incompliant invoices will be automatically rejected.
- **Rejection Reason**—Type a warning message to display when this rule is triggered (manual invoices only). If the triggered invoice has multiple projects and the triggered invoice line item has a project with categories NOT defined in the **Project Categories** parameter, the invoice will be rejected, and this warning will display when the user attempts to save the invoice. The default warning message is: "Multi-matter invoices are not accepted."
- **Project Categories**—Select one or multiple project categories that allow multiple matters to be submitted on an individual invoice.

#### 3.3.3.4 Timekeeper Billed Too Much Time

**CSM (Validation) - Timekeeper billed too much time**

---

This Rule is Active

Maximum allowed hours billed per day

Warning Message (include %s to display the specified time in hours)

Auto-adjust incompliant time to match the maximum allowed time

Check Across All Invoices

Timekeeper Billed Too Much Time Rule Settings

### Concept:

This rule ensures that all submitted timekeeper time entries on an invoice (or across invoices, depending on your settings) are less than or equal to your company's specified maximum billable hours for timekeepers for the same service date and for the appropriate time period. This validation rule is triggered on invoice creation and on invoice update (for manual invoices).

**Note:** This is an optional rule; do not set this rule if your company does not require this type of validation.

### Field Descriptions:

- **This Rule is Active**—Select this check box to activate the rule. The rule is triggered if an invoice is created with timekeeper time entries that exceed the client-specified maximum amount of time for a single service date. This means that a warning message will be inserted into non-compliant line item charges, unless the **Auto-adjust non-compliant time to match the maximum allowed time** option is checked.
- **Maximum allowed hours billed per day**—Enter the maximum number of hours that can be billed by a timekeeper each day. The default value is 8. This must be a positive, non-zero value.
- **Warning Message**—Type a warning message to display when this rule is triggered. If a line item contains fee charges exceeding the specified time limit, this warning will be inserted into the invoice line item. If you specified the Check Across All Invoices option, all invoices for the same timekeeper and service date will be combined and checked against the limit. The default warning message is:  
  
"Timekeeper daily time entry exceeds the maximum allowed time of %s hrs.", where "%s" is used to display the specified time in hours.  
  
For example, Timekeeper daily time entry exceeds the maximum allowed time of 8 hrs on:  
current invoice (line item 1, 2, 3, ... ), INV-101 (line item 1), INV-121 (line item 2,12), INV-129 (line item 20), INV-131 (line item 11),...
- **Auto-adjust non-compliant time to match the maximum allowed time**—If this option is checked, the vendor's submitted time that exceeds your company's specified time limit will be auto-adjusted to your company's maximum allowed time. The adjustment reason will be defined in the rule parameter "Adjustment reason", if one exists. The **Adjusted By** value will be *System*.
- **Check Across All Invoices**—Select this check box to enforce this rule on new invoice creation or invoice update against all existing invoice fee line items (for all non-rejected invoices). By default, this check box is cleared.

### Sample Scenario One

For example, if the **Check Across All Invoices** check box is selected and the Auto-adjust non-compliant time to match the maximum allowed time check box is cleared then the following would apply:

Assume that the invoice validation rule field, Maximum allowed hours billed per day = 8, and an invoice already exists with a fee line item for Timekeeper (John D.), with service date (03/17/2009), and Units (3 hours).

If you create an invoice that contains a fee line item for Timekeeper (John D.), with service date (03/17/2009), and Units (7 hours), the invoice validation rule would be triggered and the sum of billed hours (3 + 7 = 10) would cause the new invoice to be rejected.

### Sample Scenario Two

If both **Check Across All Invoices** and Auto-adjust non-compliant time to match the maximum allowed time check boxes are selected then the following would apply:

Assume that the invoice validation rule field, Maximum allowed hours billed per day = 8, and an invoice already exists with a fee line item for Timekeeper (John D.), with service date (03/17/2009), and Units (3 hours).

If you create an invoice that contains two fee line items for Timekeeper (John D.), with service date (03/17/2009):

- Fee line item 1: Units value is 2 hours
- Fee line item 2: Units value is 5 hours

Then the invoice validation rule would be triggered and the sum of billed hours ( $3 + 2 + 5 = 10$ ) would cause the auto-adjust condition to be applied so that the new invoice's Fee line item 2, Unit value would be auto-adjusted to 3 hours. The resulting sum of billed hours ( $3 + 2 + 3 = 8$ ) would then comply with the invoice validation rule.

The following will occur:

- Under the new invoice's fee line item 2, a warning will display, like Timekeeper daily time entry exceeds the maximum allowed time of 8 hrs on: current invoice (line item 1, 2), INV-101 (line item 1).

Warnings resulting from breaking this invoice validation rule only display on the line items of invoices you are currently trying to create or update. The warning message will list pre-existing invoice fee line items with the same Timekeeper and service date, that were used to calculate the total hours billed by a Timekeeper.

- The line item detail field for the new invoice's, fee line item 2 will display a Adjusted icon, indicating that the line item was adjusted.

**Note:** If you select both the Auto-adjust in compliant time to match the maximum allowed time and **Check Across All Invoices** check boxes, and if you create/update an invoice that contains fee line items that would break the current conditions of this rule, the auto-adjustment would only apply to the fee line items of a new invoice or invoice you are updating and not be applied to any fee line items of previously created/saved invoices.

### 3.3.3.5 Invoice Service/Expense Charges are Too Old

**Invoice service/expense charges are too old**

---

This Rule is Active

Maximum number of days between the charge date and the invoice date

Warning Message (include %d to display the calculated difference in days and %s to display the specified maximum days)

Auto-adjust in compliant charges to zero

#### Invoice Service/Expense Charges are Too Old Rule Settings

### Concept:

This rule ensures that invoices do not contain service or expense charges (i.e., line items) that are on dates that fall after your company's maximum number of days between the service date and the invoice date. For example, a company may not accept an invoice that contains service dates that are more than 180 days old. This validation rule is triggered on invoice creation and on invoice update (for manual invoices) and settings must be valid within the appropriate time period.

**Note:** This is an optional rule; do not set this rule if your company does not require this type of validation.

### Field Descriptions:

- **This Rule is Active**—Select this check box to activate the rule. If an invoice is created with service/expense dates on the invoice (i.e., line item dates) that exceed the client-specified maximum number of days between the service date and the invoice date, the rule will be triggered. This means that a warning message will be inserted into each incompliant line item charge, unless the **Auto-adjust incompliant charges to zero** option is checked.
- **Maximum number of days between the charge date and the invoice date**—Enter the number of days that your company will allow between an invoice date and an individual charge date. The default is 180 days. This must be a positive, non-zero value.
- **Warning Message**—Type a warning message to display when this rule is triggered. If a line item contains service/expense dates exceeding the specified time, this warning will be inserted into the invoice line item. The default warning message is: "Charge service or expense date is more than %d days old. Allowed maximum number of days between the charge date and the invoice date is %s days.", where "%d" is used to display the calculated difference in days and "%s" is used to display the specified maximum days.
- **Auto-adjust incompliant charges to zero**—If this option is checked, incompliant charges will be adjusted to zero. In this case, the information about this adjustment along with the adjustment reason (which should be selected from the Adjustment Reasons lookup table) will be recorded in the adjustment history. The **Adjusted By** value will be *System*.

#### 3.3.3.6 Invoice is Too Old

**Invoice is too old**

---

This Rule is Active

Maximum number of days between the invoice date and the invoice submitted date

Rejection Reason

Invoice is Too Old Rule Settings

### Concept:

This rule ensures that the calculated difference between the invoice date and the invoice creation date is less than or equal to your company's maximum number of days in which vendors are allowed to submit charges for payment. For example, a company may not accept an invoice with an invoice date that is more than 30 days from the current date. This validation rule is triggered on invoice creation and on invoice update (for manual invoices) and settings must be valid within the appropriate time period.

**Note:** *This is an optional rule; do not set this rule if your company does not require this type of validation.*

### Field Descriptions:



- **This Rule is Active**—Select this check box to activate the rule. If an invoice is submitted with an invoice date that is more than the client-specified maximum number of days between the invoice date and the current date, the rule will be triggered. Incompliant invoices will be automatically rejected.
- **Maximum number of days between the invoice date and the invoice submitted date**—Enter the number of days that your company will allow between an invoice date and the invoice submitted date. The default is 30 days. This must be a positive, non-zero value.
- **Rejection Reason**—Type a warning message to display when this rule is triggered (manual invoices only). If the invoice is submitted after the maximum number of allowed dates, this warning will appear when the user attempts to save the invoice. The default warning message is: "Invoice cannot be accepted as past due."

### 3.3.3.7 Invoice Contains Duplicate Timekeeper Charge

**CSM (Validation) - Invoice contains duplicate Timekeeper charge**

---

This Rule is Active

A duplicate fee line item charge criteria

Project

Line Item Service Date

Timekeeper

Line Item Task Code

Line Item Quantity

Warning Message (include %s to display the first line item number that contain the same charge)

Potential duplicate charge on :

Auto-adjust incompliant charges to zero

Check Across All Invoices

Invoice Contains Duplicate Timekeeper Charge Rule Settings

#### Concept:

This rule ensures that vendors do not charge more than once for the same work done on a matter for the appropriate time period. This validation rule is triggered on invoice creation and on invoice update (for manual invoices).

**Note:** This is an optional validation rule; do not set this rule if your company does not require this type of validation.

#### Field Descriptions:

- **This Rule is Active**—Select this check box to activate the rule. If two fee line items contain a duplicate charge, the rule will be triggered. This means that a warning message will be inserted into each incompliant line item charge, unless the **Auto-adjust incompliant charges to zero** option is checked.

- **A duplicate fee line item charge criteria**—Select the check boxes next to the items that may not have another duplicate fee item charge in the same invoice.
  - Project
  - Line Item Service Date
  - Timekeeper
  - Line Item Task Code
  - Line Item Quantity
- **Warning Message**—Type a warning message to display when this rule is triggered. If the rule is applied to validate only one invoice, a line item contains a duplicate fee item charge, the second and every duplicate line item after will receive this warning. The default warning message is either:
  - "Potential duplicate charge, see line item %s", where "%s" is used to display the duplicated line item number.
  - Potential duplicate charge on: <invoice number or "current invoice"> (line item <line item number>)(line item 1, 2, 3 ...), ...  
  
For example, Potential duplicate charge on: current invoice (line item 1, 2, 3, ... ), INV-101 (line item 1, 3), INV-121 (line item 1, 2), INV-129 (line item 20), INV-131 (line item 11, 14), ...  
  
In the resulting warning message "... " indicates either more than 3 line items of an invoice or more than 5 invoices break the validation rule.
- **Auto-adjust in compliant charges to zero**—If this option is checked, in compliant charges will be adjusted to zero. In this case, the information about this adjustment along with the adjustment reason (which should be selected from the Adjustment Reasons lookup table) will be recorded in the adjustment history. The **Adjusted By** value will be *System*.
- **Check Across All Invoices**—Select this check box to enforce this rule against all existing invoice fee line items (for non-rejected invoices). By default, this check box is cleared.

### Sample Scenario One

For example, if the **Check Across All Invoices** check box is selected and the Auto-adjust in compliant charges to zero check box is cleared then the following would apply:

Assume that all Duplicate Fee Line Item Charge Criteria are selected (Project, Line Item Service Date, Timekeeper, Line Item Task Code, Line Item Quantity). Also assume an invoice already exists with a fee line item for project (Accident Claim 20-304), with service date (03/17/2009), Timekeeper (John D.), Task Code (Accident), and Units/Quantity (3 hours).

If you create an invoice that contains a fee line item for project (Accident Claim 20-304), with service date (03/17/2009), Timekeeper (John D.), Task Code (Accident), and Units/Quantity (3 hours), the invoice validation rule would be triggered and the duplicate values in the Duplicate Fee Line Item Charge Criteria fields would cause the new invoice to be rejected.

A warning would be added to the fee line item of the new invoice.

### Sample Scenario Two

If both **Check Across All Invoices** and Auto-adjust incompilant charges to zero check boxes are selected then the following would apply:

Assume that all Duplicate Fee Line Item Charge Criteria are selected (Project, Line Item Service Date, Timekeeper, Line Item Task Code, Line Item Quantity). Also assume an invoice already exists with a fee line item for project (Accident Claim 20-304), with service date (03/17/2009), Timekeeper (John D.), Task Code (Accident), and Units/Quantity (3 hours).

If you create an invoice that contains a fee line item for project (Accident Claim 20-304), with service date (03/17/2009), Timekeeper (John D.), Task Code (Accident), and Units/Quantity (3 hours), the invoice validation rule would be triggered and the new invoice's fee line item Units/Quantity value would be auto-adjusted to 0 hours.

A warning would be added to the fee line item of the new invoice. The Warnings column for the new invoice, fee line item would display a green check mark, indicating that the line item was adjusted.

**Note:** *Warnings resulting from breaking this invoice validation rule only display on the line items of invoices you are currently trying to create or update. The warning message will list pre-existing invoice fee line items that meet the invoice validation rule's current Duplicate fee line item charge criteria.*

**Note:** *If you select both the Auto-adjust incompilant charges to zero and **Check Across All Invoices** check boxes, and if you create (or update) an invoice that contains fee line items that would break the current conditions of this rule, the auto-adjustment would only apply to the fee line items of the new/updated invoice and not be applied to any fee line items of previously saved invoices.*

### 3.3.3.8 Invoice Contains Duplicate Expense

**CSM (Validation) - Invoice contains duplicate expense**

This Rule is Active

A duplicate expense line item charge criteria

Project

Line Item Service Date

Line Item Expense Code

Line Item Quantity

Warning Message (include %s to display the first line item number that contain the same charge)

Potential duplicate expense on :

Auto-adjust incompilant charges to zero

Check Across All Invoices

Invoice Contains Duplicate Expense Rule Settings

#### Concept:

This rule ensures that vendors do not charge more than once for the same expense on a matter for the appropriate time period. This validation rule is triggered on invoice creation and on invoice update (for manual invoices).

**Note:** This is an optional validation rule; do not set this rule if your company does not require this type of validation.

### Field Descriptions:

- **This Rule is Active**—Select this check box to activate the rule. If two expense line items contain a duplicate charge, the rule will be triggered. This means that a warning message will be inserted into each incompliant expense line item, unless the **Auto-adjust incompliant charges to zero** option is checked.
- **A duplicate fee line item charge criteria**—Select the check boxes next to the items that may not have another duplicate expense item charge in the same invoice.
  - Project
  - Line Item Service Date
  - Line Item Expense Code
  - Line Item Quantity
- **Warning Message**—Type a warning message to display when this rule is triggered. If a line item contains a duplicate expense item charge, the second and every duplicate line item after that will receive this warning. The default warning message is either:
  - "Potential duplicate expense, see line item %s", where "%s" is used to display the duplicated line item number.
  - Potential duplicate expense on: <invoice number or "current invoice"> (line item <line item number>)(line item 1, 2, 3 ...), ...  
  
For example, Potential duplicate expense on: current invoice (line item 1, 2), INV-101 (line item 1, 3, 19, ... ), INV-121 (line item 2), INV-129 (line item 20), INV-131 (line item 14), ...
- **Auto-adjust incompliant charges to zero**—If this option is checked, incompliant charges will be adjusted to zero. In this case, the information about this adjustment along with the adjustment reason (which should be selected from the Adjustment Reasons lookup table) will be recorded in the adjustment history. The **Adjusted By** value will be *System*.
- **Check Across All Invoices**—Select this check box to enforce this rule against all existing invoice expense line items (for non-rejected invoices). By default, this check box is cleared.

### Sample Scenario One

For example, if the **Check Across All Invoices** check box is selected and the **Auto-adjust incompliant charges to zero** check box is cleared then the following would apply:

Assume that all Duplicate Expense Line Item Charge Criteria are selected (Project, Line Item Service Date, Line Item Expense Code, Line Item Quantity). Also assume an invoice already exists with a expense line item for project (Accident Claim 20-304), with service date (03/17/2009), Expense Code (Photocopy), and Units/Quantity (50,000).

If you create an invoice that contains an expense line item for project (Accident Claim 20-304), with service date (03/17/2009), Expense Code (Photocopy), and Units/Quantity (50,000), the invoice

validation rule would be triggered and the duplicate values in the Duplicate Expense Line Item Charge Criteria fields would cause the new invoice to be rejected.

A warning would be added to the expense line item of the new invoice.

## Sample Scenario Two

If both **Check Across All Invoices** and **Auto-adjust noncompliant charges to zero** check boxes are selected then the following would apply:

Assume that all Duplicate Expense Line Item Charge Criteria are selected (Project, Line Item Service Date, Line Item Expense Code, Line Item Quantity). Also assume an invoice already exists with an expense line item for project (Accident Claim 20-304), with service date (03/17/2009), Expense Code (Photocopy), and Units/Quantity (50,000).

If you create an invoice that contains an expense line item for project (Accident Claim 20-304), with service date (03/17/2009), Expense Code (Photocopy), and Units/Quantity (50,000), the invoice validation rule would be triggered and the new invoice's expense line item Units/Quantity value would be auto-adjusted to 0 units.

A warning would be added to the expense line item of the new invoice. The Warnings column for the new invoice, expense line item would display a green check mark, indicating that the line item was adjusted.

**Note:** Warnings resulting from breaking this invoice validation rule only display on the line items of invoices you are currently trying to create or update. The warning message will list pre-existing invoice expense line items that meet the invoice validation rule's current **Duplicate expense line item charge criteria**.

**Note:** If you select both the **Auto-adjust noncompliant charges to zero** and **Check Across All Invoices** check boxes, and if you create (or update) an invoice that contains expense line items that would break the current conditions of this rule, the auto-adjustment would only apply to the expense line items of the new/updated invoice and not be applied to any expense line items of previously saved invoices.

### 3.3.3.9 Invoice Line Item Description Contains Unauthorized Keywords

**Invoice line item description contains keywords**

---

This Rule is Active

Disallowed Keywords (Type in key words or word combinations for disallowed charges, separating each by a comma)

support call help

Warning Message (include %s to display the specified keywords)

Line item description contains a charge matching disallowed description: %s

Auto-adjust noncompliant charges to zero

Invoice Line Item Description Contains Unauthorized Keywords Rule Settings

## Concept:

This rule ensures that the invoice line item descriptions do not include certain client-specified keywords, for example, "support, help" during the appropriate time period. This validation rule is triggered on invoice creation and on invoice update (for manual invoices). The search for Disallowed Keywords is not case-sensitive and is performed as a wildcard search.

*Note: This is an optional validation rule; do not set this rule if your company does not require this type of validation.*

### Field Descriptions:

- **This Rule is Active**—Select this check box to activate the rule. If any of the client-specified keywords are found in the line item descriptions, the rule will be triggered. This means that a warning message will be inserted into each noncompliant line item, unless the **Auto-adjust noncompliant charges to zero** option is checked.
- **Disallowed Keywords**—Type in any keywords or word combinations that are not allowed in the line item descriptions, separating each by a comma. For example, if you type "one" then the following invoice descriptions would trigger warnings: "ONE", "TONE", "ToNeS", "Phone". Another example, if you would like to get a warning for any line items that reference "library books", then it would be recommended to type "book" rather than "library books" in order to match on all combinations such as "books from library", "a library loaned book", etc.
- **Warning Message**—Type a warning message to display when this rule is triggered. If a line item contains such words, this warning will be inserted into the line item details. The default warning message is: "Line item description contains a charge matching disallowed description: %s", where "%s" is used to display the unauthorized key words.
- **Auto-adjust noncompliant charges to zero**—If this option is checked, noncompliant line items will be adjusted to zero. In this case, the information about this adjustment along with the adjustment reason (which should be selected from the Adjustment Reasons lookup table) will be recorded in the adjustment history. The **Adjusted By** value will be *System*.

#### 3.3.3.10 Invoice currency doesn't match vendor currency

### Concept:

If this rule is enabled and the invoice currency is different from the vendor's default currency (from the TeamConnect contact record, Rates page, Default Currency), the invoice will be rejected. If the vendor's contact default currency is not defined, this rule will compare the invoice currency against the TeamConnect system default currency. This validation rule is triggered on invoice creation and on invoice update (for manual invoices).

You should not enable this rule if you expect your vendors to submit invoices in multiple currencies.

### Field Descriptions:

- **Warning Message**—Type a warning message to display when this rule is triggered. If a line item contains such words, this warning will be inserted into the line item details. The default warning message is: "Invoice currency <SUBMITTED CURRENCY CODE> does not match our records for your company: <VENDOR CURRENCY CODE>. Please contact your client's billing coordinator."

- **This Rule is Active**—Select this check box to activate the rule. By default, this rule is not active.

### 3.3.4 Frequently Asked Questions and Troubleshooting

This section includes additional general information about CSM and tips for troubleshooting CSM.

#### 3.3.4.1 Frequently Asked Questions

##### How do I enter Non-US Tax Categories?

Detailed instructions for editing Non-US Tax Categories are available in section "Defining Non-US Tax Categories" in the *Customization Guide*.

##### What kind of email notifications will I receive from Collaborati?

For clients (TeamConnect/CSM customers), the following notifications are sent:

- A welcome email with the user ID and password to connect with Collaborati.
- Email notification when a vendor(s) is added in Collaborati and associated with your (client) company.
- Email notification when a timekeeper(s) is added in Collaborati for a vendor that is associated with your (client) company. Note that if the setting, **Automatically create new Contact records with Collaborati timekeeper information**, is enabled (check) you will not receive this email notification because you will not be required to respond to the timekeeper addition.
- Email notifications will be sent to your company's Business Administrator (email address set on the CSM Settings Connection page) for the following:
  - If creation of an e-invoice in TeamConnect fails due to a run-time error in a custom rule, and the invoice is rejected.
  - If an e-invoice has been created in TeamConnect but then fails to post due to a rule triggered on Update or Post.
- Email notifications will be sent to your company's Technical Administrator (email address set on the CSM Settings Connection page) when your CSM ID (User ID and Password) is changed (on request) by Mitratesh Support.

#### 3.3.4.2 Troubleshooting

##### Invoice Transfers Incomplete on TeamConnect Using WebLogic's JDBC Driver with Oracle

**Explanation:** If you are running TeamConnect with CSM on a WebLogic application server, using the Oracle JDBC driver that ships with WebLogic, you may notice that no data is synchronizing with Collaborati, even though synchronization shows as running in the CSM Settings screen (for example, expected invoices are not received in TeamConnect). Please check the default system log for a Java ClassCastException on a java.lang.Double.

**Resolution:** If you find this exception in the logs, we recommend configuring the WebLogic data source using the latest JDBC driver from Oracle.

#### Unable to start a connection between CSM and Collaborati

**Explanation:** If you are using an Oracle 10g database for TeamConnect and you are trying to test the connection between CSM Settings and Collaborati, start a connection between CSM Settings and Collaborati, or save an authorized timekeepers list in CSM Settings, if the following error displays:

Connection refused. Unrecognized CSM identifier. The `<encrypted-database-instance-name>`, `<encrypted-database-schema-name>`, `<database-schema-creation-date>`, `<last-csm-connection-date>` do not match the authorized info in Collaborati.

...then you may need to set an additional Oracle JVM parameter.

**Resolution:** Include the following setting in the JVM argument on the server on which TeamConnect is deployed:

`-Doracle.jdbc.V8Compatible=true`

For more details, see Oracle MetaLink documentation: Doc ID: Note:360422.1.

#### Why don't I see CSM Settings or why can't I open CSM Settings?

**Explanation:** If you have upgraded from TeamConnect 2.x to the current version, you need to do the following to open CSM Settings.

1. Verify that your user is a member of the CSM Admin group. See the TeamConnect Administration Guide for more information.
2. From the TeamConnect User area, from the **All Services** tab, select **CSM Settings**.
3. From the resulting page, you need to click the **All CSM Settings** link from the left navigation pane the first time you open CSM Settings. Afterward, a table will appear in the page body with the **Settings** link available to open CSM Settings.

#### Why am I seeing warning messages on the CSM Settings Share Documents page or Vendor's Share Documents page?

**Explanation:** If you are logged in as a user that only belongs to the CSM Admin group and no other groups, you may not have sufficient rights to use the **Share Documents** page. From the **Admin** tab, you need to [add your user to a group](#) that has sufficient rights to work with TeamConnect Document records. See your system administrator for assistance.

## 3.4 Developer's Help

Welcome to the *TeamConnect® Enterprise Developer's Help*.

<a href="#">TeamConnect API Policy</a>	<a href="#">TeamConnect API</a>
<a href="#">Qualifiers and Actions</a>	<a href="#">Custom Pages</a>



[Scheduled Actions](#)[Java Samples](#)

### 3.4.1 TeamConnect API Policy

This API Policy applies when a Client licenses or upgrades to TeamConnect version 4.0 or later. For the purposes of this API Policy, Partners are service providers who successfully complete Mitrtech's Partner Certification Program and have signed a partner agreement with Mitrtech. Mitrtech acknowledges some Clients may engage a third party service provider to perform professional services on Client's behalf; a third party service provider's violation of this API Policy is deemed a violation by Client of this API Policy. Any development performed by Mitrtech's Professional Services team will be deemed compliant with this API Policy. The use of the TeamConnect API layer by Mitrtech's Clients is subject to the following terms and conditions:

#### Permitted Use

Interaction with Mitrtech products via the API layer is subject to Mitrtech licensing rules and agreements. Client acknowledges the API layer contains checkpoints which allow Mitrtech to monitor usage of the API layer. Client (including any end users allowed by Client) shall not circumvent the API layer to directly interact with Mitrtech products, databases or other components. The end users, whether a named user, limited privilege user, or any other user, of any application accessing the Mitrtech API constitute end users of Mitrtech products.

Client shall not market or sell an Application that utilizes the API layer without prior written consent of Mitrtech and paying the applicable licensing fees, if any.

#### Policy Violation

If, absent prior written consent of Mitrtech obtained via Mitrtech's standard consent procedures, Client circumvents the API layer to directly interact with Mitrtech products, databases or other components, Client has violated this API Policy. Even if Mitrtech consents to the circumvention of the API layer by Client, Client understands the circumventing product will not be warranted or supported by Mitrtech, but keeping Mitrtech apprised of the situation will help Mitrtech better advise Client on future development and upgrades.

#### Consequences of Violation

Client acknowledges circumvention of the API layer could make future upgrades impractical or cost prohibitive, and Client assumes this risk. In addition, Client understands Mitrtech does not support or warrant the specific circumventing development.

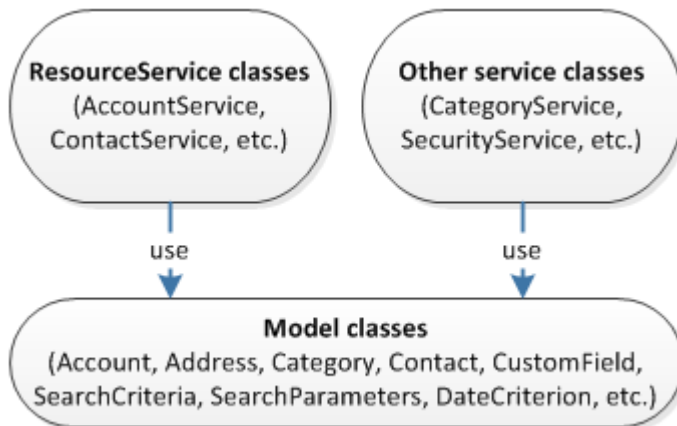
If Client uses a third party service provider, whether or not a Partner, to work on or make any product, database, or component that integrates with TeamConnect, then Client holds Mitrtech harmless from any violation of this API policy by the service provider. If Client markets or sells an Application that utilizes the API layer without obtaining Mitrtech's prior written consent and paying the applicable licensing fees, if any, Client may be liable for the unpaid license fees and subject to injunction or any other remedies allowed under the license agreement with Mitrtech.

### 3.4.2 The TeamConnect API

If the customization options available in the Designer user interface are not sufficient for your design, you can use the TeamConnect API to add functionality to areas of the application. Written in Java, the TeamConnect API consists of several classes and methods.

#### Service and Model Classes

The following diagram illustrates the structure of the API.



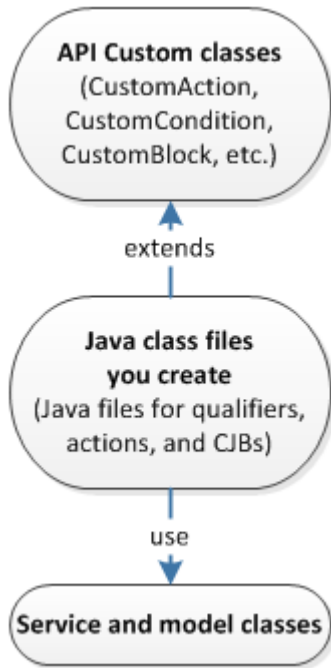
As shown in the diagram, the API includes [service classes](#) and [model classes](#). The service classes use the model classes, which contain methods for records and functionality in TeamConnect.

The service classes perform behavior and provide access to instances of an object. The service classes are divided into two different categories:

- [ResourceService classes](#)—The service classes that provides methods for creating, reading, updating, deleting, searching for, and manipulating model objects. The API includes a service class for each TeamConnect entity. For example, you use `AccountService` to access financial accounts and `UserService` to access user accounts.
- [Other service classes](#)—All other service classes provide methods for high-level functions that do not pertain to one object type. For example, you use `CategoryService` to access category objects and `SecurityService` to access security objects.

#### Custom Classes

In addition to service and model classes, the API also includes classes for custom code. The following diagram illustrates how custom classes function in the API.



You use the custom classes to create the Java class files for [automated qualifiers](#), [automated actions](#), [custom screens](#), [custom tools](#), and [scheduled custom actions](#). When writing code for these files, you extend the API custom class specific to the type of file you are creating. For example, when creating a Java class for a qualifier, you extend `CustomCondition`. When creating a Java class for a wizard custom screen, you extend `WizardCustomBlock`. After you extend the necessary class, use the service and model class methods to write custom code for the class file.

For example code of each of these types of classes, see [Getting Started with Custom Classes](#).

#### 3.4.2.1 Setting up Your Development Environment

The TeamConnect API is based on JDK 1.6 and works with versions 1.6 and 1.7. You can use any compatible development environment and compiler that you wish. The following JAR file is necessary for custom code development:

**teamconnect-enterprise-api-version.jar**, which contains most TeamConnect packages.

You can find these files in the **utilities/lib** directory of your installation media.

**Note:** In these JAR files, **version** represents the version number that differs from one TeamConnect version to the next.

To test your custom code, you must have access to an instance of TeamConnect. From TeamConnect, upload your Java classes, XML, or JAR files to TeamConnect's **Documents** area. Refer to [Adding Custom Code to TeamConnect](#) for details about adding custom code to TeamConnect.

### 3.4.2.2 API Package Classes

Written in Java, the TeamConnect API consists of several packages. Packages are methods grouped together with a common theme for organization purposes.

The API includes the following packages:

- **com.mitratech.teamconnect.enterprise.api**—Includes any general classes, mainly `Platform`.
- **com.mitratech.teamconnect.enterprise.api.callable**—Includes the four different types of callable methods that you use with `UtilityService.runAsSystemUser()`.
- **com.mitratech.teamconnect.enterprise.api.custom**—Includes the classes you extend when you are creating custom screens, automated qualifiers, and automated actions.
- **com.mitratech.teamconnect.enterprise.api.model**—Includes the classes for all API objects that are not part of the other model packages.
- **com.mitratech.teamconnect.enterprise.api.model.custom**—Includes the API model classes for all design entities.
- **com.mitratech.teamconnect.enterprise.api.model.enums**—Includes the API model classes for all enumerations.
- **com.mitratech.teamconnect.enterprise.api.model.exceptions**—Includes the API model classes for all exceptions.
- **com.mitratech.teamconnect.enterprise.api.model.parameters**—Includes the API model classes for all parameters.
- **com.mitratech.teamconnect.enterprise.api.model.search**—Includes the API model classes for searching.
- **com.mitratech.teamconnect.enterprise.api.service**—Includes all the service classes.

To see the specific contents of these packages, refer to the API Reference in the following location on your installation media:

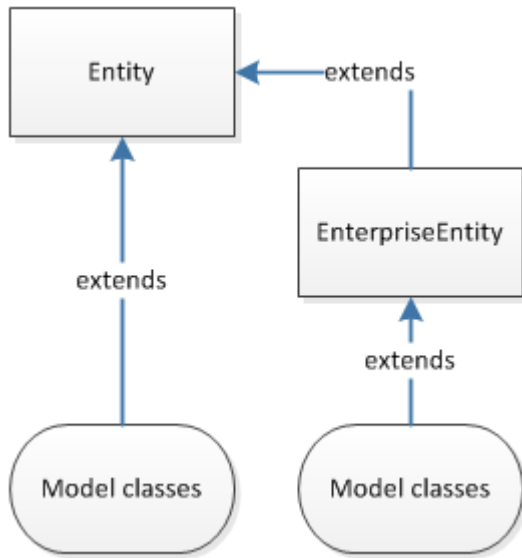
**~\utilities\javadocs\api\index.html**

#### 3.4.2.2.1 Model Classes

Use the API model classes to describe properties of TeamConnect objects and to implement actions.

**Note:** To use model classes, you may need to [access a service class](#) if you do not already have access to an instance of the object.

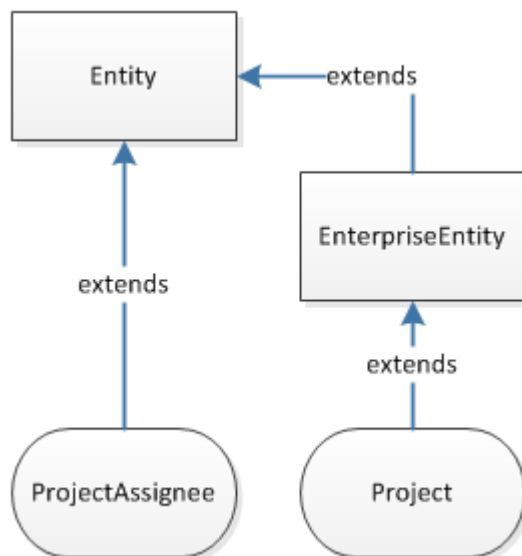
The following diagram illustrates the structure of most API model classes.



All model classes correspond to an object in the database. The TeamConnect API includes the following main model classes:

- `EnterpriseEntity`—The base class for all model classes that correspond to records of system and custom objects. The classes for these objects include methods for record features associated with the objects, such as security, custom fields, and categories. Those model classes extend `EnterpriseEntity`.
- `Entity`—The base class that provides functionality for sub-objects and other properties of `EnterpriseEntity` records, such as assignees, phone numbers, and invoice adjustments. `EnterpriseEntity` extends `Entity`, which means that `Entity` also provides a base class for system and custom objects.

For example, the `Project` model class extends `EnterpriseEntity`, which extends `Entity`. The `ProjectAssignee` model class, which provides functionality for a project, extends `Entity`. The following diagram illustrates the API model structure with these classes.



### 3.4.2.2.2 Enumerations

Enumerations, or enums, manage options in drop-down lists in TeamConnect. Unlike lookup tables, you cannot add options to these drop-down lists.

All enums are [model API classes](#) that include a list of the enum constants. The purpose of these enums is to specify the selected option for a drop-down field, so that you can use the field in API code. Enum classes always include the following two methods:

- `valueOf()`—Returns the enum constant for a given field name.
- `values()`—Returns the list of enum constants.

For example, accounts in TeamConnect have the drop-down list field of **Overdraft Type**. The `AccountOverdraftType` class includes three enum constants for the three options in this drop-down list. You use this class for the **Overdraft Type** account field.

### 3.4.2.2.3 Service Classes

You access a service class for a particular object when you are using any of the `ResourceService` methods (`create()`, `update()`, `read()`, etc.). For example, if you need to use the `update()` method to update the name of an account, you need to access `AccountService`.

**Note:** If you are adding the [custom code to a screen or rule](#), the API already has access to the object with that screen or rule. As a result, you do not need to access the service class for that object.

When you use the API, you must reference the service classes so that you can call or invoke methods that the service classes use. Accessing the service classes requires two parts:

- A main utility class specific to the module, which you can use to retrieve service implementations. `Platform` is the main utility class for the TeamConnect enterprise module. If you look at the `Platform` class in the javadocs, you see a method that you can use to call each service class for `ResourceService` and `TransationalService`. For example, to access the account service using the `Platform` class, you use the following method:

```
platform.getAccountService()
```

- A service class. Service classes provide access to API model classes. To access the account model classes using the main utility class method, you enter the following information:
  - The name of the service class
  - A variable name for the method
  - An equal sign
  - The `Platform` method for the service class

For example, to access `AccountService`, enter the following code:

```
AccountService accountService = Platform.getAccountService();
```

**Note:** Before you can make a change to an object, you must access the service class for the object. If you are using objects from more than one service class in your code, you must access the service classes for both objects.

After you call the service class(es), you can use [the API model classes](#) to describe properties of TeamConnect objects and implement actions.

#### 3.4.2.2.3.1 Resource Services

Resource services provide classes for creating, reading, updating, deleting, searching for, and manipulating model objects. Each service class corresponds to a TeamConnect entity or object type (for example, contacts, projects, accounts, invoices, etc.).

While each service class has its own set of methods that are specific to that object, `ResourceService` provides a few common functions that all service classes extend, as shown in the following table.

**ResourceService Method Actions**

Action	Methods to Use
Creating a record	<ul style="list-style-type: none"> <li>Use the <code>create()</code> method to create a record.</li> </ul> <p><b>Note:</b> When writing code to <a href="#">create a record for a rule, condition, or wizard</a>, you do not need to use the <code>create()</code> method because the changes automatically commit to the database.</p> <ul style="list-style-type: none"> <li>Use the <code>batchCreate()</code> method to create multiple records.</li> </ul>
Updating a record	<ul style="list-style-type: none"> <li>Use the <code>update()</code> method to update a record.</li> </ul> <p><b>Note:</b> When writing code to <a href="#">update a new record for a rule, condition, or wizard</a>, you do not need to use the <code>update()</code> method because the changes automatically commit to the database.</p> <ul style="list-style-type: none"> <li>Use the <code>batchUpdate()</code> method to update multiple records.</li> </ul>
Reading a record	<ul style="list-style-type: none"> <li>Use the <code>read()</code> method to read a record using its primary key.</li> <li>Use the <code>readLastSaved()</code> method to read the last saved version of a record. You can use this version to determine if a record property has any updates.</li> </ul>
Deleting a record	<p><b>Note:</b> When you want to <a href="#">delete a record that is a dependency for another record</a>, consider the best way to delete that record.</p> <ul style="list-style-type: none"> <li>Use the <code>delete()</code> methods to delete a record:</li> </ul>

	<ul style="list-style-type: none"> <li>○ One <code>delete()</code> method deletes an object for which you have a reference.</li> <li>○ The other <code>delete()</code> method deletes an object for which you have no reference and the primary key is known.</li> <li>• Use <code>batchDelete()</code> to delete multiple records. <code>batchDelete()</code> uses the <code>delete()</code> method multiple times.</li> </ul>
Creating a copy of a record	<ul style="list-style-type: none"> <li>• Use the <code>copyEntity()</code> method to create a copy of a record.</li> </ul>
Searching for a record	<ul style="list-style-type: none"> <li>• Use the <code>search()</code> methods to search for a record: <ul style="list-style-type: none"> <li>○ One <code>search()</code> method uses the search criteria to search.</li> <li>○ One <code>search()</code> method uses the search criteria and parameters to search.</li> <li>○ One <code>search()</code> method uses an existing search view to search.</li> </ul> </li> <li>• All <code>search()</code> methods use the search model classes.</li> </ul>
Determining if security settings changed	<ul style="list-style-type: none"> <li>• Use the <code>isSecurityChange()</code> method to determine if security settings for an entity changed.</li> </ul>
Retrieves a record URL	<ul style="list-style-type: none"> <li>• Use the <code>getRecordURL()</code> method to return the URL to a record on your instance.</li> </ul>

For examples of some of these functions, see the code examples for each service class:

- [AccountService](#)
- [AppointmentService](#)
- [ContactService](#)
- [DocumentService](#)
- [ExpenseService](#)
- [GroupService](#)
- [HistoryService](#)
- [InvoiceService](#)
- [ProjectService](#)
- [InvolvedService](#)
- [TaskService](#)



- [UserService](#)

In the cases of the record that triggers the qualifier or action and the record with the custom block, you do not need to call the service class to access to the record.

The following table shows whether you use `ResourceService` classes in qualifiers, actions, and screens.

**Services in Qualifiers, Actions, and Screens**

Method	Qualifiers and Actions	Custom Screens
<code>read()</code> (for the initial record)	No	No
<code>create()</code>	No	Yes
<code>update()</code>	No	Yes
<code>new&lt;object&gt;()</code> (for the new object method in the Service class)	Yes	Yes

## Qualifiers and Actions

When you create an automated qualifier or action, the code calls the `CustomCondition.condition(M)` or `CustomAction.action(M)` methods. Your code must override these methods, which allows access to the record that triggered the qualifier or action. Because you have access to the record, you do not need to call the service class to read the record.

Rules, wizards, and conditions save all changes automatically. As a result, you do not need to use the `create()` or `update()` service methods. To delete an object or create an object using the `new<Object>()` method in a service class, you must still call the service class for the object.

## Custom Screens

When you create a custom block for a record, you use the `CustomBlock.getRecord()` method to access the record. Because you have access to the record, you do not need to call the service class for the record.

Unlike with automated qualifiers and actions, custom screens do not save automatically. As a result, if you want to create a new record or update the record with the screen, you must use the `create()` or `update()` methods. Because `create()` and `update()` are `ResourceService` methods, you must call the service class of the record as part of the code.

For example, if the custom screen is on the contact object definition, but you are not creating a new record or updating an existing contact, you do not need to call `ContactService`. However, if you want to update a field in the contact record when another field changes, your custom screen code must include the call to `ContactService` so that you can use the `update()` method.

When you want to delete a record that is a dependency for another record, you must first delete the record that has the dependency, then you can delete the dependency record. For example, if you want to delete a contact that is listed as the contact for a user, delete the user first. Then you can delete the contact. Because you must use the `delete()` method to perform the deletions, you must call `UserService` and `ContactService`.

However, if you want to delete a record that is a dependency for another record as part of an automated qualifier and action, you cannot delete both as part of the qualifier or action because both deletions cannot be part of the same transaction.

#### 3.4.2.2.3.2 Other Service Classes

Any service classes that are not `ResourceService` classes initiate actions not specific to any model objects. These classes corresponds to different high-level functions in TeamConnect, including settings, security, and internationalization.

For examples of using these functions in the API, see the topics for the following classes:

- [CategoryService](#)
- [InternationalizationService](#)
- [LookupItemService](#)
- [ObjectDefinitionService](#)
- [ScheduledActionService](#)
- [SecurityService](#)
- [SettingsService](#)
- [UtilityService](#)

#### 3.4.2.3 Getting Started with Custom Classes

When you create a rule in the **Setup** or use Screen Designer, TeamConnect provides customization options in the user interface. If the user interface does not offer the type of customization you want, you use the API classes to create custom code and upload the code to TeamConnect. The TeamConnect API allows you to create the following types of custom code:

- **Conditions**—You add conditions to the **Qualifier** page of a rule or the **General** tab of a condition. Custom conditions are also known as automated qualifiers.
- **Actions**—You add actions to the **Action** page of a rule and the **Actions** page of a wizard. Custom action are also known as automated actions.
- **Blocks**—You upload custom blocks to the **Screens** directory under the **Documents** tab and select them from the **Blocks** tab to add them to a specific object definition. Custom blocks are also known as custom screens.
- **Wizard blocks**—You upload custom blocks to the **Screens** directory under the **Documents** tab and select them from the wizard's **Page Components** tab. Wizard blocks are also known as wizard custom screens.
- **Scheduled actions**—You upload scheduled actions to the **Scheduled Actions** directory under the **Documents** tab and select them from the **Scheduled Actions** tool under the **All** tab.

- **Tools**—You upload custom tools to the **Tools** directory under the **Documents** tab and select them from the **Administer Custom Tools** page.

Use the sample code in this topic as a starting place for each previously mentioned code type.

## Custom Action

This code provides a sample for a [custom action](#) on a contact rule. It also includes code for [logging Debug messages](#).

```
import com.mitratech.teamconnect.enterprise.api.custom.CustomAction;
import com.mitratech.teamconnect.enterprise.api.model.Contact;

public class ContactCustomAction extends CustomAction<Contact> {

    @Override
    public void action(final Contact contact) {
        logDebug("contact="+contact);
    }
}
```

## Custom Action with Parameters

This code provides a sample for a custom action with [parameters](#) on a contact rule.

```
import com.mitratech.teamconnect.enterprise.api.custom.CustomAction;
import com.mitratech.teamconnect.enterprise.api.model.Contact;

public class ContactCustomAction extends CustomAction<Contact> {

    @Override
    public void declareParameters() {
        getParameters().addTextParameter("name", "Label", "Default Value");
    }

    @Override
    public void action(final Contact contact) {
        logDebug("Parameter="+getParameters().getTextParameterValue("name"));
    }
}
```

## Custom Condition

This code provides a sample for a [custom condition](#) on a contact rule or condition. It also includes code for logging Debug messages.

```
import com.mitratech.teamconnect.enterprise.api.custom.CustomCondition;
import com.mitratech.teamconnect.enterprise.api.model.Contact;

public class ContactCustomCondition extends CustomCondition<Contact> {

    @Override
    public boolean condition(final Contact contact) {
        logDebug("contact="+contact);

        return true;
    }
}
```

## Custom Block

This code provides a sample for a [custom block](#) for a contact record. It also includes code for custom initialization.

```
import com.mitratech.teamconnect.enterprise.api.custom.CustomBlock;
import com.mitratech.teamconnect.enterprise.api.model.Contact;

public class MyCustomBlock extends CustomBlock<Contact>
{
    @Override
    public void initialize(java.util.Map<String, String> pageArgs) {
        // Perform some custom initialization
    }

    public String getText() {
        Contact contact = getRecord();
        // Display something on screen

        return contact.getDisplayString();
    }
}
```

## Wizard Custom Block

This code provides a sample for a [custom block for a contact wizard](#). It also includes code for parameters and custom initialization.

```
import com.mitratech.teamconnect.enterprise.api.custom.WizardCustomBlock;
import com.mitratech.teamconnect.enterprise.api.model.Contact;
```

```
public class MyCustomBlock extends WizardCustomBlock<Contact> {
    @Override
    public void initialize(java.util.Map<String, String> pageArgs) {
        // Perform some custom initialization
        getParameters().addTextParameter("name", "Label", "Default Value");
    }

    public void action() {
        Contact contact = getRecord();
        logDebug("Parameter=" + getParameters().getTextParameterValue("name"));
        logDebug("Contact=" + contact.getDisplayString());
    }
}
```

## Scheduled Custom Action

This code provides a sample for a [scheduled custom action](#) for a contact. It also includes code for logging Debug messages.

```
import com.mitratesoft.teamconnect.enterprise.api.model.Contact;
import com.mitratesoft.teamconnect.enterprise.api.custom.ScheduledCustomAction;

public class SampleActionClass extends ScheduledCustomAction {
    @Override
    public void action() {
        logDebug("message");
    }
}
```

## Custom Tool

This code provides a sample for a [custom tool](#). It also includes code for logging Debug messages.

```
import com.mitratesoft.teamconnect.enterprise.api.custom.CustomTool;
import com.mitratesoft.teamconnect.enterprise.api.model.Contact;

public class MyCustomTool extends CustomTool {
    @Override
    public void initialize(java.util.Map<String, String> pageArgs) {
        // Perform some custom initialization
    }
}
```

```

public void action() {
    Contact contact = getRecord();
    logDebug("Contact=" + contact.getDisplayString());
}
}

```

### 3.4.2.3.1 Automated Qualifiers and Actions

[Automated qualifiers](#) are files that you create to add conditional expressions to TeamConnect rules and conditions. [Automated actions](#) are files that you create to add actions to rules and wizards.

Automated qualifiers and actions display in one of the following two states:

- **Current Object State**—Object data displayed on the screen at the time when the user attempts a specified operation. In existing records, this data may include previously saved values from the database, as well as the values entered by the user before updating the record.

In new records that are being created, this data includes only the values entered by the user and automatically populated by the system. All rule types, except security, typically use the current object data. Post commit rules only use the current object state.

- **Old Object State**—Object data stored in the database at the time when the user attempts a specified operation. This applies only to existing records and excludes the values that the user may have entered before saving the record.

Security rules are the only rule type that uses exclusively the old object state. However, sometimes, other rule types may use the old object state to check if any changes have been made to the current object.

## Rule Types and Execution Order

The following table indicates which rules may have custom or automated qualifiers and actions.

Rule Types

Rule Type	Can Have Automated:		Execution Order
	Qualifier	Action	
Security	x		1
Pre-population	x	x	2
Validation	x		3
Approval	x		4
Custom Action	x	x	5

<b>Scheduled Action</b>	<b>x</b>	<b>x</b>	<b>6</b>
<b>Audit</b>	<b>x</b>	<b>x</b>	<b>7</b>
<b>Post Commit</b>	<b>x</b>	<b>x</b>	<b>8</b>

When a user attempts an operation in TeamConnect, multiple rules may trigger. If rules of different types apply to that operation, the rules execute according to the predefined order indicated in [the Rule Types table](#). You cannot modify the predefined order in which multiple rules execute.

If several rules of the same type trigger, they execute according to the order number of the rule in the user interface (for example, see [the General Tab on Rule Screens image](#)). You can change this order.

## Parameters

[Parameters](#) are arguments passed to the rule through fields in the user interface. The Java file defines how parameters appear as fields in the user interface. The user specifies parameter values in the rule interface rather than in the file. As a result, you can maintain the rule components without modifying the actual code.

### 3.4.2.3.1.1 Automated Qualifiers

You add qualifiers to the **Qualifier** page of a rule and the **General** page of a condition. All qualifiers return a Boolean to determine whether the qualifier conditions are true or false.

When qualifiers are automated, they use Java files to create the qualifier. Automated qualifiers for rules define whether the associated action executes; the action executes if the qualifier is true. Automated qualifiers for conditions define whether the condition is true.

After you create an automated qualifier, upload the file to the **Automated Qualifiers** folder in the appropriate object definition. For more details, see [Uploading Rule Component Files](#).

## When to Create Automated Qualifiers

You create an automated qualifier when you cannot create the qualifier through the user interface. Create an automated qualifier in the following situations:

- You need access to more methods than those methods available through the TeamConnect object model.

For example, a rule may require that an action executes if a project has a related involved party record with a specific role. Because projects have no one-to-many relationship with involved parties, creating this rule requires an API method to return the list of the associated involved party records.

For a sample in Java, see [Checking Category List in Related Records](#).

- You need to use Boolean operators, such as NOT, XOR, NOR, XNOR.

For more automated qualifier samples, see [Automated Qualifiers](#).

## Wizard Page Transition Rules

[Page transition rules](#) are similar to the TeamConnect rule types in [the Rule Types table](#), especially validation rules. They determine the sequence of the wizard pages by performing actions based on conditions met in the qualifier. Along with values of the main object, automated qualifiers in page transition rules may use wizard parameters and values from related objects and sub-objects defined in the templates associated with the wizard.

You cannot customize actions in page transition rules because they have a predefined transition to a specific page in the wizard. All page transition rules also have the same predefined trigger, which you do not need to define. The trigger is the transition between pages that occurs when you click the **next** button on the wizard page for the rule.

For example, if the wizard identifies a claimant, a page transition rule may use the provided information to search the database for the contact. If the system finds a matching contact, the wizard goes to the page that verifies the available contact information. If not, the wizard goes to the page that collects the contact information for a new contact record.

### 3.4.2.3.1.2 Automated Actions

You add actions to the **Action** page of a rule and the **Actions** page of a wizard. Rule actions specify what occurs if the rules meet the conditions in rule qualifiers. Wizard actions specify the actions for each wizard page. Unlike automated qualifiers that must return a boolean value, actions do NOT return values.

Automated actions files for both rules and wizards must be uploaded to the **Automated Actions** folder in the appropriate object definition. For more details, see [Uploading Rule Component Files](#).

## Rule Automated Actions

[Rule actions](#) execute when qualifiers return true. With automated rule actions, you can create new records and compare, calculate, and insert values. You use automated actions for custom and scheduled action rules.

For example, a rule may require that if a contact name changes, a history record for the contact saves with the date when the name changed and the ID of the user who made the change. For rule action samples in Java, see [Rule Actions](#).

You do not create automated actions in the following situations:

- To deny the operation that triggers a rule.
- To approve the operation that triggers a rule and its route.
- To define actions with a template, such as populating fields with defined values.

## Wizard Automated Actions

Wizard actions execute when the user clicks **next** on the wizard page with the action, moving the wizard to the next page. Like rule actions, automated wizard actions create new records and compare, calculate, and insert values.

For example, in a wizard, you may want to assign a value to a custom field of the object. Or, instead of assigning values you may want to calculate values, display additional entry fields, overwrite values



in the associated templates, validate the entered data, or create a new record. For examples of wizard page actions in Java, see [Wizard Page Actions](#).

When writing automated page actions, make sure they will not duplicate custom action rules with the **Create** trigger. Custom action rules execute when the wizard finishes and the rule qualifiers are met.

You do not need to create automated actions in wizards when you need to assign a value to a system field of the main object, its sub-objects, and related objects.

**Important:** *In wizards, do not confuse page actions with rule actions in page transition rules because you cannot automate actions in page transition rules.*

#### 3.4.2.3.1.3 Qualifier and Action Code

When creating [automated qualifiers](#) and [automated actions](#), you use the following subclasses of `CustomItem`:

- `CustomCondition`—Use when creating automated qualifiers for rules and conditions.
- `CustomAction`—Use when creating automated actions for rules and wizards.

The following table provides the basic differences between qualifiers and actions.

Qualifiers	Actions
Extend <code>CustomCondition</code> and the class of the enterprise entity,	Extend <code>CustomAction</code> and the class of the enterprise entity,
Implement the abstract method <code>CustomCondition.condition(final M record)</code> that returns true or false.	Implement the abstract method <code>CustomAction.action(final)</code> that does not return anything.
Define your logic in the <code>condition(M)</code> method. Because the qualifier returns a Boolean value, if the return value is true for rules, TeamConnect runs the action associated with this condition. If it is false, the action does not execute.	Define your logic in the <code>action(M)</code> method.

You call the `condition(M)` and `action(M)` methods when the qualifiers and actions run. Override these methods to access an instance of the object.

The following code snippet demonstrates the basic structure of a qualifier in Java. You must determine what else to import for your qualifier.

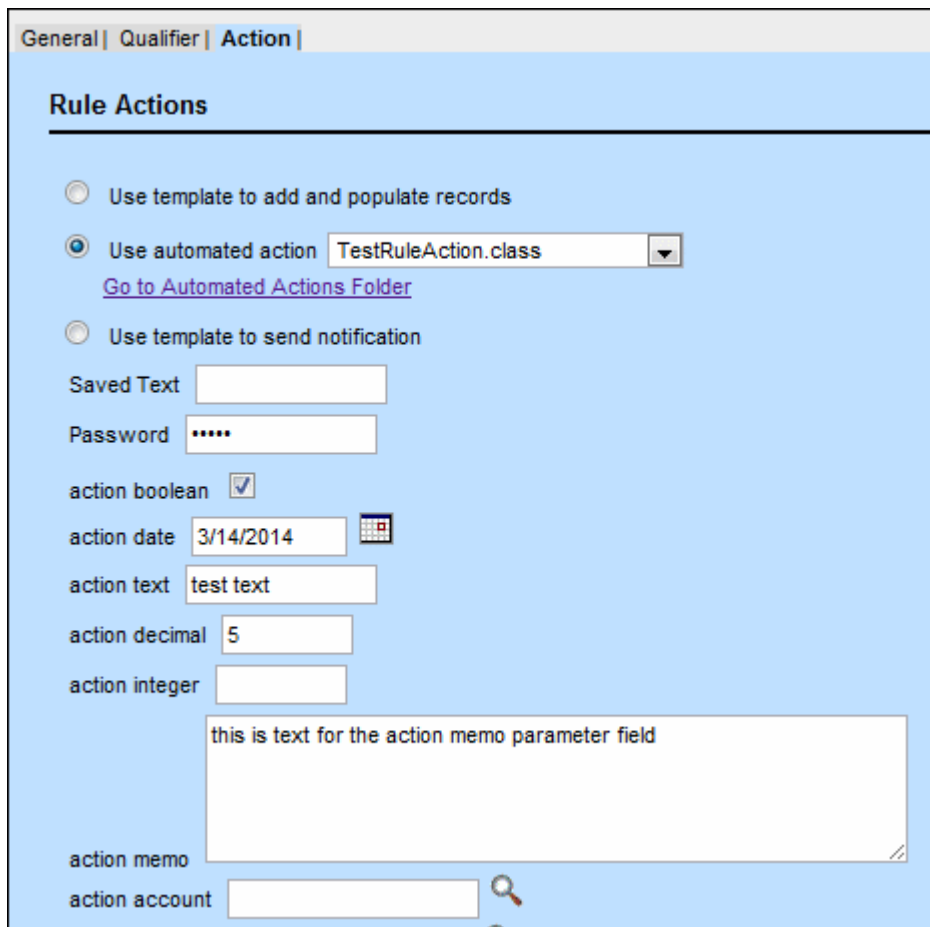
```
import com.mitratesoft.teamconnect.enterprise.api.model.Project;
import com.mitratesoft.teamconnect.enterprise.api.model.rule.CustomCondition;
public class MyCustomCondition extends CustomCondition<Project>
```

```
{
    @Override
    public boolean condition (final Project record){
        // Put your code here
    }
}
```

The following code snippet demonstrates the basic structure of an action in Java. You must determine what else to import for your action.

```
import com.mitratech.teamconnect.enterprise.api.model.Project;
import com.mitratech.teamconnect.enterprise.api.model.rule.CustomAction;
public class MyCustomAction extends CustomAction<Project>
{
    @Override
    public boolean action (final Project record) {
        // Put your code here
    }
}
```

Parameters in actions and qualifiers are values that the Java file displays as fields in the user interface, as shown in the following image.



Parameters Added from an Automated Action

If values in a qualifier or an action might change, use parameters so that the solution developer can change the value without modifying the action or qualifier file. In addition, when you use parameters in rules, you are sometimes able to use the same file for multiple rules that require different values.

For example, when an automated action creates a task with the due date based on a date in a project, the solution developer can set the number of days before the task is due through the user interface. You can also use an automated action to create new accounts for budgets that a solution developer sets from a parameter.

Set or get parameter values using the methods from the `Parameters` class.

For code samples, see [Parameterized Actions](#).

## About Parameters in Wizards

Wizard page actions use the following types of parameters:

- Wizard parameters that are transient values of different types that the wizard creates, uses, and discards afterwards. The user who designs the wizard through the user interface defines the wizard parameter name within the wizard.

- All sub-objects and related objects defined in the template, which the wizard treats as parameters. The system within the template automatically defines the names of these parameters, using the name of the object and a sequential number. For example, Account 1, History 1, Assignee 1, and Assignee 2.
- [Parameters](#) that you use in any automated action or qualifier.

## Adding and Using Parameters

Qualifiers and actions both accept parameters. Parameters are fields that appear on the **Qualifier**, **General**, and **Action** pages that someone working in the **Setup** can update. When [adding, getting, or setting parameters](#), use the `Parameters` class.

You define parameters in rules in the following way:

- The `CustomCondition.declareParameters()` or `CustomAction.declareParameters()` methods, which accept methods from the `Parameters` class.
- The add methods in the `Parameters` class, which create the parameters. The class includes a parameter for each data type and certain system objects. For example, if you want to add a project field to the **Qualifier** or **Action** page, use the `addProjectParameter()` method.

After you define parameters, you include the code for the qualifier or action using the `condition(M)` or `action(M)` methods. As part of this code, you can retrieve the value entered for a parameter in the **Setup**. In the case of the project parameter example, you use the `getProjectParameterValue()` method to retrieve the project. If you need to update the value an existing parameter within the `condition(M)` or `action(M)` methods, you can use the set methods. For example, `setProjectParameterValue()`.

If no one enters a value for the parameter, the qualifier or action uses the default value for the parameter. When someone enters a value in the **Setup** for the parameter field, the entered value overrides the default value.

The following code sample demonstrates how to add parameters to the **Action** page of a rule. The action code retrieves the value of the parameter text field with the name "text" and updates the name of a project with the value.

```
public class TestRuleAction extends CustomAction<Project> {

    @Override
    public void declareParameters() {
        parameters.addTextParameter("text", "Text", "default value");
        parameters.addPasswordParameter("password", "Password", "default password");
        parameters.addBooleanParameter("action boolean", "action boolean", false);
        Date date = new Date();
        parameters.addDateParameter("new action date", "new action date", new Date());
        parameters.addDecimalParameter("action decimal", "action decimal", BigDecimal.valueOf(.5));
        parameters.addNumberParameter("action number", "action number", Long.valueOf(5));
    }
}
```

```
parameters.addMemoParameter("action memo", "action memo", "This is memo
text field content");

Account account = platform.getAccountService().read(Long.valueOf(503));
addAccountParameter("action account", "action account", account);
}

@Override
public void action(final Project dispute) {
    dispute.setName(parameters.getTextParameterValue("text"));
    Account newAccount = platform.getAccountService().newAccount("Test
Account", BigDecimal.valueOf(1000), new Date(), new Date());
    newAccount.setNote("Here is a note that has been set");
}
}
```

The sample code in this topic is for a qualifier that checks whether the policy holder of a Policy changed. An Involved custom field in the Policy custom object definition specifies the policy holder. If the primary key of the policy holder is the same as the primary key of the policy holder in the previous state of the record, the qualifier returns true, and the action runs. Otherwise, the qualifier returns false.

Before attempting to run this rule, you must create a custom field of type Involved in the root of the Policy custom object definition. Then, you specify this custom field as a parameter for the qualifier.

## Defining the Parameter

As explained in [Parameters in Qualifiers and Actions](#), all rule parameters use the `Parameters` class and the `declareParameter()` method of a rule condition or action. Because the example rule compares the contents of an involved custom field in two states of a record, we need to be able to locate that custom field. The `Parameter.addTextParameter()` method in the example locates the field by taking three arguments:

- The label of the parameter. This text describes the custom field.
- The name of the parameter. This text identifies the custom field.
- The default value of the parameter.

Specifying this parameter results in a text box with the appropriate label in the **Qualifiers** tab of the rule. The following code sample shows how to define this text box using a parameter.

```
import com.mitratesoft.teamconnect.enterprise.api.custom.CustomCondition;
public class MyParameterizedCustomCondition extends CustomCondition<Project>
{
    @Override
    public void declareParameters() {
        parameters.addTextParameter("My Parameter Label", "myParameter",
            "defaultValue");
    }
}
```

```
}
@Override
public boolean condition(final Project dispute) {
    // Place code for all conditions here.
}
```

## Checking the Condition

After defining the parameter, the rule must be able to compare the policy holder in the previous version of the policy with the policy holder in the new version. The rule should return true if the primary keys of the custom fields' contents are the same. When writing the code, the rule must perform the following actions:

- Retrieve the associated project.
- Retrieve the old state of the record from the database.
- Retrieve the parameter using the name of the parameter, which is `fieldKey` for this sample.

Use the `getTextParameterValue()` method to retrieve the value: `String fieldKey = parameters.getTextParameterValue("fieldKey");`

## Completing the Qualifier

The following code sample includes the parameter and conditions:

### Completed Qualifier Example

```
import com.mitratesoft.teamconnect.enterprise.api.custom.CustomCondition;
import com.mitratesoft.teamconnect.enterprise.api.model.Contact;
import com.mitratesoft.teamconnect.enterprise.api.model.Project;
public class CheckingWhetherPolicyHolderWasChanged extends
CustomCondition<Project>
{
    @Override
    public void declareParameters() {
        parameters.addTextParameter("Policy custom field name",
            "policyCustomFieldName", "policyHolder");
    }
    @Override
    public boolean condition(final Project dispute) {
        // Getting the old object.
        Project oldDispute = platform.getProjectService().read(dispute);
        // Getting parameter value that indicates custom field key
        String policyCustomFieldName =
            parameters.getTextParameterValue("policyCustomFieldName");
        // Getting policy holder from the old object's custom field.
        Contact oldPolicyHolder =
            oldDispute.getInvolvedFieldValue(policyCustomFieldName).getContact();
    }
}
```

```
// Getting policy holder from the current object's custom field.
Contact currentPolicyHolder =
dispute.getInvolvedFieldValue(policyCustomFieldName).getContact();
// If the policy holder in the old object is not the policy holder in the
current object, return true.
return
!oldPolicyHolder.getPrimaryKey().equals(currentPolicyHolder.getPrimaryKey
());
}
}
```

### 3.4.2.3.2 Custom Pages

Use XML files to create [custom screens](#) and [tools](#). The XML file includes the structure, layout, and contents of a custom block. If the screen or tool has complex business requirements or if data is pulled from related records, you can use a [Custom Java Block](#) (CJB). The CJB uses the TeamConnect API to specify actions and properties for the custom block.

**Note:** You do not need a CJB if the page is only referencing system fields or existing custom fields or you just want to display plain HTML.

When implementing a CJB, stay within the following JavaBean conventions:

- The get method signature—`public PropertyType getPropertyname()` or `public boolean isPropertyname()`

For example:

```
public String getName();
public boolean isActive();
```

- The set method signature—`public void setPropertyname(PropertyType value)`

For example:

```
public void setName(String name);
public void setActive(boolean isActive);
```

## Service and Model Classes

When you add a custom screen or tool to an object, you create page sections. When custom page requires information, access the associated [service classes](#) and call specific [model methods](#) in the CJB.

For custom screens only, because the underlying record is already available to the CJB because of the `getRecord()` method, you do not need to use the `read()` method in [the record service class](#) to access the record. If you need to access a different record of the same type, then you access the service class. For example, if you create a contact record as part of the screen, access `ContactService` in the code to use the `create()` method.

**Note:** If you want to perform some initialization when TeamConnect first loads a screen or tool, you must override the `initialize()` method. For example, if you want to access the page arguments that launch a wizard, you can use the `initialize()` method to access them.

## OnClick and OnChange Functions

If the custom block includes a CJB, you execute actions in the CJB from the XML file using the following functions:

- `invokeCustomAction`—Use in custom screen XML files.
- `invokeToolAction`—Use in custom tool XML files.

Use these functions for `onClick` and `onChange` events. For example, you can use the `invokeCustomAction` function to execute an action when a user clicks a button in a screen, as shown in the following sample:

```
<input type="button" value="Execute action1" onClick="invokeCustomAction(this, 'cjb', 'action1', 'hi', 1, true)" />
```

In the previous sample, `invokeCustomAction` takes the following parameters:

- `this`—The first parameter of these functions in all custom XML files.
- `'cjb'`—The ID of the CJB defined in the XML file.
- `'action1'`—The name of the Java method in the CJB.
- `'hi', 1, true`—Arguments that the action uses.

If this button was in a custom tool, the parameters of the `invokeToolAction` function would include the following two parameters:

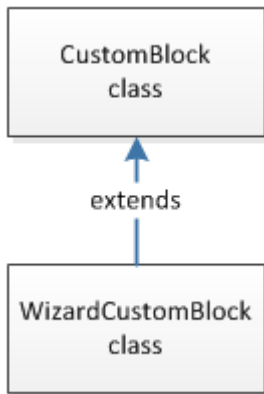
- `'action1'`—The name of the Java method in the CJB.
- `'hi', 1, true`—Arguments that the action uses.

### 3.4.2.3.2.1 Custom Screens

Custom screens extend the `CustomBlock` class and are made up of [custom blocks](#). Any block that is not part of the original functionality of TeamConnect is a custom block. Wizard custom blocks are a [type of custom block](#) that you create for wizards.

The following diagram illustrates how custom screens work in the API.





Refer to the [code sample](#) for an example of a custom screen.

## Wizard Custom Screens

As shown in the previous diagram, the `WizardCustomBlock` class extends the `CustomBlock` class when you are creating a screen for a wizard. As a result, any custom block can be a wizard custom block. However, you cannot use a wizard custom block for a custom screen because wizard blocks may contain [wizard parameters](#).

You can use wizard parameters in two ways:

- *To pass data from one page to another.*

When you add a wizard parameter to pass data, the parameter does not automatically appear in the wizard block.

To add a wizard parameter, use the `Parameters` object method for adding the specific type of parameter. For example, the following code adds text and boolean parameters.

```
getParameters().addTextParameter("text", "Text", "default value");
getParameters().addBooleanParameter("action boolean", "action boolean",
false);
```

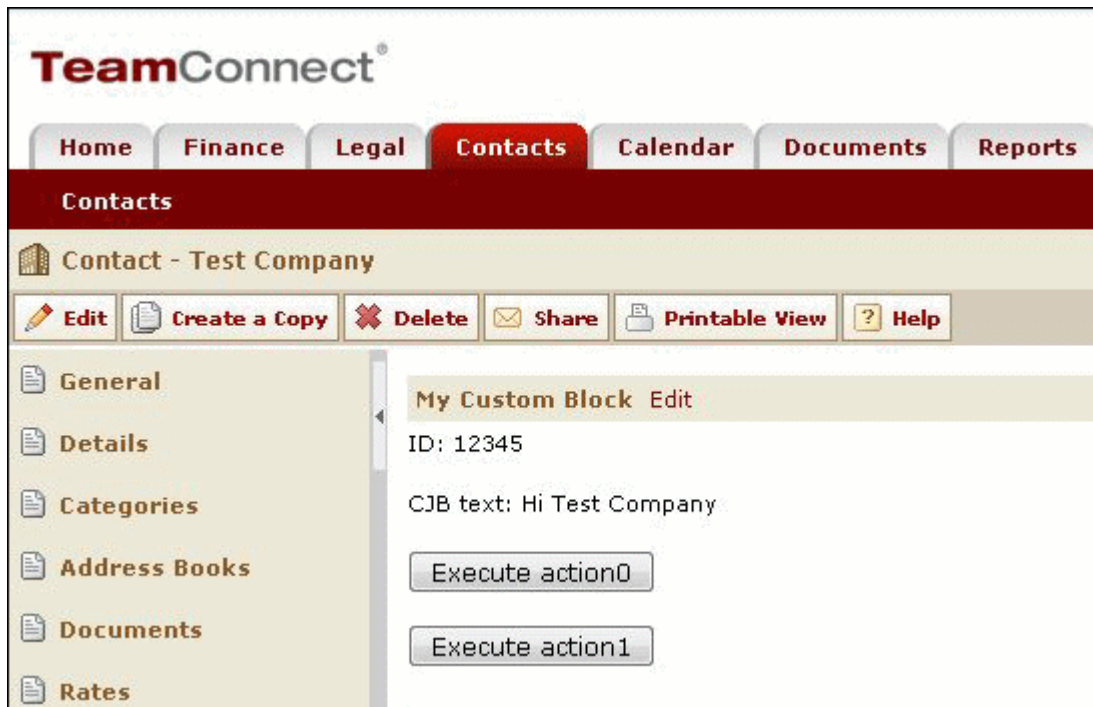
After you add the parameter to the code, you can refer to it on any other wizard page by the String name of the parameter. Retrieve the parameter using the get method for the parameter type. In the case of the previous two parameters, use the following code to retrieve the parameters.

```
getParameters().getTextParameterValue("text");
getParameters().getBooleanParameterValue("action boolean");
```

- *To display a parameter field on a wizard custom screen.*

To add a parameter to the wizard screen, you must define it on the [Page Components page](#) in the **Setup**. From the XML file, you can use the [tc:WizardParameter](#) tag to display the parameter on the wizard custom screen.

The code samples in this section provide an example of how to use CJB files to display the following block in a contact object. When you click one of the buttons in the screen, the system creates or updates a history entry.



Custom Block Example

## CJB File

The following code sample provides the CJB file for the custom block example.

```
import com.mitratesoft.teamconnect.enterprise.api.custom.CustomBlock;
import com.mitratesoft.teamconnect.enterprise.api.model.Company;
import com.mitratesoft.teamconnect.enterprise.api.model.Contact;
import com.mitratesoft.teamconnect.enterprise.api.model.History;
import com.mitratesoft.teamconnect.enterprise.api.model.Person;
import com.mitratesoft.teamconnect.enterprise.api.service.HistoryService;

// Contact CJB example
public class MyCustomBlock extends CustomBlock<Contact>
{
    @Override
    public void initialize(java.util.Map<String, String> pageArgs) {
        // Perform some custom initialization
    }

    // Display something on the screen (see XML)
    public String getText() {
```

```
        Contact contact = getRecord();
        if (contact instanceof Person) {
            return "Hi " + ((Person) contact).getFirstName();
        } else {
            return "Hi " + ((Company) contact).getName();
        }
    }

    // Execute a user action (see XML)
    public void action0() {
        if (isDebug()) {
            logDebug("action0 executed");
        }
        // Create a History
        HistoryService historyService = platform.getHistoryService();
        History newHistory = historyService.newHistory("Action0 button was
        clicked by " +
        platform.getUtilityService().getCurrentUser().getDisplayName(),
        getRecord());
        historyService.create(newHistory);
    }

    // Execute a user action with arguments (see XML)
    public void action1(String arg1, int arg2, boolean arg3) {
        try {
            // Update History
            History lastHistory =
            platform.getHistoryService().getLastHistory(getRecord());
            lastHistory.setNote(String.format("Arguments passed to method were
            arg1=%s, arg2=%d, arg3=%b", arg1, arg2, arg3));
            platform.getHistoryService().update(lastHistory);
        } catch (Exception e) {
            logError("Error in action1", e);
        }
    }
}
```

Here's how the previous CJB sample works:

- The beginning of the code extends `CustomBlock`. When you extend `CustomBlock`, the object type must match the object for which you are defining the CJB file. Because the code in this sample is for the contact object, the CJB file extends `CustomBlock<Contact>` with the following code.

```
public class MyCustomBlock extends CustomBlock<Contact>
```

- The body of the `initialize()` method includes any initial processing. The sample overrides the `initialize()` method to access the page arguments that launch the screen.

```
@Override
public void initialize(java.util.Map<String, String> pageArgs)
```

- The sample retrieves the contact record with the custom block.

```
Contact contact = getRecord();
```

**Note:** Because the CJB retrieves the contact record with this code, it does not need to performing a [ResourceService read\(\)](#) action.

- The following code displays the contact name fields on the screen after retrieving the object.

```
public String getText() {
    Contact contact = getRecord();
    if (contact instanceof Person) {
        return "Hi " + ((Person) contact).getFirstName();
    } else {
        return "Hi " + ((Company) contact).getName();
    }
}
```

- When the user clicks the `action0` button, the following code executes in response to the user action:
  - The system checks if the logger for the custom code is enabled for the **Debug** level. If true, the system logs a message.

```
if (isDebug()) {
    logDebug("action0 executed");
}
```

- The code [accesses the HistoryService class](#) using the `Platform` class.

```
HistoryService historyService = platform.getHistoryService();
```

- The system creates a new history entry with the name of the user who clicked the button.

```
History newHistory = historyService.newHistory("Action0 button was
clicked by " +
platform.getUtilityService().getCurrentUser().getDisplayName(),
getRecord());
historyService.create(newHistory);
```

- When the user clicks the action1 button, the following code executes with arguments in response to the user action:
  - The action lists the three arguments.

```
public void action1(String arg1, int arg2, boolean arg3) {
```

- The system tries retrieves the most recent history and updates the note of the entry.

```
try {  
    // Update History  
    History lastHistory =  
    platform.getHistoryService().getLastHistory(getRecord());  
    lastHistory.setNote(String.format("Arguments passed to method were  
    arg1=%s, arg2=%d, arg3=%b", arg1, arg2, arg3));  
    platform.getHistoryService().update(lastHistory);  
}
```

- If the system cannot perform an update, the code logs an error.

```
catch (Exception e) {  
    logError("Error in action1", e);  
}
```

## XML File

The following code sample provides the XML file for the custom block example. See [Creating Block XML Files](#) for more information about creating XML files.

```
<?xml version="1.0" encoding="UTF-8"?>  
<tc:transform version="4.0" xmlns:tc="http://www.w2.prg/1999/XSL/Transform">  
  <tc:blockTemplate>  
    <tc:useClass id="cjb" name="MyCustomBlock"/>  
    <!-- Display a system field -->  
    ID: <tc:field name="enterpriseEntity.idNumber" />  
    <br/><br/>  
    <!-- Display a dynamic value -->  
    CJB text: <tc:out value="{cjb.text}" />  
    <br/><br/>  
    <!-- Execute a cjb action without arguments -->  
    <input type="button" value="Execute action0"  
    onClick="invokeCustomAction(this, 'cjb', 'action0')" />  
    <br/><br/>  
    <!-- Execute a cjb action with arguments -->  
    <input type="button" value="Execute action1"  
    onClick="invokeCustomAction(this, 'cjb', 'action1', 'hi', 1, true)" />
```

```
</tc:blockTemplate>
</tc:transform>
```

Here's how the previous XML file works:

- The `tc:useClass` tag references the Java class, as shown in the following code.

```
<tc:useClass id="cjb" name="MyCustomBlock"/>
```

This tag includes the following attributes:

- `name`—The name of the Java class.
- `id`—An alias for the Java class. The `invokeCustomAction` function refers to the Java class using this alias.
- The following code displays a number system field and a text field with a dynamic value.

```
ID: <tc:field name="enterpriseEntity.idNumber" />
CJB text: <tc:out value="{cjb.text}" />
```

- The following code executes an action without arguments and an action with arguments. This code provides an example of how to use the [invokeCustomAction function](#). When you click a button, you call `invokeCustomAction`, which submits the page and causes the method in the Java class to execute. Methods can also execute with arguments, as shown in the second `invokeCustomAction` function.

```
<input type="button" value="Execute action0"
onClick="invokeCustomAction(this, 'cjb', 'action0')" />
<input type="button" value="Execute action1"
onClick="invokeCustomAction(this, 'cjb', 'action1', 'hi', 1, true)" />
```

#### 3.4.2.3.2.2 Custom Tools

Custom tools extend the `CustomTool` class. Code for custom tools is [similar to code for custom screens](#).

Refer to the [code sample](#) for an example of a custom tool.

The code samples in this section provide an example of how to use CJB files to display the following block in a custom tool.



Custom Tool Example

## Java File

The following code sample provides the CJB file for the custom block example.

```
import com.mitratesoft.teamconnect.enterprise.api.custom.CustomTool;
import com.mitratesoft.teamconnect.enterprise.api.model.Account;

public class MyCustomTool extends CustomTool
{
    public void createAccount() {
        String accountName = "New Account";
        BigDecimal allocationLimit = new BigDecimal(1000);
        Date startDate = new Date();
        Date endDate = DateUtils.addYears(startDate, 1);
        Account newAccount = platform.getAccountService().newAccount(accountName,
            allocationLimit, startDate, endDate);
        platform.getAccountService().create(newAccount);
    }
}
```

Here's how the previous CJB sample works:

- The beginning of the code extends `CustomTool`, as shown in the following code.

```
public class MyCustomTool extends CustomTool
```

- The sample [accesses the AccountService class](#) using the `Platform` class and creates a new account object called **New Account**.

```
String accountName = "New Account";
```

```
BigDecimal allocationLimit = new BigDecimal(1000);
Date startDate = new Date();
Date endDate = DateUtils.addYears(startDate, 1);
Account newAccount = platform.getAccountService().newAccount(accountName,
allocationLimit, startDate, endDate);
```

- The sample executes the `create()` method to create a new account from the tool.

```
platform.getAccountService().create(newAccount);
```

## XML File

The following code sample provides the XML file for the custom tool example. See [Creating Block XML Files](#) for more information about creating XML files.

```
<?xml version="1.0" encoding="UTF-8"?>
<tc:transform version="1.0" xmlns:tc="http://www.w3.org/1999/XSL/Transform">
  <input id="createAccount" name="createAccount" type="button"
onclick="invokeToolAction('createAccount', 'createAccount');"
title="<teamconnect:message key='button.yes' />" value="<teamconnect:message
key='button.yes' />" />
</tc:transform>
```

Here's how the previous XML sample works:

- When you save this XML file, you give it the same name as the Java file. The system knows that the XML file refers to the Java file of the same name.
- The `<input>` tag includes the following parts:
  - The ID, the name, and the type of `<input>` tag.

```
id="createAccount" name="createAccount" type="button"
```

- The `onClick` event with the [invokeToolAction function](#). When you click the button, you call `invokeToolAction`, which submits the page and causes the method in the Java class to execute.

```
onclick="invokeToolAction('createAccount', 'createAccount');
```

- The title and value of the button with a custom key that you can internationalize. To use this code, replace `button.yes` with a custom internationalization key for this button.

```
title="<teamconnect:message key='button.yes' />"
value="<teamconnect:message key='button.yes' />"
```



### 3.4.2.3.3 Scheduled Actions

The **Scheduled Actions** tool controls actions that are scheduled to execute at specific times in the future. When you add a new scheduled action from the tool, you can [select a Java class](#) from the **Action** drop-down.

All scheduled actions that you [create through the API](#) automatically save to the **Scheduled Actions** tool.

In the API, the following interfaces represent scheduled actions:

- `ScheduledActionService`—Includes methods that you use to create, delete, activate, and retrieve scheduled actions.
- `ScheduledActionStatus`—Includes methods that you use to return information about each scheduled action.
- `Parameter`—Includes methods that you use to return information about [parameter fields](#) for the scheduled action.
- `Schedule`—Includes methods that you use to return information about the start date, end date, and how often the scheduled action occurs.

#### 3.4.2.3.3.1 Adding Java Classes for Scheduled Actions

You use the API to create a Java class for the scheduled action. After you have a file for the Java class, you can add it to the **Scheduled Actions** folder in TeamConnect and select it from the **Actions** drop-down list.

The action in the following code sample retrieves a text parameter, creates a new person contact, and adds an address to the contact.

```
import com.mitratesoft.teamconnect.enterprise.api.model.Contact;
import com.mitratesoft.teamconnect.enterprise.api.model.Address;
import com.mitratesoft.teamconnect.enterprise.api.service.ContactService;
import com.mitratesoft.teamconnect.enterprise.api.custom.ScheduledCustomAction;

public class SampleActionClass extends ScheduledCustomAction {

    @Override
    public void action() {
        //Retrieve the contact service from platform.
        final ContactService contactService = platform.getContactService();

        //retrieves a text parameter
        String paramCity = getParameters().getTextParameterValue("city");

        //create a new contact of type person.
        Contact cont = contactService.newPerson("firstName", "lastName");
        //add a business address to the new contact.
```

```
Address address =
cont.addAddress("BUS1").setCity(paramCity).setState("TX");

//create new contact
Contact createdContact = contactService.create(cont);
}
}
```

#### 3.4.2.3.2 Using the API with Scheduled Actions

You can use `ScheduledActionService` to do most of the actions you can do on the **Scheduled Actions** tool page. Also like the tool, you cannot update scheduled actions. You can only delete or deactivate existing actions.

When you create a scheduled action using the API, you reference a Java class in the **Scheduled Actions** folder. If the Java class is not in the folder, you can add it through the API using [DocumentService](#). The name of the Java class in the following code sample is [ActionClass](#).

```
public class MyRule extends CustomAction<Project> {

    @Override
    public void action(Project project) {
        //the action class for the scheduled action.
        Class<ActionClass> actionClass = ActionClass.class;

        // Retrieve the service
        ScheduledActionService scheduledActionService =
platform.getScheduledActionService();

        // Build parameters
        Parameters parameters = scheduledActionService.newParameters();
        parameters.addBooleanParameter("isAThing", "This is a thing", false);
        parameters.addProjectParameter("triggeringProject", "Project responsible
for the action", project);

        //create a new weekly schedule.
        Calendar cal = Calendar.getInstance();
        cal.set(2014, 06, 19);
        Schedule schedule = Schedule.createWeekly(cal.getTime(), 2);
        ScheduledActionStatus generatedUniqueId =
scheduledActionService.scheduleAction(actionClass, parameters, schedule);
    }
}

public class ActionClass extends ScheduledCustomAction {
```

```
@Override
public void action() {
    // Repeated action code
}
}
```

If you want to get a scheduled action from the tool, you can use the `getScheduledAction()` method, as shown in the following sample.

```
ScheduledActionStatus scheduledActionStatus =
scheduledActionService.getScheduledAction("uniqueName");
```

If you want to deactivate a scheduled action using the API, you can use the `deactivate()` method, as shown in the following sample.

```
scheduledActionService.deactivate("uniqueName");
```

#### 3.4.2.4 Creating New Records

Specific conditions frequently require the creation of new records. For example, a business rule may require that when a value changes in a project, the [rule automated action](#) to create a new history record.

When creating new records, you use the following methods:

- The `new<Object>()` methods in the resource service classes to create new objects. For example, `newExpense()` in `ExpenseService`.

**Note:** When writing code to create a new record for a rule, you do not need to use the `create()` method because the rule automatically commits the changes to the database.

- The `ResourceService.create()` method to save the created object to the database after you create the empty object.
- Object-specific methods that you can set in the new empty object using [the API model interfaces](#) for the objects.
- The `UtilityService.runAsSystemUser()` method to create the record regardless of the current user's rights. For more details, see [Running Code as the System User](#).

In addition to the [resource service samples](#), refer to [Creating a History Record](#), [Creating a Child Project](#), and [Creating Related Task Record](#) for more samples about creating a record in Java.

#### 3.4.2.5 Object Definitions

An implementation of TeamConnect contains a variety of object definitions that are classified in one of the following ways:

- [System Object Definitions](#)—Objects that exist as part of the core functionality of TeamConnect, before any customization. A TeamConnect implementation may utilize some, all, or none of the system objects. System objects include Account, Appointment, Contact, Document, Expense, History, Invoice, Involved, and Task.
- [Custom Object Definitions](#)—Objects defined for a TeamConnect implementation. Custom objects could be Claims, Cases, Matters, Litigations, among others. They can have parent-child relationships with one another or you can define a custom object as an embedded object of another custom object, which is a child object with limited functionality.

An instance of an object is a record. For more details about system or custom objects, see [About Objects](#).

## ObjectDefinitionService Interface

If you want to retrieve object definitions using the API, you can use `ObjectDefinitionService`. As part of the `ObjectDefinitionService` interface, you can use the `getObjectDefinition()` method to retrieve an object definition. If the API cannot find an object definition that fits the unique code, the method throws an exception. The following code sample uses the `getObjectDefinition()` method to retrieve the contact object definition.

```
ObjectDefinition contactObjectDefinition =
objectDefinitionService.getObjectDefinition("CONT");
```

In the following code sample, you can use the `getObjectDefinition()` method to retrieve a custom object definition with the unique code of MATT.

```
ObjectDefinition matterObjectDefinition =
objectDefinitionService.getObjectDefinition("MATT");
```

**Note:** Use the following code to [access ObjectDefinitionService](#): `ObjectDefinitionService objectDefinitionService = platform.getObjectDefinitionService();`

## ObjectDefinition Interface

The `ObjectDefinitionService` methods use the `ObjectDefinition` class to return information about object definitions. `ProjectObjectDefinition` and `InvolvedObjectDefinition` also extends `ObjectDefinition`. The `ObjectDefinition` class includes the settings on the following pages of an object definition in the **Setup**:

- **General**—The `hasInvolved()`, `isContactCentric()`, and `isNewBtnNotShownInRelObjBlock()` methods refer to settings that might appear on the **General** page, depending on the type of object definition. If an object definition does not have the associated setting, the method returns `false`.
- **Wizards**—The `getRequiresWizardForCreate()` method refers to whether or not the setting on the **Wizards** page specifies that a wizard is required to create a new record.

### 3.4.2.5.1 System Objects

System objects are the out-of-the-box objects included with TeamConnect. System objects allow you to do basic functions, such as creating contacts or invoices.

All system objects in the API each have a [resource service class](#). These resource service classes use the [API model interfaces](#) to work with system fields. All system fields have get and set methods in their owner interfaces. You can also get or set [custom fields](#) on system objects.

#### 3.4.2.5.1.1 Accounts

[Accounts](#) electronically track posted transactions, mainly expenses, tasks, and invoices. Use the `AccountService` class to access accounts in the API.

The API provides the following model interfaces for accounts:

- The `Account` class, which contains information about the account.
- The `AccountInvolvedType` class, which contains the enums for posting an account to a contact.
- The `AccountOverdraftType` class, which contains the enums for how the account handles overdraft amounts.
- The `AccountPostingCriteria` class, which contains information about the account and its posting criteria.
- The `AccountPostingStatus` class, which contains the enums that indicate whether an expense, invoice, or task is posted to an account.
- The `AccountProjectType` class, which contains the enums for posting a project to an account.
- The `AccountTransaction` class, which contains information about an account transaction.
- The `AccountVendorType` class, which contains the enums for posting a vendor to an account.

Refer to [the AccountService samples](#) for examples of working with accounts in the API.

Use `AccountService` when working with account records.

Whenever a Service operation requires `AccountService`, set up `AccountService` using the following code:

```
AccountService accountService = platform.getAccountService();
```

### Creating an Account

The following sample creates an account record. During account creation, you can set objects in the account record, such as the account name.

```
Account newAccount = accountService.newAccount("name", allocationLimit,  
startOnDate, endOnDate);
```

## Creating a Child Account

The following sample creates a child account record. During child account creation, you can use API interface methods to set objects in the account, such as the parent account.

```
Account newChildAccount = accountService.newAccount("name",
allocationLimit, startOnDate, endOnDate);
newChildAccount.setParentAccount(parentAccount);
```

## Reading an Account

The following sample reads an account record. When you read an account, you retrieve the account so that you can make changes to it, such as activating or transferring money from it.

```
//The id initialized here is the id of an existing account
Long id = 1234567890L;
Account account = accountService.read(id);
```

## Activating an Account

The following sample activates an account record.

```
accountService.activateAccount(accountToBeActivated);
```

## Deactivating an Account

The following sample deactivates an account record.

```
accountService.deactivateAccount(account);
```

## Allocating Money to an Account

The following sample allocates money to an account record. When you allocate money to an account, a parent account must already exist.

```
//Allocating money to an account - $5000
accountService.allocateMoney(account, amount, "description");
```

## Transferring Money between Accounts

The following sample transfers money between two accounts. When you transfer money, a parent and child account must already exist.

```
accountService.transferMoney(accountFrom, accountTo, amount, "description");
```

## Withdrawing money from an Account

The following sample withdraws money from an account record.

```
//Withdrawing from an account - $5000
accountService.withdrawMoney(account, new BigDecimal("5000"), "Withdraw to
zero");
```

## Deleting an Account

The following sample deletes an account record. When you delete an account, the account must not have a balance.

```
//Account cannot have a balance before deletion. Make necessary withdrawals.
accountService.withdrawMoney(parentAccount, new BigDecimal("1000"), "Withdraw to
zero");
accountService.withdrawMoney(childAccount, new BigDecimal("234"), "Withdraw to
zero");

//When deleting a parent account with attached child accounts, the child
accounts also cannot have a balance. The deletion of a parent will delete its
child accounts as well.
accountService.delete(parentAccount);
```

### 3.4.2.5.1.2 Appointments

[Appointments](#) are scheduled events or meetings with a date, time, and attendees. Use the `AppointmentService` class to access appointments in the API.

The API provides the following model interfaces for appointments:

- The `Appointment` class, which contains information about an appointment.
- The `AppointmentAttendee` class, which contains information about an appointment attendee user and type.
- The `AttendanceType` class, which contains the enums that indicate if a contact will attend an appointment.

Refer to [the AppointmentService samples](#) for examples of working with appointments in the API.

Use `AppointmentService` when working with account records.

Whenever a Service operation requires `AppointmentService`, set up `AppointmentService` using the following code:

```
AppointmentService appointmentService = platform.getAppointmentService();
```

## Creating an Appointment

The following sample creates an appointment. During appointment creation, you can use API interface methods to set objects in the appointment, including the start date, end date, and category.

```
// Retrieve the appointment service from the platform provided by the API
AppointmentService appointmentService = platform.getAppointmentService();

// Set up the initial required arguments to create a valid appointment
long fourDaysInMilliseconds = 86400000 * 4;
Date startOn = new Date();
Date endOn = new Date(startOn.getTime() + fourDaysInMilliseconds);

// Create a new appointment, passing in the arguments created above.
Appointment appointment = appointmentService("Appointment Subject", startOn,
endOn);

// Any non-required fields on the appointment may be set or added now
appointment.setAllDay(true);
appointment.addCategory("APPT_CATE_CUST");
```

## Updating an Appointment

The following sample updates an appointment. When you update an appointment, you can use API interface methods to set objects in the appointment, including the primary key, new attendees, and custom fields.

```
// Retrieve the appointment service from the platform provided by the API.
AppointmentService appointmentService = platform.getAppointmentService();

// Retrieve the desired appointment.
long primaryKeyForAppointment = 2;
Appointment appointment = appointmentService.read(primaryKeyForAppointment);

// Modify fields.
appointment.addAttendee(platform.getUserService().findUserByUsername("TestUser"
), AttendanceType.WILL_ATTEND);
appointment.setBooleanFieldValue("customBooleanField", false);
```

## Deleting an Appointment

The following sample deletes an appointment.

```
// Retrieve the appointment service from the platform provided by the API.
```



```
AppointmentService appointmentService = platform.getAppointmentService();

// An appointment can be deleted simply by passing in the primary key.
long primaryKeyForAppointment = 8;
appointmentService.delete(primaryKeyForAppointment);

// Or, you can pass in the appointment object.
Appointment appointment = appointmentService.read(primaryKeyForAppointment);
appointmentService.delete(appointment);

// In a rule, the delete will be committed when the rule completes.
// Otherwise, the delete will be committed immediately.
```

## Searching for an Appointment

The following sample searches for all appointments in Room 101.

```
// Retrieve the appointment service from the platform provided by the API.
AppointmentService appointmentService = platform.getAppointmentService();

// Find all appointments located in Room 101
SearchCriteria criteria = new SearchCriteria(new
StringCriterion(Appointment.LOCATION).equalTo("Room 101"));

// Sort results by start date
SearchParameters parameters = new SearchParameters(new
SortField(Appointment.START_ON));

// Execute search and return results
return appointmentService.search(criteria, parameters);
```

### 3.4.2.5.1.3 Contacts

Contacts are people or companies for which your organization needs to store information, including employees, vendors, outside counsel, claimants, injured parties, witnesses, and agencies. Use the `ContactService` class to access contacts in the API.

The API provides the following model interfaces for contacts:

- The `Address` class, which contains information about a contact address.
- The `Company` class, which contains information that applies to only company contacts and extends the `Contact` class.
- The `Contact` class, which contains information that applies to all contacts.
- The `ContactRelation` class, which contains information about a contact relation.

- The `DefaultRate` class, which contains information about a default rate for a contact.
- The `Email` class, which contains information about a contact email address.
- The `FaxNumber` class, which contains information about a contact fax number.
- The `InternetAddress` class, which contains information about a contact web address.
- The `InvoiceTaskRate` class, which contains information about an invoice task rate for a contact.
- The `Person` class, which contains information that applies to only person contacts and extends the `Contact` class.
- The `PhoneNumber` class, which contains information about a contact phone number.
- The `Skill` class, which contains information about a contact skill.
- The `TaskRate` class, which contains information about a task rate for a contact.

Refer to [the ContactService samples](#) for examples of working with contacts in the API.

Use `ContactService` when working with contact records.

Whenever a Service operation requires `ContactService`, set up `ContactService` using the following code:

```
ContactService contactService = platform.getContactService();
```

## Creating a Person Contact

The following sample creates a person contact record. During contact creation, you can use API interface methods to set objects in the contact record, including the contact address.

```
Person personContact = ContactService.newPerson("Bob", "Dole");
Address address = person.addAddress("ADDR_HOME");
address.setStreet("1234 Fake St.");
address.setCity("Austin");
address.setState("TX");
address.setPostalCode("78704");
address.setCountry("USA");
```

## Creating a Company Contact

The following sample creates a company contact record. During contact creation, you can use API interface methods to set objects in the contact record, including the contact address and email.

```
Company companyContact = ContactService.newCompany("myLittleCompany");
Address address = company.addAddress("ADDR_BUS1");
address.setStreet("1234 Fake St.");
```

```
address.setCity("Austin");
address.setState("TX");
address.setPostalCode("78704");
address.setCountry("USA");
Email email = company.addEmail("MAIL_BUS1", "you@yourcompany.com");
```

## Updating a Contact

The following sample updates a contact record. When you update a contact, you can use API interface methods to update objects in the contact record, including the contact fax number, email, and web address.

```
//Adding or updating a business fax number
FaxNumber faxNumber = person.addFaxNumber("FAXX_BUS1", "5123827322");

//Adding or updating a personal email
Email email = person.addEmail("MAIL_PER1", "you@yourcompany.com");

//Adding or updating a mobile phone number
PhoneNumber phoneNumber = person.addPhoneNumber("PHON_MOBI", "5123827322");

//Adding or updating a business web address
InternetAddress internetAddress = person.addInternetAddress("INET_BWEB",
"www.mitratech.com");

//Adding or updating a default skill
Skill skill = person.addSkill("SKIL_DEFA", 2);
```

## Reading a Contact

The following sample reads a contact record. When you read a contact, you retrieve the contact so that you can make changes to it, such as updating or deleting it.

```
//The id initialized here is the id of an existing person contact
long personId = 49L;
Person personContact = ContactService.readPerson(personId);
//The id initialized here is the id of an existing company contact
long companyId= 23L;
Company companyContact = ContactService.readCompany(companyId);
```

## Deleting a Contact

The following sample deletes a contact record. When you delete a contact, if the contact is associated with a user or any other object, remove the association first.

```
ContactService.delete(personContact);  
ContactService.delete(companyContact);
```

#### 3.4.2.5.1.4 Documents

In addition to the **Documents** tab where you can save any document, all records have a **Documents** page where you can save documents to that record. When [working with documents](#) in the API, you have the ability to perform the same functionality that you can use with documents in TeamConnect. Use the `DocumentService` class to access with documents in the API.

The API provides the following model interfaces for documents:

- The `Document` class, which contains information about a document. A document is anything that can appear in the documents list.
- The `Folder` class, which contains information about a document folder.
- The `Hyperlink` class, which contains a hyperlink.
- The `Shortcut` class, which contains a shortcut.
- The `File` class, which contains information about a file.
- The `FileContentType` class, which contains information about the file type of a document.
- The `FileVersion` class, which contains information for a file version.

Refer to [the DocumentService samples](#) for examples of working with documents in the API.

Use `DocumentService` when working with document records.

Whenever a Service operation requires `DocumentService`, set up `DocumentService` using the following code:

```
DocumentService documentService = platform.getDocumentService();
```

### Creating a Document Record

The following sample creates a document. During document creation, you can add the document to a folder, create a shortcut for the folder, and check in the file, among other things.

```
DocumentService documentService = platform.getDocumentService();  
  
// Get attachment folder of a record  
Folder attachmentFolder =  
documentService.getAttachmentFolderForDocumentOwner(getRecord());  
  
// Create a folder under the attachment folder  
Folder folder = attachmentFolder.addFolder("New Folder");
```

```
// Create a file directly under the attachment folder
FileContentType contentType =
platform.getLookupItemService().getFileContentTypeByFileExtension("txt");
byte[] fileContent = new byte[0];
File file = attachmentFolder.addFile("New File", fileContent, contentType);

// Create a hyperlink
Hyperlink hyperlink = folder.addHyperlink("Mitratesh", "http://
www.mitratesh.com");

// Create a shortcut to a folder
Shortcut shortcutToFolder = folder.addShortcut(folder);

// Create a shortcut to a file
Shortcut shortcutToFile = folder.addShortcut(file);

//Checkout file
documentService.checkOut(file);

//Checkin file
documentService.checkIn(file, fileContent, "check in comment");
```

## Reading a Document Using its Path

The following sample returns a document using its folder path.

```
// Returns a document from the repository using its folder path
Document document = documentService.findDocumentByPath("path");
```

## Retrieving the Root Folder

The following sample returns the root document folder.

```
// Returns the root document folder
Folder rootFolder = documentService.getRootFolder();
```

## Copying a Document

The following sample copies a document to a new folder.

```
// Returns an exact duplicate of a given entity object
Document duplicateDocument = documentService.copyEntity(originalDocument);
```

```
// Copies a document to a new parent folder
documentService.copyDocument(document, parentFolder);
```

## Moving a Document

The following sample moves a document to a new folder.

```
// Changes the location of a file
documentService.moveDocument(document, parentFolder);
```

## Checking Out a Document

The following sample checks out a file and locks it so that other users cannot check it out.

```
// Checks out and locks a file for a single user to update
documentService.checkOut(file);
```

## Undoing a Document Checkout

The following sample reverses a document checkout. The document remains unchanged and has no version number.

```
// Reverses the checkout of a file.
documentService.undoCheckOut(file);
```

## Checking in a Document

The following sample checks in a document by converting the existing file to the new file version and replacing the file contents the byte array of the new contents.

```
// Checks in a file.
documentService.checkIn(file, data, "versionText");
```

## Reverting a Document to a Previous Version

The following sample reverts a document to a previous version by converting the existing file to a different file version and replacing the file contents with the byte array of the new contents.

```
// Revert a document to a prior version
FileVersion version = documentService.getFileVersions(file).get(0);
documentService.revertTo(version);
```

### 3.4.2.5.1.5 Expenses

[Expenses](#) are internal costs for your business as a whole or that you can associate with an account. Use the `ExpenseService` class to access expenses in the API.

The API provides the `Expense` model interfaces for information about an expense.

Refer to [the ExpenseService samples](#) for examples of working with expenses in the API.

Use `ExpenseService` when working with expense records.

Whenever a Service operation requires `ExpenseService`, set up `ExpenseService` using the following code:

```
ExpenseService expenseService = platform.getExpenseService();
```

## Creating an Expense

The following sample creates an expense record. During expense creation, you can use API interface methods to set objects in the expense record, including the user who is responsible for the expense and a description of the expense.

```
//Expense description is a mandatory field
Expense expense = expenseService.newExpense("Brand New Test Description");
User user = platform.getUserService().findUserByUsername("Expenser");
expense.setExpensedBy(user);
//Date will be converted to midnight GMT, then back to the local date/time
relative to GMT midnight
Calendar cal = Calendar.getInstance();
cal.set(2014, 01, 30);
expense.setExpenseDate(new CalendarDate(cal));
```

## Updating an Expense

The following sample updates an expense record. When you update an expense, you can use API interface methods to update objects in the expense, including the expense name.

```
expense.setShortDescription("Expense Description Update");
//Adding new unit price for expensed item ($150) and quantity of units (2)
expense.setUnitPrice(BigDecimal.valueOf(150));
expense.setQuantity(BigDecimal.valueOf(2));
```

## Reading an Expense

The following sample reads an expense record. When you read an expense, you can retrieve the last saved version of the expense to post, delete, or do something else with the record.

```
//The expenseHandle here is a preexisting Expense object  
Expense lastSavedExpense = expenseService.readLastSaved(expenseHandle);
```

## Posting an Expense

The following sample posts an expense record. When you post an expense, you can use API interface methods to prepare an expense for posting.

```
//Ensure that these options are enabled in an account to post an expense  
account.setAllowPosting(true);  
account.getAccountPostingCriteria().setAllowExpense(true);  
expenseService.postExpense(expense);
```

## Voiding an Expense

The following sample voids an expense record.

```
//An expense must be posted to be voided  
expenseService.voidExpense(expense);
```

## Deleting an Expense

The following sample deletes an expense record.

```
expenseService.delete(expense);
```

## Getting Transactions for an Expense

The following sample returns a list of transactions for an expense record.

```
List<AccountTransaction> transactions =  
expenseService.getTransactionsForExpense(expense);
```

## Getting Transactions for an Expense Associated with an Account

The following sample returns a list of transactions for an expense record that is associated with an account.

```
List<AccountTransaction> transactions =  
expenseService.getTransactionsForExpense(account, expense);
```

## Searching for Expenses

The following sample searches for expenses with a start date of January 2012.



```
Calendar cal = Calendar.getInstance();
cal.set(2012, 1, 1);
Date startDate = cal.getTime();

Calendar cal = Calendar.getInstance();
cal.set(2012, 1, 31);
Date endDate = cal.getTime();

DateCriterion dateCriterion = new
DateCriterion(Expense.EXPENSE_DATE).between(startDate, endDate);

List<Expense> expenses = platform.getExpenseService().search(new
SearchCriteria(dateCriterion));
```

#### 3.4.2.5.1.6 Groups

In TeamConnect, a [group](#) exists independently of its group account record, meaning that if you need to get or set any information for a group, you must use the group account. All TeamConnect users have their own group accounts (instances of `GroupAccount`) that include information such as the group display name and description. Use the `GroupService` class to access with groups in the API.

The API provides the following model interfaces for groups:

- The `Group` class, which contains the display name of the group. You can use this object to return the group name.
- The `GroupAccount` class, which contains all the fields for the group account in TeamConnect. You can use this object to get or set information for a user.

Because every display name is unique, the group name connects a group to its group account. When you create a new group, TeamConnect generates a key that you can also use to identify the group. Use the `entity.getPrimaryKey` method if you want to identify the group with its key.

Because group rights determine the code you can run, it is sometimes necessary to [check whether the current user is part of a group](#). You can get the current user using the `UtilityService.getCurrentUser()` method.

Refer to [the GroupService samples](#) for examples of working with groups in the API.

Use `GroupService` when working with group records.

Whenever a Service operation requires `GroupService`, set up `GroupService` using the following code:

```
GroupService groupService = platform.getGroupService();
```

## Creating a Group

The following sample creates a group. To create a group, you need to create a group account.

During group creation, you can use API interface methods to set objects in the group, such as its description.

```
//create a new group account with specified unique name and display name.
GroupAccount groupAccount = groupService.newGroupAccount("GroupUniqueName",
"GroupDisplayName");
groupAccount.setDescription("Description");
```

## Updating a Group

The following sample updates a group. When you update a group, you can use API interface methods to update the group account, such as its description.

```
//get group with specified name.
Group group = groupService.getGroupForName("GroupName");

//retrieve the group account from the group.
GroupAccount groupAccount = groupService.getGroupAccountForGroup(group);
groupAccount.setDescription("newDescription");
```

## Reading a Group

The following sample reads a group. When you read a group, you retrieve the group account so that you can make changes to it, such as updating or deleting it.

```
//read group and user accounts.
GroupAccount groupAccount = groupService.read(1L);
UserAccount userAccount = userService.read(1L);

//get the group and user from their corresponding accounts.
Group group = groupService.getGroupForGroupAccount(groupAccount);
User user = userService.getUserForUserAccount(userAccount);

groupService.addUserToGroup(group, user);
```

## Finding a Group

The following sample finds a particular group. You can use the unique to find the group.

```
//get group based on its unique key.
Group group = groupService.getGroupForKey("uniqueKey");
//get all groups.
List<Group> allGroups = groupService.getAllGroups();
//get all users for a group.
```

```
List<User> usersForGroup = groupService.getAllUsersForGroup(group);
//get all user accounts for group account.
GroupAccount groupAccount = groupService.read(1L);
List<UserAccount> userAccountsForGroupAccount =
groupService.getAllUserAccountsForGroupAccount(groupAccount);
//get all groups for a user.
User user = userService.findUserByUsername("username");
List<Group> groupsForUser = groupService.getGroupsForUser(user);
```

## Deleting a Group

The following sample deletes a group by deleting the group account.

```
GroupAccount groupAccount = groupService.read(1L);
groupService.delete(groupAccount);
//same as groupService.delete(groupAccount.getPrimaryKey());;
```

### 3.4.2.5.1.7 History Entries

All records have a **History** or **Narratives** page, where you can save an entry with a general comment or information about an update to the record. When [working with history entries](#) in the API, you view current entries or create new entries. Use the `HistoryService` class to work with history entries in the API.

The API provides the `History` model interface for information about specific entries.

Refer to [the HistoryService samples](#) for examples of working with histories entries in the API.

Use `HistoryService` when working with history records.

Whenever a Service operation requires `HistoryService`, set up `HistoryService` using the following code:

```
HistoryService historyService = platform.getHistoryService();
```

## Creating a History Entry

The following sample creates a history entry. During history creation, you can use API interface methods to set objects in the entry, such as the history description and the category.

```
// Creates an empty history object.
History newHistory = historyService.newHistory("history", parentEntity);
```

## Reading a History Entry

The following sample returns a history entry using its primary key.

```
// Reads the Entity object based on the primary key provided
historyService.read(primaryKey);
```

## Finding a History Entry

The following samples return all history entries for a particular entity and the most recent entry for an entity.

```
List<History> histories = historyService.getHistories(entity);
// Return most recent history with specified category and for a particular
entity
History lastHistory = historyService.getLastHistory(entity, "HIST_STAT");
```

## Deleting a History Entry

The following samples delete a history entity.

```
// Deletes the Entity object
historyService.delete(history);
// Deletes the Entity object based on the primary key provided
historyService.delete(primaryKey);
```

### 3.4.2.5.1.8 Invoices

[Invoices](#) are bills that your organization receives from vendors and that you can post against accounts and projects. When working with invoices in the API, use the `InvoiceService` class.

The API provides the following model interfaces for invoices:

- The `Invoice` class, which contains information about the invoice.
- The `InvoiceAdjustment` class, which is the base class for all invoice adjustment classes and contains basic information about the adjustment.
- The `InvoiceAdjustmentTarget` class, which contains the enums that indicate the amount being adjusted for the invoice summary.
- The `InvoiceAdjustmentType` class, which contains the enums that indicate the type of invoice summary adjustment.
- The `InvoiceCategoryAdjustment` class, which contains information about the invoice category adjustment, such as the task or expense code.
- The `InvoicePostingStatus` class, which contains the enums that indicate the posting status of the invoice.
- The `InvoiceHeaderAdjustment` class, which contains information about whether fees, expenses, or the total amount is the target of the header adjustment.

- The `InvoiceTimekeeperAdjustment` class, which contains information about the timekeeper and the project associated with the adjustment.
- The `InvoiceType` class, which contains the enums that indicate the type of invoice.
- The `LineItem` class, which contains information about a line item.
- The `LineItemType` class, which contains the enums that indicate the type of line item.
- The `AdjustmentInformation` class, which contains information about the adjustments.
- The `AdjustmentMethod` class, which contains the enums that indicate the method for adjusting the amount.

Refer to [the InvoiceService samples](#) for examples of working with invoices in the API.

Use `InvoiceService` when working with invoice records.

Whenever a Service operation requires `InvoiceService`, set up `InvoiceService` using the following code:

```
InvoiceService invoiceService = platform.getInvoiceService();
```

## Creating an Invoice with Line Items

The following sample creates an invoice and adds line items. During invoice creation, you can use API interface methods to set objects in the invoice record and line items, including the line item rate, quantity, discount, and total.

```
// vendor is an existing contact
Company vendor = platform.getContactService().readCompany(45L);

CalendarDate invoiceDate = new CalendarDate();
Invoice invoice = invoiceService.newInvoice("456", vendor, invoiceDate);
//the line item original rate.
BigDecimal originalRate = new BigDecimal("23.00");
//the line item original quantity.
BigDecimal originalQuantity = new BigDecimal("500");
//the line item original discount.
BigDecimal originalDiscount = new BigDecimal("17.56");
//the line item original total.
BigDecimal originalTotal = new BigDecimal("34805743570");

LineItem newLineItem = invoiceService.newExpenseLineItem(invoice, invoiceDate,
"LNI$_EXPS_OCEX_E100_E101", originalRate,
originalQuantity).setOriginalDiscount(originalDiscount).setOriginalTotal(originalTotal);
```

## Updating and Adjusting an Invoice

The following sample updates and adjusts an invoice. When you update an invoice, you can use API interface methods to set objects in the invoice record, including the start date. You also use API interface methods to adjust an invoice, such as for reducing an amount to a new amount.

```
CalendarDate periodStartDate = new CalendarDate();
invoice.setPeriodStartDate(periodStartDate);
AdjustmentInformation information =
new AdjustmentInformation(AdjustmentMethod.REDUCE_BY_AMOUNT, new
BigDecimal("45.00"));
invoiceService.adjustInvoiceFees(invoice, information);
```

## Adjusting Invoice Line Items

The following sample adjusts an invoice line item. When you adjust a line item, you can use API interface methods to adjust the line item, such as changing a line item amount to a new amount.

```
AdjustmentInformation information =
new AdjustmentInformation(AdjustmentMethod.NEW_AMOUNT, new BigDecimal("22.56"));
invoiceService.adjustLineItemRate(lineItemToAdjust, information);
```

## Reading an Invoice

The following sample reads an invoice record. When you read an invoice, you retrieve the invoice so that you can make changes to it, such as updating or posting to it.

```
// id of existing invoice.
long id = 4982340L;
Invoice invoice = invoiceService.read(id);
```

## Searching for an Invoice

The following sample searches for an invoice record.

```
// Returns all invoices that have an invoice date within the last 10 days
SearchCriteria criteria = new SearchCriteria(new
RelativeDateCriterion(Invoice.INVOICE_DATE).withinLast(10));
List<Invoice> invoices = invoiceService.search(criteria);
```

## Posting an Invoice

The following sample posts an invoice.

```
invoiceService.postInvoice(invoiceToBePosted);
```

## Voiding an Invoice

The following sample voids an invoice.

```
invoiceService.voidInvoice(invoiceToBeVoided);
```

## Deleting an Invoice

The following sample voids an invoice.

```
invoiceService.delete(invoiceToBeDeleted);
```

### 3.4.2.5.1.9 Tasks

[Tasks](#) are internal assignments that you can bill to accounts if necessary. When working with tasks in the API, use the `TaskService` class.

The API provides the following model interfaces for tasks:

- The `Task` class, which contains information about the task.
- The `TaskAssignee` class, which contains information about the user assigned to a task.
- The `TaskPriority` class, which contains the enums that indicate the priority of the task.
- The `TaskWorkStatus` class, which contains the enums that indicate the status of the task.

Refer to [the TaskService samples](#) for examples of working with tasks in the API.

Use `TaskService` when working with task records.

Whenever a Service operation requires `TaskService`, set up `TaskService` using the following code:

```
TaskService taskService = platform.getTaskService();
```

## Creating a Task

The following sample creates a task record. During task creation, you can use API interface methods to set objects in the task record, including the current assignee, associated project, and task rate.

```
Project project = platform.getProjectService().read(0L);
User assignee = platform.getUserService().findUserByUsername("task assignee");
Task task = taskService.newTask("This is a new task", assignee);
task.setActualHours(BigDecimal.TEN);
task.setRateAmount(BigDecimal.ONE);
task.setNote("Note: This is a new task");
task.setProject(project);
```

## Reading a Task

The following sample reads a task record. When you read a task, you retrieve the task so that you can make changes to it, such as updating or posting to it.

```
//the primary key of an existing task
long id = 12345678910L;
Task task = taskService.read(id);
```

## Searching for Tasks

The following sample searches for tasks with a start date of January 2012.

```
Date startDate = new SimpleDateFormat("yyyy-MM-dd").parse("2012-01-01");
Date endDate = new SimpleDateFormat("yyyy-MM-dd").parse("2012-01-31");
DateCriterion dateCriterion = new DateCriterion(new FieldPath(Task.START_DATE));

List<Task> tasksStartingInJanuary2012 = taskService.search(new
SearchCriteria(dateCriterion.between(startDate, endDate)));
```

## Posting a Task

The following sample posts a task record. When you post a task, you can use API interface methods to update the task at the same time.

```
//the primary key of an existing task
long id = 12345678910L;
Task task = taskService.read(id);
task.setCompletedDate(new CalendarDate());
task.setCompletedPercent(BigDecimal.valueOf(100));
taskService.postTask(task);
```

## Voiding a Task

The following sample voids a task record.

```
//the primary key of an existing task
long id = 12345678910L;
Task task = taskService.read(id);
taskService.voidTask(task);
```

## Reassigning a Task

The following sample reassigns a task to another user.

```
//the primary key of an existing task
```



```
long taskId = 12345678910L;
Task task = taskService.read(taskId);

//the primary key of an existing user
long userId = 12345678910L;
User user = platform.getUserService().findUserByUsername("new assignee");

taskService.reassign(task, user);
```

## Deleting a Task

The following sample deletes a task record.

```
//the primary key of an existing task
long taskId = 12345678910L;
Task task = taskService.read(id);

taskService.delete(task);
```

## Getting a List of Transactions for a Task

The following sample returns a list of transactions for a task record.

```
//the primary key of an existing task
long taskId = 12345678910L;
Task task = taskService.read(id);

List<AccountTransaction> transactions =
taskService.getTransactionsForTask(task);
```

## Getting a List of Transactions for a Task associated with an Account

The following sample returns a list of transactions for a task record that is associated with an account.

```
//the primary key of an existing task
long taskId = 12345678910L;
Task task = taskService.read(id);

//the primary key of an existing account
long accountId = 12345678910L;
Account account = accountService.read(id);
```

```
List<AccountTransaction> transactions =
taskService.getTransactionsForTask(account, task);
```

#### 3.4.2.5.1.10 Users

In TeamConnect, a [user](#) exists independently of its user account record, meaning that if you need to get or set any information for a user, you must use the user account. All TeamConnect users have their own user accounts (instances of `UserAccount`) that include information such as username, password, and account status. Each user account is also linked to a contact record (instance of `Contact`) through the `getContact()` method. Use the `UserService` class to access with users in the API.

The API provides the following model interfaces for users:

- The `User` class, which contains some read-only fields for the user. You can use this object to specify a user.
- The `UserAccount` class, which contains all the fields for the user account in TeamConnect. You can use this object to get or set information for a user.
- The `UserType` class, which contains the enums that indicate the type of user.

Because every username is unique, the username connects a user to its user account. When you create a new user, TeamConnect generates a key that you can also use to identify the user. Use the `entity.getPrimaryKey` method if you want to identify the user with its key.

Because you can change the user who is running the code, it is sometimes necessary to [identify the current user](#). You can get the current user using the `UtilityService.getCurrentUser()` method.

Refer to [the UserService samples](#) for examples of working with users in the API.

Use `UserService` when working with user records.

Whenever a Service operation requires `UserService`, set up `UserService` using the following code:

```
UserService userService = platform.getUserService();
```

## Creating a User

The following sample creates a user record. You can use the `newUser()` and `newUserAccount()` methods to create a user account and its associated user, the `saveUserAccount()` method to save the user, and the `getUserForUserAccount()` method to extract the user from its account.

During user creation, you can use API interface methods to set objects in the user record, including the corresponding contact, username, and password. Before creating a user, a corresponding contact record must already exist in TeamConnect or you must create a new contact using [ContactService](#). As shown in the sample, when creating a user record, you must also create the user account.

```
//read or create a contact
Contact contact = contactService.readPerson(1L);
```

```
//create a limited privilege user account.
UserAccount limitedUserAccount =
userService.newLimitedPrivilegeUserAccount("username", "password", contact,
true);
limitedUserAccount.setActive(true);
limitedUserAccount.setChangePasswordNextLogin(true);
limitedUserAccount.setShortDescription("Description for limited privilege user
account");
//get user from limited privilege user account.
User limitedUser = userService.getUserForUserAccount(limitedUserAccount);

//create a normal user account.
UserAccount normalUserAccount = userService.newNormalUserAccount("username",
"password", contact);
normalUserAccount.setActive(true);
normalUserAccount.setChangePasswordNextLogin(true);
normalUserAccount.setShortDescription("Description for normal user account");
//get user from normal user account.
User normalUser = userService.getUserForUserAccount(normalUserAccount);

//create a super user account.
UserAccount SuperPowerfulUserAccount =
userService.newSuperUserAccount("username", "password", contact);
SuperPowerfulUserAccount.setActive(true);
SuperPowerfulUserAccount.setChangePasswordNextLogin(true);
SuperPowerfulUserAccount.setShortDescription("Description for a super powerful
user account");
//get user from super user account.
User superUser = userService.getUserForUserAccount(SuperPowerfulUserAccount);
```

## Updating a User

The following sample updates a user record. When you update a user, you can use API interface methods to update fields in the user record, including the username, password, and active setting.

```
//read or create a user account.
UserAccount userAccount = userService.read(1L);
userAccount.setUsername("newUsername");
userAccount.setPassword("newPassword");
userAccount.setActive(true);
```

## Reading a User

The following sample reads a user record. When you read a user, you retrieve the user so that you can make changes to the user account, such as updating or deleting it.

```
//read a user account.  
UserAccount userAccount = userService.read(1L);  
//get the user from the user account.  
User user = userService.getUserForUserAccount(userAccount);
```

## Finding a User

The following sample find a particular user record. You can use a username to find the user.

```
User user = userService.findUserByUsername("username");
```

## Deleting a User

The following sample deletes a user record.

```
UserAccount userAccount = userService.read(1L);  
userService.delete(userAccount);  
//same as userService.delete(userAccount.getPrimaryKey());
```

### 3.4.2.5.2 Custom Objects

Custom objects are objects not included with TeamConnect Enterprise. The solution developer typically creates custom objects or they are included as part of a TeamConnect module, such as TeamConnect Legal. Custom objects are also called projects or matters.

Custom objects in the API have a [resource service class](#) for [projects](#) and [Involved parties](#). These classes use the [API model interfaces](#) to work with the system fields of custom objects. All system fields have get and set methods in their owner interfaces. You can also get or set [custom fields](#) on custom objects.

#### 3.4.2.5.2.1 Projects

[Projects](#) refer to custom objects with information that meets the business requirements of your organization. When working with projects in the API, use the `ProjectService` class.

The API provides the following model interfaces for projects:

- The `Project` class, which contains information about the project.
- The `ProjectAssignee` class, which contains information about a user assignees to a project.
- The `ProjectRelation` class, which contains information about a contact related to the Involved party.

Refer to [the ProjectService samples](#) for examples of working with projects in the API.

Use `ProjectService` when working with project records.

Whenever a Service operation requires `ProjectService`, set up `ProjectService` using the following code:

```
ProjectService projectService = platform.getProjectService();
```

## Adding an Embedded Project to a Parent Project

The following samples update a project record by adding an embedded project to the project. As shown in these samples, you can use `Project` methods to add the embedded project.

```
EmbeddedEntity embeddedProject = project.addEmbeddedEntity("unique code");
//or
EmbeddedEntity embeddedProject = project.addEmbeddedEntityWithName("unique
code", "name");
//or
EmbeddedEntity embeddedProject =
project.addEmbeddedEntityWithNumberString("unique code", "id number as a
string");
//or
EmbeddedEntity embeddedProject = project.addEmbeddedEntity("unique code",
"name", "id number as a string");
```

## Changing a Project's Phase

The following sample changes the phase of a project and updates the record. While making these changes, you can use API interface methods to update objects in the project record, such as the project name.

```
project.setContact(contact);
project.setMainAssignee(assignee);
projectService.changePhaseWithUpdate(project, "phase code of the phase to change
to");
```

## Reading Child Projects for a Record

The following sample returns the child projects of a parent project.

```
List<Project> childProjects = projectService.getChildProjects(project, "unique
code of the child Custom Object Definition");
```

## Searching for a Project

The following samples return an existing project. The second sample includes a `ProjectService.search()` method. This `search()` method is different from the

`ResourceService.search()` methods because the `ProjectService.search()` method includes the unique code of an existing project as a parameter.

```
//Reading a project
//The id initialized here is the id of an existing project
Long id = 1234567890L;
Project project = projectService.read(id);

//Searching for a project
List<Project> search("unique code", "search view key");
```

## Deleting a Project

The following sample deletes an existing project.

```
//The id initialized here is the id of an existing project
Long id = 1234567890L;
projectService.delete(id);
```

### 3.4.2.5.2.2 Involved Parties

[Involved parties](#) are project participants that may not be TeamConnect users. When working with Involved parties in the API, use the `InvolvedService` class.

The API provides the following model interfaces for Involved parties:

- The `Involved` class, which contains information about an Involved party.
- The `InvolvedRelation` class, which contains information.

Refer to [the InvolvedService samples](#) for examples of working with Involved parties in the API.

Use `InvolvedService` when working with Involved party records.

Whenever a Service operation requires `InvolvedService`, set up `InvolvedService` using the following code:

```
InvolvedService involvedService = platform.getInvolvedService();
```

## Adding an Involved Party to a Project

The following sample adds a new Involved party to a project.

During Involved party creation, you can set objects in the Involved party record, including the Involved party unique code and the associated project.

```
Involved newInvolved = involvedService.newInvolved(project, contact,
```

```
"INPA_EXPE");
```

## Reading an Involved Party

The following sample returns an Involved party.

```
//The id initialized here is the id of an existing project  
Long id = 1234567890L  
Involved involved = involvedService.read(id);
```

## Searching for an Involved Party

The following sample creates an Involved party record. The sample includes an `InvolvedService.search()` method. This `search()` method is different from the `ResourceService.search()` methods because the `InvolvedService.search()` method includes the unique code of an existing project as a parameter.

```
List<Involved> involveds = involvedService.search("unique code", "search view  
key");
```

## Deleting an Involved Party

The following samples delete an Involved party record.

```
involvedService.delete(involved);  
//or  
//The id initialized here is the id of an existing project  
Long id = 1234567890L;  
involvedService.delete(id);
```

### 3.4.2.5.3 Related Records

Any system or custom object record can have [related records](#). For example, contact and appointment records can have documents or histories related to them. In addition, project records can have related appointments, documents, histories, accounts, tasks, expenses, and child project records.

Do not confuse related objects with [sub-objects](#), such as a contact's addresses or assignees of a project. To retrieve sub-objects, you must use the appropriate methods in the sub-object's owner interface.

The methods that you use to get related records depend on the types of related records. [Projects have some methods](#) for obtaining related records. In certain cases, to retrieve some related records, you must [perform a search](#). Search for the following types of related records:

- The following related records of system objects and projects: tasks, accounts, appointments, expenses, invoices, history entries, and documents.

- Child accounts. While you can access the parent account from the child using `Account.getParentAccount()`, access child accounts of a parent by searching.

#### 3.4.2.5.3.1 Getting Related Records of Projects

When you edit a field with a related record in a project record, you retrieve that field using the `Project` interface. When you cannot edit a related record field in the project record, the `ProjectService` interface has methods for retrieving information related to a project. Use the following methods to obtain related records for a project:

- `ProjectService.getChildProjects()`—Retrieves all child projects records of a particular type related to a `Project` record.
- `ProjectService.getInvolved()`—Retrieves all Involved party records related to a `Project` record.
- `Project.getParentProject()`—Retrieves the parent project of a `Project` record.
- `Project.getEmbeddedProjects()`—Retrieves all embedded project records of a particular type related to a `Project` record.
- `Project.getRelations()`—Retrieves all projects related to a `Project` record. The API retrieves information about each relation using the `ProjectRelation` class.

**Note:** In some cases, when you need to narrow down the list of related records you are looking for, you may need to [search](#).

For complete samples in Java, see the [Looping through Related Records of Projects](#) and [Copying Values from Parent to Multiple Child Records](#) samples.

#### 3.4.2.5.3.2 Using Custom Fields and Primary Keys to Relate Records

Certain situations require you to uniquely identify a system object record, such as a task or history record, that does not have a relationship to the current object record. To work around this situation, you can add a custom field to a record and store the primary key of another record to create a relationship. When you need to retrieve a record, you can use the primary key in this field. Typically, users do not see or modify this field.

For example, you might have a rule that creates a history record when someone creates a particular task for a project. After the task is complete, another rule might retrieve the history record and update it. In this situation, the project's history and the task have no direct relationship. To create a relationship, you can use the first rule to create a custom field in the task object definition and store the primary key of the history record. When you need to retrieve that history record in the second rule, you can use the custom field value.

The following code snippet demonstrates how the first rule retrieves the primary key of the history record and stores it in the custom field of a task:

```
// Store the primary key of the history in the record's custom field
task.setNumberFieldValue("history", new BigDecimal(history.getPrimaryKey()));
```

The following code snippet demonstrates how a different rule retrieves the history record using its primary key in the custom field of a task record:



```
// Retrieving the primary key of the history from the record's custom field
Long primaryKey = record.getNumberFieldValue("history").longValue();

// Retrieve the related history record
History retrievedHistory = platform.getHistoryService().read(primaryKey);
```

#### 3.4.2.5.4 Sub-Objects

[Sub-objects](#) contain information that exists only within a record. You cannot access the sub-objects outside that record. For example, [categories](#), appointment attendees, contact addresses, invoice line items, and project assignees are all sub-objects.

Some sub-objects are also associated with the [system lookup tables](#) that define the individual types of sub-objects. For example, the **Fax Type** system lookup table defines the different types of fax numbers that you can add to a contact record.

### Sub-Objects in the API

Each sub-object is part of a main object. You retrieve sub-objects using a get method in the main object class. For example, if you want to retrieve a list of attendees for an appointment, use the `Appointment.getAttendees` method. If you want to retrieve the phone numbers for a contact, use the `Contact.getPhoneNumbers` method.

You can also add and remove sub-objects using the corresponding methods in the owner interfaces. For example, if you want to add an attendee to an appointment, use the `Appointment.addAttendee` method. If you want to remove an attendee from an appointment, use the `Appointment.removeAttendee` method.

### Sub-Object Classes

The following table includes all the frequently used sub-objects and their classes.

Sub-Objects of Main Team Connect Objects	
Main Object Class	Specific Sub-Object Classes
Account	(None)
Appointment	AppointmentAttendee
Contact	Address Email FaxNumber InternetAddress PhoneNumber DefaultRate TaskRate InvoiceTaskRate

	ContactRelation Skill
Document	(None)
Expense	(None)
History	(None)
Involved	InvolvedRelation
Invoice	LineItem
Project	ProjectAssignee Phase ProjectRelation
Task	TaskAssignee

#### 3.4.2.5.4.1 Project and Task Assignees

Assignees are sub-objects of users assigned to a project or task. The methods for assignees in respective `Task` and `Project` interfaces reflect the differences between task and project assignees.

##### Project and Task Assignee Differences

Project Assignees	Task Assignees
Project assignees are instances of the <code>ProjectAssignee</code> interface.	Task assignees are instances of the <code>TaskAssignee</code> interface.
Projects can have multiple or no assignees. Projects with assignees always have one main assignee.	Tasks must have a current assignee and they can only have one assignee at a time.  By default, the user who creates a task record is set as the assignee of that task.
A project has a list of assignees that includes the main assignee, other active assignees, and inactive (unassigned) assignees.	A task includes a list of all previously assigned users as well as the current assignee.
You can remove (or unassign) a project assignee without adding a new one.	When you reassign a task, it adds the new assignee as the current assignee and unassigns the previously set assignee.
Each project assignee has an assignee role.	Task assignees do not have roles.

Instances of `ProjectAssignee` or `TaskAssignee` are always associated with a user. When you add an assignee to a task or project, you must provide a user in order to create the assignee.

For example, the `Project.addAssignee()` method assigns a project to an existing user, as shown in the following sample, and returns `ProjectAssignee` with the assigned user.

```
project.addAssignee("ROLE_TREE_POSI", user);
```

#### 3.4.2.5.4.2 Contact Sub-Objects

Contacts have many sub-objects, including the sub-objects with the following interfaces:

- `Address`
- `Email`
- `FaxNumber`
- `InternetAddress`
- `PhoneNumber`
- `DefaultRate`
- `TaskRate`
- `InvoiceTaskRate`
- `ContactRelation`
- `Skill`

In some sub-object lists for a contact, one of the list items is the primary item. Specific methods for getting and setting the primary sub-objects are available in the `Contact` interface. For example, the phone and fax sub-objects have a primary phone number and a primary fax number. You can retrieve these primary sub-objects with the `getPrimaryPhoneNumber()` and `getPrimaryFaxNumber()` methods.

All of the sub-objects of contact have types, which describe the values you enter. Most sub-objects store these types in the [system lookup tables](#), with the exception of the rate sub-objects. The rate sub-objects (`DefaultRate`, `TaskRate`, and `InvoiceTaskRate`) are categories and instances of `CategoryDefintion`.

#### 3.4.2.5.5 Categories

[Categories](#) have the following two purposes:

- They provide a simple way to organize and search for records.
- They organize custom fields by associating each custom field with a category. For [custom fields](#) to appear at all times, you can associate them with the root category. Users can display or hide custom fields by selecting or deselecting the associated categories.

In every object definition, the starting point for the category hierarchy is the root category. The hierarchy also includes a primary category, which the user sets manually. Upon record creation, the root category is set as the primary category.

## Categories in the API

In the API, you represent categories with the following interfaces:

- `CategoryService`—The general category class that you use to obtain information about the categories for an object definitions or a tree position.

**Note:** Use the following code to [access `CategoryService`](#):  

```
CategoryService categoryService  
= platform.getCategoryService();
```

- `Category`—The class that you use to obtain information about added categories and their hierarchical positions in a record, including any children and parent categories. A `Category` is an instance of a `CategoryDefinition`.
- `CategoryDefinition`—The class that you use to obtain information about categories that exist for an object and their associated custom fields.
- `EnterpriseEntity`—The class that includes methods that system objects and other interfaces extend. This class includes the `getAllCategories()`, `getPrimaryCategory()`, `hasCategory()`, and `setPrimaryCategory()` methods.
- Object model interfaces with categories—These class also have methods for getting and setting categories. For example, the `Invoice` class has the following methods: `addCategory()`, `getCategories()`, and `removeCategory()`. Other object model interfaces may have different category methods.

**Note:** In instances of an *Involved object*, categories are known as "involved roles."

## Category Tree Positions

A full tree position uniquely identifies a category. The root category's full tree position is the same as the unique code of the object definition. The full tree position of non-root categories is a combination of the following with underscores as delimiters:

- The full tree position of the category's parent category
- The partial tree position of the category itself

For example, category B with partial tree position BBBB is under category A with partial tree position AAAA, which is under the root category of the Appointment object definition with the unique code APPT. As a result, the full tree position of category B is APPT\_AAAA\_BBBB, and the full tree position of category A is APPT\_AAAA.

### 3.4.2.5.5.1 Adding or Deleting Categories in Records

If you want to select a category for a record, you can use the `addCategory()` method from the object's model class. The following code sets External, with a tree position of CONT\_EXTE, as a category for a contact:

```
contact.addCategory("CONT_EXTE");
```

If you want to deselect a category from a record, you can use the `removeCategory()` method from the object's model class. The following code removes External from the list of selected categories for a contact:

```
contact.removeCategory("CONT_EXTE");
```

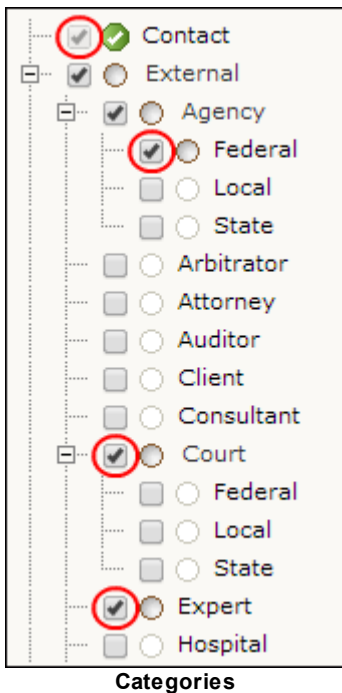
For a sample, see [Updating Added Categories](#).

#### 3.4.2.5.5.2 Returning Categories for a Record

If you want to know the categories that are selected for a record, you can use the `EnterpriseEntity.getAllCategories()` method. The following code returns all selected categories, including the parent categories, for a contact:

```
List<Category> allCategories = contact.getAllCategories();
```

If you want to know the lowest-level categories that are selected for a record, you can use the `getCategories()` method from the object's model class. The lowest-level categories include all the categories that do not have any child categories. If a selected category has lower-level categories that are not selected, `getCategories()` only returns the selected category. For example, the circled categories in the following image are the lowest-level selected categories.



If the contact record has the categories selected in the previous image, the following code returns the **Contact**, **Federal**, **Court**, and **Expert** categories for the contact:

```
List<Categories> categories = contact.getCategories();
```

In this example, the code does not return **Agency** because **Federal** is a lower level category, and it returns **Court** because none of the lower level categories for **Court** are selected.

#### 3.4.2.5.5.3 Getting and Setting Primary Categories

If you want to set the primary category of an object, you can use the `EnterpriseEntity.setPrimaryCategory()` method. The following code sample provides an example of how to create a new contact and set its primary category:

```
// Retrieve the contact service
ContactService contactService = platform.getContactService();

// Create a new, blank contact. The vendor record is a contact which has
// previously been created. Set some categories for the contact record
Contact contact = contactService.newCompany("vendor");
contact.setPrimaryCategory("CONT_EXTE_LAWF");
contact.addCategory("CONT_EXTE_VEND");

// Save the contact to the database
contactService.create(contact);
```

If you want to return the primary category of an object, you can use the `EnterpriseEntity.getPrimaryCategory()` method. The following code sample returns the primary category of a contact:

```
Category primaryCategory = contact.getPrimaryCategory();
```

For a complete sample, see the [Checking the Default Role in Related Involved Records sample](#).

#### 3.4.2.5.5.4 Checking Records for Specific Categories

To return whether a record has a particular category, you can use the `hasCategory()` method and the full tree position of the category. The following code sample checks if a contact has the "Vendor" category:

```
boolean isVendor = contact.hasCategory("CONT_EXTE_VEND");
```

For a sample, see the [Checking Lists of Added Categories sample](#).

#### 3.4.2.5.5.5 Custom Fields

[Custom fields](#) exist for capturing your organization's custom information in TeamConnect. You create custom fields within object definitions under [categories](#) in an object. Within records in TeamConnect, users make custom fields available by selecting the categories with that custom field. Custom fields that appear at all times are under the root category.

In the TeamConnect API, the following interfaces represent custom field information:

- `EnterpriseEntity` and `Project`—You use methods in these classes to [get and set custom fields](#), depending on the custom field type.
- `CustomField`—You use this class to retrieve information about a custom field. You mainly use this custom field and its subclasses when [constructing search criteria](#).

You can get and set most custom fields using methods in the `EnterpriseEntity` interface. The `Project` interface has methods to get and set custom fields of `Involved` type.

To get or set a value in a custom field, you need the following information to locate the necessary field:

- (If the field is not under a root category) The [full tree position of the category](#) for the field.  
You can find the full tree position of a category in the UI of the TeamConnect Setup.
- The name of the custom field, not its label.  
You can find field names on the **Custom Fields** tab of the corresponding object definition in the TeamConnect user interface.
- The type of custom field.  
Depending on the field type, you use [different methods to get or set the value](#) in the custom field.
- (If the custom field is of type List) The full tree position of the [lookup table item in the list](#).

The following table lists all the methods from the `EnterpriseEntity` and `Project` interfaces that allow you to get or set custom field values according to field type.

**Custom Field Types and Method Names**

Field Type	get and set Methods	Class(es) with Methods
<b>Check Box (Boolean)</b>	<code>getBooleanFieldValue()</code> <code>setBooleanFieldValue()</code>	<code>EnterpriseEntity</code>
<b>Date</b>	<code>getDateFieldValue()</code> <code>setDateFieldValue()</code>	<code>EnterpriseEntity</code> <code>CalendarDate</code>
<b>Involved</b>	<code>getInvolvedFieldValue()</code> <code>setInvolvedFieldValue()</code>	<code>Project</code>
<b>List</b>	<code>getLookupFieldValue()</code> <code>setLookupFieldValue()</code>	<code>EnterpriseEntity</code>
<b>Memo (Text)</b>	<code>getMemoFieldValue()</code> <code>setMemoFieldValue()</code>	<code>EnterpriseEntity</code>
<b>Number</b>	<code>getNumberFieldValue()</code> <code>setNumberFieldValue()</code>	<code>EnterpriseEntity</code>
<b>Custom Object</b>	<code>getProjectFieldValue()</code> <code>setProjectFieldValue()</code>	<code>EnterpriseEntity</code>

<b>Text</b>	<pre>getTextFieldValue() setTextFieldValue()</pre>	EnterpriseEntity
-------------	--	------------------

You can retrieve custom fields from any of the objects that extend `EnterpriseEntity`, as shown in the following code:

```
record.getCalendarDateFieldValue("fieldName");
record.getLookupFieldValue("fieldName");
```

If a custom field is part of a non-Root category, you must include the tree position when retrieving the value of the custom field:

```
record.getBooleanFieldValue("TREE_POSI", "fieldName");
record.getLookupFieldValue("TREE_POSI", "fieldName");
```

In the following code sample, you can get the value of the **LossAmount** custom field of type Number, which has a category with the full tree position `CLAM_FEAT`:

```
record.getNumberFieldValue("LossAmount", "CLAM_FEAT");
```

For a complete sample, see the [Comparing Custom Field Qualifiers sample](#).

You can set different kinds of values for custom fields of objects that extend `EnterpriseEntity`. To set a value in a list custom field, you can enter a field name and the value using `EnterpriseEntity` methods, as shown in the following code:

```
record.setBooleanFieldValue(fieldName, true);
record.setLookupFieldValue(categoryTreePosition, fieldName, "ITEM");
```

In the following code sample, you can set the value of the memo field **OtherDistractions** and use its category full tree position of `CLAM_FEAT`:

```
record.setMemoFieldValue("CLAM_FEAT", "OtherDistractions", "Claimant was reading a map.");
```

For a complete sample, see the [Checking and Setting Values in Custom Fields of Type List sample](#).

#### 3.4.2.5.6 Lookup Table Fields

Lookup tables provide options in List fields of TeamConnect. Each option in a List field is a lookup table item. TeamConnect has two types of lookup tables:

- System lookup tables that TeamConnect includes by default. You can modify them by adding and removing items, but you cannot delete them.



- Custom lookup tables that you can create as custom fields. Unlike system lookup tables, you can delete custom lookup tables. However, determine whether other design components are dependent on the custom lookup table before deleting it.

Because items in a lookup table have hierarchical relationships between them, you can use full tree positions to uniquely identify all lookup table items. Tree positions for [system lookup tables](#) and [custom lookup tables](#) are different.

## Lookup Tables in the API

In the API, you represent lookup tables and their items with the following interfaces:

- [LookupItemService](#)—Includes methods that you use to return information about lookup tables and items. For example, you can use the `getCustomLookupItems()` method to retrieve a list of items in a custom lookup table:

```
lookupItemService.getCustomLookupItems("CODE");
```

**Note:** Use the following code to [access LookupItemService](#): `LookupItemService lookupItemService = platform.getLookupItemService();`

- [LookupItem](#)—Includes methods for retrieving information about lookup table items. For example, the following code uses the `getStoredValue()` method to retrieve the stored value of a lookup table item:

```
contact.getLookupFieldValue("categoryFullTreePosition",  
"fieldName").getStoredValue();
```

- [EnterpriseEntity](#)—Includes methods that system objects and other interfaces extend. For example, you can use the `getLookupFieldValue()` method with the `Contact` model interface to return the value for a custom lookup field:

```
contact.getLookupFieldValue(String treePosition, String fieldName);
```

## LookupItemService Interface

Most `LookupItemService` methods return information about the following types of lookup tables and lookup table items:

- System lookup tables—You can retrieve all system lookup tables. Each system lookup table in TeamConnect has [a LookupItemService method](#). For example, use `lookupItemService.getContactPhoneNumberTypes()` to return a list of the lookup table items for the phone number field in a contact.
- Custom lookup tables—You can retrieve a custom lookup table and item. `getCustomLookupItemForTreePosition()` returns one item in a lookup table, and `getCustomLookupItems()` returns the list of items in a custom lookup table.
- Currency—You can retrieve and update the currency lookup table. `getCurrencyItemList()` returns the list of currencies in the lookup table, and `updateCurrencyRates()` adds a currency entry to the lookup table.

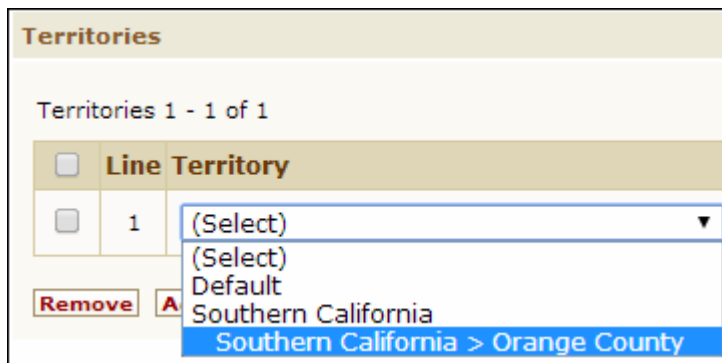
- Expenses—You can retrieve information about an expense category. `readLineItemExpenseCode()` returns an item in the expense category lookup table, and `readLineItemExpenseUnitPrice()` and `readExpenseRateScaleValue()` return rates and units associated with a category.
- Document types—You can retrieve document types. For example, `getAllDocumentTypes()` returns the list of document types in a lookup table.

#### 3.4.2.5.6.1 System Lookup Table Tree Positions

The full tree position of a system lookup table item is a combination of the following codes with underscores as delimiters:

- The full tree position of the lookup item's parent item.
- The partial tree position of the item.

In the following image, the **Territory Type** lookup table in contact records has the code of TERR. If **Orange County** has the partial tree position OCNT and **Southern California** has the partial tree position SOCA, the full tree position of Orange County is TERR\_SOCA\_OCNT.



**Territory System Lookup Table**

The following code samples provide examples of how you can use the tree position to retrieve a system lookup table item.

```
LookupItemService lookupItemService = platform.getLookupItemService();

// Retrieving the "Apple" item within the Fruits custom table
CustomLookupItem apple =
lookupItemService.getCustomLookupItemForTreePosition("FRUI_APPL");

// Retrieve all items in the Fruits custom table
List<CustomLookupItem> fruits = lookupItemService.getCustomLookupItems("FRUI");

// Retrieving the "Mobile" item within the Phone Type system table;
SystemLookupItem mobile =
lookupItemService.getSystemLookupItemForTreePosition("PHON_MOBI");
```

The following table includes all system lookup tables, their associated `LookupItemService` method, and their unique codes. Use the unique code of each system lookup table at the beginning of the full

tree position for a lookup table item. The table also lists API model interfaces associated with the lookup tables. If the lookup tables are not associated with a string field, the API class column does not apply.

#### System Lookup Table Names

Friendly name	Associated LookupItemService method	Associated API class	Unique code
<b>Activity Item</b>	getActivityItems()	N/A	ACTI
<b>Address Type</b>	getContactAddressTypes()	Address	ADDR
<b>Contact Relation Type</b>	getContactRelationTypes()	ContactRelation	CONR
<b>Country Item</b>	getCountries()	N/A	COUN
<b>Email Type</b>	getContactEmailTypes()	Email	MAIL
<b>Fax Type</b>	getContactFaxNumberTypes()	FaxNumber	FAXX
<b>Internet Address Type</b>	getContactInternetAddressTypes()	InternetAddress	INET
<b>Invoice Rejection Reason</b>	getInvoiceRejectionReason()	N/A	INRR
<b>Phone Type</b>	getContactPhoneNumberTypes()	PhoneNumber	PHON
<b>Project Assignee Type</b>	getProjectAssigneeRoles()	ProjectAssignee	Unique code of the respective custom object definition.
<b>Project Relation Type</b>	getProjectRelationTypes()	ProjectRelation	PRJR
<b>Resource Type</b>	getAppointmentResourceTypes()	N/A	RESO
<b>Skill Type</b>	getContactSkillTypes()	Skill	SKIL
<b>Territory Type</b>	getContactTerritoryTypes()	N/A	TERR
<b>State Item</b>	getStatesForCountryCode()	N/A	STAT

## 3.4.2.5.6.2 Custom Lookup Table Tree Positions

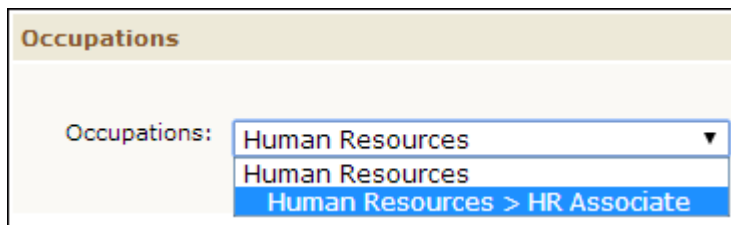
When you create a custom lookup table, you specify the unique code of the lookup table in the form of a 4-character alphanumeric combination. For example, the unique code of the **Occupations** custom lookup table might be OCCU.

The full tree position of the root node includes the unique code of the custom lookup table and ROOT. For example, the full tree position of the root node of the **Occupations** lookup table is OCCU\_ROOT.

The full tree position of non-root custom lookup table items include the following with underscores as delimiters:

- The unique code of the custom lookup table.
- The partial tree position of the root node (ROOT).
- The partial tree position of any parent items.
- The partial tree position of the item.

For example, the **Occupations** custom lookup table in the following image has an **HR Associate** item with a partial tree position of HRAS and a **Human Resources** item with the partial tree position HURE. As a result, the full tree position of the **HR Associate** item is OCCU\_ROOT\_HURE\_HRAS.



Occupations Custom Lookup Table

The following code samples provide examples of how you can use the tree position to retrieve a custom lookup table item.

```
CustomLookupItem customItem =
lookupItemService.getCustomLookupItemForTreePosition("treePosition");

//Retrieving "Big Budget Hits" custom item in the "Released Movies" custom table
//Table Code: RMOV, Item Tree Position: BBHT
CustomLookupItem movie =
lookupItemService.getCustomLookupItemForTreePosition("RMOV_ROOT_BBHT");

//Retrieving "Comedy" child item of the "Big Budget Hits" custom item
//Table Code: RMOV, Item Tree Position: CMDY
CustomLookupItem comedy =
lookupItemService.getCustomLookupItemForTreePosition("RMOV_ROOT_BBHT_CMDY");
```

### 3.4.2.5.6.3 Getting or Setting Values in List Custom Fields

The following sections demonstrate how to get or set values in List custom fields.

## Getting Values in Lists

The `EnterpriseEntity.getLookupFieldValue()` method retrieves the item selected for a List custom field. You can use this value, which is a `LookupItem` object, to retrieve the full tree position of the lookup item. The following code provides an example of how you can retrieve the full tree position of a List custom field in the contact object:

```
contact.getLookupFieldValue("APPT_AAAA_BBBB", "HR Associate").getKey();
```

In addition to contacts, you can use this code for any [objects that extend `EnterpriseEntity`](#), such as projects or expenses.

## Setting Values in Lists

When you set values in List custom fields, you must include the following information:

- The [full tree position of the category](#) for the custom field.
- The name of the List custom field.
- The [full tree position of the item](#) in the lookup table.

You can use the `EnterpriseEntity.getLookupFieldValue()` without the full tree position of the category, as shown in the following code sample.

```
contact.setLookupFieldValue("HR Associate", "OCCU_ROOT_HURE_HRAS");
```

If you know the tree position, you can also use it as one of the parameters for the method:

```
contact.setLookupFieldValue("APPT_AAAA_BBBB", "HR Associate",  
"OCCU_ROOT_HURE_HRAS");
```

For a complete sample of custom list fields in Java, see [Checking and Setting Values in Custom Fields of Type List](#).

### 3.4.2.6 Common API Functions

If you need to perform common functions using the API, you can use the `UtilityService` interface. This interface encompasses miscellaneous methods that you might use throughout your code. These methods do not set anything in TeamConnect, but instead they retrieve information or run your code as a different user.

**Note:** Use the following code to [access `UtilityService`](#):  

```
UtilityService utilityService =  
platform.getUtilityService();
```

`UtilityService` includes methods for the following types of actions:

- **Converting or changing formats**—For example, you can use the `booleanValueForString()` method to convert a string to a boolean and the `formatDateByUserSetting()` method to format the date based on the user's date settings, as shown in the following code samples.

```
UtilityService utilityService = platform.getUtilityService();

//Converting strings to booleans and returning boolean true.
utilityService.booleanValueForString("YES");
utilityService.booleanValueForString("TRUE");
utilityService.booleanValueForString("1");

//Converting strings to booleans and returning boolean false.
utilityService.booleanValueForString("NO");
utilityService.booleanValueForString("FALSE");
utilityService.booleanValueForString("0");

//Formatting the date for the user based on the current settings for that
User user = utilityService.getCurrentUser();
Date today = new Date();
String formattedDate = utilityService.formatDateByUserSetting(today, user);
```

- **Retrieving information based on the current day, time, or user**—For example, you can also use the `getCurrentUser()` method [to retrieve the user object for the user currently running the code](#). You can also use the `getBeginningOfToday()` method to return the start of the current day in the current user's time zone, as shown in the following code sample.

```
Date date = utilityService.getBeginningOfToday();
```

- **Retrieving information about single-project object definitions**—For example, you can use the `getSingletonForObjectDefinition()` method to return the single project for an object definition, as shown in the following code sample.

```
// Unique 4-letter code for custom project.
String projectDefinitionUniqueCode = "PROJ";
Project singletonProject =
utilityService.getSingletonForObjectDefinition(projectDefinitionUniqueCode);
```

- **Retrieving system information**—For example, you can use the `getDatabaseInfo()` to return information about the TeamConnect database, as shown in the following sample.

```
DatabaseInfo databaseInfo = utilityService.getDatabaseInfo();
String serverName = databaseInfo.getServerName();
String databaseName = databaseInfo.getDatabaseName();
Date databaseCreatedOn = databaseInfo.getDatabaseCreatedOn();
String version = databaseInfo.getVersion();
DatabaseProduct product = databaseInfo.getDatabaseProduct();
```

- **Comparing two different days**—For example, you can use the `isDayAfter()` method to determine whether one date occurs before or after another date, as shown in the following sample about comparing the created on date for two different invoices.

```
//find two invoices
Long invoiceId1 = 1L;
Invoice invoice1 = invoiceService.read(invoiceId1);
Long invoiceId2 = 2L;
Invoice invoice2 = invoiceService.read(invoiceId2);
//Determine if invoice2 was created after invoice1
if (utilityService.isDayAfter(invoice2.getCreatedOn(),
invoice1.getCreatedOn()))
{
    //do something
}
```

- **Sorting a list**—You can use the `sortUsingPath()` method to sort a list, as shown in the following sample.

```
List<User> users = platform.getUserService().getAllUsers();
//sort users based on the contact primary city. In ascending order.
FieldPath pathToContactsPrimaryCity = new
FieldPath(User.CONTACT).add(Contact.PRIMARY_ADDRESS).add(Address.CITY);
boolean isAscending = true;
List sortedUsers= platform.getUtilityService().sortUsingPath(users,
pathToContactsPrimaryCity, isAscending);
```

- **Running code as a different user**—You can use the `runAsSystemUser()` method [to run code as the system user](#). You can also use the `runAsUser()` method to run code as a specific user, as shown in the following sample.

```
//get current user
User user = utilityService.getCurrentUser();
//run a call as the user
utilityService.runAsUser(user, new Callable() {
    @Override
    public void call() {
        //do something
    }
});
```

#### 3.4.2.6.1 Running Code as the System User

Because every record is subject to security, change to a user you know has the appropriate rights to ensure that your code runs. To execute code regardless of your [functional or record security rights](#), you can run code as if you are the system user.

**Note:** If you want code to fail if the current user does not have the appropriate rights, you do not need to run code as the system user.

The system user, which has the username "system" and contact name "system, system" includes the following characteristics:

- Has all rights at all times.
- Cannot be added to a group.
- Cannot be denied security rights in a record's **Security** tab.
- Can always execute any method in the API.

To run code as the system user inside a call, use `utilityService.runAsSystemUser()`.

`UtilityService` includes four `runAsSystemUser()` methods. The method you use depends on whether you want to return data and/or execute additional code if the call throws an exception.

The following code snippet indicates how to execute code as a system user without returning any results:

```
utilityService.runAsSystemUser(new Callable() {
    public void call() {
        // Place your code here.
    }
});
```

The following code snippet indicates how to execute code as a system user and return data:

```
String username = utilityService.runAsSystemUser(new
CallableWithReturn<String>() {
    public String call() {
        String answer = "";
        // Place your code here.
        return answer;
    }
});
```

The following code snippet indicates how to execute code as a system user without returning any results and also executing additional code if the call throws an exception:

```
try {
    utilityService.runAsSystemUser(new
CallableWithException<FileNotFoundException> callable() {
        public void call() throws FileNotFoundException {
            throw new FileNotFoundException();
        }
    });
}
```



```
} catch (FileNotFoundException e) {  
    System.out.println("Couldn't find file!");  
}
```

You can combine the last two code snippets to execute code that returns data or that executes additional code in the case of an error.

#### 3.4.2.6.2 Getting Current User Information

The current user is the user who is triggering a rule or invoking an action. When you want code to execute regardless of the current user's rights, you change the current user to [the system user](#). Because the user running the code can change, you may need to retrieve the current user.

To retrieve the user object for the current user, use the `utilityService.getCurrentUser()` method.

The following code sample retrieves the username of the current user. This code is useful when the current user is a required condition. You can use the username to determine whether the current user meets the condition.

```
String username = UtilityService.getCurrentUser().getUsername();
```

The following code sample retrieves the contact record of the current user.

```
User currentUser = utilityService.getCurrentUser();  
Contact userContact = userService.getUserAccountForUser(user).getContact();
```

The following code sample retrieves the groups to which the current user belongs.

```
utilityService.getCurrentUser().getGroupList();
```

The following code sample checks the list of groups for the current user to find out if the user is part of a specific group. You can use this code as a template to check other conditions in addition to the group membership of the current user.

```
// Initialize services.  
UtilityService utilityService = platform.getUtilityService();  
GroupService groupService = platform.getGroupService();  
  
//get group list for user.  
User currentUser = utilityService.getCurrentUser();  
List<Group> groupsForUser = groupService.getGroupsForUser(currentUser);  
  
//Check to see if specific group is part of the list.  
for (Group group : groupsForUser) {  
    if (group.getDisplayName().equals("specifiedGroupName")) {  
        return false;  
    }  
}
```

```
    }  
  }  
  return true;  
}
```

### 3.4.2.7 Security Rights

TeamConnect includes the following types of security settings:

- [Record security](#)—The security settings of a record. You handle record security from the **Security** page in a record.
- [Functional security](#)—The rights of each user or group account to perform specific operations or functions on TeamConnect objects. Functional security affects all records of a specific object type.

**Note:** Use the following code to [access SecurityService](#):  
`SecurityService SecurityService = platform.getSecurityService();`

#### 3.4.2.7.1 Record Security

You can use [record security](#) in the API to update record permissions and determine whether a user or group can access a record. The `SecurityAccess` interface includes all the methods for retrieving and updating permissions for users and groups. To call these methods, use the `GroupSecurityAccess` and `UserSecurityAccess` interfaces, which extend `SecurityAccess`.

Updating the security of a record has two parts:

- **Creating the security object**—Before you can update the security of a record, you must create a security object for the user or group. This security object is specific to the record and the user or group. When you create this security object, you also allow or deny the user or group read access to the record. Create this object with the `allowUserAccess()`, `allowGroupAccess()`, `denyUserAccess()`, or `denyGroupAccess()` methods in the `EnterpriseEntity` interface.

**Note:** A record can only have one security object for each user or group. If you try to create a security object for a user or group that already exists, you receive an error.

- **Updating security**—After you create the security object, you can use the `SecurityAccess` methods through the `UserSecurityAccess` and `GroupSecurityAccess` interfaces to update permissions. If you want to update a security object that already exists, use the `getUserSecurityAccessList()` or `getGroupSecurityAccessList()` methods in `EnterpriseEntity` to retrieve a record's security object from a list.

## Allowing Security Access

To allow access to a record, create the security object and give the user or group read access.

For example, if you are adding group rights to a record, you use the `EnterpriseEntity.allowGroupAccess()` method to create a security object and grant read access to the group. After you retrieve the group, you can use methods part of the `GroupSecurityAccess` interface to update permissions, as shown in the following code sample.

```
public void allowGroupSecurityAccessObject(Contact record) {
    // Creates a record-level security object which (by default) grants the
    // group read access to the record
    Group group = platform.getGroupService().getGroupForName("test group");
    GroupSecurityAccess gsa = record.allowGroupAccess(group);

    // To allow more permissions, specify them individually
    gsa.addUpdate().addDelete().addChangeSecurityAccess();
}
```

If group rights for a record already exist, you cannot use the `allowGroupAccess()` method to retrieve the security object. If you want to update the group rights of the record, you must retrieve it from the list of records for the group's security objects using the `getGroupSecurityAccessList()` method.

## Denying Security Access

To deny access to a record, create the security object and deny the user or group specific rights.

For example, if you want to deny group access to a record, you must create the security object using the `EnterpriseEntity.denyGroupSecurityAccess()` method. In addition, when you create the group, you can specify which security permissions you want to deny for the group's access to that record, as shown in the following code sample:

```
public void denyGroupSecurityAccessObject(Contact record) {
    //Creates a record-level security object which includes the specified
    //permissions
    Group group = platform.getGroupService().getGroupForName("demo group");
    GroupSecurityAccess gsa = record.denyGroupAccess(group, true, true, true,
    true);

    // Check if the object contains the delete permission
    if (gsa.isDelete()) {
        record.setNote("Can be deleted by " + group.getDisplayName());
    }
}
```

**Note:** Specifying *true* for the Boolean parameters of the `denyGroupAccess()` method denies access to a particular right, but specifying *false* does not allow or deny access.

If you want to update a group's rights after creating the security object, you can use the `SecurityAccess` methods through the `GroupSecurityAccess` interface.

### 3.4.2.7.2 Functional Security

Users with similar functions are typically organized into user groups (instances of `GroupAccount`). The [rights assigned to the group](#) reflect the functional security rights of the users in the group. You

cannot modify functional rights using the TeamConnect API, only through the user interface. If you want to know if a user has a particular functional right, you can use the following interfaces:

- `SecurityService`—Provides methods that can check a user's rights within TeamConnect.
- `OperationType`—Provides enumerations that include all the rights a user can have.

Some of the `SecurityService` methods use the `OperationType` class. For example, the following code checks if a user can create an account:

```
boolean hasEntityOperationRights =
platform.getSecurityService().isEntityOperationEnabled(user,
OperationType.CREATE, "ACCT");
```

In the previous code snippet, the `securityService.isEntityOperationEnabled()` method checks if the given user, called **anyUser**, has the necessary rights to create an account object. If the user has the rights, this code returns `true`.

In addition to a method for checking rights at the object level, `SecurityService` also includes methods for checking whether a user has rights to categories, tools, and user invoked rules:

- `isCategoryOperationEnabled()` checks rights for categories.
- `isToolAccessEnabled()` checks rights for system or custom tools.
- `isUserInvokedAccessEnabled()` checks rights for user invoked rules.
- `isEntityOperationEnabled()` checks rights for objects.

### 3.4.2.8 System and User Settings

You can use the API to retrieve and update system and user settings. [System settings](#) are settings an administrator updates from the **Admin Settings**. [User settings](#) are settings that a user updates on their **Preferences** page. When working with these settings in the API, you use `SettingsService` with the `SettingKey` enums.

## SettingsService Interface

`SettingsService` provides methods to retrieve and update system and user settings. This interface includes the following types of methods:

- *Methods that retrieve the value for a setting*—Most `SettingsService` methods retrieve a value for a specific setting. For example, the following method returns the filename of the custom logo image.

```
settingsService.getSystemCustomLogoImage();
```

For settings that do not have a specific get method, you can use the `getSystemSetting()` or `getUserSetting()` methods. You can also use these methods for settings that do have a get method. For example, the following code returns the filename of the custom logo image, just as the `getSystemCustomLogoImage()` method does.

```
settingsService.getSystemSetting(SettingsKey.CUSTOM_LOGO_IMAGE);
```

**Note:** The `getSystemSetting()` and `getUserSetting()` methods always return `String` values. Convert each value to another data type, such as an `Integer` or `Boolean`, if necessary. Get methods for specific settings, such as `getSystemMaxUploadSize()`, return the appropriate data type for the `TeamConnect` field.

- **Methods that set the value of a setting**—You use the `setSystemSetting()` and the `setUserSetting()` methods to set the values for system and user settings. `SettingsService` does not provide methods for setting specific settings.

**Note:** Use the following code to [access SettingsService](#): `SettingsService settingsService = platform.getSettingsService();`

## Two Types of Setting Parameters

Two methods of the same name exist for each of the following previously mentioned methods: `getSystemSetting()`, `getUserSetting()`, `setSystemSetting()`, and `setUserSetting()`. These two methods are different because they each use two different types of parameters:

- The enums in the `SettingKey` class. This class includes enums for all user and system settings.
- The key of a custom setting. You can only [add a custom setting](#) through the `TeamConnect` API, not from the `TeamConnect Setup`.

For an example of the different types of parameters, the next two code samples use the two different `setSystemSetting()` methods.

In the following sample, `setSystemSetting()` uses a `SettingKey` enum as a parameter. Using the `SEARCH_MAX_TIME` setting key, this code updates the **Maximum Search time (seconds)** field in the **Admin Settings** to a value of 25.

```
settingsService.setSystemSetting(SettingKey.SEARCH_MAX_TIME, "25");
```

The following sample includes a custom field as a parameter for `setSystemSetting()`. This code updates a custom field in the **Admin Settings**, specified by the **Custom System Setting** parameter, to a value of 25.

```
settingsService.setSystemSetting("Custom System Setting", "25");
```

### 3.4.2.8.1 Adding Custom Settings

You can add user and system custom settings to `TeamConnect` using the API. When you add a custom setting, you can enter custom code so that the setting works throughout `TeamConnect`.

To add custom settings, use the `setSystemSetting()` and `setUserSetting()` methods that [receive the custom setting name parameter](#). Even though custom setting values save as `String` values, you can convert the setting to another data type as part of the code.

For example, if you want to create a new user setting with the **New Custom Setting** key, use the following code:

```
// Adds a new custom setting to the system.
// Making the same call after the setting is initially added updates the value
like it does for non-custom settings.
platform.getSettingsService().setSystemSetting("New Custom Setting", "10");
// Setting values save as a String, so you can convert the String to an Integer
or another data type.
// You can enter custom code to check the value of the newly added setting and
impact parts of TeamConnect.
if(Integer.valueOf(platform.getSettingsService().getSystemSetting("newCustomSett
ing")) > 5) {
    // Custom code here
}
```

**Note:** If *New Custom Setting* already exists when you enter the previous code, the setting updates with the new value.

### 3.4.2.9 Internationalization

TeamConnect provides ways for you to customize system and custom data for multiple languages and regions. In addition to changing the language in the user interface, you can also change the currency and the time and date display for a specified region.

`InternationalizationService` is the main interface for working with different locales and currencies. None of the methods for this interface change anything in your version of TeamConnect. However, you can use the methods to retrieve information based on specified locales. If an `InternationalizationService` method does not require a locale, the code uses the [current user's locale](#).

`InternationalizationService` includes methods for the following types of actions:

- **Converting currency**—You can use the `convertCurrency()` method to convert one type of currency to the system currency. The following code sample converts 100 British Pounds to the system currency.

```
BigDecimal value = new BigDecimal(100);
BigDecimal valueInOtherCurrency =
platform.getInternationalizationService().convertCurrency(value, "GBP");
```

- **Retrieving the currency based on a code**—You can use the `getCurrency()` method to retrieve a currency. The following code sample returns the currency object for GBP, which includes the currency's code, symbol, and exchange rate.

```
Currency currency = internationalizationService.getCurrency("GBP");
```

- **Retrieving values for a system or user setting**—For example, you can use the `getCurrentUserLocale()` method to return the locale of the current user, as shown in the following code sample.

```
Locale currencyUserLocale =
internationalizationService.getCurrentUserLocale();
```

- **Returning a localized version of a string or number**—For example, you can use the `localizeNumber()` method to determine the format of a number based on the locale. The following code sample returns 5.555,55 for the French locale.

```
String localizedNumberString = internationalizationService.localizeNumber(new
BigDecimal(5555.55), Locale.FRANCE);
```

You can also use the `localize()` method with a custom message to [format the date](#).

- **Sorting based on a locale**—Based on the current user's locale, you can use the `sortCategories()` method or the `sortLookupItems()` method to sort items with the same display order alphabetically, as shown in the following code sample.

```
List<LookupItem> sortedLocalizedLookupItems =
internationalizationService.sortCategories(unsortedLookupItems);
```

**Note:** To update the display order, use the `CategoryDefinition` interface for categories and the `LookupItem` interface for lookup items.

**Note:** Use the following code to [access InternationalizationService](#): `InternationalizationService internationalizationService = platform.getInternationalizationService();`

### 3.4.2.10 Searching for Records

In your API code, you may need to perform actions on records that meet specified criteria, particularly as part of a rule specification. For example, before integrating with a billing system, a rule may search the database for all task records added to a matter within a period of time.

[Search in the API](#) using `ResourceService.search()`, which provides similar functionality to [search views in the TeamConnect user interface](#).

Because searching may drastically affect performance, verify that there is no other way to get the desired list of records. If you must conduct a search, especially against a large number of records, narrow your search criteria and consider providing a time-out.

Before using the `search()` methods, consider using available predefined methods to get related records. For example, the `User` interface has the `getContact()` method to obtain the contact of a user. The `Project` interface has methods for [obtaining a list of child or involved records](#).

The following table provides examples of records that custom code is likely to include as part of a search and whether or not an alternative exists.

To Search or Not to Search

If you are looking for	Use search?		Use this instead
	Yes	No	

All contacts.	<b>x</b>		
Any documents (but not document folders).	<b>x</b>		
Any history records.	<b>x</b>		
Any of these system object records related records to a <code>Project</code> record: <ul style="list-style-type: none"> <li>• Accounts</li> <li>• Appointments</li> <li>• Expenses</li> <li>• Tasks</li> </ul>	<b>x</b>		
Child or embedded records of a <code>Project</code> record.		<b>x</b>	<code>Project.getEmbeddedProjects</code> <code>ProjectService.getChildProjects</code>
Child records of an <code>Account</code> record.	<b>x</b>		
Current user.		<b>x</b>	<code>UtilityService.getCurrentUser()</code>
Involved record associated with an account.	<b>x</b>		
Old object.		<b>x</b>	<code>ResourceService.read()</code>
Parent record of a related <code>Account</code> record.		<b>x</b>	<code>Account.getParentAccount()</code>
Parent record of a related <code>Project</code> record.		<b>x</b>	<code>Project.getParentProject()</code>
Project of a related system object record.		<b>x</b>	<code>SystemObjectName.getProject()</code> For example, <code>Task.getProject()</code>
Project record of an <code>Involved</code> record.		<b>x</b>	<code>Involved.getProject()</code>
Record to which a <code>History</code> record is related.		<b>x</b>	<code>History.getParentObject()</code>



Sub-objects.		<b>x</b>	Use the appropriate methods defined in the owner interface. For details, see <a href="#">Sub-Objects</a> .
--------------	--	----------	--

### 3.4.2.10.1 Creating an API Search View

When you are creating code for searching, you must include the following information in your search code:

- One of three `ResourceService.search()` methods to execute the search.
  - A method that uses the search view key to return the results of a search view.
  - A method that uses filter criteria to return search results.
  - A method that uses filter criteria and search parameters to return search results.
- Methods part of the search classes to enter [search criteria](#) and [parameters](#).
- The resource service class and concrete objects to specify the objects you are searching. For example, if you are searching for accounts, [access AccountService](#) so that you can use `Account` methods as part of the search.

When you include all this information, you can [execute the search](#).

#### 3.4.2.10.1.1 Search Criteria

Creating search criteria is the same as entering **Filter Criteria** for a custom search. To create filter criteria, use one of the two `search()` methods that accept `SearchCriteria`. Depending on your filter criteria, you can construct `SearchCriteria` in one of three ways:

- `SearchCriteria()` — [Searching for one type of record](#).
- `SearchCriteria(FieldCriterion fieldCriterion)` — [Searching with one filter criteria](#).
- `SearchCriteria(SearchExpression expr1, SearchExpression expr2, SearchExpression... expressions)` — [Searching with multiple filter criteria](#).

### Searching for One Type of Record

To return all records of a specific type, use an empty `SearchCriteria()`. For example, if you want to return all records for one project type, pass in the project's unique code and `SearchCriteria()`, as shown in the following sample.

```
// Using an empty search criteria to return all Disputes
List<Project> allDisputes = platform.getProjectService().search("DISP", new
SearchCriteria());
```

### Searching with One Filter Criteria

To search with one filter criteria, use `SearchCriteria(FieldCriterion fieldCriterion)`. Use `FieldCriterion` to enter the fields you are searching on and its [subclasses when you know the](#)

[data types](#) of those fields. Along with the field, you include the value of the field that determines the search results.

For example, if you want to search for all appointments in Room 101, you can use the code in the following sample.

```
// Retrieve the appointment service from the platform provided by the API.
AppointmentService appointmentService = platform.getAppointmentService();

// Find all appointments located in Room 101
StringCriterion locationCriterion = new
StringCriterion(Appointment.LOCATION).equalTo("Room 101");
SearchCriteria criteria = new SearchCriteria(locationCriterion);

// Execute search
List<Appointment> results = appointmentService.search(criteria);
```

The previous sample uses the following code to build search criteria:

- `Appointment.LOCATION` for [searching on a system field](#).
- `equalTo("Room 101")` for [setting the operator](#).

## Searching with Multiple Filter Criteria

To search with multiple filter criteria, use `SearchCriteria(SearchExpression expr1, SearchExpression expr2, SearchExpression... expressions)`. `SearchExpression` allows you to search during the following scenarios.

### All or Any Criteria as Part of the Results

You may want to search with more than one filter criteria and you want all or any of the criteria to be part of the results. For all the criteria, use the `and()` method. For any of the criteria, use the `or()` method.

The following code sample provides an example of using the `or()` method to join date criteria and combining the date criteria with the `and()` method.

```
// Retrieve the appointment service from the platform provided by the API.
AppointmentService appointmentService = platform.getAppointmentService();

// Find all appointments located in Room 101
SearchCriteria criteria = new SearchCriteria(new
StringCriterion(Appointment.LOCATION).equalTo("Room 101"));

// Find appointments starting within four days or ending in the next two days
RelativeDateCriterion startOnCriterion = new
RelativeDateCriterion(Appointment.START_ON).next(4);
```

```
RelativeDateCriterion endOnCriterion = new
RelativeDateCriterion(Appointment.END_ON).next(2);

// Join the two date criteria, specifying 'or' logic
SearchCriteria dateCriteria = startOnCriterion.or(endOnCriterion);

// Join the date criteria and location criteria
criteria.and(dateCriteria);

// Execute search and return results
List<Appointment> results = appointmentService.search(criteria);
```

The previous sample includes the following code:

- The first search expression uses one filter criteria to specify that the results return all appointments in Room 101.

```
SearchCriteria criteria = new SearchCriteria(new
StringCriterion(Appointment.LOCATION).equalTo("Room 101"));
```

- The second search expression specifies relative date criteria using the `next()` method and combines them with the `or()` method to specify whether the search returns the applicable start date or end date.

```
RelativeDateCriterion startOnCriterion = new
RelativeDateCriterion(Appointment.START_ON).next(4);
RelativeDateCriterion endOnCriterion = new
RelativeDateCriterion(Appointment.END_ON).next(2);

SearchCriteria dateCriteria =
startOnCriterion.or(endOnCriterion);
```

- The sample combines both of the previous expressions.

```
criteria.and(dateCriteria);
```

- The [code executes](#).

```
List<Appointment> results = appointmentService.search(criteria);
```

### Complex Criteria to Determine Results

You may want to specify whether some and which criteria should be, should possibly be, or should not be part of the results. You can use a different method for each search criteria:

- You can specify that the search must include two search conditions with `and()`.
- You can specify that either search condition should appear in the search results using `or()`.

- You can specify conditions for results that you do not want to include using `not()`.

The following code sample provides an example of the previous scenario.

```
// Get service
AppointmentService appointmentService = platform.getAppointmentService();

// Group One: In Room 101 AND All Day
StringCriterion locationCriterion = new
StringCriterion(Appointment.LOCATION).equalTo("Room 101");
BooleanCriterion allDayCriterion = new
BooleanCriterion(Appointment.ALL_DAY).isTrue();
SearchCriteria andGroup = locationCriterion.and(allDayCriterion);

// Group Two: Started within the last 5 days OR ends in the next 3 days
RelativeDateCriterion startOnCriterion = new
RelativeDateCriterion(Appointment.START_ON).withinLast(5);
RelativeDateCriterion endOnCriterion = new
RelativeDateCriterion(Appointment.END_ON).next(3);
SearchCriteria orGroup = startOnCriterion.or(endOnCriterion);

// Composite Group: NOT in both group one and group two
SearchCriteria notGroup = new SearchCriteria(andGroup, orGroup).not();

// Execute search and return results
return appointmentService.search(notGroup);
```

The previous sample includes the following code:

- The first search expression specifies all day-long appointments in Room 101.

```
StringCriterion locationCriterion = new
StringCriterion(Appointment.LOCATION).equalTo("Room 101");
BooleanCriterion allDayCriterion = new
BooleanCriterion(Appointment.ALL_DAY).isTrue();
SearchCriteria andGroup = locationCriterion.and(allDayCriterion);
```

- The second search expression specifies all appointments that started in the last 5 days or end in the next 3 days.

```
RelativeDateCriterion startOnCriterion = new
RelativeDateCriterion(Appointment.START_ON).withinLast(5);
RelativeDateCriterion endOnCriterion = new
RelativeDateCriterion(Appointment.END_ON).next(3);
SearchCriteria orGroup = startOnCriterion.or(endOnCriterion);
```

- The third search expression specifies all appointments not part of the other two expressions.

```
SearchCriteria notGroup = new SearchCriteria(andGroup, orGroup).not();
```

- The sample searches for all results that the third expression returns as part of [its execution](#).

```
appointmentService.search(notGroup);
```

**Note:** This code uses the `and(SearchExpression expression)` method instead of the `and()` method because it combines multiple filter criteria with additional search expressions. If the AND or OR applies to all filter criteria, use the `and()` or `or()` methods.

When you have one field that is directly part of a record, you can construct a criterion with the `SearchableField`.

**Note:** If the record does not have direct access to the field but can access it through a field in the record, you must [construct a criterion with a FieldPath](#).

The type of criterion depends on the type of field. The API provides the following criterion types with `SearchableField` constructors:

- `BooleanCriterion`—For Boolean fields.
- `CategoryCriterion`—For categories.
- `DateCriterion`—For date fields.
- `DecimalCriterion`—For BigDecimal fields.
- `EnumerationCriterion`—For enum-based fields.
- `FieldCriterion`—For all fields.
- `IntegerCriterion`—For integer fields.
- `LookupItemCriterion`—For lookup table fields.
- `ObjectCriterion`—For account, contact, project, and user fields.
- `RelativeDateCriterion`—For integers before and after a certain date.
- `StringCriterion`—For string fields.

The `SearchableField` you use depends on whether the field is a system or custom field.

## System Field Searching

When you want to search for a system field, use the static field in the model class for the object. You can find a list of static fields for each class in the javadocs: [~\utilities\javadocs\api\index.html](#)

For example, if you are searching on the **Location** field in appointments, the `LOCATION` static field is part of the `Appointment` class. As a result, the `SearchableField` is `Appointment.LOCATION`. Before you [add the operator](#), the search field of the criterion looks like the following code.

```
StringCriterion locationCriterion = new StringCriterion(Appointment.LOCATION)
```

## Custom Field Searching

When you want to search for custom fields, you can use the `CustomField` class to retrieve information about fields for a search.

**Note:** When searching on system fields, use the static system fields in the class for that object. For example, to search with the **Location** field in appointments, use `Appointment.LOCATION`.

Use the `CustomField` class when you have to include custom fields in [search criteria](#). You can use this interface with custom fields of all data types. However, if you are working with projects, Involved parties, or lookup table custom fields, you have the following additional classes:

- `ProjectCustomField`
- `LookupCustomField`
- `InvolvedCustomField`

The following code sample provides an example of how to search using a custom field as the search criteria.

```
// Get instances of the services we'll need
ProjectService projectService = platform.getProjectService();
CategoryService categoryService = platform.getCategoryService();

// Retrieve the custom field
CustomField<String> textField =
categoryService.getCustomField("fullTreePositionofCategory", "customFieldName");

// Find records with the given custom field set to "Test value"
StringCriterion stringCriterion = new StringCriterion(textField).equalTo("Test
value");
SearchCriteria searchCriteria = new SearchCriteria(stringCriterion);

// Execute search and return results
return projectService.search("DISP", searchCriteria);
```

Field paths contain more than one [SearchableField](#). Use `FieldPath` to construct a field when the record does not have direct access to the field you want to search.

For example, if you want to search with the username of an appointment attendee, you cannot access the user directly from the appointment. The `Appointment` class contains the static `ATTENDEES` field. Once you have access to the attendees, you can access the static `USER` field with the `AppointmentAttendee` class. After you have access to the user, you can access the static `USERNAME` field with the `User` class. As a result, the `FieldPath` is `(Appointment.ATTENDEES).add(AppointmentAttendee.USER).add(User.USERNAME)`;

Before you [add the operator](#), the field path looks like the following code.

```
FieldPath fieldPath = new
FieldPath(Appointment.ATTENDEES).add(AppointmentAttendee.USER).add(User.USERNAME
);
```

The following code includes an example of a search with two criteria that use field paths.

```
AppointmentService appointmentService = platform.getAppointmentService();

// Create a path to the Appointment's Project's Created On
FieldPath projectCreatedOn = new
FieldPath(Appointment.PROJECT).add(Project.CREATED_ON);

// Create a path to the Appointment's Attendee's Attendance Type
FieldPath attendeeAttendanceType = new
FieldPath(Appointment.ATTENDEES).add(AppointmentAttendee.ATTENDANCE_TYPE);

// Search for appointments related to projects created in the last thirty days
RelativeDateCriterion relativeDateCriterion = new
RelativeDateCriterion(projectCreatedOn).withinLast(30);

// Search for appointments that have attendees that will not attend
EnumerationCriterion<AttendanceType> enumerationCriterion = new
EnumerationCriterion<AttendanceType>(attendeeAttendanceType).equalTo(AttendanceT
ype.WILL_NOT_ATTEND);

// Criteria constructed from field paths behave exactly like criteria
constructed from fields, and can be combined just the same
SearchCriteria criteria = new SearchCriteria(relativeDateCriterion,
enumerationCriterion);

// Execute search
List<Appointment> appointments = appointmentService.search(criteria);
```

You can also use field paths with custom field searching, as shown in the following sample.

```
ProjectService projectService = platform.getProjectService();
CategoryService categoryService = platform.getCategoryService();

// Field Paths can also point to custom fields
CustomField<String> customField = categoryService.getCustomField("CONT", "custom
field");
FieldPath contactCustomField = new FieldPath(Project.CONTACT).add(customField);

// Search for projects based on contact custom field
```

```
StringCriterion stringCriterion = new
StringCriterion(contactCustomField).equalTo("value");
SearchCriteria criteria = new SearchCriteria(stringCriterion);

// Execute search
List<Project> projects = projectService.search("DISP", criteria);
```

Setting operators in the API are similar to setting operators in TeamConnect custom searches. To search with operators, [use the `FieldCriterion` class](#) and all the classes that extend it.

`FieldCriterion` includes methods for operators that you can use on all fields.

Because each data type has different operators, each data type class has its own set of methods for using operators. These methods are part of the following classes:

- `BooleanCriterion`—Operators for Boolean fields.
- `CategoryCriterion`—Operators for categories.
- `DateCriterion`—Operators for date fields.
- `DecimalCriterion`—Operators for BigDecimal fields.
- `EnumerationCriterion`—Operators for enum-based fields.
- `IntegerCriterion`—Operators for integer fields.
- `LookupItemCriterion`—Operators for lookup table fields.
- `ObjectCriterion`—Operators for account, contact, project, and user fields.
- `RelativeDateCriterion`—Operators for integers before and after a certain date.
- `StringCriterion`—Operators for string fields.

In the case of the example for system field searching without the operator, the entire criterion looks like the following code with the operator:

```
StringCriterion locationCriterion = new
StringCriterion(Appointment.LOCATION).equalTo("Room 101");
```

#### 3.4.2.10.1.2 Search Parameters

Use search parameters to control the search, such as the page size and the sort order of the search results. To control your search, [use the `search\(\)` method](#) that accepts the `SearchParameters` class.

As you create the code for searching, use the search parameter constructors. The constructors include a combination of the following search parameters:

- `beginIndex`—The first page of results the search returns.
- `limit`—The number of search results to return on a page.
- `useUnsavedChanges`—An indication of whether the search should run against changes not yet committed to the database.



- `sortFields`—The fields for sorting the search results.

If a constructor does not include one of these parameters, the search uses the default for the parameter. Retrieve or set the default settings using the `SearchParameters` `get` and `set` methods (`getBeginIndex()`, `setLimit()`, etc.).

The following list provides examples of some `SearchParameters` constructors. For a full list of constructors, refer to the `SearchParameters` class in the javadocs.

- `SearchParameters(SortField... sortFields)`

You can specify how you want the search results to sort. This constructor creates a parameter with one or more sort fields, the default limit, and the default begin index. The following code sample sorts the results by start date in ascending order.

```
// Sort results by start date
SearchParameters parameters = new SearchParameters(new
SortField(Appointment.START_ON));
```

- `SearchParameters(java.lang.Integer limit)`

You can limit the number of results in a search. This constructor creates a parameter with the number of search results, the default begin index, and no sort fields. The following code sample limits the number of results to 10 records.

```
// Limit results to 10 records
SearchParameters parameters = new SearchParameters(10);
```

- `SearchParameters(boolean useUnsavedChanges)`

You can specify whether a search should return results from the database or changes not yet commented to the database. This constructor creates a parameter that specifies which to search, the default limit and begin index, and no sort fields. The following code sample specifies that the search results will include changes not yet saved to the database.

```
// Includes unsaved changes in search results
SearchParameters parameters = new SearchParameters(true);
```

#### 3.4.2.10.2 Executing a Search

When you have all the information you need to return search results, whether it is the [search view key](#) or the [search criteria](#) and [parameters](#), you can execute the search.

The following code sample executes a search with search criteria and parameters. This sample returns search results for appointments in Room 101 sorted by start date.

```
public class AppointmentSample extends List<Appointment> {

@Override
    public void action(Appointment appointment) {
    }
}
```

```
public void searchAppointment() {

    // Retrieve the appointment service from the platform provided by the
    API.
    AppointmentService appointmentService = platform.getAppointmentService();

    // Find all appointments located in Room 101
    StringCriterion locationCriterion = new StringCriterion(new
    FieldPath(Appointment.LOCATION)).equalTo("Room 101");
    SearchCriteria criteria = new SearchCriteria(locationCriterion);

    // Sort results by start date
    SearchParameters parameters = new SearchParameters(new SortField(new
    FieldPath(Appointment.START_ON)));

    // Execute search
    List<Appointment> results = appointmentService.search(criteria,
    parameters);

}
}
```

### 3.4.2.11 Logging

TeamConnect's logging functionality allows the system to generate logging messages for information or troubleshooting. As a TeamConnect developer, you can specify messages for log levels during customization, design, and production.

For example, when you are writing and testing code, include **Debug** messages so that you can see what occurs when your code executes. When code is in production, messages with a higher level of severity, such as **Warn** or **Error**, indicate less frequent but more critical problems in your code.

Logs capture messages based on the logger levels that system administrators or solution developers [define in TeamConnect](#). TeamConnect only logs messages for the level defined and the levels of higher severity. As a result, if you include a message in your code for a lower level, TeamConnect does not log the message to the logger. For example, loggers with a **Warn** level include messages with levels of **Warn**, **Error**, and **Fatal**. If your code includes a message for the **Info** logger level, the message does not log.

#### Loggers for API Code

TeamConnect has [several predefined loggers](#). Depending on the type of code you are writing, TeamConnect logs messages to the corresponding logger:

- Code for an automated qualifier or action logs messages to the **Rule** logger.
- Code for a custom screen logs messages to the **Custom Blocks** logger.
- Code for a custom tool logs messages to the **Tools** logger.

## Logging Messages from the API

Complete the following steps for your custom code to generate log messages:

1. [On the Logging page of the Admin Settings](#), define the logging level for the type of [logger associated with the code](#).
2. Specify [logging messages](#) in your code using the `CustomItem` or `Platform` logging methods.
3. Add code that [checks the logging levels](#) in TeamConnect before running logging methods.

### 3.4.2.11.1 Specifying Logging Messages

The following classes include methods for logging:

- `CustomItem`—Includes the logging methods for code that extends `CustomItem`, which means automated qualifiers, automated actions, custom screens, and custom tools.
- `Platform`—Includes the logging methods for code that does not extend `CustomItem`.

Both classes include the same logging methods at the following [logger levels](#):

- **Debug**
  - `logDebug(String message)`
  - `logDebug(String message, Throwable t)`
- **Info**
  - `logInfo(String message)`
  - `logInfo(String message, Throwable t)`
- **Warn**
  - `logWarn(String message)`
  - `logWarn(String message, Throwable t)`
- **Error**
  - `logError(String message)`
  - `logError(String message, Throwable t)`

The following code includes examples of methods that log messages:

```
String username = "username";
Date time = new Date();

// Log some debugging information
logDebug("rule executed by " + username + " at " + time);
// Also possible via Platform
platform.logDebug("rule executed by " + username + " at " + time);

// Log an informational message
```

```
logInfo("rule executed by " + username + " at " + time);
// Also possible via Platform
platform.logInfo("rule executed by " + username + " at " + time);

// Log a warning
logWarn("Missing some information");
// Also possible via Platform
platform.logWarn("Missing some information");

// Log an error
logError("This is a problem");
// Also possible via Platform
platform.logError("This is a problem");
```

#### 3.4.2.11.2 Checking Logger Levels

Running code with [logging methods](#) can have an impact on performance. You can prevent code from running unnecessarily by first checking whether the specified logger level is enabled on the **Logging** page of **Admin Settings**.

The API provides the following methods for checking whether the **Debug** or **Info** logger levels are enabled:

- `isDebug()` to check for the **Debug** logger level.
- `isInfo()` to check for the **Info** logger level.

**Note:** *These levels have methods for checking because levels of a higher severity are not intended for code in production.*

The following code for logging **Debug** and **Info** log messages checks whether the logger levels are turned on before running the log methods.

```
// Log some debugging information
logDebug("rule executed");
// Check the log level before executing complex debugging statements to improve
performance
if(isDebug()) {
logDebug("rule executed by " + username + " at " + time);
}

// Log an informational message
logInfo("rule executed");
// Check the log level before executing complex informational statements to
improve performance
if(isInfo()) {
logInfo("rule executed by " + username + " at " + time);
}
```

```
}
```

### 3.4.3 Java Samples

This reference section provides examples of automated qualifiers, automated actions, and searching samples written in Java. If desired, you can copy these samples and use them as models or starting points for your own custom code.

The concepts that are demonstrated in these samples are introduced in [The TeamConnect API](#), [Automated Qualifiers and Actions](#), and [Searching for Records](#).

#### Sample Values

Each complete sample includes a description and a table listing all of the example values in the sample. The values in each sample are intended to demonstrate the kind of values needed, such as unique codes of object definitions, full tree positions of categories, and custom field names.

#### System User In These Samples

Many of the following code samples switch to the system user to enable the code to be executed regardless of the current user's rights.

However, you must determine whether switching to the system user is necessary in your own code based on the requirements for your custom code and your custom TeamConnect design. For more details about the system user, see [Running Code as the System User](#).

#### 3.4.3.1 Partial Samples to Demonstrate Concepts

The following code samples are incomplete. They demonstrate concepts as noted in the heading and description. These concepts are presented in the context of rule qualifiers, but you can apply them in other types of custom code. You can use these qualifiers with any rule type.

You can find the following partial code samples in this section:

- [Checking Lists of Added Categories in Records](#)
- [Looping Through Related Records of Projects](#)
- [Getting Child Projects of Specific Custom Object](#)

#### Checking Lists of Added Categories in Records

The following rule qualifier code sample in Java shows how you can check whether a specific category is one of the multiple categories added to a record.

##### Java Sample: Checking Lists of Added Categories

```
import java.util.List;
import com.mitratesoft.teamconnect.enterprise.api.model.Category;
import com.mitratesoft.teamconnect.enterprise.api.model.Project;
public class CheckingValuesInRelatedRecords3 extends CustomCondition<Project> {
```

```
@Override
public boolean condition(Project record) {
    // Getting list of categories added to the project
    List<Category> categories = record.getCategories();
    for (Category category : categories) {
        // Looking for some specific category
        if (category.getTreePosition().equals("FULL_TREE_POSITION")) {
            return false;
        }
    }
    return true;
}
}
```

## Looping Through Related Records of Projects

Related records of projects, including involved records and child projects (custom objects), have specific methods for getting them. These methods are described in [Related Records](#).

You retrieve a list of Involved parties or all child projects:

### Java Sample: Looping Through Related Records of Projects

```
public void action(Project project) {
    // Get list of child objects for project for a specific custom object
    definition
    List<Project> children =
    platform.getProjectService().getChildProjects(project, "uniqueCode");

    // Loop through list of child objects
    for (Project child : children) {
        // Do something with the child here
    }

    // Get list of embedded entities for the same specific custom object
    definition
    List<EmbeddedEntity> embeddedEntities=
    project.getEmbeddedEntities("uniqueCode");

    // Loop through list of embedded entities
    for (EmbeddedEntity entity : embeddedEntities) {
        // Do something with the entity here
    }
}
```

## Getting Child Projects of Specific Custom Objects

The following code sample obtains a list of child or embedded objects of a specific custom object definition using its unique code. The method for getting the list takes the unique code of that object definition as an argument:

**Java Sample: Getting Child Projects of Specific Custom Objects**

```
public void action(Project project) {
    // Get list of child objects for project for a specific custom object
    // definition
    List<Project> children =
    platform.getProjectService().getChildProjects(project, "uniqueCode");
    // Loop through list of child objects
    for (Project child : children) {
        // Do something with the child here
    }
    // Get list of embedded entities for the same specific custom object
    // definition
    List<EmbeddedEntity> embeddedEntities=
    project.getEmbeddedEntities("uniqueCode");
    // Loop through list of embedded entities
    for (EmbeddedEntity entity : embeddedEntities) {
        // Do something with the entity here
    }
}
```

### 3.4.3.2 Automated Qualifiers

You can use the following examples of automated qualifiers written in Java with any rule type.

Depending on whether you need the old object or the current object, different methods are necessary for getting values of an object. In these examples, the values in the current object are used unless otherwise stated.

You can find the following automated qualifier samples in this section:

- [Checking System Field Values](#)
- [Comparing Custom Field Values](#)
- [Checking Whether System Field Values Were Changed](#)
- [Checking Whether Custom Field Values Were Changed](#)
- [Checking Roles in Related Records](#)
- [Checking Default Category in Related Records](#)
- [Checking Values in Vendor's Contact Record](#)
- [Checking Whether Custom Field Has a Value](#)

### Checking System Field Values

Both the total amount and activity item are system fields in task records (instances of `Task`). Therefore, the interface `Task` has specific methods to get their values.

The activity items in task records are system lookup table items. For more details, see [Lookup Tables and List Fields](#).

<b>Business Rule</b>	No tasks that are over \$10,000 with the activity item Printing can be posted without approval.
<b>Qualifier</b>	Checks whether: <ul style="list-style-type: none"> <li>• The total task amount is over \$10,000.</li> <li>• The task's activity item is Printing.</li> </ul>
<b>Sample Values</b>	<p><b>Object Definition:</b></p> <p>TASK (system object Task)</p> <p><b>Task Activity Item:</b></p> <p>ACTI_PRNT (Printing)</p> <p><b>Categories:</b></p> <p>None</p> <p><b>Custom Fields:</b></p> <p>None</p>

#### Java Qualifier: Checking System Field Values

```
public class VerifyingSystemFieldValues extends CustomCondition<Task> {
    @Override
    public boolean condition(Task task) {
        // Check whether the activity item of the task is Printing and the total
        // amount of the task is greater than or equal to 10,000
        return task.getActivityItem().equals("ACTI_PRNT") &&
            task.getTotalAmount().doubleValue() >= 10000;
    }
}
```

## Comparing Custom Field Values

Automated qualifiers are frequently used to compare field values. You can compare the values of any fields as long as the values are of the same data type. This example demonstrates how to compare the values from two custom fields of type Date.

<b>Business Rule</b>	Served company records do not save if the values in the <b>Service Date</b> and <b>Received Date</b> custom fields are not the same.
----------------------	--



<b>Qualifier</b>	Checks whether the date values are in two custom fields of the same type.
<b>Sample Values</b>	<p><b>Object Definition:</b> LITI (custom object Litigation)</p> <p><b>Categories:</b> LITI (root category of Litigation)</p> <p><b>Custom Fields:</b> ServiceDate (Date) ReceivedDate (Date)</p>

#### Java Qualifier: Comparing Custom Field Values

```
public class ComparingTwoCustomFieldValues extends CustomCondition<Project> {
    @Override
    public boolean condition(Project project) {
        // Getting the value of the service date field
        Date serviceDate = project.getDateFieldValue("LITI", "serviceDate");

        // Getting the value of the received date field
        Date receivedDate = project.getDateFieldValue("LITI", "receivedDate");

        // Return true if they are not equal
        return !serviceDate.equals(receivedDate);
    }
}
```

## Checking Whether System Field Values Changed

Automated qualifiers in business rules often compare the values of a record that displayed when the record opened with the values entered by the currently logged-in user before saving the record.

The following sample checks whether the contact-centric field of a record has changed.

<b>Business Rule</b>	Main policy holder cannot change in policy records without approval.
<b>Qualifier</b>	Checks the value on the screen in the <b>Policy Holder</b> contact-centric field against the value in the database.
<b>Sample Values</b>	<p><b>Object Definition:</b> Policy or any custom object definition</p> <p><b>Categories:</b></p>

None

**Custom Fields:**

None

**Java Qualifier: Checking Whether System Field Values Changed**

```

public class CheckingWhetherRecordValuesWereChanged extends
CustomCondition<Project> {
    @Override
    public boolean condition(Project project) {
        // Getting the object as it currently exists in the database - that is,
        // the 'old' object
        Project initialProject =
        platform.getProjectService().readLastSaved(project);

        // Return true if the contact associated with the project has been
        // changed
        return project.getContact() != null &&
        initialProject.getContact().getPrimaryKey() !=
        project.getContact().getPrimaryKey();
    }
}

```

**Checking Whether Custom Field Values Changed**

This sample demonstrates how to check whether the value in a custom field changed. Notice that the method of getting the database value and the value entered during the current session is done in the same as in the previous sample.

<b>Business Rule</b>	Main policy holder cannot change in policy records without approval.
<b>Qualifier</b>	Checks the value on the screen in the <b>Policy Holder</b> custom field against the value in the database.
<b>Sample Values</b>	<b>Object Definition:</b> PLCY (custom object Policy) <b>Categories:</b> PLCY (root category of Policy) <b>Custom Fields:</b> PolicyHolder (Involved)

**Java Qualifier: Checking Whether Custom Field Values Changed**

```

public class CheckingWhetherCustomFieldValuesWereChanged extends

```

```

CustomCondition<Project> {
    @Override
    public boolean condition(Project project) {
        // Getting the object as it currently exists in the database - that is,
        // the 'old' object
        Project initialProject =
            platform.getProjectService().readLastSaved(project);

        // Getting new policy holder
        Involved currentPolicyHolder = project.getInvolvedFieldValue("PLCY",
            "PolicyHolder");

        // Getting old policy holder
        Involved initialPolicyHolder =
            initialProject.getInvolvedFieldValue("PLCY", "PolicyHolder");

        // Return true if the policy holder has been changed
        return currentPolicyHolder.getPrimaryKey() !=
            initialPolicyHolder.getPrimaryKey();
    }
}

```

## Checking Roles in Related Records

You can use automated qualifiers to check the values in related records of the current object.

<b>Business Rule</b>	No claim records can update if they do not have an involved record with the role Claimant.
<b>Qualifier</b>	Checks if an involved record has the specified role.
<b>Sample Values</b>	<p><b>Object Definition:</b> Policy or any custom object definition</p> <p><b>Related Object of Current Object:</b> CLIN (Involved)</p> <p><b>Category in Related Object:</b> CLIN_CLMT (Involved role Claimant)</p> <p><b>Custom Fields:</b> None</p>

In the following sample, after getting the list of Involved records related to the current project object, the rule checks the list of categories (involved roles) added to each involved record. If a role does not exist, the qualifier returns true.

**Java Qualifier: Checking Roles in Related Involved Records**

```

public class CheckingValuesInRelatedRecords extends CustomCondition<Project> {
    @Override
    public boolean condition(Project project) {
        List<Involved> involvedList =
            platform.getProjectService().getInvolved(project);
        for (Involved involved : involvedList) {
            List<Category> roles = involved.getCategories();
            for (Category role : roles) {
                if (role.getTreePosition().equals("CLIN_CLMT")) {
                    // Return false if the project has any involved with the role
                    // of Claimant
                    return false;
                }
            }
        }
        // Return true if the project has no involved with the role of Claimant
        return true;
    }
}

```

**Checking Default Category in Related Records**

The following sample gets the list of Involved records related to the current object and checks the default category (role) of each Involved record. If none of the Involved records have a particular category, the qualifier returns true.

<b>Business Rule</b>	Claim records can only update if they have an Involved record with the role Claimant.
<b>Qualifier</b>	Checks if there is an Involved record with the specified role as its default role (category).
<b>Sample Values</b>	<p><b>Object Definition:</b> Claim or any custom object definition</p> <p><b>Related Object of Current Object:</b> CLIN (Involved)</p> <p><b>Category in Related Object:</b> CLIN_CLMT (Involved role Claimant)</p> <p><b>Custom Fields:</b> None</p>

**Java Qualifier: Checking the Default Role in Related Involved Records**

```

package com.mitratesoft.teamconnect.enterprise.api.model.rule;
import java.util.List;
import com.mitratesoft.teamconnect.enterprise.api.model.Involved;
import com.mitratesoft.teamconnect.enterprise.api.model.Project;
import com.mitratesoft.teamconnect.enterprise.api.service.ProjectService;
public class CheckingValuesInRelatedRecords2 extends CustomCondition<Project> {
    @Override
    public boolean condition(Project project) {
        // Getting the project service
        ProjectService projectService = platform.getProjectService();
        // Getting the list of involved records related to the project
        List<Involved> involvedList = projectService.getInvolved(project);
        for (Involved involved : involvedList) {
            // If the default category is Claimant, return false
            if
                (involved.getPrimaryCategory().getTreePosition().equals("CLIN_CLMT"))
            {
                return false;
            }
        }
        return true;
    }
}

```

## Checking Values in Vendor's Contact Record

The following qualifier checks whether the vendor of an invoice has a specific checkbox selected in the contact record. The qualifier then checks whether the number of line items in the invoice is more than the number that were in the previously saved version of the invoice.

<b>Business Rule</b>	If the contact selected in the <b>Vendor</b> field of an invoice has the <b>Volume Discount</b> checkbox selected in the contact record, recalculate the task line item amount using the specified <b>Discount Percentage</b> field.
<b>Qualifier</b>	Checks the value in the <b>Volume Discount</b> checkbox in the contact record of the vendor.
<b>Sample Values</b>	<p><b>Object Definition:</b> INVC (system object Invoice)</p> <p><b>Related Object:</b> CONT (system object Contact)</p> <p><b>Categories of Related Object:</b> CONT_VNDR (category of Contact)</p>

**Custom Fields of Related Object:**

volumeDiscount (checkbox)

The following sample is an example of checking values in vendor's contact record.

**Java Qualifier: Checking Values in Vendor's Contact Record**

```
public class CheckingValueInVendorContactRecord extends CustomCondition<Invoice>
{
    @Override
    public boolean condition(Invoice invoice) {
        // Get invoice service object
        InvoiceService invoiceService = platform.getInvoiceService();

        // Getting old invoice object
        Invoice oldInvoice = invoiceService.readLastSaved(invoice);
        Contact vendor = invoice.getVendor();

        // Return true if the vendor has volume discount set to true and the line
        item count on the invoice has increased
        return vendor.getBooleanFieldValue("CONT_VNDR", "volumeDiscount") &&
            invoiceService.getLineItemCount(invoice) >
            invoiceService.getLineItemCount(oldInvoice);
    }
}
```

**Checking Whether Custom Field Has a Value**

The following qualifier checks whether a custom field that was previously null has a value, by comparing the value in the old object with the value in the current object.

This qualifier can be used with the action described in [Creating Related Task Record](#).

<b>Business Rule</b>	When the <b>Trial Date</b> field in a Matter updates with a value, create a task for the main assignee to complete the executive summary, with the due date 15 days before the trial date.
<b>Qualifier</b>	Checks the value in the <b>Trial Date</b> field in a matter record.
<b>Sample Values</b>	<p><b>Object Definition:</b> MATT (custom object Matter)</p> <p><b>Categories:</b> MATT (root category of Matter)</p> <p><b>Custom Fields:</b></p>

trialDate (Date)
------------------

**Java Qualifier: Checking Whether Custom Field Has a Value**

```
public class CreatingARelatedTaskRecordQualifier extends
CustomCondition<Project> {
    @Override
    public boolean condition(Project project) {
        // Get old object
        Project oldProject = platform.getProjectService().read(project);

        // Get current trial date
        CalendarDate trialDate = project.getCalendarDateFieldValue("MATT",
"trialDate");

        // Get old trial date
        CalendarDate oldTrialDate = project.getCalendarDateFieldValue("MATT",
"trialDate");

        // Return true if the current trial date is not null and the previous
        trial date was null
        return trialDate != null && oldTrialDate == null;
    }
}
```

**3.4.3.3 Automated Actions**

This section provides samples of automated actions written in Java in the following sections:

- [Rule Actions](#)
- [Parameterized Actions](#)
- [Wizard Page Actions](#)

**3.4.3.3.1 Rule Actions**

You can use the following samples of automated actions written in Java with rules of type Custom Action:

- [Copying Values from Parent to Multiple Child Records](#)
- [Creating a History Record](#)
- [Updating Added Categories](#)
- [Checking and Setting Values in Custom Fields of Type List](#)
- [Creating Related Task Record](#)

For parameterized rule action samples, see [Parameterized Actions](#). For wizard page action samples, see [Wizard Page Actions](#).

## Copying Values from Parent to Multiple Child Records

The following Java action gets the values of two custom fields of type date in the current object record (custom object Complaint) and copies the values of these fields into date fields in the child records of a particular custom object definition (custom object Served Company).

<b>Business Rule</b>	When the <b>Service Date</b> and <b>Received Date</b> custom fields in the parent complaint record update, the same values must copy to the corresponding <b>Service Date</b> and <b>Received Date</b> fields in the child served company records.
<b>Action</b>	Copy the values from the date custom fields in the parent complaint record to the corresponding date custom fields in all of its child served company records.
<b>Sample Values</b>	<p><b>Object Definition:</b></p> <p>COMP (custom object Complaint)</p> <p>SECO (custom object Served Company, child of Complaint)</p> <p><b>Categories:</b></p> <p>COMP (root category of Complaint)</p> <p>SECO (root category of Served Company)</p> <p><b>Custom Fields:</b></p> <p>In Complaint:</p> <ul style="list-style-type: none"> <li>• serviceDate (Date)</li> <li>• receivedDate (Date)</li> </ul> <p>In Served Company:</p> <ul style="list-style-type: none"> <li>• serviceDate (Date)</li> <li>• receivedDate (Date)</li> </ul>

### Java Action: Copying Values from Parent to Multiple Child Records

```
public class CopyingValuesFromParentToMultipleChildRecords extends
CustomAction<Project> {
    @Override
    public void action(final Project project) {
        // Run as system user to bypass security
        platform.getUtilityService().runAsSystemUser(new Callable() {
            @Override
            public void call() {
```



```

// Get service date and received date field values from parent
CalendarDate serviceDate =
project.getCalendarDateFieldValue("COMP", "serviceDate");
CalendarDate receivedDate =
project.getCalendarDateFieldValue("COMP", "receivedDate");

// Get a set of child objects for a specific child object
definition
List<Project> servedCompanies =
platform.getProjectService().getChildProjects(project, "SECO");

for (Project company : servedCompanies) {
    // Set the values from the parent project fields onto each
    child
    company.setCalendarDateFieldValue("SECO", "serviceDate",
serviceDate);
    company.setCalendarDateFieldValue("SECO", "receivedDate",
receivedDate);
}
}
});
}
}

```

## Creating a History Record

The following Java action creates a history record and relates it to the current object, which is a contact record. This action would be executed whenever a contact's name is updates.

<b>Business Rule</b>	<p>If the name of a contact modifies, write the following information to the contact History, under the <b>Data Change</b> category:</p> <ul style="list-style-type: none"> <li>• Old name of the contact.</li> <li>• New name of the contact.</li> <li>• Date when the change occurred.</li> <li>• ID of the user who made the change.</li> </ul>
<b>Action</b>	<p>Create a new history record with the following information in its fields:</p> <p><b>Text:</b> Contact name changed from &lt;old name&gt; to &lt;new name&gt;. Changed by &lt;user ID&gt;.</p> <p><b>Note:</b> <i>The date of the change was made is automatically recorded in the <b>Created On</b> system field of the history record.</i></p> <p><b>Default Category:</b> Data Change</p>

	<p><b>Parent:</b> Contact record where name change took place.</p> <p><b>Entered by:</b> User who made the change.</p>
<b>Sample Values</b>	<p><b>Object Definition:</b> contact (system object Contact)</p> <p><b>Categories:</b> HIST_DACH (category of History)</p> <p><b>Custom Fields:</b> None</p>

#### Java Action: Creating a History Record

```
public class CreatingAHistoryRecord extends CustomAction<Contact> {

    @Override
    public void action(final Contact contact) {
        // Run as system user
        platform.getUtilityService().runAsSystemUser(new Callable() {
            @Override
            public void call() {
                // Get old contact
                final Contact oldContact =
                    platform.getContactService().readLastSaved(contact);

                // Create a string for history text using old and new values of
                the contact's name
                String historyText = String.format("Contact name changed from %s
                to %s",
                    oldContact.getDisplayString(), contact.getDisplayString());

                // Create a history record and related it to the contact
                History history =
                    platform.getHistoryService().newHistory(historyText, contact);

                // Set the primary category of the history to Data Change
                history.setPrimaryCategory("HIST_DACH");
            }
        });
    }
}
```

## Updating Added Categories

The following Java action removes specific categories from the current object record and adds other categories.

In this example, when the value of the **Pre/Post** custom field in a claim record updates, the corresponding **Pre** or **Post** category is added. Each project can have only one of the two categories at a time. This action automatically sets the categories according to the value in the **Pre/Post** field.

<b>Business Rule</b>	When the <b>Pre/Post</b> field value updates, the corresponding <b>Pre</b> or <b>Post</b> category is added to a Claim project. Each project can have only one of the two categories at a time.
<b>Action</b>	Delete the old category and add the new category.
<b>Sample Values</b>	<p><b>Object Definition:</b> CLAM (custom object Claim)</p> <p><b>Categories:</b> CLAM (root category of Claim) CLAM_POST CLAM_PREE</p> <p><b>Custom Fields:</b> PrePost (List)</p> <p><b>Custom Field Value:</b> PRPO_ROOT_PREE (full tree position of lookup item)</p>

#### Java Action: Updating Added Categories

```
package com.mitratesoft.teamconnect.enterprise.api.model.rule;
import com.mitratesoft.teamconnect.enterprise.api.callable.Callable;
import com.mitratesoft.teamconnect.enterprise.api.model.Project;
public class UpdatingAddedCategories extends CustomAction<Project> {
    @Override
    public void action(final Project record) {
        // The record must be declared as final if it is going to be referenced
        // in a Callable
        // Get an instance of the utility service in order to run code as system
        // user
        platform.getUtilityService().runAsSystemUser(new Callable() {
            public void call() {
                // Getting the value of the Pre/Post custom field
                String newPrePostKey = record.getProjectFieldValue("CLAM",
                    "PrePost").getUniqueKey();
                if (newPrePostKey.equals("PRPO_ROOT_PREE")) {
                    record.removeCategory("CLAM_POST");
                }
            }
        });
    }
}
```

```

        record.addCategory("CLAM_PREE");
    } else {
        record.removeCategory("CLAM_PREE");
        record.addCategory("CLAM_POST");
    }
    }
    });
}
}

```

## Checking and Setting Values in Custom Fields of Type List

The following Java action automatically sets the value of one custom field according to the value selected in another custom field.

<b>Business Rule</b>	If the value in the <b>Ice Cream</b> custom field is Chocolate Cake, Chocolate Chip Cookie Dough, Moose Tracks, or Chocolate Peanut Butter, select <b>Good</b> in the <b>Tastiness</b> custom field. Otherwise, select <b>Okay</b> .
<b>Action</b>	Set the <b>Tastiness</b> custom field value to <b>Good</b> or <b>Okay</b> .
<b>Sample Values</b>	<p><b>Object Definition:</b></p> <p>project (custom object)</p> <p><b>Custom Fields:</b></p> <p>icecream (List)</p> <p>Tastiness (String)</p> <p><b>Custom Field List Values:</b></p> <p>In custom lookup table icecream:</p> <ul style="list-style-type: none"> <li>• ICRM_ROOT_CHOC_CAKE</li> <li>• ICRM_ROOT_CCHP_CDGH</li> <li>• ICRM_ROOT_MOOS_TRKS</li> <li>• ICRM_ROOT_CHOC_PBTR</li> </ul> <p>In custom lookup table Tastiness:</p> <ul style="list-style-type: none"> <li>• Good</li> <li>• Okay</li> </ul>

### Java Action: Checking and Setting Values in Custom Fields of Type List

```
public class IceCreamCondition extends CustomCondition<Project> {
```

```

@Override
public boolean condition(Project project) {
    // Getting the value of the ice cream custom lookup field
    String flavor = project.getLookupFieldValue("icecream").getKey();
    /* Check whether the value of the field is one of the four ice cream
    flavors that contains chocolate */
    if (flavor.equals("CHOC_CAKE") || flavor.equals("CCHP_CDGH") ||
    flavor.equals("MOOS_TRKS") || flavor.equals("CHOC_PBTR")) {
        // If value is one of the four values, set "Tastiness" custom field to
        Good.
        project.setTextFieldValue("Tastiness", "Good");
    } else {
        // If value is NOT one of the four values, set "Tastiness" custom
        field to Okay.
        project.setTextFieldValue("Tastiness", "Okay");
    }

    /* This code can also be done without using keys, such as getting the
    value of a text field which is not going to have a key associated with
    it. */

    // Getting the value of the ice cream custom lookup field
    String flavorText = project.getTextFieldValue("icecream");
    /* Check whether the value of the field is one of the four ice cream
    flavors that contains chocolate */
    return (flavorText.equals("CHOCOLATE CAKE") ||
    flavorText.equals("CHOCOLATE CHIP COOKIE DOUGH") ||
    flavorText.equals("MOOSE TRACKS") || flavorText.equals("CHOCOLATE PEANUT
    BUTTER"));
}
}

```

## Creating Related Task Record

The following Java action creates a task record, relates it to the current object, and sets the default category. The due date of the task is set to 15 days before the date in a custom field in the current object.

<b>Business Rule</b>	When the <b>Trial Date</b> field in a matter record updates with a value, create a task for the main assignee to complete the executive summary, with the due date 15 days before the trial date.
<b>Action</b>	Create a task with the following information in its fields:  <b>Parent Project:</b>

	<p>The Matter project</p> <p><b>Assignee:</b> Main project assignee</p> <p><b>Due On:</b> 15 days before the trial date</p> <p><b>Description:</b> Complete executive summary</p> <p><b>Default Category:</b> Trial Preparation</p>
<b>Sample Values</b>	<p><b>Object Definition:</b> MATT (custom object Matter)</p> <p><b>Related Object Created:</b> Task (system object)</p> <p><b>Categories:</b> MATT (root category of Matter) TASK_TRPR (Task category)</p> <p><b>Custom Fields:</b> trialDate (Date field in Matter, created under the root category)</p>

#### Java Action: Creating Related Task Record

```
public class CreatingRelatedTaskRecord extends CustomAction<Project> {
    @Override
    public void action(final Project project) {
        // Run as system user to bypass security
        platform.getUtilityService().runAsSystemUser(new Callable() {
            @Override
            public void call() {
                // Creating task and assigning it to main assignee of current
                object
                Task task = platform.getTaskService().newTask("Complete executive
                summary", project.getMainAssignee().getUser());
                CalendarDate trialDate = project.getCalendarDateFieldValue("MATT",
                "trialDate");

                // Set the due on date to fifteen days before the trial date
                Calendar dueOn = Calendar.getInstance();
                dueOn.setTime(trialDate);
            }
        });
    }
}
```

```

        dueOn.add(Calendar.DAY_OF_YEAR, -15);
        task.setDueDate(new CalendarDate(dueOn));

        // Set primary category to trial preparation
        task.setPrimaryCategory("TASK_TRPR");
    }
});
}
}

```

#### 3.4.3.3.2 Parameterized Actions

Rule parameters appear in the user interface and allow the administrator to enter values for the rule. Having parameters in rules is useful when certain values in a qualifier or an action may change and you prefer to have the administrator change the value through the user interface rather than modify the file. Parameterized rules are also useful because you can use the same file for multiple rules that each require different values.

Parameters for a wizard page action work in the same way as a rule, except they appear in the wizard.

This section includes the following parameterized action samples:

- [Specifying Due Date for Task Record Using Parameter](#)
- [Creating Project with Related History Using Parameters](#)

### Specifying Due Date for Task Record Using Parameter

The following sample automatically creates a task record when the user is saving a claim record for the first time. The due date of the task is set automatically according to the number of days specified by the administrator in the parameter, which appears as a field on the Rule screen in the user interface.

<b>Business Rule</b>	When the <b>Police Report Issued</b> checkbox is selected in a Claim, the system creates a task.
<b>Action</b>	<p>Creates a task to obtain a copy of the report with the following information in its fields:</p> <p><b>Parent Project:</b> the Claim project.</p> <p><b>Assignee:</b> Main project assignee.</p> <p><b>Description:</b> Obtain a copy of the police report.</p> <p><b>Default Category:</b> Follow-up:FNOL.</p> <p><b>Parameter:</b> A Number field with the name "numberOfDaysAfter" and the label "The number of days between the date the claim was created and the due date"</p>

	<b>Due On:</b> Parameter numberOfDaysAfter.
<b>Sample Values</b>	<p><b>Object Definition:</b> Claim or any custom object definition</p> <p><b>Rule Parameter Created:</b> numberOfDaysAfter (Integer)</p> <p><b>Related Object Created:</b> TASK (system object Task)</p> <p><b>Categories in Related Object:</b> TASK_FLUP_FNOL</p> <p><b>Custom Fields:</b> None</p>

#### Java Parameterized Action: Specifying Due Date for Task Record Using Parameter

```
public class SpecifyingDueDateForTaskRecord extends CustomAction<Project> {

    @Override
    public void declareParameters() {
        parameters.addNumberParameter("numberOfDaysAfter", "The number of days
        between the date the claim was created and the due date",
        Long.valueOf(1));
    }

    @Override
    public void action(final Project project) {
        // Run as system user to bypass security
        platform.getUtilityService().runAsSystemUser(new Callable() {
            @Override
            public void call() {
                // Get the value of the rule parameter
                int numberOfDaysAfter =
                parameters.getNumberParameterValue("numberOfDaysAfter").intValue()
                ;

                // Create a task and assigning it to the main assignee of the
                project
                Task task = platform.getTaskService().newTask("Obtain a copy of
                the police report", project.getMainAssignee().getUser());

                // Relate the task to the project
                task.setProject(project);
            }
        });
    }
}
```



```

// Set the due date based on the number of days specified in the
rules parameter
Calendar dueOn = Calendar.getInstance();
dueOn.setTime(project.getCreatedOn());
dueOn.add(Calendar.DAY_OF_YEAR, numberOfDaysAfter);

// Set the primary category of the task
task.setPrimaryCategory("TASK_FLUP_FNOL");
    }
});
}
}

```

### Creating Project with Related History Using Parameters

The following Java action uses multiple parameters. Based on these parameter values, the action creates a Package Request custom object record. It then determines whether to create a related history record based on a boolean parameter, and if so, creates the history record with the correct values.

<b>Business Rule</b>	When the value in the <b>Accident State</b> custom field in a new Claim is New York, a letter Package Request project must be created to generate the appropriate notices for the insured and claimant.
<b>Action</b>	<p>Create a Package Request with the following information:</p> <p><b>Requested for:</b> Claim record.</p> <p>Parameter 1:</p> <p>Text field named "categories" and labeled "Type the full tree position of the Letter Package Request categories, separated by commas:"</p> <p>Parameter 2:</p> <p>Number field with the name "numberOfDaysAfter" and the label "Type the number of days from the moment the rule is triggered to specify the date when the letters must be sent out:"</p> <p>Parameter 3:</p> <p>Check Box field with the name "isAuditCreated" and the label "Specify whether an audit file should be created for each request:"</p> <p>Using Parameter 1, add the specified categories to the package request record.</p> <p><b>Send Date:</b> Parameter 2.</p> <p><b>Create Audit File:</b> Parameter 3.</p>

	<ul style="list-style-type: none"> <li>If yes—create a history record with following data in its fields:           <p><b>Text:</b> NY Package request generated.</p> <p><b>Default Category:</b> Audit</p> <p><b>Parent:</b> the Claim project.</p> <p><b>Entered by:</b> TeamConnectAdmin.</p> </li> <li>If no—no other records are created.</li> </ul>
<b>Sample Values</b>	<p><b>Object Definition:</b></p> <p>Claim or any custom object definition</p> <p><b>Related Objects Created:</b></p> <p>PKRE (custom object Package Request)</p> <p>TNHistory (the audit file)</p> <p><b>Wizard Parameters:</b></p> <p>categories (Text)</p> <p>numberOfDaysAfter (Number)</p> <p>isAuditCreated (Check Box)</p> <p><b>Categories:</b></p> <p>PKRE (root category of Package Request)</p> <p>HIST_AUDI (History category)</p> <p><b>Custom Fields:</b></p> <p>In PKRE:</p> <p>requestedFor (custom field of type Custom Object, pointing to the same custom object definition as the one for which the rule is written)</p> <p>sendDate (Date)</p>

#### Java Parameterized Action: Creating Project with Related History Using Parameters

```
public class CreatingProjectWithRelatedHistory extends CustomAction<Project> {

    @Override
    public void declareParameters() {
        parameters.addTextParameter("categories", "Full tree position of the
Letter Package Request categories, separated by commas", null);
        parameters.addNumberParameter("numberOfDaysAfter", "Number of days
between the rule triggering and the date the letters must be sent out",
Long.valueOf(1));
    }
}
```

```
parameters.addBooleanParameter("isAuditCreated", "Whether an audit file
should be created for each package request", null);
}

@Override
public void action(final Project project) {
    // Get the parameter values
    final int numberOfDaysAfter =
parameters.getNumberParameterValue("numberOfDaysAfter").intValue();
    final boolean isAuditCreated =
parameters.getBooleanParameterValue("isAuditCreated");
    final String[] categories =
parameters.getTextParameterValue("categories").split(Pattern.quote(","));

    // Run as system user to bypass security
platform.getUtilityService().runAsSystemUser(new Callable() {
    @Override
    public void call() {
        // Creating new package request
        Project packageRequest =
platform.getProjectService().newProject("PKRE");

        // Setting value of Requested For custom field to the current
project object
packageRequest.setProjectFieldValue("PKRE", "requestedFor",
project);

        for (String category : categories) {
            packageRequest.addCategory(category);
        }

        Calendar sendDate = Calendar.getInstance();
sendDate.setTime(new Date());
sendDate.add(Calendar.DAY_OF_YEAR, numberOfDaysAfter);
packageRequest.setDateFieldValue("PKRE", "sendDate",
sendDate.getTime());

        if (isAuditCreated) {
            // Create history with primary category set to audit
            History history =
platform.getHistoryService().newHistory("Package request
generated", project);
            history.setPrimaryCategory("HIST_AUDI");
        }
    }
});
```

```

    }
}

```

### 3.4.3.3.3 Wizard Page Actions

The construction of a [wizard](#) action is almost the same as the construction of a rule action, except the action identifies and uses parameters in different ways. For details about wizard page actions, see [Wizard Automated Actions](#).

You can use the following examples of wizard page actions in Java. These actions execute when the wizard user clicks next to go to the next wizard page.

- [Creating a Child Project](#)
- [Setting User with Specific Role in Project as Task Assignee](#)
- [Filtering Users for Manual Selection](#)
- [Automatically Selecting Project Assignees](#)

## Creating a Child Project

Automated actions in wizards can create related records for a main record. While you can also create related records through a template, by creating related records through a page action, you can specify that the related record only create when certain conditions exist.

In this example, when the **Any Injuries?** checkbox is selected in a wizard, the system creates a child BI Claim. The following wizard action automatically creates a child BI Claim project.

<b>What User Does On Wizard Page</b>	User selects the <b>Any Injuries?</b> checkbox (a wizard parameter of type Boolean) in the First Notice of Loss wizard
<b>Wizard Page Action</b>	Create a child Claim with the default category Bodily Injury
<b>Sample Values</b>	<p><b>Object Definition:</b></p> <p>CLAM (custom object Claim)</p> <p><b>Categories:</b></p> <p>CLAM_BOIN</p> <p><b>Custom Fields:</b></p> <p>None</p> <p><b>Wizard Parameters:</b></p> <p>anyInjury (boolean)</p>

### Java Wizard Action: Creating a Child Project

```
public class CreatingAChildProject extends CustomAction<Project> {
```

```

@Override
public void action(final Project project) {
    // Get wizard parameter
    final boolean isAnyInjury =
        parameters.getBooleanParameterValue("anyInjury");

    // Run as system user to bypass security
    platform.getUtilityService().runAsSystemUser(new Callable() {

        @Override
        public void call() {
            // Create new claim as a child of the project
            Project claim = platform.getProjectService().newProject("CLAM");
            claim.setParentProject(project);

            // Add BI category as primary
            claim.setPrimaryCategory("CLAM_BOIN");
        }
    });
}
}

```

## Setting User with Specific Role in Project as Task Assignee

The wizard action automatically assigns the task to the first user who has the **Proposed Attorney** assignee role in the matter.

<b>What User Does On Wizard Page</b>	User adds an assignee to the matter wizard.
<b>Wizard Page Action</b>	Checks the role of the assignee, and if a user has the <b>Proposed Attorney</b> role, add this user as the assignee to the template-created task record associated with the wizard.
<b>Sample Values</b>	<p><b>Object Definition:</b></p> <p>MATT (custom object Matter)</p> <p><b>Assignee Roles:</b></p> <p>MATT_PRAT (Proposed Attorney)</p> <p><b>Categories:</b></p> <p>None</p>

**Custom Fields:**

None

**Wizard Parameters:**

project (project)

**Related Object:**

task (a Task)

**Java Wizard Action: Setting User with Specific Role as Assignee in Task**

```

public class SettingUserWithRoleAsAssigneeInRelatedTask extends
CustomAction<Task> {

    @Override
    public void action(final Task task) {
        // Retrieve project wizard parameter
        final Project project = parameters.getProjectParameterValue("project");

        // Run as system user to bypass security
        platform.getUtilityService().runAsSystemUser(new Callable() {

            @Override
            public void call() {
                // Get list of project assignees
                List<ProjectAssignee> assignees = project.getAssignees();

                for (ProjectAssignee assignee : assignees) {
                    // Assign the project assignee with the 'Proposed Attorney'
                    role
                    if (assignee.getAssigneeRole().equals("MATT_PRAT")) {
                        platform.getTaskService().reassign(task,
                        assignee.getUser());
                        break;
                    }
                }
            }
        });
    }
}

```

**Filtering Users for Manual Selection**

The following wizard page action assigns a list of users to a project when they have a certain primary category in their contact records.

<b>What User Does On Wizard Page</b>	User selects <b>Bodily Injury</b> as the default category for a claim record.
<b>Wizard Page Action</b>	Selects all BI adjusters and lists them as assignees on the project.
<b>Sample Values</b>	<p><b>Object Definition:</b> CLAM (custom object Claim)</p> <p><b>Assignee Roles:</b> ADJU (BI Adjuster)</p> <p><b>Categories:</b> CONT_ADJU_BIAD (category of Contact)</p> <p><b>Custom Fields:</b> None</p> <p><b>Wizard Parameters:</b> None</p>

#### Java Wizard Action: Filtering Users for Manual Selection

```
public class FilteringUsersForManualSelection extends CustomAction<Project> {

    @Override
    public void action(final Project project) {

        // Run as system user to bypass security
        platform.getUtilityService().runAsSystemUser(new Callable() {

            @Override
            public void call() {

                List<User> users = platform.getUserService().getAllUsers();

                for (User user : users) {

                    // If the user's contact's primary category is BI Adjuster, add
                    // to adjusters
                    if
                    (user.getContact().getPrimaryCategory().getTreePosition().equal
                    s("CONT_ADJU_BIAD")) {

                        project.addAssignee("ADJU", user);

                    }

                }

            }

        });
    }
}
```

```

}
}

```

## Automatically Selecting Project Assignees

The following wizard page action first checks whether the claim record created by the wizard has the category BI. It then gets a list of users by checking the default category (Bodily Injury) and the value in the **Is available** checkbox in the user's contact record. The action also assigns the first available BI Adjuster to the claim and then clears the **Is available** checkbox.

<b>What User Does On Wizard Page</b>	User adds the <b>Bodily Injury</b> category to a claim record created through a wizard.
<b>Wizard Page Action</b>	Assigns the first available BI Adjuster to the claim by checking the default category (Bodily Injury) and the value in the <b>Is available</b> checkbox in the user's contact record. Then, clears the <b>Is available</b> checkbox.
<b>Sample Values</b>	<p><b>Object Definition:</b></p> <p>CLAM (custom object Claim)</p> <p><b>Assignee Roles:</b></p> <p>CLAM_ADJU (Claim Adjuster)</p> <p><b>Categories:</b></p> <p>CLAM_BDIN (category in Claim)</p> <p>CONT_ADJU_BIAD (category in Contact)</p> <p>CONT_ADJU (category in Contact)</p> <p><b>Custom Fields:</b></p> <p>isAvailable (checkbox in Contact)</p> <p><b>Wizard Parameters:</b></p> <p>None</p>

### Java Wizard Action: Automatically Selecting Project Assignees

```

public class AutomaticallySelectingProjectAssignee extends CustomAction<Project>
{

    @Override
    public void action(final Project project) {
        try {
            platform.getUtilityService().runAsSystemUser(new
                CallableWithException<Exception>() {

```



```
@Override
public void call() throws Exception {
    // Check whether BI Category has been added to the claim record
    boolean isBI = project.hasCategory("CLAM_BDIN");

    // Get all users
    List<User> users = platform.getUserService().getAllUsers();

    for (User user : users) {
        // Check primary category and value of Is Available field on
        // each user's contact
        Contact contact = user.getContact();
        if (contact.getPrimaryCategory().equals("CONT_ADJU_BIAD") &&
            contact.getBooleanFieldValue("CONT_ADJU", "isAvailable")) {
            // Add the user as a project assignee with the role BI
            // Adjuster
            project.addAssignee("CLAM_ADJU", user);

            // Make the contact no longer available
            contact.setBooleanFieldValue("CONT_ADJU", "isAvailable",
                false);
        }
    }
}

});
} catch (Exception e) {
    // Log a message with the exception attached
    logWarn("The category CLAM_BDIN was not added to Claim", e);
}
}
```

#### 3.4.3.4 Searching Samples

You can use the following searching examples written in Java in your custom code. In these examples, the values in the current object are used unless otherwise stated.

You can find the following search samples in this section:

- [Searching for Child Accounts with Certain Posting Criteria](#)
- [Getting a List of Tasks Completed within a Certain Time Period](#)

#### Searching for Child Accounts with Certain Posting Criteria

The following code searches for all accounts with the same parent account and a specific project set as their posting criteria.

<b>Search Requirements</b>	Find all child accounts of the Matter XYZ Budget account that have a specific litigation record with the unique ID 7/15/2005-6548 set as their posting criteria.
<b>Search Criteria</b>	<p>AND statement with the following qualifiers:</p> <ul style="list-style-type: none"> <li>• Parent account Matter XYZ Budget.</li> <li>• Litigation project with the unique ID 7/15/2005-6548 set as posting criterion.</li> </ul>
<b>Sample Values</b>	<p><b>System Field Attributes:</b></p> <p>NAME (of the parent account)</p> <p>NUMBER_STRING (ID of the project)</p> <p><b>Categories:</b></p> <p>None</p> <p><b>Custom Field Attributes:</b></p> <p>None</p>

#### Searching for Child Accounts with Certain Posting Criteria

```
// An example of building more complicated field paths
AccountService accountService = platform.getAccountService();

// Search by the name of the parent account
FieldPath parentAccountName = new
FieldPath(Account.PARENT_ACCOUNT).add(Account.NAME);
StringCriterion parentAccountNameCriterion = new
StringCriterion(parentAccountName).equalTo("Matter XYZ Budget");

// Search by the number string of the project
FieldPath projectNumberString = new
FieldPath(AccountPostingCriteria.PROJECT).add(Project.NUMBER_STRING);
StringCriterion projectNumberStringCriterion = new
StringCriterion(projectNumberString).equalTo("7/15/2005-6548");

// Join the criteria together
SearchCriteria criteria = new SearchCriteria(parentAccountNameCriterion,
projectNumberStringCriterion);

List<account> accounts = accountService.search(criteria);
```

## Getting a List of Tasks Completed within a Certain Time Period

The following code gets a list of all tasks completed within the specified time period.

<b>Search Requirements</b>	Find all tasks John Smith completed within the month of June 2005.
<b>Search Criteria</b>	<p>AND statement with the following qualifiers:</p> <ul style="list-style-type: none"> <li>• User object of the task assignee.</li> <li>• Tasks that are marked as completed.</li> <li>• The start date of the specified period, 06/01/2005.</li> <li>• The end date of the specified period, 06/30/2005.</li> </ul>
<b>Sample Values</b>	<p><b>System Field Attributes:</b></p> <p>CURRENT_ASSIGNEE</p> <p>COMPLETED_DATE</p> <p><b>Categories:</b></p> <p>None</p> <p><b>Custom Field Attributes:</b></p> <p>None</p>

### Searching for All Tasks Completed within a Certain Period

```
TaskService taskService = platform.getTaskService();

// Find tasks only for the given user
UserCriterion userCriterion = new
UserCriterion(Task.CURRENT_ASSIGNEE).equalTo(assignee);

// Find tasks completed after the given start date and before the given end date
DateCriterion dateCriterion = new
DateCriterion(Task.COMPLETED_DATE).between(6/01/2005, 6/30/2005);

// Join together the two criteria, using 'and' logic by default
SearchCriteria searchCriteria = new SearchCriteria(userCriterion,
dateCriterion);

// Execute search and return results
List<Task> tasks = taskService.search(searchCriteria);
```

## 3.5 Web Services Help

Welcome to the *TeamConnect® Enterprise Web Services Help*.

<a href="#">Common Web Service Functions</a>	<a href="#">Repository Methods</a>
<a href="#">Abstract Classes</a>	<a href="#">Code Samples</a>

### 3.5.1 Getting Started

Welcome to TeamConnect Web Services. To get started building application interfaces with the TeamConnect system, follow these steps:

1. Get comfortable with high-level concepts in this chapter.
2. Decide which features of TeamConnect® Enterprise you'd like to use. Take a look through the section [Supported Operations](#).
3. Use the [Web Service API Reference](#) chapter to get a further idea of the types of requests you can make per repository or TeamConnect Web Service.
4. Use your 3rd party SOAP toolkit to generate client-side source files and API for the TeamConnect Web Service .WSDL and schema (.XSD) files provided.

The resulting client-side source files (in the programming language you will use for your client application) will provide more specific syntax for the repository, method, parameter, return names/values to use in your client application.

5. Create your client application.

#### 3.5.1.1 Overview

TeamConnect Web Services provide repositories (for ex. ContactRepository) that are interfaces to make Web Service requests. The result of a TeamConnect Web Service request will generally be data insertion to, update to, retrieval from, or deletion from a TeamConnect database.

To work with TeamConnect records supported by Web Services, you build TeamConnect Web Service client programs that use the TeamConnect Web Service API. You'll need to use a SOAP toolkit (such Apache CXF) that connects to and interacts with TeamConnect® Enterprise.

You can write a client program in the programming language of your choice. Write a client program to send a request to one of the TeamConnect Web Services such as the DocumentRepository or ContactRepository.

You don't need to install the TeamConnect Web Service API. You only need to install software for your programming language and the SOAP toolkit you will use to write client programs. For example, if you will write client programs in Java, you need to install the Java development kit and a SOAP toolkit such as Apache CXF.

TeamConnect Web Service operations are defined in WSDL (Web Services Definition Language) files. The TeamConnect Web Service source files (.WSDL and .XSD) are provided in the

TeamConnect installation package. You need to host the .WSDL and .XSD files on a web server or application server. To connect to a TeamConnect Web Service, you need to declare the URL to the .WSDL files in your client programs. This guide provides a format standard for the .WSDL URLs. Refer to your SOAP toolkit documentation for pre-defined functions provided to connect to a TeamConnect Web Service.

After establishing a connection to the WSDL, the client program can send requests to the TeamConnect Web Service to perform any operations supported by the TeamConnect Web Service. For example, after connecting to the ContactRepository, a client program could send an insertContact request to add a contact record.

A client program sends a SOAP request which the TeamConnect Web Service processes and sends a response. The request and response messages are XML files with a header and body.

For TeamConnect Web Service requests, the header must include a valid TeamConnect user name and password (also note that the user should belong to a group with sufficient rights to perform the requested operations). The header also lists classes that are referred to in the message. The message body specifies the requested operation with related parameters.

The WSDL source files provided with TeamConnect define the requests that a Web Service can process. The WSDL file includes the operations that a TeamConnect Web Service can perform, required parameters per operation, and the response per operation. SOAP header elements and errors thrown are also described in the WSDL file.

Typically before you begin to use TeamConnect Web Services, you need to download a SOAP toolkit or framework (such as Apache CXF) that interprets WSDL files, and also encodes and decodes XML requests and responses. When a TeamConnect Web Service receives a request, it sends an XML response. The SOAP toolkit will parse the response in a format appropriate to the programming language of the client program.

In addition, the SOAP toolkit generates stubs or client proxies that know how to locate TeamConnect® Enterprise. Usually one stub or client proxy is generated for each Web Service and your client application is written against the stubs.

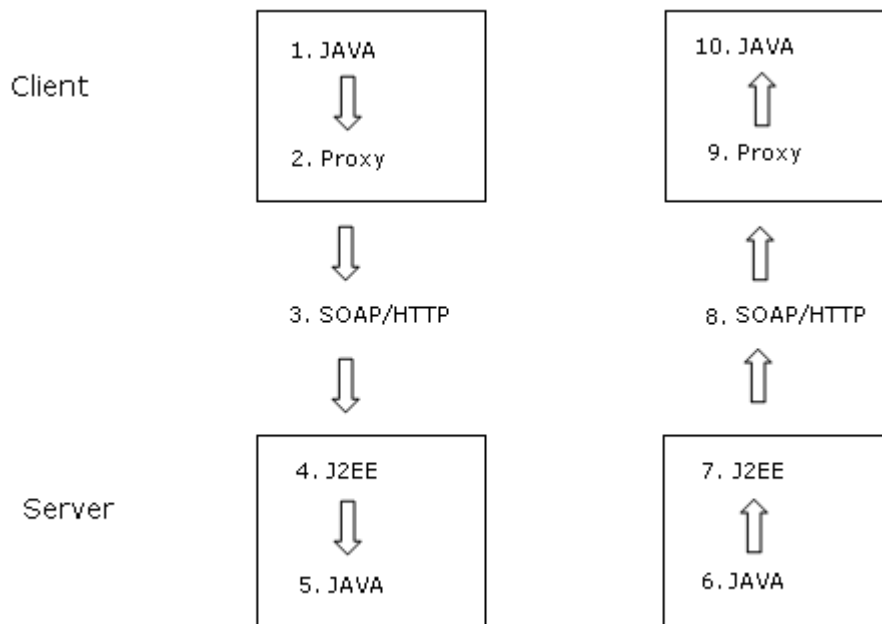
The toolkit you would use depends on the programming language you are using. Depending on the toolkit you use, you may not need to write any XML or you may write some of the XML messages. Note that the TeamConnect Web Service API uses document/literal style SOAP 1.1 (not rpc/encoded style) and WSDL 1.1.

**Note:** *When using TeamConnect Web Service functions, note that TeamConnect Rules associated with the operation you are performing such as update or insert will apply (for example, field validations).*

**Important:** *Samples in this guide use JAVA syntax (specific to the Apache CXF wsdl2java-generated source). If you are using a different programming language, you must get a corresponding 3rd party SOAP toolkit to convert the provided .WSDL and .XSD files to the appropriate source code. If you are using a programming language other than JAVA and SOAP toolkit other than Apache CXF, you should feel comfortable with the SOAP toolkit vendor's support.*

**Note:** *The code samples provided in this guide are not complete applications. You may need to refer to 3rd party software documentation for the tool you use to generate client-side code for how*

to perform authentication and how to create client proxies for repositories. For example, see [Putting Components Together in a Client Application](#) for more information.



**Data Flow Diagram for JAVA Application Using Web Service API**

#### 3.5.1.1.1 Supported Operations

The following table summarizes supported Web Service operations by TeamConnect object type.

##### Supported Web Service Operations

TeamConnect Object Type	Supported Operations
Accounts	<ul style="list-style-type: none"> <li>• insert</li> <li>• read</li> <li>• readChildAccounts</li> <li>• readRecentlyViewed</li> <li>• update</li> <li>• readByCriteria</li> <li>• activateAccount</li> <li>• deactivateAccount</li> <li>• allocateMoney</li> <li>• transferMoney</li> </ul>

	<ul style="list-style-type: none"><li>• withdrawMoney</li><li>• delete</li></ul>
Appointments	<ul style="list-style-type: none"><li>• insert</li><li>• read</li><li>• update</li><li>• readByCriteria</li><li>• readRecentlyViewed</li><li>• delete</li></ul>
Contacts	<ul style="list-style-type: none"><li>• insert</li><li>• read</li><li>• update</li><li>• readByCriteria</li><li>• delete</li><li>• readRecentlyViewed</li></ul>
Documents	<ul style="list-style-type: none"><li>• insert</li><li>• update</li><li>• read</li><li>• readByPath</li><li>• readChildDocuments</li><li>• readChildDocumentForName</li><li>• readDocumentFolderForDocumentOwner</li><li>• readByCriteria</li><li>• checkIn</li><li>• checkOut</li><li>• undoCheckout</li><li>• copy</li><li>• createShortcut</li><li>• createSubFolder</li><li>• createHyperlinkDefaultCategory</li><li>• createHyperlink</li></ul>

	<ul style="list-style-type: none"><li>• move</li><li>• revert</li><li>• delete</li><li>• readRecentlyViewed</li><li>• setAsRecentlyViewed</li></ul>
Expenses	<ul style="list-style-type: none"><li>• insert</li><li>• read</li><li>• update</li><li>• readByCriteria</li><li>• post</li><li>• void</li><li>• delete</li><li>• readRecentlyViewed</li></ul>
Group Accounts	<ul style="list-style-type: none"><li>• insert</li><li>• read</li><li>• update</li><li>• readByCriteria</li><li>• delete</li><li>• readRecentlyViewed</li></ul>
History records	<ul style="list-style-type: none"><li>• insert</li><li>• read</li><li>• update</li><li>• readByCriteria</li><li>• delete</li><li>• readRecentlyViewed</li></ul>
Invoice	<ul style="list-style-type: none"><li>• insert</li><li>• read</li><li>• update</li><li>• readByCriteria</li></ul>



	<ul style="list-style-type: none"><li>• post</li><li>• void</li><li>• delete</li><li>• readRecentlyViewed</li><li>• adjustInvoiceHeaders</li><li>• readActiveApprovals</li><li>• readCompletedApprovals</li><li>• readInvoiceApprovalsPendingOnPost</li></ul>
Involved	<ul style="list-style-type: none"><li>• insert</li><li>• read</li><li>• readInvolvedsForProject</li><li>• readInvolvedsByCriteria</li><li>• update</li><li>• delete</li></ul>
Projects	<ul style="list-style-type: none"><li>• insert</li><li>• read</li><li>• update</li><li>• readChildProjectsForEntityType</li><li>• readByCriteria</li><li>• changePhase</li><li>• delete</li><li>• readRecentlyViewed</li><li>• readProjectEntityTypes</li><li>• readProjectIntegrationSearches</li><li>• readProjectsUsingSearch</li></ul>
Tasks	<ul style="list-style-type: none"><li>• insert</li><li>• read</li><li>• update</li><li>• readByCriteria</li><li>• post</li></ul>

	<ul style="list-style-type: none"> <li>• void</li> <li>• reassign</li> <li>• delete</li> <li>• readRecentlyViewed</li> </ul>
User Accounts	<ul style="list-style-type: none"> <li>• insert</li> <li>• read</li> <li>• update</li> <li>• readByCriteria</li> <li>• delete</li> <li>• readRecentlyViewed</li> </ul>

## What is Not Supported

See [Operations Not Supported In Web Services](#) for a list of functions that need to be performed by using the TeamConnect application user interface (UI).

### 3.5.1.2 Key Concepts

This section provides general concepts or conventions to understand before you start to use TeamConnect Web Services.

#### 3.5.1.2.1 Repository Classes

Before you begin, you should understand that each TeamConnect object type is associated with a repository class (for example, ContactRepository). Methods are provided per repository for common TeamConnect functions. For example:

- insertContact (ContactCreate contact)
- updateContact (ContactUpdate contact)
- readContact (String uniqueKey, List properties)
- readContactsByCriteria (SearchCriteria criteria, Int limit, List properties)
- deleteContact (String uniqueKey)

**Note:** For a list of available properties that can be retrieved per record, see the elements listed in the original types.xsd file under the webservices directory. For example, the complexTypes, person, has a list of elements whose names are available properties to read or search from TeamConnect .

One HTTP request results from each repository class call. However, there are multiple subclasses associated with each TeamConnect object type (or repository class) that you use to store data (such as field values) when inserting (creating), updating, reading, or deleting TeamConnect records.

In a client program you instantiate a repository class once but you can use that repository instance multiple times to work with multiple records. You use the type classes or record unique keys as parameters to identify which record you are working with per request.

#### 3.5.1.2.2 Selective Record Updating and Reading

To maximize performance, methods to update type classes (TeamConnect records) and read type classes (TeamConnect records) allow you to specify the properties you want to update or read.

When you update a type class (TeamConnect record) using a Web Service, you identify the record to update by passing its unique key. The update request only updates your specified properties. Unlike a record update made by a TeamConnect rule, the record does not need to be retrieved first in order to change a property. This feature offers better performance through fewer HTTP requests and smaller data packets.

When you read a type class (TeamConnect record) using a Web Service, you must specify which properties to return. Only the specified property values are returned with a corresponding object. This feature improves performance through smaller data packets.

*Note: When a record is read by a TeamConnect rule, the returned object includes all of its property values.*

*Important: When reading records, any properties you do not include in the List<String> properties parameter may be returned with a null or 0 (zero) value. For example, if you are reading a contact record and specify three property values to return out of 24 properties available, then the remaining 21 properties will be returned with either null or 0 values. Note that you should ignore the null and 0 values for properties you did not explicitly retrieve.*

#### 3.5.1.2.3 Data Types

After you use a 3rd party tool (for example, Apache CXF wsdl2java) to generate client-side source code, you can view the resulting classes that provide information about data types in the corresponding directory: ...\\com\\mitratech\\teamconnect\\webservice\\type

### Dates and Time Zone Independence

When working with date fields (properties of type Date), dates can be either time-zone independent or time-zone dependent. You need to refer to the types.xsd and find the property's corresponding element type. If the element type is "dateTime" then the date value is time-zone dependent. If the type is "date" then the date value is time-zone independent.

#### 3.5.1.2.4 Request and Response Formats

Each repository method (request) call in your client application will be translated by the client proxy (provided by the SOAP toolkit) into an XML message with a SOAP wrapper. Each repository method results in one SOAP request over HTTP to the Web Service (TeamConnect ) server.

**Note:** Each SOAP request requires authentication to the TeamConnect system. See [SOAP Message Security Header](#) for more information.

#### 3.5.1.2.5 Error Handling

There are a few types of errors you may receive.

- Permission Denied errors will result if the user ID provided during authentication doesn't have sufficient access rights to perform the requested functions.
- Client-side errors. For example, an error would result if you tried to create a record but didn't populate one of its required fields.
- Server-side errors. After you generate client-side source code from the WSDL files, the `WebServiceFaultDetail` class provides a property, `detailedErrorMessage`, that is a wrapper for server-side error messages.

For server-side errors, the server throws the error, which is wrapped in an .XML message. This message is returned to the client and translated to a .JAVA object (or the appropriate programming language object equivalent) with the exception.

#### 3.5.1.2.6 Record Unique Keys

**Note:** This section describes the current convention of a record's unique key. This definition may change in the future.

When updating, reading, or deleting records, you need to pass the desired record's unique key. The unique key format is like: `TC_Object_Definition_Unique_Code + "_" + TC_record_primary_key` (for example, an invoice's unique key might look like **INVC\_3055**).

**Note:** The unique key value is not stored directly in the TeamConnect database. It is constructed by two values stored separately in the database.

If you don't know a record's unique key and want to read or retrieve its properties, you can alternatively use the `readXByCriteria` request. For example, to read a contact you can use the `contactRepository.readContactsByCriteria` request. You can either specify the properties to read in the target record directly as a `readContactsByCriteria` parameter or you can just retrieve the `uniqueKey` property in the search results and pass this value in the `readContact` request. Typically you will search or use the `readXByCriteria` request to get a record's unique key. Then you can use the unique key for update, read, or delete operations.

#### 3.5.1.3 Requirements

- A TeamConnect® Enterprise instance
- A valid TeamConnect® Enterprise user ID and password. This user must have sufficient rights to perform operations you plan to use through membership in a group with corresponding rights. For more information, see [Account Administration](#) for group record rights (global user rights, global admin rights, global setup rights) and group category and custom field rights.

For projects/matters (custom objects), you will need to manually grant access rights to your group per custom object under the Group **Record Rights** and **Category Rights** areas.

- TeamConnect Web Service source files (.WSDL and .XSD files), located at the following installation path (after running the TeamConnect Installer):

C:\Program Files\Mitratech\TeamConnect \*\utilities\webservices\

(where \* would be replaced by the current release version, for example,

C:\Program Files\Mitratech\TeamConnect 3.4 SP1\utilities\webservices\ )

- An SDK for the programming language you want to use.

If you are creating a Java client application, you must use a JDK version that is supported by TeamConnect. Refer to the TeamConnect Release Notes for supported versions.

- A third-party SOAP toolkit to generate the Web Service client API and code from the Web Service WSDL files. For example, Apache CXF. The SOAP toolkit should also provide the framework for creating a client proxy.

**Note:** *Samples provided in this guide are based on client-side JAVA code generated using Apache CXF 2.1. If you are using a different tool to generate client-side code, you need to reference the resulting source files.*

#### 3.5.1.3.1 Generating Client Source-code and API

This section describes the generation of the Web Service Client API and source-code using your SOAP toolkit.

**Note:** *If you will write a JAVA client program and use the Apache CXF toolkit/framework, then skip to the bottom of this section for information about locating the client API.*

Depending on which programming language and SOAP toolkit you plan to use, you will need to use the toolkit to convert the TeamConnect Web Service .WSDL and .XSD files to client-side code (for example .JAVA source). This code serves as the Client API for the TeamConnect Web Service classes, methods, properties, and return values you would write to in your client program.

Copy the TeamConnect Web Service files (.WSDL and schema or .XSD) from the TeamConnect installation path at C:\Program Files\Mitratech\TeamConnect \*\utilities\webservices\ (where \* is the version number) to a different working directory (where the path does not include spaces or special characters). The following is a list of TeamConnect Web Service files:

- account-repository.wsdl
- account-repository-schema.xsd
- appointment-repository.wsdl
- appointment-repository-schema.xsd
- contact-repository.wsdl
- contact-repository-schema.xsd
- document-repository.wsdl

- document-repository-schema.xsd
- expense-repository.wsdl
- expense-repository-schema.xsd
- group-account-repository.wsdl
- group-account-repository-schema.xsd
- history-repository.wsdl
- history-repository-schema.xsd
- invoice-repository.wsdl
- invoice-repository-schema.xsd
- involved-repository.wsdl
- involved-repository-schema.xsd
- project-repository.wsdl
- project-repository-schema.xsd
- repository-fault-schema.xsd
- task-repository.wsdl
- task-repository-schema.xsd
- types.xsd
- user-account-repository.wsdl
- user-account-repository-schema.xsd

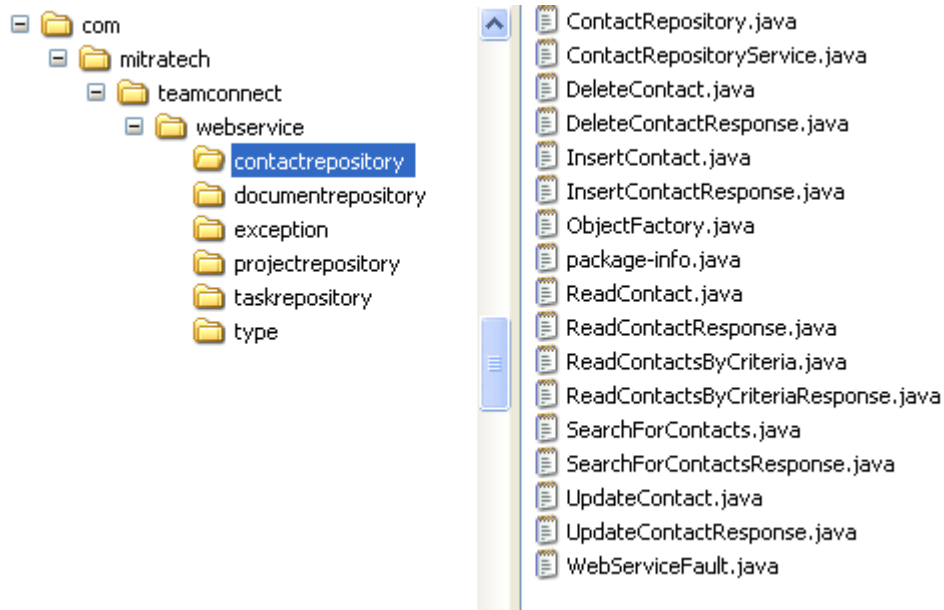
Use the tool provided in the 3rd party SOAP toolkit to convert the .WSDL files to the source code corresponding to the programming language you plan to code.

After generating the client-side source code, you must compile the code. Then package the binaries into a .JAR file (or appropriate package for your programming language, where the root directory should be \com).

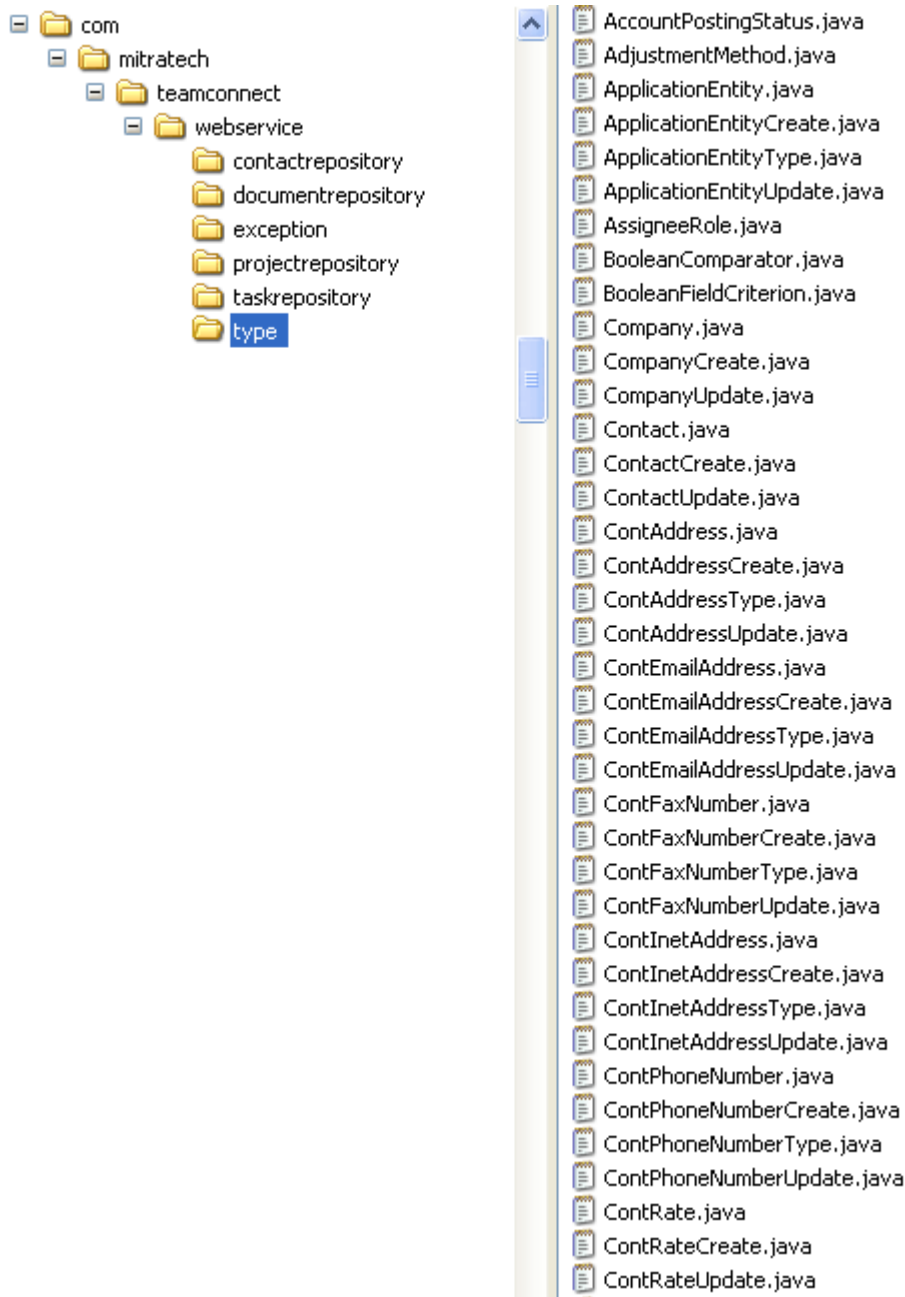
You need to add the resulting .JAR file (or appropriate package for your programming language) to your CLASSPATH environment variable (or appropriate environment variable for your programming language) before you can compile a custom application.

### **Client Source Code for JAVA and Apache CXF**

For example, after generating client-side source code from the contact-repository.wsdl file, a directory structure like the following results:



**Sample Client-side Source Code Directory (Contact Repository)**



Sample Client-side Source Code Directory (Types)

Continuing with the example of contacts, find the objects that you will typically use in the following generated client-side source code directories:

`\com\mitratech\teamconnect\webservice\contactrepository`

`\com\mitratech\teamconnect\webservice\type`

## Client API for JAVA and Apache CXF



If you plan to write a JAVA client program and use the Apache CXF toolkit/framework, then you only need to generate the client-side code. You do not need to compile the code because the TeamConnect Installer provides the client API in the form of a .JAR file located in the installation path:

```
C:\Program Files\Mitratech\TeamConnect *\utilities\webservices\teamconnect-webservice-client-*.jar
```

(where \* and \*\* will be the current version of TeamConnect )

#### 3.5.1.4 Creating an Application

The following section describes components required for a client application.

- Client Proxy
  - Reference to Client API path
  - Web Service URL
- Security Header (Interceptor)
- Client Application

##### 3.5.1.4.1 Client Proxy

The 3rd party SOAP toolkit should provide a class that creates a client proxy to translate the programming language used in your client application to the XML message format defined in the WSDL files. In your client application, you need to instantiate the given client proxy.

#### Client API Path

For each client proxy you also need to reference the class name of the client API that will be used. You can do this by passing the target entity name to populate the resulting client API qualified class name. For example, you'd construct the string for the class path to a client interface like :

```
"com.mitratech.teamconnect.webservice." + entityName.toLowerCase() + "repository." + entityName + "Repository"
```

(where depending on the repository type you are working with, you would pass in the corresponding entityName, such as Contact)

For more a more detailed example, see the [Client Proxy Sample Code](#) section of the [Client Application Components \(Java/Apache CXF\)](#) appendix.

#### Web Service URL

For each client proxy you also need to reference the URL of the TeamConnect Web Service that will be used. You can do this by passing the target entity name to populate the resulting TeamConnect Web Service URL. For example, you'd construct the string for a TeamConnect Web Service URL like :

```
"http://" + SERVER + ":" + PORT + "/" + APP NAME + "/webservice/" + StringUtils.uncapitalize(entityName) + "Repository"
```

(where depending on the repository type you are working with, you would pass in the corresponding entityName, such as Contact)

The format of the resulting Web Service URL would be like:

*http://localhost:8080/teamconnect/webservice/contactRepository*

For more a more detailed example, see the [Client Proxy Sample Code](#) section of the [Client Application Components \(Java/Apache CXF\)](#) appendix.

#### 3.5.1.4.2 SOAP Message Security Header

For each client proxy you can define an interceptor with SOAP message security header content (for TeamConnect authentication). The interceptor should be added to the client proxy once and afterward, the corresponding security header will be automatically inserted into each XML message resulting from a Web Service request.

The interceptor for the message security header needs to contain the following:

- A string variable for the TeamConnect user name used to authenticate to the system.  
If you are writing a Web Service client application in a Software as a Service (SaaS) environment, the user name should use the format: user@domain
- A string variable for the TeamConnect user password used to authenticate to the system. During message transfer the password will be sent as unencrypted clear text.  
If you need to transmit the password in a more secure manner, you can set up SSL between your client application and the TeamConnect instance.

For a more detailed example, see the [Security Headers Sample Code](#) section of the [Client Application Components \(Java/Apache CXF\)](#) appendix.

The TeamConnect Web API is compliant with the WS-Security 1.1 OASIS standard. Currently only authentication using the UsernameToken is supported. For more information, see:

[http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wss](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss)

### Sample SOAP Message Security Header for Authentication

In your custom application, you should define a security header for SOAP/HTTP requests that looks like:

```
<soap:Header>
<wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-secext-1.0.xsd" soap:mustUnderstand="1">
<wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/
01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="UsernameToken-1809471" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
<wsse:Username xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-secext-1.0.xsd">administrator</
wsse:Username>
<wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-
```

```
200401-wss-username-token-profile-1.0#PasswordText"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">password</wsse:Password>
</wsse:UsernameToken>
</wsse:Security>
</soap:Header>

//administrator should be replaced with a valid TeamConnect user name with
sufficient rights to perform required functions.

//password should be replaced with the administrator user's password.
```

#### 3.5.1.4.3 Putting Components Together in a Client Application

In a client application there are two key items to do:

- Instantiate a client proxy and type cast the proxy with the Web Service client interface that corresponds to the entity type you pass as parameter. For example, if you use the entityName, Contact, then you'd cast the proxy as type ContactRepository.
- Add the interceptor for SOAP security header to the client proxy.

Afterward you would use the client proxy to make Web Service request calls.

For more a more detailed example, see the [Client Application Sample Code](#) section of the [Client Application Components \(Java/Apache CXF\)](#) appendix.

#### 3.5.1.4.4 Common Functions

This section shows how you can use TeamConnect® Enterprise Web Services for common functions such as creating, updating, reading, searching, deleting records.

Each TeamConnect Web Service is represented by a repository WSDL file. Each Web Service corresponds to a TeamConnect entity or object type (for example, contacts, projects, accounts, invoices). There are a few recurring methods or operations that are common across all TeamConnect® Enterprise Web Services:

- Creating (Inserting) a record
- Updating a record
- Reading a record
- Searching for records
- Deleting a record

Some TeamConnect® Enterprise Web Services may have additional functions that are particular to that object, for example, posting or voiding records (such as invoices, expenses, tasks), and uploading documents, etc.

**Note:** *Some common functionality is not supported for involved party records.*

#### 3.5.1.4.4.1 Creating Records

Each TeamConnect Web Service provides a method for creating a corresponding record. The general form of the method is `insertObject` (`objectDataType`). For example, for the Contact repository the corresponding method is `insertContact` (`ContactCreate`). The resulting record's unique key is returned.

#### 3.5.1.4.4.2 Updating Records

Each TeamConnect Web Service provides a method for updating a corresponding record. The general form of the method is `updateObject` (`ObjectUpdate`). For example, for the Contact repository the corresponding method is `updateContact` (`ContactUpdate`). Among the properties of the `ContactUpdate` object that you need to set, the unique key of the target record to update is required. There is no return value.

An `updateObject` method does not retrieve all of a record's data before updating its properties. The target record to update is identified by its unique key, then you use set methods to update existing property values or populate empty property values.

To clear existing property values and reset the values to null, use the `getClearedProps` method provided in the `ObjectUpdate` class. For example, to update a contact of type person by clearing existing property values, use the `PersonUpdate.getClearedProps().add(Item)` method. Otherwise use the `PersonUpdate.getClearedProps().addAll(Items)` method where **Items** is a list of contact property names whose values to clear.

#### 3.5.1.4.4.3 Reading Records

Each TeamConnect Web Service provides a method for reading a record of its object type. The general form of the method is `readObject` (`String uniqueKey, List<String> properties`). For example, for the Contact repository the corresponding method is `readContact` (`String uniqueKey, List<String> properties`). The corresponding contact record is returned with only the specified property values.

You can find the list of available properties in the types.wsdl file. For example, to find the properties available for contact records, look for the complex types: `company`, `contact`, or `person`. Each element for those types represents a property that can be returned on read. Similarly you can find the properties in the corresponding client-side source code (for example, in the `Contact`, `Person`, `Company`, `ContAddress`, `ContEmailAddress` objects, etc.). For more information, refer to [Web Service API Reference](#).

#### 3.5.1.4.4.4 Searching Records

Each TeamConnect Web Service provides a method for searching records by an object type and by additional search criteria. The general form of the method is `readObjectsByCriteria` (`SearchCriteria criteria, int limit, List<String> properties`). For example, for the Contact repository the corresponding method is `readContactsByCriteria` (`SearchCriteria criteria, int limit, List<String> properties`). A list of record objects that meet the search criteria is returned with only the specified property values.

*Note: When defining the list of property values to return, remember that some of the entity properties are objects. In these cases you need to explicitly specify the property of the property (where the property type is object) to return. For an example, see [Code Snippet for readContactsByCriteria \(specify return properties of contact property of type object\)](#).*

The search method provided in TeamConnect® Enterprise Web Services is designed to search among a moderate number of records (for example, if you expect less than 100 records in the search results). For the limit parameter, it is recommended to enter 100 as the maximum number of records to return. If you need to search across a larger volume of records, you may want to use the application UI.

A detailed description of how to define search criteria is provided below. The properties parameter in the readContactsByCriteria method is defined and used in the same way as for the readContact method (described above).

## Search Criteria

You need to define search criteria. From the \types directory you need to use the FieldSearchClause object to hold your search criteria. For each search, you need to define the following:

- one or more field criterion

Field criterion are defined according to the data type for the field you are searching. For example, you can define field criterion for comparing a field that is a String type using the StringComparator and StringFieldCriterion objects.

- an operator (for example, AND or OR) that defines how the field criterion are combined in the search if you are using multiple field criteria

Each field criterion is made up of three parts: a field path (pointer to the database table/column that you will query), a comparator (defines how you are comparing the database record values against a literal value), and a literal value. The parts of a field criterion are illustrated in the following search fields screenshot.

The screenshot shows a search form with three numbered labels above it: 1., 2., and 3. Below these labels, the text 'First Name:' is followed by a dropdown menu containing 'Equal To' and a text input field containing 'John'.

**Example of a Contact Search by First Name**

In the contact search screenshot, First Name (1.) corresponds to the field path; Equal To (2.) corresponds to the comparator; and John (3.) is the literal value. Corresponding sample code for search criteria is in the Code Snippet section. See [Searching for Contacts](#) for a JAVA example of how to set search criteria.

## Searching Records By Custom Field

When defining search criteria where you want to search for a record by a custom field value, you need to define the searchKeyPath using the following format: TCCategoryTreePosition + "\_\_" + detailFieldName (where there are two underscores, "\_", between the tree position and detail field name).

For example, fieldPathExpression.setSearchKeyPath("DISP\_\_CorrectiveActionDeadlineDI");

For a more detailed example, see [Code Snippet for readContactsByCriteria \(search by custom field value\)](#).

**Note:** If you will search by a custom field value that is associated with a child category, then the TCCategoryTreePosition should be constructed like parentObjectRootCategoryTreePosition + "\_" + childCategoryTreePosition + "\_" + detailFieldName (where there is a single

*underscore, "\_", between the root category tree position and its child category tree positions. Also note that an object's root category tree position is the same as the object definition's unique code.*

#### 3.5.1.4.4.5 Working with Categories

Web Services support enabling categories for a record, disabling categories for a record, and reading categories (and category properties) for a record.

### Before You Begin

You need to get the following information before working with categories using Web Services:

- Category unique key—This value is constructed like <projectUniqueCode + "\_" + categoryTreePosition>

For example, if you are adding a parent category, Contracts, to a dispute record, then the category unique key would be "DISP\_CONT"

(where the tree position for Contracts is "CONT"; and the unique code for the Dispute object is "DISP")

Another example would be if you are adding a child category, Confidentiality Breach, to a dispute record, then the category unique key would be "DISP\_CONT\_COBR"

(where the following apply:

- the tree position for the child category, Confidentiality Breach, is "COBR"
  - the tree position for the parent category, Contracts, is "CONT"
  - the unique code for the Dispute object is "DISP")
- Category name

### Enabling Categories

When creating or updating a record, you can enable categories for the record. You can enable multiple categories for a record with the exception of invoice line items, tasks, and expenses.

***Important:*** *A category must already exist in TeamConnect before you can enable the category for a record using Web Services. Web Services do not support creating a category for an object definition. Use the TeamConnect Designer UI area to create new categories.*

When creating records that are system object or custom object types, the TeamConnect system automatically sets the default category to the corresponding object definition's root category (for example an account record's root category is Account). As a result, you are not required to set a category for records with the exception of invoice line items.

When creating an invoice line item record where the line item type is set to EXPENSE, you need to explicitly set one Expense Category. Otherwise, when creating an invoice line item record where the line item type is set to FEE, you need to explicitly set one Fee/Task Category.

**Caution:** If you set multiple categories for a line item, only the last set category will be saved.

**Note:** If a validation rule exists that requires new projects or matters to be saved with a category other than the root category, you may need to enable or add a category to the record when you create it using Web Services.

From the Category type class, use the setUniqueKey (String) method to identify which category you want to add to (enable for) a record. From an *ObjectCreate* class, you would use the getCategories() method to add an instance of the Category class for the type of record you are creating. For example:

```
Category cat = new Category();
cat.setUniqueKey("CONT_EXTE");
PersonCreate person = PersonCreate();
person.getCategories().add(cat);
```

Use the getCategories() method to:

- enable categories for a record during record creation or
- enable additional categories for a record during record update

## Reading Primary Categories

If a record has multiple categories and you are only interested in getting the primary category, include primaryCategory in the properties parameters.

## Disabling Categories

When updating a record, use the getCategoryDeleteUniqueKeys() method to disable a category for that record. For example:

```
PersonUpdate person = PersonUpdate();
person.getCategoryDeleteUniqueKeys().add("CONT_INTR");
\\where "CONT_INTR" is the unique key of the category to disable
```

## Reading Categories

Reading categories for a record is different from reading other record properties in that in the properties (List of string property names) parameter, you include "categories". The returned record object will include a list of category objects for that record (including all of that record's custom fields).

**Important:** Reading categories is the exception to the general rule that when you specify a record's properties to read, for properties that are objects, you need to specify the non-object property of the parent object property to retrieve.

For example:

```
Person readPerson = (Person) contactRepository.readContact(uniqueKey,
getPropertiesToRead());
}
```

```
private List<String> getPropertiesToRead() {
    List<String> properties = new ArrayList<String>();
    properties.add("firstName");
    properties.add("lastName");
    properties.add("address.type");
    properties.add("categories");
    return properties;
}
```

#### 3.5.1.4.4.6 Working with Custom Fields

Web Services support populating, editing, clearing, and reading custom field values.

### Before You Begin

A custom field must already exist in TeamConnect before you can add the custom field to a record using Web Services. Use the TeamConnect Designer UI area to create new custom fields (for an object/category combination).

**Important:** Each custom field is dependent on a record category. Before you can work with a custom field in a record, its corresponding category (or category hierarchy) must be enabled in that record. For more information, see [Working with Categories](#).

You also need to get the following information before working with categories using Web Services:

- Category unique key—This value is constructed like <projectUniqueCode + "\_" + categoryTreePosition>

For example, if you are adding a parent category, Contracts (with tree position "CONT") that was created under the Dispute object (with unique code "DISP"), then the category unique key would be "DISP\_CONT".

Another example would be if you are adding a child category, Confidentiality Breach (with tree position "COBR"), which is a child of the Contracts category under the Dispute object, then the unique key would be "DISP\_CONT\_COBR".

- Category name
- Custom field type (for example, Boolean, DateTime, Decimal, Involved, Note, Project, Text, TableItem)
- Custom field name

### Populating and Updating Custom Field Values

To populate a custom field value, you would enable the corresponding category for a record. There are several type classes available to instantiate a class or container for the properties of a custom field. Each of these type classes are associated with a field data type and include:

BooleanCustomField, DateTimeCustomField, DecimalCustomField, InvolvedCustomField, NoteCustomField, ProjectCustomField, TableItemCustomField, TextCustomField. Instantiate the appropriate custom field type class. Use the provided methods, setFieldName(String value) and



`setValue(boolean value)` to identify the custom field name to populate and to assign the field value for, respectively (where the `setValue` parameter type would vary).

Each custom field is associated with a category. From the `Category` type class, several methods are provided to add the custom field data object described in the paragraph above, including `getBooleanCustomFields()`; `getDateTimeCustomFields()`; `getDecimalCustomFields()`; `getInvolvedCustomFields()`; `getNoteCustomFields()`; `getProjectCustomFields()`; `getTableItemCustomFields()`; `getTextCustomFields()`.

A sample for populating a boolean custom field (associated with a corresponding category) for a record follows:

```
PersonCreate person = createPerson();
Category cat2 = new Category();
cat2.setUniqueKey("CONT");
BooleanCustomField synched = new BooleanCustomField();
synched.setFieldName("Synchronized");
synched.setValue(true);
cat2.getBooleanCustomFields().add(synched);
person.getCategories().add(cat2);
```

Use the custom field type class' `setValue()` method to populate or update a custom field's value.

## Clearing Custom Field Values

Use the custom field type class' `setValue()` method to clear a custom field's value. For example:

```
PersonUpdate person = PersonUpdate();
Category cat2 = new Category();
cat2.setUniqueKey("CONT");
BooleanCustomField synched = new BooleanCustomField();
synched.setFieldName("Synchronized");
synched.setValue();
//in this case set the value to null to clear the current custom field value.
//if the custom field were a String type, you'd set the value to ""
cat2.getBooleanCustomFields().add(synched);
person.getCategories().add(cat2);
```

## Reading Custom Field Values

When reading a record, specify the `categories` property as part of the return values. The resulting record's `categories` (property) object will also contain all associated custom fields and related custom field properties.

### 3.5.1.4.4.7 Deleting Records

Each TeamConnect Web Service provides a method for deleting a corresponding record. The general form of the method is `deleteObject` (String uniqueKey). For example, for the Contact repository the corresponding method is `deleteContact` (String uniqueKey). There is no return value.

## 3.5.2 Web Service API Reference

This chapter provides a reference for the TeamConnect Web Service repositories, their request methods, parameters, and return values. Summary information is also provided for the data objects or record property containers typically specified as request parameters.

### Data Object Summary Tables

This guide does not include detailed property descriptions for data objects (containers for record property values). Most of these properties and their data restrictions are documented in the Customization guide, Object Model Reference section. In the [API Reference Summary](#) section of this chapter, table summaries listing the data objects typically used within a repository's requests or returned from the server are provided. Those table summaries include references to the corresponding Object Model Reference object names.

To locate more detailed descriptions of a TeamConnect Web Service data object's property, find the corresponding Object Model Reference object name. Search for that Object Model Reference object name in the Customization guide or TeamConnect Designer area help. From the description of each Object Model Reference object, you can typically match the TeamConnect Web Service data object property name directly to the attribute name for the Object Model Reference object (although exceptions may exist).

***Note:** For example, if you were interested in learning more information about the property, **name**, for the Web Service data object, **account**, then you would search the Customization help/guide for the Object Model Reference attribute table, "TAccount" and within the resulting help/guide section, look for the Attribute, **name**, and its related Comments.*

There are a set of data objects used to set search criteria. Each of these data objects can generally be used to search multiple record types so they are listed separately from the repository summary tables. See [Data Objects Used in Search Criteria](#) for more information.

### System Lookup Table Items

There are a set of data objects that you use to populate a record (repository) property value(s) using existing items from TeamConnect system lookup tables. These data objects (classes) are subclasses of the lookupItem abstract class.

To add a system lookup table item to populate a record property, use the method, setStoredValue (String storedValue), where the storedValue should be the TeamConnect system lookup table item's Tree Position value. Note that the system lookup table item should already be defined in TeamConnect before you can use the item with Web Services.

For an example, [Code Snippet for creating a contact of type person and populating a property from a system lookup table](#).

#### 3.5.2.1 API Reference Summary

This section provides the following information per repository (Web Service interface):

- A table of requests
- A table of required fields to create a record with the repository

- A table of related data objects (record property containers)

**Note:** There are some remaining data objects (for example, related to search criteria), that are currently not documented in this section.

### 3.5.2.1.1 AccountRepository

#### AccountRepository Requests Summary

Request	Description
<a href="#">activateAccount</a>	Activates an account.
<a href="#">allocateMoney</a>	Deposits funds to an account.
<a href="#">deactivateAccount</a>	Deactivates an account.
<a href="#">deleteAccount</a>	Deletes an account record.
<a href="#">insertAccount</a>	Creates an account record.
<a href="#">readAccount</a>	Reads and returns an account record. Specified account properties are returned with the record.
<a href="#">readAccountsByCriteria</a>	Searches for account records that meet a given search criteria. Specified account properties are returned with the resulting records.
<a href="#">readChildAccounts</a>	Reads child account records of a given parent account.
<a href="#">readRecentlyViewedAccounts</a>	Reads and returns recently viewed account records.
<a href="#">transferMoney</a>	Transfers funds between accounts.
<a href="#">updateAccount</a>	Updates an existing account with the specified properties.
<a href="#">withdrawMoney</a>	Withdraws funds from an account.

#### AccountRepository Data Objects Summary

Data Object	Description
-------------	-------------

<p>1 account</p> <p>2 accountCreate</p> <p>3 accountUpdate</p>	<p>1 Account record property container returned when you read or search accounts.</p> <p>2 Account record property container used to set a new account's properties. This object is passed as a request parameter when you create (insert) an account.</p> <p>3 Account record property container used to update or clear an account's properties. This object is passed as a request parameter when you update an account.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">TAccount</a> object.</p>
<p><b>accountType</b></p>	<p>An enumeration used to set the account type. Options include:</p> <ul style="list-style-type: none"> <li>• BUDGET</li> <li>• RESERVE</li> </ul>
<p><b>accountOverdraftType</b></p>	<p>An enumeration used to set the account overdraft type. Options include:</p> <ul style="list-style-type: none"> <li>• NON_NEGATIVE</li> <li>• ALLOW_NEGATIVE</li> <li>• OVERDRAFT</li> </ul>
<p><b>accountInvolvedType</b></p>	<p>An enumeration used to set restrictions on the posting criteria for an account. The effect of this enumeration's value depends on other account posting criteria you have set in either the accountCreate or accountUpdate properties.</p> <p>If posting criteria is applied to tasks or expenses, use this enumeration to restrict account posting by the task/expense's associated contact. If posting criteria is applied to invoice line items, use this enumeration to restrict account posting by the line item's associated timekeeper.</p> <p>Options include:</p> <ul style="list-style-type: none"> <li>• ANY_INVOLVED</li> <li>• ONE_INVOLVED</li> </ul>
<p><b>accountProjectType</b></p>	<p>An enumeration used to set restrictions on the posting criteria for an account. The effect of this enumeration's value depends on other account posting criteria you have set in either the accountCreate or accountUpdate properties.</p>

	<p>Use this enumeration to restrict account posting by the task/expense/invoice line item's associated project.</p> <p>Options include:</p> <ul style="list-style-type: none"> <li>• ANY_PROJECT</li> <li>• ONE_PROJECT</li> <li>• BY_TYPE</li> </ul>
<b>accountVendorType</b>	<p>An enumeration used to set restrictions on the posting criteria for an account. The effect of this enumeration's value depends on other account posting criteria you have set in either the accountCreate or accountUpdate properties.</p> <p>Use this enumeration to restrict account posting by the invoice line item's associated vendor.</p> <p>Options include:</p> <ul style="list-style-type: none"> <li>• ANY_VENDOR</li> <li>• ONE_VENDOR</li> </ul>

## 3.5.2.1.2 AppointmentRepository

**AppointmentRepository Requests Summary**

<b>Request</b>	<b>Description</b>
<a href="#"><u>insertAppointment</u></a>	Saves a new appointment to the repository.
<a href="#"><u>updateAppointment</u></a>	Updates a previously saved appointment.
<a href="#"><u>readAppointment</u></a>	Retrieves a appointment from the repository based on a unique key. For a list of available properties, see the type class, appointment.
<a href="#"><u>readAppointmentsByCriteria</u></a>	Retrieves a set of appointments based on specified criteria. If the criteria is empty, then the search will be for all appointments. The specified limit takes precedence over the current system setting for the maximum number of search results to obtain. For a list of available properties, see the type class, appointment.
<a href="#"><u>deleteAppointment</u></a>	Deletes the appointment specified by the unique key.

[readRecentlyViewedAppointments](#)

Read and return recently reviewed appointments.

#### AppointmentRepository Data Objects Summary

Data Object	Description
1 appointment 2 appointmentCreate 3 appointmentUpdate	1 Appointment record property container returned when you read or search appointments. When defining a repository search's return properties, refer to the property names defined in this object.  2 Appointment record property container used to set a new appointment's properties. This object is passed as a request parameter when you create (insert) an appointment.  3 Appointment record property container used to update or clear an appointment's properties. This object is passed as a request parameter when you update an appointment.  <b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">TAppointment</a> object.
1 appointmentResource 2 appointmentResourceCreate 3 appointmentResourceUpdate	1 Appointment resource object property container returned when you read or search appointment resources.  2 Appointment resource object property container used to set a new appointment resource's properties.  3 Appointment resource object property container used to update or clear an appointment resource's properties.
<b>appointmentResourceType</b>	Subclass of lookupItem. Use the provided get methods to read an appointment resource's associated type.
<b>lookupItem</b>	Abstract class that provides get methods to read properties associated with TeamConnect system lookup table items.
1 attendee 2 attendeeCreate 3 attendeeUpdate	1 Attendee object property container returned when you read or search attendees.  2 Attendee object property container used to set a new attendee's properties.  3 Attendee object property container used to update or clear an attendee's properties.

<b>attendanceType</b>	<p>An enumeration used to indicate an attendee's plans to attend an appointment.</p> <p>Options include:</p> <ul style="list-style-type: none"> <li>• WILL_ATTEND</li> <li>• TENTATIVE</li> <li>• WILL_NOT_ATTEND</li> <li>• UNKNOWN</li> </ul>
-----------------------	---

## 3.5.2.1.3 ContactRepository

**ContactRepository Requests Summary**

<b>Request</b>	<b>Description</b>
<a href="#">deleteContact</a>	Deletes the contact specified by the unique key.
<a href="#">insertContact</a>	Saves a new Contact object to the repository.
<a href="#">readContact</a>	Retrieves a Contact from the repository based on a unique key. For a list of available properties, see the type class, contact.
<a href="#">readContactsByCriteria</a>	Retrieves a set of contacts based on specified criteria. If the criteria is empty, then the search will be for all contacts. The specified limit takes precedence over the current system setting for the maximum number of search results to obtain. For a list of available properties, see the type class, contact.
<a href="#">ReadRecentlyViewedContacts</a>	Retrieve a list of recently viewed contacts.
<a href="#">updateContact</a>	Updates a previously saved contact.

**ContactRepository Data Objects Summary**

<b>Data Object</b>	<b>Description</b>
1 Company 2 CompanyCreate 3 CompanyUpdate	<p>All three objects are subclasses of the related Contact objects. The Company objects are used to work with contact records of type, Company.</p> <p>1 Company contact record property container returned when you read or search companies. When defining a repository search's</p>

	<p>return properties, refer to the property names defined in this object.</p> <p>2 Company contact record property container used to set a new company's properties. This object is passed as a request parameter when you create (insert) a company.</p> <p>3 Company contact record property container used to update or clear a company's properties. This object is passed as a request parameter when you update a company.</p> <p><b>Note:</b> <i>CompanyCreate is a subclass of ContactCreate. CompanyUpdate is a subclass of ContactUpdate.</i></p> <p><b>Note:</b> <i>For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">TContact</a> object.</i></p>
<p>1 Person</p> <p>2 PersonCreate</p> <p>3 PersonUpdate</p>	<p>All three objects are subclasses of the related Contact objects. The Person objects are used to work with contact records of type, Person.</p> <p>1 Person contact record property container returned when you read or search person contacts. When defining a repository search's return properties, refer to the property names defined in this object.</p> <p>2 Person contact record property container used to set a new person contact's properties. This object is passed as a request parameter when you create (insert) a person contact.</p> <p>3 Person contact record property container used to update or clear a person contact's properties. This object is passed as a request parameter when you update a person contact.</p> <p><b>Note:</b> <i>PersonCreate is a subclass of ContactCreate. PersonUpdate is a subclass of ContactUpdate.</i></p> <p><b>Note:</b> <i>For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">TContact</a> object.</i></p>
<p>1 ContAddress</p> <p>2 ContAddressCreate</p> <p>3 ContAddressUpdate</p>	<p>All three objects are used to work with contact mailing addresses within contact records.</p> <p>1 Contact address object property container returned when you read or search contacts, including the ContAddress property.</p>



	<p>2 Contact address object property container used to set properties for a contact address.</p> <p>3 Contact address object property container used to update or clear properties for a contact address.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JContAddress</a> object.</p>
<b>contAddressType</b>	Subclass of lookupItem. Use the provided get methods to read the properties related to a contact address type.
<b>countryItem</b>	Subclass of lookupItem. Use the provided get methods to read the properties related to a contact address country.
<p>1 ContEmailAddress</p> <p>2 ContEmailAddressCreate</p> <p>3 ContEmailAddressUpdate</p>	<p>All three objects are used to work with contact email addresses within contact records.</p> <p>1 Contact email address object property container returned when you read or search contact email addresses.</p> <p>2 Contact email address object property container used to set properties for a contact email address.</p> <p>3 Contact email address object property container used to update or clear properties for a contact email address.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JContEmail</a> object.</p>
<b>contEmailAddressType</b>	Subclass of lookupItem. Use the provided get methods to read the properties related to a contact email address' associated type.
<p>1 ContFaxNumber</p> <p>2 ContFaxNumberCreate</p> <p>3 ContFaxNumberUpdate</p>	<p>All three objects are used to work with contact fax numbers within contact records.</p> <p>1 Contact fax number object property container returned when you read or search contact fax numbers.</p> <p>2 Contact fax number object property container used to set properties for a new contact fax number.</p> <p>3 Contact fax number object property container used to update or clear properties for a contact fax number.</p>

	<p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JContFax</a> object.</p>
<b>contFaxNumberType</b>	Subclass of lookupItem. Use the provided get methods to read the properties related to a contact fax number's associated type.
<ul style="list-style-type: none"> <li>1 ContInetAddress</li> <li>2 ContInetAddressCreate</li> <li>3 ContInetAddressUpdate</li> </ul>	<p>All three objects are used to work with contact internet addresses (website URL's) within contact records.</p> <ul style="list-style-type: none"> <li>1 Contact internet address object property container returned when you read or search contact internet addresses.</li> <li>2 Contact internet address object property container used to set properties for a new contact internet address.</li> <li>3 Contact internet address object property container used to update or clear properties for a contact internet address.</li> </ul> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JContInetAddress</a> object.</p>
<b>contInetAddressType</b>	Subclass of lookupItem. Use the provided get methods to read the properties related to a contact email address' associated type.
<ul style="list-style-type: none"> <li>1 ContPhoneNumber</li> <li>2 ContPhoneNumberCreate</li> <li>3 ContPhoneNumberUpdate</li> </ul>	<p>All three objects are used to work with contact phone numbers within contact records.</p> <ul style="list-style-type: none"> <li>1 Contact phone number object property container returned when you read or search contact phone numbers.</li> <li>2 Contact phone number object property container used to set properties for a new contact phone number.</li> <li>3 Contact phone number object property container used to update or clear properties for a contact phone number.</li> </ul> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JContPhone</a> object.</p>
<b>contPhoneNumberType</b>	Subclass of lookupItem. Use the provided get methods to read the properties related to a contact phone number's associated type.
<ul style="list-style-type: none"> <li>1 ContDefaultRate</li> <li>2 ContDefaultRateCreate</li> </ul>	All three objects are used to work with contact default rates within contact records.

<p>3 ContDefaultRateUpdate</p>	<p>1 Contact default rate object property container returned when you read or search contact default rates.</p> <p>2 Contact default rate object property container used to set properties for a new contact default rate.</p> <p>3 Contact default rate object property container used to update or clear properties for a contact default rate.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JContRate</a> object.</p>
<p>1 contTaskRate 2 contTaskRateCreate 3 contTaskRateUpdate</p>	<p>All three objects are used to work with contact task rates within contact records.</p> <p>1 Contact task rate object property container returned when you read or search contacts.</p> <p>2 Container used to set properties for contact invoice task rates.</p> <p>3 Contact rate object property container used to update or clear properties for contact invoice task rates.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JContRate</a> object.</p>
<p>1 contInvoiceTaskRate 2 contInvoiceTaskRateCreate 3 contInvoiceTaskRateUpdate</p>	<p>All three objects are used to work with contact invoice task rates within contact records.</p> <p>1 Contact invoice task rate object property container returned when you read or search contacts.</p> <p>2 Container used to set properties for contact invoice task rates.</p> <p>3 Contact rate object property container used to update or clear properties for contact invoice task rates.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JContRate</a> object.</p>
<p>1 contDefaultRate 2 contDefaultRateCreate 3 contDefaultRateUpdate</p>	<p>All three objects are used to work with default contact rates within contact records.</p> <p>1 Default contact rate object property container returned when you read or search contacts.</p> <p>2 Default contact rate object property container used to set properties for a default contact rate.</p>

	<p>3 Default contact rate object property container used to update or clear properties for a default contact rate.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JContRate</a> object.</p>
<b>currencyItem</b>	<p>Subclass of lookupItem. This class corresponds to the TeamConnect Multi-currency lookup table's items.</p> <p>Use the provided get methods to read the properties related to either a contact rate's currency or invoice currency.</p> <p>The currencyItem name property corresponds to the TeamConnect multi-currency lookup table Currency code.</p> <p>The currencyItem name property</p>
<p>1 ContRelation</p> <p>2 ContRelationCreate</p> <p>3 ContRelationUpdate</p>	<p>All three objects are used to work with contact relations within contact records.</p> <p>1 Contact relation object property container returned when you read or search contact relations.</p> <p>2 Contact relation object property container used to set properties for a new contact relation.</p> <p>3 Contact relation object property container used to update or clear properties for a contact relation.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JContRelation</a> object.</p>
<b>contRelationType</b>	<p>Subclass of lookupItem. Use the provided get methods to read the properties related to a contact relation's associated type.</p>
<b>relationDirection</b>	<p>An enumeration used to identify the direction between two relations. Options include:</p> <ul style="list-style-type: none"> <li>• FROM_THIS</li> <li>• TO_THIS</li> </ul>
<p>1 ContSkill</p> <p>2 ContSkillCreate</p> <p>3 ContSkillUpdate</p>	<p>All three objects are used to work with contact skills within contact records.</p> <p>1 Contact skill object property container returned when you read or search contact skills.</p>

	<p>2 Contact skill object property container used to set properties for a new contact skill.</p> <p>3 Contact skill object property container used to update or clear properties for a contact skill.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JContSkill</a> object.</p>
<b>contSkillType</b>	Subclass of lookupItem. Use the provided get methods to read the properties related to a contact skill's associated type.
<p>1 ContTerritory</p> <p>2 ContTerritoryCreate</p> <p>3 ContTerritoryUpdate</p>	<p>All three objects are used to work with contact territories within contact records.</p> <p>1 Contact territory object property container returned when you read or search contact territories.</p> <p>2 Contact territory object property container used to set properties for a new contact territory.</p> <p>3 Contact territory object property container used to update or clear properties for a contact territory.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JContTerritory</a> object.</p>
<b>contTerritoryItem</b>	Subclass of lookupItem. Use the provided get methods to read the properties related to a contact territory.

## 3.5.2.1.4 DocumentRepository

**DocumentRepository Requests Summary**

Request	Description
<a href="#">insertDocument</a>	Saves a new Document object to the repository.
<a href="#">createShortcut</a>	Creates a shortcut or link to a document (file, folder, or existing shortcut) with a given unique key. The parentFolderUniqueKey defines the folder where the shortcut will be created.

<a href="#"><u>createSubFolder</u></a>	Creates a sub-folder with given name in given parent folder.
<a href="#"><u>createHyperlinkDefaultCategory</u></a>	Creates a hyperlink and sets its category to the default Document category.
<a href="#"><u>createHyperlink</u></a>	Creates a hyperlink.
<a href="#"><u>updateDocument</u></a>	Updates a previously saved document.
<a href="#"><u>readDocument</u></a>	Retrieves a Document from the repository based on a unique key. For a list of available properties, see the type class, Document.
<a href="#"><u>readDocumentsByCriteria</u></a>	Retrieves a set of documents based on specified criteria. If the criteria is empty, then the search will be for all documents. The specified limit takes precedence over the current system setting for the maximum number of search results to obtain. For a list of available properties, see the type class, Document.
<a href="#"><u>readChildDocuments</u></a>	<p>Performs two functions:</p> <ul style="list-style-type: none"> <li>• For a given document file (parent), gets older document versions of that file.</li> <li>• For a given folder, gets documents within that folder.</li> </ul> <p>Returns documents with only specified property values. For a list of available properties, see the type class, document.</p>
<a href="#"><u>readChildDocumentForName</u></a>	Retrieves a document from the repository given the file name and the unique ID of its parent folder.
<a href="#"><u>readDocumentFolderForDocument Owner</u></a>	Retrieves a document folder from the repository given the unique key of the folder's owner (user).
<a href="#"><u>readDocumentByPath</u></a>	Retrieves a Document from the repository based on a folder path.
<a href="#"><u>checkOut</u></a>	Attempts to check out a document with a given unique key.
<a href="#"><u>undoCheckOut</u></a>	Attempts to undo check out of a given document.

<a href="#">checkIn</a>	Attempts to check in a document with a given unique key.
<a href="#">revert</a>	Attempts to revert a given document to a given old version.
<a href="#">moveDocument</a>	Moves a document to a new parent folder.
<a href="#">copyDocument</a>	Copies a document to a new parent folder.
<a href="#">deleteDocument</a>	Deletes the persistent document specified by the unique key.
<a href="#">ReadRecentlyViewedDocuments</a>	Retrieves a list of recently viewed documents.
<a href="#">setDocumentAsRecentlyViewed</a>	Set a document as recently viewed.
<a href="#">readFolderHierarchy</a>	Retrieves a hierarchical list of document folders for one or more projects.

#### DocumentRepository Data Objects Summary

Data Object	Description
1 document 2 documentCreate 3 documentUpdate	<p>1 Document record property container returned when you read or search documents. When defining a repository search's return properties, refer to the property names defined in this object.</p> <p>2 Document record property container used to set a new document's properties. This object is passed as a request parameter when you create (insert) a document.</p> <p>3 Document record property container used to update or clear a document's properties. This object is passed as a request parameter when you update a document.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">TDocument</a> object.</p>
<b>documentContentType</b>	<p>This object defines the document file type. In an application you would typically use this object's properties to read or search documents.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to this object, see the <a href="#">Object Model Reference</a> for the <a href="#">YDocuContentType</a> object.</p>

<b>documentType</b>	<p>An enumeration used to indicate the document type.</p> <p>Options include:</p> <ul style="list-style-type: none"> <li>• FILE</li> <li>• FOLDER</li> <li>• HYPERLINK</li> <li>• SHORTCUT</li> <li>• HISTORY—the documentType for previous versions of a document file. Note that a document with documentType HISTORY is different from a history record.</li> </ul> <p><b>Note:</b> For more information about the data restrictions for properties that belong to this object, see the <a href="#">Object Model Reference</a> for the <a href="#">YDocuContentType</a> object.</p>
---------------------	--

## 3.5.2.1.5 ExpenseRepository

**ExpenseRepository Requests Summary**

Request	Description
<a href="#">deleteExpense</a>	Deletes an expense record.
<a href="#">insertExpense</a>	Creates an expense record.
<a href="#">postExpense</a>	Posts an expense record.
<a href="#">readExpense</a>	Reads and returns an expense record. Specified expense properties are returned with the record.
<a href="#">readExpensesByCriteria</a>	Searches for expense records that meet a given search criteria. Specified invoice properties are returned with the resulting records.
<a href="#">readRecentlyViewedExpenses</a>	Read and return recently viewed expense records.
<a href="#">updateExpense</a>	Updates an expense record. Specified invoice properties are updated.
<a href="#">voidExpense</a>	Voids an expense record.

**ExpenseRepository Data Objects Summary**



Data Object	Description
<b>accountPostingStatus</b>	<p>An enumeration used to track whether a finance record has posted to an account. Typically in an application, you would read or search by this object's value. You would not explicitly set a financial record's posting status.</p> <p>Options include:</p> <ul style="list-style-type: none"> <li>• NOT_SUBMITTED</li> <li>• POSTED</li> <li>• FAILED</li> </ul>
<p>1 expense</p> <p>2 expenseCreate</p> <p>3 expenseUpdate</p>	<p>1 Expense record property container returned when you read or search expenses.</p> <p>2 Expense record property container used to set a new expense's properties. This object is passed as a request parameter when you create (insert) an account.</p> <p>3 Expense record property container used to update or clear an expense's properties. This object is passed as a request parameter when you update an expense.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">TExpense</a> object.</p>

## 3.5.2.1.6 GroupAccountRepository

**GroupAccountRepository Requests Summary**

Request	Description
<a href="#">deleteGroupAccount</a>	Deletes the group account specified by the unique key.
<a href="#">insertGroupAccount</a>	Saves a new group account object to the repository.
<a href="#">readGroupAccount</a>	Retrieves a group account from the repository based on a unique key. For a list of available properties, see the type class, groupAccount.
<a href="#">readGroupAccountsByCriteria</a>	Retrieves a set of group accounts based on specified criteria. If the criteria is empty, then the search will be for all group accounts. The specified limit takes precedence over the current system setting for the maximum number of

	search results to obtain. For a list of available properties, see the type class, groupAccount.
<a href="#">readRecentlyViewedGroupAccounts</a>	Read and return recently viewed group accounts.
<a href="#">updateGroupAccount</a>	Updates a previously saved group account.

#### GroupAccountRepository Data Objects Summary

Data Object	Description
1 groupAccount 2 groupAccountCreate 3 groupAccountUpdate	<p>1 Group account record property container returned when you read or search groups.</p> <p>2 Group account record property container used to set a new group's properties. This object is passed as a request parameter when you create (insert) a group.</p> <p>3 Group account record property container used to update or clear a group's properties. This object is passed as a request parameter when you update a group.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">YGroup</a> object.</p>

#### 3.5.2.1.7 HistoryRepository

##### HistoryRepository Requests Summary

Request	Description
<a href="#">insertHistory</a>	Saves a new history record to the repository.
<a href="#">readHistoriesByCriteria</a>	Retrieves a set of records based on specified criteria. If the criteria is empty, then the search will be for all records. The specified limit takes precedence over the current system setting for the maximum number of search results to obtain. For a list of available properties, see the type class, history.
<a href="#">readHistory</a>	Retrieves a record from the repository based on a unique key. For a list of available properties, see the type class, history.
<a href="#">updateHistory</a>	Updates a previously saved history record.

<a href="#">deleteHistory</a>	Deletes the record specified by the unique key.
<a href="#">readRecentlyViewedHistories</a>	Read recently viewed histories.

**HistoryRepository Data Objects Summary**

Data Object	Description
1 history 2 historyCreate 3 historyUpdate	<p>1 History record property container returned when you read or search history records. When defining a repository search's return properties, refer to the property names defined in this object.</p> <p>2 History record property container used to set a new history record's properties. This object is passed as a request parameter when you create (insert) a history record.</p> <p>3 History record property container used to update or clear a history record's properties. This object is passed as a request parameter when you update a history record.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">THistory</a> object.</p> <p><b>Note:</b> <i>historyOwner</i> refers to the unique key (String) of a parent record that the history describes.</p>

## 3.5.2.1.8 InvoiceRepository

**InvoiceRepository Requests Summary**

Request	Description
<a href="#">adjustInvoiceHeader</a>	Adjust an invoice on the header level.
<a href="#">deleteInvoice</a>	Deletes an invoice record.
<a href="#">insertInvoice</a>	Creates an invoice record.
<a href="#">postInvoice</a>	Posts an invoice record.
<a href="#">readActiveApprovals</a>	Read active approvals for an invoice.
<a href="#">readCompletedApprovals</a>	Read completed approvals for an invoice.

<a href="#">readInvoice</a>	Reads and returns an invoice record. Specified invoice properties are returned with the record.
<a href="#">readInvoiceApprovalsPendingOnPost</a>	Read all pending invoice approvals that are pending on post.
<a href="#">readInvoicesByCriteria</a>	Searches for invoice records that meet a given search criteria. Specified invoice properties are returned with the resulting records.
<a href="#">readRecentlyViewedInvoices</a>	Read recently viewed invoices.
<a href="#">updateInvoice</a>	Updates an invoice record. Specified invoice properties are updated.
<a href="#">voidInvoice</a>	Voids an invoice record.

#### InvoiceRepository Data Objects Summary

Data Object	Description
<b>adjustmentMethod</b>	<p>An enumeration used to identify the way an invoice summary is adjusted. Options include:</p> <ul style="list-style-type: none"> <li>• REDUCE_BY_AMOUNT</li> <li>• REDUCE_BY_PERCENT</li> <li>• NEW_AMOUNT</li> </ul> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JInvHeaderAdjustment</a> object.</p>
<b>currencyItem</b>	See <a href="#">currencyItem</a> .
1 invoiceAdjustment 2 invoiceAdjustmentCreate 3 invoiceAdjustmentUpdate	<p>All three objects are used to work with invoice summary adjustments within invoice records.</p> <p>1 Invoice summary adjustment object property container returned when you read or search invoices.</p> <p>2 Invoice summary adjustment object property container used to set properties for an invoice.</p> <p>3 Invoice summary adjustment object property container used to update an invoice.</p>

	<p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JInvHeaderAdjustment</a> object.</p>
<b>invoiceAdjustmentTarget</b>	<p>An enumeration used to identify which property of the invoice summary to adjust. Options include:</p> <ul style="list-style-type: none"> <li>• TOTAL_FEES</li> <li>• TOTAL_EXPENSES</li> <li>• TOTAL_INVOICE</li> </ul>
<p>1 invoice 2 invoiceCreate 3 invoiceUpdate</p>	<p>1 Invoice record property container returned when you read or search invoices.</p> <p>2 Invoice record property container used to set a new invoice's properties. This object is passed as a request parameter when you create (insert) an invoice.</p> <p>3 Invoice record property container used to update or clear an invoice's properties. This object is passed as a request parameter when you update an invoice.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">TInvoice</a> object.</p>
<b>invoiceNonUSTax</b>	<p>Invoice Non-US Taxes object property container returned when you read or search invoices.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned object, see the <a href="#">Object Model Reference</a> for the <a href="#">TInvoice</a> object.</p>
<b>invoiceNonUSTaxType</b>	<p>Subclass of lookupItem. Use the provided get methods to read the properties related to an invoice's Non-US Tax category.</p> <p><b>Note:</b> An invoice can have multiple Non-US Tax amounts.</p>
<b>invoicePostingStatus</b>	<p>An enumeration used to track whether a finance record has posted to an account. Typically in an application, you would read or search by this object's value. You would not explicitly set a invoice's posting status.</p> <p>Options include:</p>

	<ul style="list-style-type: none"> <li>• NOT_SUBMITTED</li> <li>• POSTED</li> <li>• FAILED</li> <li>• REJECTED</li> </ul>
<p>1 <code>lineItemAdjustment</code></p> <p>2 <code>lineItemAdjustmentCreate</code></p> <p>3 <code>lineItemAdjustmentUpdate</code></p>	<p>All three objects are used to work with invoice line item adjustments within invoice records.</p> <p>1 Line item adjustment object property container returned when you read or search invoices.</p> <p>2 Line item adjustment object property container used to set properties for an invoice line item adjustment.</p> <p>3 Line item adjustment object property container set properties for an invoice line item adjustment.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JInvLineItem</a> object.</p>
<b>lineItemAdjustmentReason</b>	<p>Subclass of <code>lookupItem</code>. Use the provided get methods to read the properties related to a line item adjustment reason. This class corresponds to the options defined by the TeamConnect system lookup table, Invoice Rejection Reason.</p>
<b>lineItemAdjustmentTarget</b>	<p>An enumeration that identifies the line item property to adjust. Options include:</p> <ul style="list-style-type: none"> <li>• RATE</li> <li>• QUANTITY</li> <li>• DISCOUNT</li> <li>• TOTAL</li> </ul>
<p>1 <code>lineItem</code></p> <p>2 <code>lineItemCreate</code></p> <p>3 <code>lineItemUpdate</code></p>	<p>1 Line Item record property container returned when you read or search invoices.</p> <p>2 Line Item record property container used to create a line item and add it to an invoice.</p> <p>3 Line Item record property container used to update a line item.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JInvLineItem</a> object.</p>

<b>lineItemType</b>	An enumeration that sets the line item type. Options include: <ul style="list-style-type: none"> <li>• EXPENSE</li> <li>• FEE</li> </ul>
<b>taskActivityItem</b>	Subclass of lookupItem. Use the provided get methods to read the properties related to a invoice line item task activity option. This class corresponds to options defined in the TeamConnect system lookup table, Activity Item.

## 3.5.2.1.9 LookupTableSource

**LookupTableSource Requests Summary**

Request	Description
<a href="#">readSystemLookupTable</a>	Retrieves a list of properties for a system table code. You use the properties to read values from lookup tables.

## 3.5.2.1.10 ProjectRepository

Use the ProjectRepository to manage project, matters, or custom object records (for example, create, update, read, search, delete). The corresponding custom object must already be defined in the TeamConnect Designer area Object Definitions screen. Corresponding categories for the custom object must also be defined in the TeamConnect Designer area Object Definitions screen.

**ProjectRepository Requests Summary**

Request	Description
<a href="#">changePhase</a>	Changes the phase of a project record. The phase must already be defined in TeamConnect.
<a href="#">deleteProject</a>	Deletes a project record.
<a href="#">insertProject</a>	Creates a project record.
<a href="#">readChildProjectsForEntityType</a>	Reads and returns a project record's child projects by a given category. Specified project properties are returned with the resulting records.
<a href="#">readProject</a>	Reads and returns a project record. Specified project properties are returned with the record.
<a href="#">readProjectEntityTypes</a>	Get a list of project entity types in TeamConnect.

<a href="#">readProjectIntegrationSearches</a>	Get a list of search views that are of type "Integration", each returned search view should have a unique key, a name, and its associated entity type.
<a href="#">readProjectsByCriteria</a>	Searches for project records that meet a given search criteria. Specified project properties are returned with the resulting records.
<a href="#">readProjectsUsingSearch</a>	Get a list of records that match the search criteria given a unique key of search view of "Integration" type.
<a href="#">readRecentlyViewedProjects</a>	Read recently viewed projects.
<a href="#">updateProject</a>	Updates a project record. Specified project properties are updated.

#### ProjectRepository Data Objects Summary

Data Object	Description
<b>assigneeRole</b>	Use this object to read or search projects by associated assignee role.  <i>Note: Project assignee role values must be pre-defined in TeamConnect UI in the Project (custom object) object definition.</i>
1 embeddedEntity 2 embeddedEntityCreate 3 embeddedEntityUpdate	1 Embedded object record property container returned when you read or search projects. 2 Embedded object record property container used to set properties of an embedded object record. 3 Embedded object record property container used to set properties of an embedded object record.  <i>Note: For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">WObjdProjectInfo</a> object.</i>
<b>embeddedEntityType</b>	Use this object's uniqueCode property to identify the existing embedded object definition to work with.
<b>phase</b>	Use this object's properties to search or read projects. Also use this object's properties to search or read projects by the <b>phaseInterval</b> (object) property.



<p><b>phaseInterval</b></p>	<p>The phaseInterval object is a container for properties about a phase's duration (for example, the date when a phase became active). In an application, typically you would</p> <p>Project phase interval object property container returned when you read or search projects.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned object, see the <a href="#">Object Model Reference</a> for the <a href="#">JProjPhase</a> object.</p>
<p>1 projectAssignee 2 projectAssigneeCreate 3 projectAssigneeUpdate</p>	<p>1 Project assignee record property container returned when you read or search projects.</p> <p>2 Project assignee record property container used to set a project's properties.</p> <p>3 Project assignee record property container used to set a project's properties.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JProjAssignee</a> object.</p>
<p>1 project 2 projectCreate 3 projectUpdate</p>	<p>1 Project (custom object) record property container returned when you read or search projects.</p> <p>2 Project (custom object) record property container used to set a new project's properties. This object is passed as a request parameter when you create (insert) a project.</p> <p>3 Project (custom object) record property container used to update or clear a project's properties. This object is passed as a request parameter when you update a project.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">TProject</a> object.</p>
<p>1 projRelation 2 projRelationCreate 3 projRelationUpdate</p>	<p>1 Project relation record property container returned when you read or search projects.</p> <p>2 Project relation record property container used to set a project's properties.</p> <p>3 Project relation record property container used to set a project's properties.</p>

	<b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JProjRelation</a> object.
<b>projRelationType</b>	Subclass of lookupItem. Use the provided get methods to read the properties related to a project relation's associated type.
<b>relationDirection</b>	For more information, see <a href="#">relationDirection</a> .

## 3.5.2.1.11 InvolvedRepository

Use this repository to work with involved party records or contacts who are involved in a project (matter).

**InvolvedRepository Requests Summary**

Request	Description
<a href="#">deleteInvolved</a>	Deletes an involved party record.
<a href="#">insertInvolved</a>	Inserts or creates an involved party record (associated with a project/matter).
<a href="#">readInvolved</a>	Reads and returns an involved record. Specified involved properties are returned with the record.
<a href="#">readInvolvedsByCriteria</a>	Searches for involved party records that meet a given search criteria. Specified involved party properties are returned with the resulting records.
<a href="#">readInvolvedsForProject</a>	Reads and returns a list of involved party records given the unique id of an associated project (matter). Specified involved properties are returned with the record.
<a href="#">updateInvolved</a>	Updates an involved record. Properties that can be updated include categories and custom field values. Existing property values can be cleared.

**InvolvedRepository Data Objects Summary**

Data Object	Description
1 involved 2 involvedCreate	1 Involved record property container provides properties returned when you read or search involved parties. When defining a repository search's

3 involvedUpdate	<p>return properties, refer to the property names defined in this object.</p> <p>2 Involved record property container used to set a new involved party's properties.</p> <p>3 Involved record object property container used to update or clear an involved party's properties.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">TInvolved</a> object.</p>
------------------	--

## 3.5.2.1.12 TaskRepository

**TaskRepository Requests Summary**

Request	Description
<a href="#">deleteTask</a>	Deletes a task record.
<a href="#">insertTask</a>	Creates a task record.
<a href="#">postTask</a>	Posts a task record.
<a href="#">readRecentlyViewedTasks</a>	Read recently viewed tasks.
<a href="#">readTask</a>	Reads and returns a task record. Specified task properties are returned with the record.
<a href="#">readTasksByCriteria</a>	Searches for task records that meet a given search criteria. Specified task properties are returned with the resulting records.
<a href="#">reassign</a>	Reassigns a task record to a different user.
<a href="#">updateTask</a>	Updates a task record. Specified task properties are updated.
<a href="#">voidTask</a>	Voids a task record.

**TaskRepository Data Objects Summary**

Data Object	Description
<a href="#">accountPostingStatus</a>	See <a href="#">accountPostingStatus</a> .

<p>1 taskAssignee 2 taskAssigneeCreate 3 taskAssigneeUpdate</p>	<p>1 Task assignee record property container provides properties returned when you read or search tasks. When defining a repository search's return properties, refer to the property names defined in this object.</p> <p>2 Task assignee record property container used to set properties for a Task assignee.</p> <p>3 Task assignee record object property container used to update or clear properties for a Task assignee.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">JTaskAssignee</a> object.</p>
<p><b>taskPriority</b></p>	<p>An enumeration used to indicate the priority of a task. Options include:</p> <ul style="list-style-type: none"> <li>• HIGHEST</li> <li>• HIGH</li> <li>• NORMAL</li> <li>• LOW</li> <li>• LOWEST</li> </ul>
<p>1 task 2 taskCreate 3 taskUpdate</p>	<p>1 Task record property container returned when you read or search tasks.</p> <p>2 Task record property container used to set a new task's properties. This object is passed as a request parameter when you create (insert) a task.</p> <p>3 Task record property container used to update or clear a task's properties. This object is passed as a request parameter when you update a task.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">TTask</a> object.</p>
<p><b>taskWorkStatus</b></p>	<p>An enumeration used to indicate the status of a task (like level of completion). Options include:</p> <ul style="list-style-type: none"> <li>• NOT_STARTED</li> <li>• STARTED</li> <li>• COMPLETED</li> </ul>

## 3.5.2.1.13 UserAccountRepository

**UserAccountRepository Requests Summary**

Request	Description
<a href="#">insertUserAccount</a>	Saves a new user account object to the repository.
<a href="#">updateUserAccount</a>	Updates a previously saved user account.
<a href="#">readUserAccount</a>	Retrieves a user account from the repository based on a unique key. For a list of available properties, see the type class, userAccount.
<a href="#">readUserAccountsByCriteria</a>	Retrieves a set of user accounts based on specified criteria. If the criteria is empty, then the search will be for all user accounts. The specified limit takes precedence over the current system setting for the maximum number of search results to obtain. For a list of available properties, see the type class, userAccount.
<a href="#">deleteUserAccount</a>	Deletes the user account specified by the unique key.
<a href="#">readRecentlyViewedUserAccounts</a>	Read recently viewed user accounts.

**UserAccountRepository Data Objects Summary**

Data Object	Description
1 userAccount 2 userAccountCreate 3 userAccountUpdate	<p>1 User account record property container returned when you read or search users.</p> <p>2 User account record property container used to set a new user's properties. This object is passed as a request parameter when you create (insert) a user.</p> <p>3 User account record property container used to update or clear a user's properties. This object is passed as a request parameter when you update a user.</p> <p><b>Note:</b> For more information about the data restrictions for properties that belong to the above-mentioned objects, see the <a href="#">Object Model Reference</a> for the <a href="#">YUser</a> object.</p> <p><b>Note:</b> To update a user's password through Web Services, you need to authenticate with that user name and password.</p>

<b>userType</b>	An enumeration used to indicate a user's access to public and private records. Options include: <ul style="list-style-type: none"> <li>• SUPER</li> <li>• NORMAL</li> <li>• LIMITED</li> </ul>
-----------------	--

### 3.5.2.2 Repository Method Details

This section provides more detailed information about the methods or requests supported for repositories.

#### 3.5.2.2.1 AccountRepository Method Details

This section describes the AccountRepository methods, including parameters and return values. For the method to insert a new record, a list of required properties is provided.

##### 3.5.2.2.1.1 insertAccount

Inserts or creates an account record.

### Parameters

#### insertAccount Parameter Descriptions

Parameter Name	Data Type	Description
<b>accountCreate</b>	<a href="#">accountCreate</a>	Data object

### Required Properties from accountCreate

The following table lists properties from the accountCreate data object that must be populated in order to create an account. Also note that the same properties must be populated when you are updating an account, using the accountUpdate data object.

#### accountCreate Required Properties

Property Name	Data Type	Description
<b>name</b>	string	The account name.
<b>allocationLimit</b>	decimal	The maximum amount that can be allocated to the account.
<b>startOn</b>	dateTime	The account start date.

<b>endOn</b>	dateTime	The account end date.
--------------	----------	-----------------------

## Response

### insertAccount Response Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	The unique key of the newly created account record.

#### 3.5.2.2.1.2 updateAccount

Updates an account record.

When you send an updateAccount request, certain account property values must be populated. For more information, see [the accountCreate Required Properties table](#).

## Parameters

### updateAccount Parameter Descriptions

Parameter Name	Data Type	Description
<b>account</b>	<a href="#">accountUpdate</a>	Data object

## Response

None

#### 3.5.2.2.1.3 readAccount

Gets requested properties for an existing account given its unique key.

## Parameters

### readAccount Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to read.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting account record.

## Response

### readAccount Response Descriptions

Parameter Name	Data Type	Description
account	<a href="#">account</a>	An account record. Only specified property values are returned with the record.

### See Also:

[account](#) for more information about the properties you can retrieve for the invoice

#### 3.5.2.2.1.4 readAccountsByCriteria

Retrieves a set of accounts based on specified search criteria. If the criteria is empty, then the search will retrieve all accounts. The specified limit takes precedence over the current system setting for the maximum number of search results. For a list of available properties, see the type class, [account](#).

## Parameters

### readAccountsByCriteria Parameter Descriptions

Parameter Name	Data Type	Description
criteria	searchCriteria	Search criteria or conditions to search for.
limit	int	A numeric value for the maximum number of records to return among search results.
properties	List<String>	A list of String values that identify the property names whose values to return with the resulting account records.

## Response

### readAccountsByCriteria Response Descriptions

Parameter Name	Data Type	Description
accounts	List< <a href="#">account</a> >	A list of account objects. Only specified property values are returned with the record.

### See Also:



[account](#) for more information about the properties you can retrieve for the account

#### 3.5.2.2.1.5 readChildAccounts

Retrieves child accounts and their requested properties for a given parent account's unique key.

### Parameters

#### readChildAccounts Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the parent account record whose child accounts to read.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting account records.

### Response

#### readChildAccounts Response Descriptions

Parameter Name	Data Type	Description
<b>account</b>	<a href="#">account</a>	An account record. Only specified property values are returned with the record.

### See Also:

[account](#) for more information about the properties you can retrieve for the invoice

#### 3.5.2.2.1.6 activateAccount

Activates an account record.

### Parameters

#### activateAccount Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the account record to activate.

### Response

None

#### 3.5.2.2.1.7 deactivateAccount

Deactivates an account record.

### Parameters

#### deactivateAccount Parameter Descriptions

Parameter Name	Data Type	Description
uniqueKey	string	Unique key of the account record to deactivate.

### Response

None

#### 3.5.2.2.1.8 allocateMoney

Deposits funds to an account.

### Parameters

#### allocateMoney Parameter Descriptions

Parameter Name	Data Type	Description
accountUniqueKey	string	Unique key of the record to read.
amount	decimal	Monetary amount to deposit to the account.
description	String	Description of the transaction.

### Response

None

#### 3.5.2.2.1.9 transferMoney

Transfers funds between accounts.

### Parameters

#### transferMoney Parameter Descriptions

Parameter Name	Data Type	Description
<b>fromAccountUniqueKey</b>	string	Unique key of the account record to transfer money from.
<b>toAccountUniqueKey</b>	string	Unique key of the account record to transfer money to.
<b>amount</b>	decimal	Monetary amount to transfer between the accounts.
<b>description</b>	String	Description of the transaction.

## Response

None

### 3.5.2.2.1.10 withdraw Money

Withdraws funds from a given account.

## Parameters

### withdrawMoney Parameter Descriptions

Parameter Name	Data Type	Description
<b>accountUniqueKey</b>	string	Unique key of the record to withdraw funds from.
<b>amount</b>	decimal	Monetary amount to withdraw.
<b>description</b>	String	Description of the transaction.

## Response

None

### 3.5.2.2.1.11 deleteAccount

Deletes the account specified by the unique key.

## Parameters

### deleteAccount Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to delete.

## Response

There is no return value.

### 3.5.2.2.1.12 readRecentlyViewedAccounts

Read and return recently viewed accounts.

## Parameters

### readRecentlyViewedAccounts Parameter Descriptions

Parameter Name	Data Type	Description
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting account records.

## Response

### readRecentlyViewedAccounts Response Descriptions

Parameter Name	Data Type	Description
<b>accounts</b>	List<account>	A list of account objects. Only specified property values are returned with the record.

### 3.5.2.2.2 AppointmentRepository Method Details

This section describes the AppointmentRepository methods, including parameters and return values. For the method to insert a new record, a list of required properties is provided.

#### 3.5.2.2.2.1 insertAppointment

Inserts or creates an appointment record.

## Parameters

### insertAppointment Parameter Descriptions

Parameter Name	Data Type	Description
appointment	<a href="#">appointmentCreate</a>	Data object

## Required Properties from appointmentCreate

The following table lists properties from the accountCreate data object that must be populated in order to create an account. Also note that the same properties must be populated when you are updating an account, using the accountUpdate data object.

### appointmentCreate Required Properties

Property Name	Data Type	Description
subject	string	The name or reason for the appointment. This field supports up to 250 characters.
startOn	dateTime	The appointment start date and time.
endOn	dateTime	The appointment start date and time.
attendeeCreates	List< <a href="#">attendeeCreate</a> >	Add one or more attendees to an appointment using this list of attendeeCreate data objects.
categories	List<category>	Enable one or more categories using this list of categories.  <b>Note:</b> The categories you identify must already be defined in TeamConnect through the Appointment Object Definition.

## Response

### insertAppointment Response Descriptions

Parameter Name	Data Type	Description
uniqueKey	string	The unique key of the newly created appointment record.

## See Also:

[appointmentCreate](#) for more information about the properties you can set for the appointment

## 3.5.2.2.2.2 updateAppointment

Updates an appointment record.

When you send an updateAppointment request, certain appointment property values must be populated. For more information, see [appointmentCreate Required Properties](#).

## Parameters

### updateAppointment Parameter Descriptions

Parameter Name	Data Type	Description
appointment	<a href="#">appointmentUpdate</a>	Data object

## Response

None

## See Also:

[appointmentUpdate](#) for more information about the properties you can set for the appointment

## 3.5.2.2.2.3 readAppointment

Gets requested properties for an existing appointment given its unique key.

## Parameters

### readAppointment Parameter Descriptions

Parameter Name	Data Type	Description
uniqueKey	string	Unique key of the record to read.
properties	List<String>	A list of String values that identify the property names whose values to return with the resulting appointment record.

## Response

### readAppointment Response Descriptions

Parameter Name	Data Type	Description
appointment	<a href="#">appointment</a>	An appointment record. Only specified property values are returned with the record.

**See Also:**

[appointment](#) for more information about the properties you can retrieve for the appointment

## 3.5.2.2.2.4 readAppointmentsByCriteria

Retrieves a set of appointments based on specified search criteria. If the criteria is empty, then the search will retrieve all appointments. The specified limit takes precedence over the current system setting for the maximum number of search results. For a list of available properties, see the type class, appointment.

**Parameters****readAppointmentsByCriteria Parameter Descriptions**

Parameter Name	Data Type	Description
<b>criteria</b>	searchCriteria	Search criteria or conditions to search for.
<b>limit</b>	int	A numeric value for the maximum number of records to return among search results.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting appointment records.

**Response****readAppointmentsByCriteria Response Descriptions**

Parameter Name	Data Type	Description
<b>appointments</b>	List< <a href="#">appointment</a> >	A list of appointment objects. Only specified property values are returned with the record.

**See Also:**

[appointment](#) for more information about the properties you can retrieve

## 3.5.2.2.2.5 deleteAppointment

Deletes the appointment specified by the unique key.

**Parameters****deleteAppointment Parameter Descriptions**

Parameter Name	Data Type	Description
uniqueKey	string	Unique key of the record to delete.

## Response

There is no return value.

### 3.5.2.2.2.6 readRecentlyViewedAppointments

Read and return recently viewed appointments.

## Parameters

### readRecentlyViewedAppointments Parameter Descriptions

Parameter Name	Data Type	Description
properties	List<String>	A list of String values that identify the property names whose values to return with the resulting appointment records.

## Response

### readRecentlyViewedAppointments Response Descriptions

Parameter Name	Data Type	Description
appointments	List<appointment >	A list of appointment objects. Only specified property values are returned with the record.

### 3.5.2.2.3 ContactRepository Method Details

This section describes the ContactRepository methods, including parameters and return values. For the method to insert a new record, a list of required properties is provided.

#### 3.5.2.2.3.1 insertContact

Inserts or creates a contact record.

## Parameters

### insertContact Parameter Descriptions



Parameter Name	Data Type	Description
contact	ContactCreate	Data object

## Required Properties from personCreate

The following table lists properties from the personCreate data object that must be populated in order to create a person. Also note that the same properties must be populated when you are updating a person, using the personUpdate data object.

### personCreate Required Properties

Property Name	Data Type	Description
firstName	string	The first name of the person.
lastName	string	The last name of the person.

## Required Properties from companyCreate

The following table lists properties from the companyCreate data object that must be populated in order to create a company. Also note that the same properties must be populated when you are updating a company, using the companyUpdate data object.

### companyCreate Required Properties

Property Name	Data Type	Description
name	string	The name of the organization.

## Response

### insertContact Response Descriptions

Parameter Name	Data Type	Description
uniqueKey	string	The unique key of the newly created contact record.

## See Also:

For more information about the properties you can set for the contact, see:

- [contactCreate](#)
- [CompanyCreate](#)
- [PersonCreate](#)

## 3.5.2.2.3.2 updateContact

Updates a contact record.

When you send an updateContact request, certain contact property values must already exist. For more information, see [personCreate Required Properties](#) and [companyCreate Required Properties](#).

**Note:** To update a value that does not exist for a contact, use a Create data object to insert the value. For example, if an existing contact does not have a phone number, use ContPhoneNumberCreate to insert a new phone number. After that, use ContPhoneNumberUpdate to update the phone number.

## Parameters

updateContact Parameter Descriptions

Parameter Name	Data Type	Description
contact	ContactUpdate	Data object

## Response

None

## See Also:

For more information about the properties you can set for the contact, see:

- [contactUpdate](#)
- [CompanyUpdate](#)
- [PersonUpdate](#)

## 3.5.2.2.3.3 readContact

Gets requested properties for an existing contact given its unique key.

## Parameters

readContact Parameter Descriptions

Parameter Name	Data Type	Description
uniqueKey	string	Unique key of the record to read.
properties	List<String>	A list of String values that identify the property names whose values to return with the resulting contact record.

## Response

### readContact Response Descriptions

Parameter Name	Data Type	Description
<b>contact</b>	Contact	A contact record. Only specified property values are returned with the record.

### See Also:

[contact](#) for more information about the properties you can retrieve for the contact

#### 3.5.2.2.3.4 readContactsByCriteria

Retrieves a set of contacts based on specified search criteria. If the criteria is empty, then the search will retrieve all contacts. The specified limit takes precedence over the current system setting for the maximum number of search results. For a list of available properties, see the type class, `contact`.

## Parameters

### readContactsByCriteria Parameter Descriptions

Parameter Name	Data Type	Description
<b>criteria</b>	searchCriteria	Search criteria or conditions to search for.
<b>limit</b>	int	A numeric value for the maximum number of records to return among search results.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting contact records.

## Response

### readContactsByCriteria Response Descriptions

Parameter Name	Data Type	Description
<b>contacts</b>	List<Contact>	A list of contact objects. Only specified property values are returned with the record.

### See Also:

[contact](#) for more information about the properties you can retrieve

#### 3.5.2.2.3.5 deleteContact

Deletes the contact specified by the unique key.

### Parameters

#### deleteContact Parameter Descriptions

Parameter Name	Data Type	Description
uniqueKey	string	Unique key of the record to delete.

### Response

There is no return value.

#### 3.5.2.2.3.6 ReadRecentlyView edContacts

Read and return recently viewed contacts.

### Parameters

#### readRecentlyViewedContacts Parameter Descriptions

Parameter Name	Data Type	Description
properties	List<String>	A list of String values that identify the property names whose values to return with the resulting contact records.

### Response

#### readRecentlyViewedContacts Response Descriptions

Parameter Name	Data Type	Description
contacts	List<contact>	A list of contact objects. Only specified property values are returned with the record.

#### 3.5.2.2.4 DocumentRepository Method Details

This section describes the DocumentRepository methods, including parameters and return values. For the method to insert a new record, a list of required properties is provided.

## 3.5.2.2.4.1 insertDocument

Inserts or creates a document record (file).

Depending on the type of document you want to create, use one of the following requests:

- For document files, use the insertDocument request with an input parameter (data object of type documentCreate) that defines the properties of the document to create.
- For shortcuts, use createShortcut with input parameters that define which existing document to create a shortcut to (documentUniqueKey) and the target folder location for the shortcut (parentFolderUniqueKey).
- For subfolders, use createSubFolder with input parameters that define the new folder's name (name) and the target folder location for the shortcut (parentUniqueKey).

Separate requests are provided for creating shortcuts and subfolders.

## Parameters

### insertDocument Parameter Descriptions

Parameter Name	Data Type	Description
documentCreate	<a href="#">documentCreate</a>	Data object

## Required Properties from documentCreate

The following table lists properties from the documentCreate data object that must be populated in order to upload a document file. Also note that the same properties must be populated when you are updating a document, using the documentUpdate data object.

### documentCreate Required Properties

Property Name	Data Type	Description
name	string	The name of the document. For example, the file name.
contentTypeUnique Key	string	The file format. The following values can be entered: <ul style="list-style-type: none"> <li>• Adobe Acrobat</li> <li>• Audio(aiff)</li> <li>• Audio(wav)</li> <li>• Binary or bin</li> <li>• Data Mapping File</li> <li>• Gif</li> <li>• HTM</li> </ul>

		<ul style="list-style-type: none"> <li>• HTML</li> <li>• Image(jpg)</li> <li>• Image(pcx)</li> <li>• Image(tiff)</li> <li>• JAR File</li> <li>• JSP</li> <li>• JavaClass</li> <li>• JavaScriptCode</li> <li>• Microsoft Access</li> <li>• Microsoft Excel</li> <li>• Microsoft Power Point</li> <li>• Microsoft Word</li> <li>• Quicktime Video</li> <li>• Rich Text Format</li> <li>• Standard Email</li> <li>• Text</li> <li>• Video(mpg)</li> <li>• XML</li> <li>• Zip</li> </ul>
<b>content</b>	base64Binary	The binary content of a file.
<b>type</b>	documentType	See <a href="#">documentType</a> for more information.
<b>parentFolderUnique Key</b>	string	The unique key of the document folder that this file will be located in.

## Response

### insertDocument Response Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	The unique key of the newly created document record.

## See Also:

[documentCreate](#) for more information about the properties you can set for a document

#### 3.5.2.2.4.2 createShortcut

Inserts or creates a shortcut to an existing TeamConnect document.

### Parameters

#### createShortcut Parameter Descriptions

Parameter Name	Data Type	Description
<b>documentUniqueKey</b>	string	Unique key of the document to create a shortcut for.
<b>parentFolderUniqueKey</b>	string	Unique key of the folder to create a shortcut with.

### Response

#### createShortcut Response Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	The unique key of the shortcut.

#### 3.5.2.2.4.3 createSubFolder

Inserts or creates a subfolder in a given parent folder.

### Parameters

#### createSubFolder Parameter Descriptions

Parameter Name	Data Type	Description
<b>name</b>	string	Name of the folder to create.
<b>parentUniqueKey</b>	string	Unique key of the folder to create a subfolder within.

### Response

#### createSubFolder Response Descriptions

Parameter Name	Data Type	Description
----------------	-----------	-------------

<b>uniqueKey</b>	string	The unique key of the subfolder.
------------------	--------	----------------------------------

#### 3.5.2.2.4.4 createHyperlinkDefaultCategory

Creates a hyperlink in a given parent folder. The hyperlink category is set to the default category for the Document object definition.

### Parameters

#### createHyperlinkDefaultCategory Parameter Descriptions

Parameter Name	Data Type	Description
<b>parentUniqueKey</b>	string	Unique key of the folder to create a subfolder within.
<b>url</b>	string	The website URL that the hyperlink will link to.
<b>name</b>	string	Name of the hyperlink. For example, in the application UI, the hyperlink name displays but the URL does not.
<b>contactUniqueKey</b>	string	Unique key of the contact record for the contact who creates the hyperlink.

### Response

#### createHyperlinkDefaultCategory Response Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	The unique key of the hyperlink.

#### 3.5.2.2.4.5 createHyperlink

Creates a hyperlink in a given parent folder. The hyperlink category is set to a specified Document category.

### Parameters

#### createHyperlink Parameter Descriptions

Parameter Name	Data Type	Description
----------------	-----------	-------------



<b>parentUniqueKey</b>	string	Unique key of the folder to create a subfolder within.
<b>url</b>	string	The website URL that the hyperlink will link to.
<b>name</b>	string	Name of the hyperlink. For example, in the application UI, the hyperlink name displays but the URL does not.
<b>documentCategoryUniqueKey</b>	string	Unique key of the existing Document category to set as the hyperlink category.
<b>contactUniqueKey</b>	string	Unique key of the contact record for the contact who creates the hyperlink.

## Response

### createHyperlink Response Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	The unique key of the hyperlink.

#### 3.5.2.2.4.6 updateDocument

Updates a document record (file, hyperlink, shortcut, or folder).

When you send an updateDocument request, certain document property values must be populated. For more information, see [documentCreate Required Properties](#).

**Note:** Although it is not customary in this guide to provide detailed documentation of the data objects, it should be noted that the `shortcutToDocumentUniqueKey` property of the `documentUpdate` data object is the unique key of the shortcut to the current document that you're updating.

## Parameters

### updateDocument Parameter Descriptions

Parameter Name	Data Type	Description
<b>document</b>	<a href="#">documentUpdate</a>	Data object

## Response

None

### 3.5.2.2.4.7 readDocument

Gets requested properties for an existing document given its unique key.

## Parameters

### readDocument Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to read.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting document record.

## Response

### readDocument Response Descriptions

Parameter Name	Data Type	Description
<b>document</b>	<a href="#">document</a>	A document record. Only specified property values are returned with the record.

### 3.5.2.2.4.8 readDocumentsByCriteria

Retrieves a set of documents based on specified search criteria. If the criteria is empty, then the search will retrieve all documents. The specified limit takes precedence over the current system setting for the maximum number of search results. For a list of available properties, see the type class, document.

## Parameters

### readDocumentsByCriteria Parameter Descriptions

Parameter Name	Data Type	Description
<b>criteria</b>	searchCriteria	Search criteria or conditions to search for.
<b>limit</b>	int	A numeric value for the maximum number of records to return among search results.

<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting invoice records.
-------------------	--------------	---

## Response

### readDocumentsByCriteria Response Descriptions

Parameter Name	Data Type	Description
<b>documents</b>	List< <a href="#">document</a> >	A list of document objects. Only specified property values are returned with the record.

#### 3.5.2.2.4.9 readDocumentByPath

Gets a document data object and the specified property values for a given path.

## Parameters

### readDocumentByPath Parameter Descriptions

Parameter Name	Data Type	Description
<b>path</b>	string	Path to the target document, including the document file name and extension. For example:  Top Level\Attachments\Disputes\DISP-000018 \IntakeRecord-3372.doc
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting document.

## Response

### readDocumentByPath Response Descriptions

Parameter Name	Data Type	Description
<b>document</b>	<a href="#">document</a>	A document record. Only specified property values are returned with the record.

## 3.5.2.2.4.10 readChildDocuments

This request can perform two functions based on the document unique key parameter.

- If you pass the unique key of a document file (parent), the response includes older document versions of that file.
- If you pass the unique key of a folder, the response includes documents within that folder.

## Parameters

### readChildDocuments Parameter Descriptions

Parameter Name	Data Type	Description
<b>parentUniqueKey</b>	string	Either the unique key of the document record whose children (previous document versions) to return or the unique key of a folder.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting document.

## Response

### readChildDocuments Response Descriptions

Parameter Name	Data Type	Description
<b>documents</b>	List< <a href="#">document</a> >	<ul style="list-style-type: none"> <li>• If the parentUniqueKey was a document file unique key, the list of documents includes previous versions of the given document.</li> <li>• If the parentUniqueKey was a document folder unique key, the list of documents includes the files, folders, shortcuts, and hyperlinks in the given folder.</li> </ul> <p>A document record. Only specified property values are returned with the record.</p>

## 3.5.2.2.4.11 readChildDocumentForName

Gets requested properties for an existing document given the unique key of its parent folder and the document's file name.

## Parameters

### readChildDocumentForName Parameter Descriptions

Parameter Name	Data Type	Description
----------------	-----------	-------------

<b>uniqueKey</b>	string	The unique key of the target document's parent folder.
<b>name</b>	string	The document file name.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting document.

## Response

### readChildDocumentForName Response Descriptions

Parameter Name	Data Type	Description
<b>documents</b>	<a href="#">document</a>	A document record. Only specified property values are returned with the record.

### 3.5.2.2.4.12 readDocumentFolderForDocumentOwner

Gets requested properties for an existing document folder given the unique key of its owner (user).

## Parameters

### readDocumentFolderForDocumentOwner Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	The unique key of the document folder owner (user).
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting document.

## Response

### readDocumentFolderForDocumentOwner Response Descriptions

Parameter Name	Data Type	Description
<b>documents</b>	<a href="#">document</a>	A document record. Only specified property values are returned with the record.

## 3.5.2.2.4.13 checkIn

Checks in a document that had been previously checked out.

When a new version of a document is checked in, its unique key does not change; however, the previous version of the document is saved under a new unique key.

## Parameters

### checkIn Parameter Descriptions

Parameter Name	Data Type	Description
<b>documentUniqueKey</b>	string	The unique key of the document file. This should be the same unique key passed when you checked out the document.
<b>content</b>	base64Binary	The document file content.
<b>versionText</b>	string	The version number to assign to this document.

## Response

None

## See Also:

[checkOut](#), [revert](#) for related methods

## 3.5.2.2.4.14 checkOut

Checks out a document. This function is typically used for version control of a file in TeamConnect and to lock a file while it is being edited.

## Parameters

### checkOut Parameter Descriptions

Parameter Name	Data Type	Description
<b>documentUniqueKey</b>	string	The unique key of the document file to check out.

## Response

None

## See Also:

[checkIn](#), [revert](#) for information about related methods

#### 3.5.2.2.4.15 undoCheckOut

Attempts to undo check out of a given document.

### Parameters

#### undoCheckOut Parameter Descriptions

Parameter Name	Data Type	Description
<b>documentUniqueKey</b>	string	The unique key of a document record that has been checked out.

### Response

None

### See Also:

[checkOut](#) for information about related methods

#### 3.5.2.2.4.16 revert

To revert a document to an older document version, you need to know the unique key of that older document version. You can accomplish this by getting a list of older document versions by calling the readChildDocuments method, then find the older version to revert back to from the list.

### Parameters

#### revert Parameter Descriptions

Parameter Name	Data Type	Description
<b>documentUniqueKey</b>	string	Document file unique key that identifies the version of the document (for example the current version) to replace.
<b>oldKey</b>	string	Document file unique key that identifies the version of the document to revert to (sets this file to the current version).

### Response

None

**See Also:**

[checkOut](#), [checkIn](#) for information about related records

## 3.5.2.2.4.17 moveDocument

Moves a document to a new parent folder.

**Parameters****moveDocument Parameter Descriptions**

Parameter Name	Data Type	Description
documentUniqueKey	string	The unique key of the document record to move.
newParentFolderUniqueKey	string	The unique key of the folder where the target document should be moved.

**Response**

None

## 3.5.2.2.4.18 copyDocument

Copies a document to a new parent folder.

**Parameters****copyDocument Parameter Descriptions**

Parameter Name	Data Type	Description
documentUniqueKey	string	The unique key of the document record to copied.
newParentFolderUniqueKey	string	The unique key of the folder where the document copy should be created.

**Response**

None



## 3.5.2.2.4.19 deleteDocument

Deletes the persistent document specified by the unique key.

### Parameters

#### deleteDocument Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to delete.

### Response

There is no return value.

## 3.5.2.2.4.20 ReadRecentlyView edDocuments

Read and return recently viewed documents.

### Parameters

#### readRecentlyViewedDocuments Parameter Descriptions

Parameter Name	Data Type	Description
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting document records.

### Response

#### readRecentlyViewedContacts Response Descriptions

Parameter Name	Data Type	Description
<b>documents</b>	List<document>	A list of document objects. Only specified property values are returned with the record.

## 3.5.2.2.4.21 setDocumentAsRecentlyView ed

Set document to recently viewed status.

### Parameters

#### setDocumentAsRecentlyViewed Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	String	Unique key of the document record set as recently viewed.

## Response

There is no return value.

### 3.5.2.2.4.22 readFolderHierarchy

Retrieves a hierarchal list of document folders for one or more projects.

## Parameters

### readFolderHierarchy Parameter Descriptions

Parameter Name	Data Type	Description
<b>projUniqueKey</b>	List<string>	A list of unique keys for each project.
<b>properties</b>	List<string>	A list of strings that identify the property names of the values to return for each folder.

## Response

### readFolderHierarchy Response Descriptions

Parameter Name	Data Type	Description
<b>readFolderHierarchy</b>	List<DocumentTreeNode>	A list of tree nodes. Each node includes the property values for the folders in the project and specifies the hierarchy of the folders. This parameter only returns specified property values.

### 3.5.2.2.5 ExpenseRepository Method Details

This section describes the ExpenseRepository methods, including parameters and return values. For the method to insert a new record, a list of required properties is provided.

## 3.5.2.2.5.1 insertExpense

Inserts or creates an expense record.

## Parameters

### insertExpense Parameter Descriptions

Parameter Name	Data Type	Description
expense	<a href="#">expenseCreate</a>	Data object

## Required Properties from expenseCreate

The following table lists properties from the expenseCreate data object that must be populated in order to upload an expense. Also note that the same properties must be populated when you are updating an expense, using the expenseUpdate data object.

### expenseCreate Required Properties

Property Name	Data Type	Description
shortDescription	string	The expense description.
expenseDate	date	The expense date. For example, when the expense was incurred or the date the record was made.
expensedByUniqueKey	string	The unique key of the user responsible for the expense.

## Response

### insertExpense Response Descriptions

Parameter Name	Data Type	Description
uniqueKey	string	The unique key of the newly created record.

## See Also:

[expenseCreate](#) for more information about the properties you can set for the expense

## 3.5.2.2.5.2 updateExpense

Update an expense record.

When you send an `updateExpense` request, certain expense property values must be populated. For more information about the required properties when creating or updating expenses, see [the `expenseCreate` Required Properties table](#).

## Parameters

### updateExpense Parameter Descriptions

Parameter Name	Data Type	Description
<code>expense</code>	<a href="#">expenseUpdate</a>	Data object

## Response

There is no return value.

## See Also:

[expenseUpdate](#) for more information about the properties you can change for expenses

### 3.5.2.2.5.3 readExpense

Gets requested properties for an existing expense given its unique key.

## Parameters

### readExpense Parameter Descriptions

Parameter Name	Data Type	Description
<code>uniqueKey</code>	string	Unique key of the record to read.
<code>properties</code>	List<String>	A list of String values that identify the property names whose values to return with the resulting expense record.

## Response

### readExpense Response Descriptions

Parameter Name	Data Type	Description
<code>expense</code>	<a href="#">expense</a>	An expense record. Only specified property values are returned with the record.

## See Also:

[expense](#) for more information about the properties you can retrieve for the invoice

#### 3.5.2.2.5.4 readExpensesByCriteria

Retrieves a list of expenses based on specified search criteria. If the criteria is empty, then the search will retrieve all expenses. The specified limit takes precedence over the current system setting for the maximum number of search results. For a list of available properties, see the type class, `expense`.

### Parameters

#### readExpensesByCriteria Parameter Descriptions

Parameter Name	Data Type	Description
<b>criteria</b>	searchCriteria	Search criteria or conditions to search for.
<b>limit</b>	int	A numeric value for the maximum number of records to return among search results.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting expense records.

### Response

#### readExpensesByCriteria Response Descriptions

Parameter Name	Data Type	Description
<b>expenses</b>	List< <a href="#">expense</a> >	A list of expense objects. Only specified property values are returned with the record.

### See Also:

[expense](#) for more information about the properties you can retrieve for the expense

#### 3.5.2.2.5.5 postExpense

Posts a given expense.

### Parameters

#### postExpense Parameter Descriptions

Parameter Name	Data Type	Description
----------------	-----------	-------------

<b>uniqueKey</b>	string	Unique key of the record to post.
------------------	--------	-----------------------------------

## Response

There is no return value.

### 3.5.2.2.5.6 voidExpense

voids a given expense.

## Parameters

### voidExpense Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to void.

## Response

There is no return value.

### 3.5.2.2.5.7 deleteExpense

Deletes the expense specified by the unique key.

## Parameters

### deleteExpense Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to delete.

## Response

There is no return value.

### 3.5.2.2.5.8 readRecentlyView edExpenses

Read and return recently viewed expenses.

## Parameters

### readRecentlyViewedDocuments Parameter Descriptions

Parameter Name	Data Type	Description
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting expense records.

## Response

### readRecentlyViewedContacts Response Descriptions

Parameter Name	Data Type	Description
<b>expenses</b>	List<expense>	A list of expense objects. Only specified property values are returned with the record.

### 3.5.2.2.6 GroupAccountRepository Method Details

This section describes the GroupAccountRepository methods, including parameters and return values. For the method to insert a new record, a list of required properties is provided.

#### 3.5.2.2.6.1 insertGroupAccount

Inserts or creates a group account record.

## Parameters

### insertGroupAccount Parameter Descriptions

Parameter Name	Data Type	Description
<b>groupAccount</b>	<a href="#">groupAccountCreate</a>	Data object

## Required Properties from groupAccountCreate

The following table lists properties from the groupAccountCreate data object that must be populated in order to upload a groupAccount. Also note that the same properties must be populated when you are updating a groupAccount, using the groupAccountUpdate data object.

### groupAccountCreate Required Properties

Property Name	Data Type	Description
<b>uniqueName</b>	string	A unique name for the group. Values can include letters and numbers. Special characters are not allowed.

<b>displayName</b>	string	The group name. Maximum length: 50 characters.
<b>userAccountUniqueKeys</b>	List<String>	A list of unique key values for the user accounts to add to the group.

## Response

### insertGroupAccount Response Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	The unique key of the newly created group account record.

## See Also:

[groupAccountCreate](#) for more information about the properties you can set for the group

### 3.5.2.2.6.2 updateGroupAccount

Updates a group account record.

When you send an updateGroupAccount request, certain contact property values must be populated. For more information, see [groupAccountCreate Required Properties](#).

## Parameters

### updateGroupAccount Parameter Descriptions

Parameter Name	Data Type	Description
<b>groupAccount</b>	<a href="#">groupAccountUpdate</a>	Data object

## Response

None

## See Also:

[groupAccountUpdate](#) for more information about the properties you can set for the user

### 3.5.2.2.6.3 readGroupAccount

Gets requested properties for an existing group account given its unique key.

## Parameters



**readGroupAccount Parameter Descriptions**

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to read.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting group account record.

**Response****readGroupAccount Response Descriptions**

Parameter Name	Data Type	Description
<b>groupAccount</b>	<a href="#">groupAccount</a>	A group account record. Only specified property values are returned with the record.

**See Also:**

[groupAccount](#) for more information about the properties you can retrieve for the group

## 3.5.2.2.6.4 readGroupAccountsByCriteria

Retrieves a set of group accounts based on specified search criteria. If the criteria is empty, then the search will retrieve all group accounts. The specified limit takes precedence over the current system setting for the maximum number of search results. For a list of available properties, see the type class, [groupAccount](#).

**Parameters****readGroupAccountsByCriteria Parameter Descriptions**

Parameter Name	Data Type	Description
<b>criteria</b>	searchCriteria	Search criteria or conditions to search for.
<b>limit</b>	int	A numeric value for the maximum number of records to return among search results.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting group account records.

## Response

### readGroupAccountsByCriteria Response Descriptions

Parameter Name	Data Type	Description
groupAccounts	List< <a href="#">groupAccount</a> >	A list of group account objects. Only specified property values are returned with the record.

## See Also:

[groupAccount](#) for more information about the properties you can retrieve

### 3.5.2.2.6.5 deleteGroupAccount

Deletes the group account specified by the unique key.

## Parameters

### deleteGroupAccount Parameter Descriptions

Parameter Name	Data Type	Description
uniqueKey	string	Unique key of the record to delete.

## Response

There is no return value.

### 3.5.2.2.6.6 readRecentlyViewedGroupAccounts

Read and return recently viewed accounts.

## Parameters

### readRecentlyViewedDocuments Parameter Descriptions

Parameter Name	Data Type	Description
properties	List<String>	A list of String values that identify the property names whose values to return with the resulting account records.

## Response

### readRecentlyViewedContacts Response Descriptions

Parameter Name	Data Type	Description
<b>groupAccounts</b>	List<accounts>	A list of account objects. Only specified property values are returned with the record.

### 3.5.2.2.7 HistoryRepository Method Details

This section describes the HistoryRepository methods, including parameters and return values. For the method to insert a new record, a list of required properties is provided.

#### 3.5.2.2.7.1 insertHistory

Inserts or creates a history record. Although a history is a unique and searchable record, you will typically associate a history with another record (for example, an account), using the history to track changes made to the account.

## Parameters

### insertHistory Parameter Descriptions

Parameter Name	Data Type	Description
<b>history</b>	<a href="#">historyCreate</a>	Data object

## Required Properties from historyCreate

The following table lists properties from the historyCreate data object that must be populated in order to create a history record. Also note that the same properties must be populated when you are updating a history record, using the historyUpdate data object.

### historyCreate Required Properties

Property Name	Data Type	Description
<b>archivedOn</b>	dateTime	The date the history record is created. For example, the current date.
<b>shortDescription</b>	string	Description of the history or note about the related record's changes. Maximum: 250 characters.
<b>parentObject</b>	historyOwner	The unique key of the record to associate the history record with.

## Response

### insertHistory Response Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	The unique key of the newly created history record.

### See Also:

[historyCreate](#) for more information about the properties you can set for the history

#### 3.5.2.2.7.2 updateHistory

Updates a history record.

When you send an updateHistory request, certain history property values must be populated. For more information, see [historyCreate Required Properties](#).

### Parameters

#### updateHistory Parameter Descriptions

Parameter Name	Data Type	Description
<b>history</b>	<a href="#">historyUpdate</a>	Data object

### Response

None

### See Also:

[historyUpdate](#) for more information about the properties you can set for the account

#### 3.5.2.2.7.3 readHistory

Gets requested properties for an existing history given its unique key.

### Parameters

#### readHistory Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to read.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting history record.

## Response

readHistory Response Descriptions

Parameter Name	Data Type	Description
history	<a href="#">history</a>	A history record. Only specified property values are returned with the record.

### See Also:

[history](#) for more information about the properties you can retrieve for the record

#### 3.5.2.2.7.4 readHistoriesByCriteria

Retrieves a set of history records based on specified search criteria. If the criteria is empty, then the search will retrieve all history records. The specified limit takes precedence over the current system setting for the maximum number of search results. For a list of available properties, see the type class, [history](#).

## Parameters

readHistoriesByCriteria Parameter Descriptions

Parameter Name	Data Type	Description
criteria	searchCriteria	Search criteria or conditions to search for.
limit	int	A numeric value for the maximum number of records to return among search results.
properties	List<String>	A list of String values that identify the property names whose values to return with the resulting history records.

## Response

readHistoriesByCriteria Response Descriptions

Parameter Name	Data Type	Description
histories	List< <a href="#">history</a> >	A list of history objects. Only specified property values are returned with the record.

### See Also:

[history](#) for more information about the properties you can retrieve for the record

## 3.5.2.2.7.5 deleteHistory

Deletes the history record specified by the unique key.

### Parameters

**deleteHistory Parameter Descriptions**

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to delete.

### Response

There is no return value.

## 3.5.2.2.7.6 readRecentlyView edHistories

Read and return recently view history records.

### Parameters

**readRecentlyViewedHistories Parameter Descriptions**

Parameter Name	Data Type	Description
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting history records.

### Response

**readRecentlyViewedHistories Response Descriptions**

Parameter Name	Data Type	Description
<b>histories</b>	List< <a href="#">history</a> >	A list of history objects. Only specified property values are returned with the record.

## 3.5.2.2.8 InvoiceRepository Method Details

This section describes the InvoiceRepository methods, including parameters and return values. For the method to insert a new record, a list of required properties is provided.

## 3.5.2.2.8.1 insertInvoice

Inserts or creates an invoice record.

To create an invoice, use the insertInvoice request with an input parameter (data object of type InvoiceCreate) that defines the properties of the invoice to create.

The response to creating an invoice is the unique key value of the new record. The unique key uniquely identifies a record and is used as a parameter in operations like updating and reading records.

Typically when approval rules (workflow) have been created for invoices, there may be scenarios when an approver rejects an invoice and you may resubmit another version of an existing invoice. When you create or update an invoice that has the same number, date, and vendor as an existing rejected invoice, the new or updated invoice will be saved with an auto-incremented version. For example, if the original rejected invoice's version was 1.0 then the resubmitted invoice's version will automatically be set to 1.1. For more information about rejected invoices, see the User Guide section on Invoices.

## Parameters

### insertInvoice Parameter Descriptions

Parameter Name	Data Type	Description
invoiceCreate	<a href="#">invoiceCreate</a>	Data object

## Required Properties from invoiceCreate

The following table lists properties from the invoiceCreate data object that must be populated in order to create an invoice. Also note that the same properties must be populated when you are updating an invoice, using the invoiceUpdate data object.

### invoiceCreate Required Properties

Property Name	Data Type	Description
numberString	string	The invoice number.
invoiceDate	date	The invoice date. For example, the date the invoice was submitted.
vendorUniqueKey	string	The account start date.

## Required Properties from lineItemCreate

The following table lists properties from the lineItemCreate data object that must be populated in order to create a line item. Although you can create or update an invoice with no line items, this is very impractical. Note that there are different required properties depending on the type of line item you are creating or editing.

Also note that the same properties must be populated when you are updating an invoice, using the `invoiceUpdate` data object with the `lineItemUpdate` data object.

#### lineItemCreate Required Properties

Property Name	Data Type	Description
<b>itemNumber</b>	int	Numeric ID that will set the display order of the line item in the application UI.
<b>serviceDate</b>	date	date for the expense or services provided.
<b>projectUniqueKey</b>	string	The unique key of the project associated with this line item.
<b>categories</b>	List<category >	<ul style="list-style-type: none"> <li>If the type property is expense, the category should be a line item expense category.</li> <li>If the type property is fee, the category should be a line item task category.</li> </ul>
<b>type</b>	lineItemType	The line item type (EXPENSE or FEE).
<b>timekeeperUniqueKey</b>	string	This property is only required if the type is Fee. Unique key of the timekeeper's contact record.
<b>activityUniqueKey</b>	string	This property is only required if the type is Fee. Tree position of the item from the Activity Item system lookup table to associate with the Fee/Task.
<b>originalRate</b>	decimal	<ul style="list-style-type: none"> <li>If the specified type is expense, the unit price for an expense.</li> <li>If the specified type is fee, the original rate for services rendered.</li> </ul>
<b>originalQuantity</b>	decimal	<ul style="list-style-type: none"> <li>If the specified type is expense, the number of units for an expense.</li> <li>If the specified type is fee, the number of hours of a service.</li> </ul>

## Response

#### insertInvoice Response Descriptions

Parameter Name	Data Type	Description
----------------	-----------	-------------



<b>uniqueKey</b>	string	The unique key of the newly created invoice record.
------------------	--------	---

**See Also:**

[invoiceCreate](#) for more information about the properties you can set for the invoice

[lineItemCreate](#) for more information about the properties you can set for line items

## 3.5.2.2.8.2 updateInvoice

Update an invoice record.

When you send an updateInvoice request, certain invoice property values must be populated. For more information about the required properties when creating or updating line items, see [the invoiceCreate Required Properties table](#). In addition, when updating an invoice it is most practical for the invoice to include at least one line item. For more information about the required properties when creating or updating line items, see [the lineItemCreate Required Properties table](#).

**Parameters**

updateInvoice Parameter Descriptions

Parameter Name	Data Type	Description
invoice	<a href="#">invoiceUpdate</a>	Data object

**Response**

There is no return value.

**See Also:**

[invoiceUpdate](#) for more information about the properties you can change for invoices

[lineItemUpdate](#) for more information about the properties you can set for line items

## 3.5.2.2.8.3 readInvoice

Gets requested properties for an existing invoice given its unique key.

**Parameters**

readInvoice Parameter Descriptions

Parameter Name	Data Type	Description
----------------	-----------	-------------

<b>uniqueKey</b>	string	Unique key of the record to read.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting invoice record.

## Response

### readInvoice Response Descriptions

Parameter Name	Data Type	Description
<b>invoice</b>	<a href="#">invoice</a>	An invoice record. Only specified property values are returned with the record.

## See Also:

[invoice](#) for more information about the properties you can retrieve for the invoice

[lineltem](#) for more information about the properties you can retrieve for line items

### 3.5.2.2.8.4 readInvoicesByCriteria

Retrieves a list of invoices based on specified search criteria. If the criteria is empty, then the search will retrieve all invoices. The specified limit takes precedence over the current system setting for the maximum number of search results. For a list of available properties, see the type class, invoice.

## Parameters

### readInvoicesByCriteria Parameter Descriptions

Parameter Name	Data Type	Description
<b>criteria</b>	searchCriteria	Search criteria or conditions to search for.
<b>limit</b>	int	A numeric value for the maximum number of records to return among search results.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting invoice records.

## Response

### readInvoicesByCriteria Response Descriptions

Parameter Name	Data Type	Description
invoices	List< <a href="#">invoice</a> >	A list of invoice objects. Only specified property values are returned with the record.

**See Also:**

[invoice](#) for more information about the properties you can retrieve for the invoice

[lineItem](#) for more information about the properties you can retrieve for line items

## 3.5.2.2.8.5 postInvoice

Posts a given invoice.

**Parameters**

## postInvoice Parameter Descriptions

Parameter Name	Data Type	Description
uniqueKey	string	Unique key of the record to post.

**Response**

There is no return value.

## 3.5.2.2.8.6 voidInvoice

voids a given invoice.

**Parameters**

## voidInvoice Parameter Descriptions

Parameter Name	Data Type	Description
uniqueKey	string	Unique key of the record to void.

**Response**

There is no return value.

## 3.5.2.2.8.7 deleteInvoice

Deletes the invoice specified by the unique key.

## Parameters

### deleteInvoice Parameter Descriptions

Parameter Name	Data Type	Description
uniqueKey	string	Unique key of the record to delete.

## Response

There is no return value.

## 3.5.2.2.8.8 readRecentlyViewedInvoices

## Parameters

### readRecentlyViewedInvoices Parameter Descriptions

Parameter Name	Data Type	Description
properties	List<String>	A list of String values that identify the property names whose values to return with the resulting invoice records.

## Response

### readRecentlyViewedInvoices Response Descriptions

Parameter Name	Data Type	Description
invoices	List<invoice>	A list of invoice objects. Only specified property values are returned with the record.

## 3.5.2.2.8.9 adjustInvoiceHeader

## Parameters

### adjustInvoiceHeader Parameter Descriptions

Parameter Name	Data Type	Description
----------------	-----------	-------------

<b>invoiceUniqueKey</b>	String	Unique key of the record to adjust.
<b>method</b>	adjustmentMethod	Method for adjustment (REDUCE_BY_AMOUNT, REDUCE_BY_PERCENT, or NEW_AMOUNT).
<b>target</b>	invoiceAdjustmentTarget	Invoice target to be adjusted (TOTAL_FEES, TOTAL_EXPENSES, or TOTAL_INVOICE).
<b>value</b>	decimal	The amount for the adjustment.
<b>reasonKey</b>	string	Tree position of the line item adjustment reasons table.
<b>commentsToVendor</b>	string	Comments to the vendor.
<b>internalComments</b>	string	Internal comments for the adjustment.

## Response

There is no return value.

### 3.5.2.2.8.10 readActiveApprovals

## Parameters

### readActiveApprovals Parameter Descriptions

Parameter Name	Data Type	Description
<b>invoiceUniqueKey</b>	String	Unique key of the record to read.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting approval records.

## Response

### readActiveApprovals Response Descriptions

Parameter Name	Data Type	Description
<b>approvals</b>	List<approval>	A list of approval objects. Only specified property values are returned with the record.

## 3.5.2.2.8.11 readCompletedApprovals

**Parameters****readCompletedApprovals Parameter Descriptions**

Parameter Name	Data Type	Description
invoiceUniqueKey	String	Unique key of the record to read.
properties	List<String>	A list of String values that identify the property names whose values to return with the resulting approval records.

**Response****readCompletedApprovals Response Descriptions**

Parameter Name	Data Type	Description
approvals	List<approval>	A list of approval objects. Only specified property values are returned with the record.

## 3.5.2.2.8.12 readInvoiceApprovalsPendingOnPost

Read all pending invoice approvals that are pending on post.

**Parameters****readInvoiceApprovalsPendingOnPost Parameter Descriptions**

Parameter Name	Data Type	Description
limit	int	A numeric value for the maximum number of records to return among search results.
properties	List<String>	A list of String values that identify the property names whose values to return with the resulting approval records.

**Response****readInvoiceApprovalsPendingOnPost Response Descriptions**

Parameter Name	Data Type	Description
----------------	-----------	-------------

<b>approvals</b>	List<approval >	A list of approval objects. Only specified property values are returned with the record.
------------------	--------------------	--

### 3.5.2.2.9 LookupTableSource Method Details

This section describes the LookupTableSource methods, including parameters and return values.

#### 3.5.2.2.9.1 readSystemLookupTable

Retrieves a list of properties for a system table code. You use the properties to read values from lookup tables.

## Parameters

### readSystemLookup Parameter Descriptions

Parameter Name	Data Type	Description
<b>SystemTableCode</b>	string	<p>Unique four-letter code for the system table types. Refer to the following list of codes for each type:</p> <ul style="list-style-type: none"> <li>• RESO—Appointment resource</li> <li>• ADDR—Contact address</li> <li>• MAIL—Contact email address</li> <li>• INET—Contact internet address</li> <li>• PHON—Contact phone number</li> <li>• FAXX—Contact fax number</li> <li>• CONR—Contact relation</li> <li>• SKIL—Contact skill</li> <li>• TERR—Contact territory</li> <li>• ACTI—Line item task activity</li> <li>• LTAR—Line item adjustment reason</li> <li>• PRJR—Project relation</li> <li>• COUN—Country</li> <li>• STAT—State</li> <li>• CURR—Currency</li> <li>• NUST—Invoice non-US tax</li> <li>• MLTT—Matter level tax</li> </ul>

## Response

### readSystemLookup Response Descriptions

Parameter Name	Data Type	Description
<b>WSLookupTable</b>	WSLookupTable	An item that includes a list of properties for a lookup table item. The list includes the following properties: <ul style="list-style-type: none"> <li>Item name</li> <li>Order item displays</li> <li>Stored value of the item</li> <li>Active status of the item</li> <li>Subtypes of item</li> </ul>

#### 3.5.2.2.10 ProjectRepository Method Details

This section describes the ProjectRepository methods, including parameters and return values. For the method to insert a new record, a list of required properties is provided.

##### 3.5.2.2.10.1 insertProject

Inserts or creates a project record.

## Parameters

### insertProject Parameter Descriptions

Parameter Name	Data Type	Description
<b>projectCreate</b>	<a href="#">projectCreate</a>	Data object

## Required Properties from projectCreate

The following table lists properties from the projectCreate data object that must be populated in order to create a project. Also note that the same properties must be populated when you are updating a project, using the projectUpdate data object.

### projectCreate Required Properties

Property Name	Data Type	Description
<b>entityTypeUniqueKey</b>	string	Unique 4-digit alphanumeric code that identifies the custom object. Corresponds to the TeamConnect Unique Code value.



<b>name</b>	string	The project record name.
<b>idNumber</b>	string	The project record ID number.
<b>categories</b>	List<Category>	The categories to add or enable for the project record.

## Response

### insertProject Response Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	The unique key of the newly created project record.

## See Also:

[projectCreate](#) for more information about the properties you can set for the project

[projectAssigneeCreate](#) for more information about adding assignees to the project and the corresponding assignee properties you can set

[projRelationCreate](#) for more information about adding relations to the project and the corresponding relation properties you can set

### 3.5.2.2.10.2 updateProject

Updates a project or matter record.

When you send an updateProject request, certain project property values must be populated. For more information, see [the projectCreate Required Properties table](#).

## Parameters

### updateProject Parameter Descriptions

Parameter Name	Data Type	Description
<b>projectUpdate</b>	<a href="#">projectUpdate</a>	Data object

## Response

None

## See Also:

[projectUpdate](#) for more information about the properties you can set for the invoice

[projectAssigneeUpdate](#) for more information about the properties you can update for assignees  
[projRelationUpdate](#) for more information about the properties you can update for relations

### 3.5.2.2.10.3 readProject

Gets requested properties for an existing project given its unique key.

## Parameters

### readProject Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to read.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting project record.

## Response

### readProject Response Descriptions

Parameter Name	Data Type	Description
<b>project</b>	<a href="#">project</a>	A project record. Only specified property values are returned with the record.

## See Also:

[project](#) for more information about the properties you can retrieve  
[projectAssignee](#) for more information about the properties you can retrieve for assignees  
[projRelation](#) for more information about the properties you can retrieve for relations  
[phaseInterval](#) for more information about the properties you can retrieve for phases

### 3.5.2.2.10.4 readProjectsByCriteria

Retrieves a set of projects based on specified search criteria. If the criteria is empty, then the search will retrieve all projects. The specified limit takes precedence over the current system setting for the maximum number of search results. For a list of available properties, see the type class, `project`.

## Parameters

### readProjectsByCriteria Parameter Descriptions

Parameter Name	Data Type	Description
<b>criteria</b>	searchCriteria	Search criteria or conditions to search for.
<b>limit</b>	int	A numeric value for the maximum number of records to return among search results.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting invoice records.

## Response

### readProjectsByCriteria Response Descriptions

Parameter Name	Data Type	Description
<b>projects</b>	List< <a href="#">project</a> >	A list of project objects. Only specified property values are returned with the record.

## See Also:

[project](#) for more information about the properties you can retrieve

[projectAssignee](#) for more information about the properties you can retrieve for assignees

[projRelation](#) for more information about the properties you can retrieve for relations

[phaseInterval](#) for more information about the properties you can retrieve for phases

### 3.5.2.2.10.5 readChildProjectsForEntityType

Retrieves requested properties of child project records for a given parent project record. You must also specify the project type of child projects to retrieve.

## Parameters

### readChildProjectsForEntityType Parameter Descriptions

Parameter Name	Data Type	Description
<b>parentProjectUniqueKey</b>	string	Unique key of the parent project record whose child project records to read.
<b>projectEntityTypeUniqueCode</b>	string	Unique code or tree position for the default category of the child project records to return.

		The unique code will be used like a search criteria and child projects that have the specified unique code set as default will be returned.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting projects.

## Response

### readChildProjectsForEntityType Response Descriptions

Parameter Name	Data Type	Description
<b>project</b>	<a href="#">project</a>	A project record. Only specified property values are returned with the record.

## See Also:

[project](#) for more information about the properties you can retrieve

[projectAssignee](#) for more information about the properties you can retrieve for assignees

[projRelation](#) for more information about the properties you can retrieve for relations

[phaseInterval](#) for more information about the properties you can retrieve for phases

### 3.5.2.2.10.6 changePhase

Changes a project records phase. You must get the valid phase's unique key from the TeamConnect application interface.

## Parameters

### changePhase Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the project whose phase to change.
<b>phaseUniqueKey</b>	string	Unique code of the phase to set for the project.  <b>Note:</b> You need to know beforehand <a href="#">the defined phase intervals</a> (for supported phase transitions) and phase unique codes.

## Response

None

### 3.5.2.2.10.7 deleteProject

Deletes the project specified by the unique key.

## Parameters

### deleteProject Parameter Descriptions

Parameter Name	Data Type	Description
uniqueKey	string	Unique key of the record to delete.

## Response

None

### 3.5.2.2.10.8 readRecentlyView edProjects

## Parameters

### readRecentlyViewedProjects Parameter Descriptions

Parameter Name	Data Type	Description
properties	List<String>	A list of String values that identify the property names whose values to return with the resulting project records.

## Response

### readRecentlyViewedProjects Response Descriptions

Parameter Name	Data Type	Description
projects	List< <a href="#">project</a> >	A list of project objects. Only specified property values are returned with the record.

### 3.5.2.2.10.9 readProjectEntityTypes

Get a list of project entity types in TeamConnect.

## Parameters

**readProjectEntityTypes Parameter Descriptions**

Parameter Name	Data Type	Description
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting project entity type records.

**Response****readProjectEntityTypes Response Descriptions**

Parameter Name	Data Type	Description
<b>projectEntityType</b> <b>s</b>	List<projectEntityTyp e>	A list of project entity types. Only specified property entity type values are returned with the record.

## 3.5.2.2.10.10 readProjectIntegrationSearches

Get a list of search views that are of type "Integration". Each returned search view should have a unique key, a name, and its associated entity type.

**Response****readProjectIntegrationSearches Response Descriptions**

Parameter Name	Data Type	Description
<b>projectSearches</b>	List<projectSearch>	A list of project searches that are of type "Integration".

## 3.5.2.2.10.11 readProjectsUsingSearch

**Parameters****readProjectsUsingSearch Parameter Descriptions**

Parameter Name	Data Type	Description
<b>projectEntityTypeUniqueKey</b>	string	Unique 4-digit alphanumeric code that identifies the custom object. Corresponds to the TeamConnect Unique Code value.
<b>searchUniqueKey</b>	string	Unique key of the project search of type "Integration".

<b>limit</b>	int	A numeric value for the maximum number of records to return among search results.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting project records.

## Response

### readProjectEntityTypes Response Descriptions

Parameter Name	Data Type	Description
<b>projects</b>	List<project>	A list of project objects. Only specified property values are returned with the record.

#### 3.5.2.2.11 InvolvedRepository Method Details

This section describes the InvolvedRepository methods, including parameters and return values. For the method to insert a new record, a list of required properties is provided.

##### 3.5.2.2.11.1 insertInvolved

Inserts or creates an involved party record. Note that when an involved record is created with the active status set to true, the activeDate will automatically be set to the current date.

## Parameters

### insertInvolved Parameter Descriptions

Parameter Name	Data Type	Description
<b>involved</b>	<a href="#">involvedCreate</a>	Data object

## Required Properties from involvedCreate

The following table lists properties from the involvedCreate data object that must be populated in order to create an involved party. Also note that the same properties must be populated when you are updating an involved party, using the involvedUpdate data object.

### involvedCreate Required Properties

Property Name	Data Type	Description
---------------	-----------	-------------

<b>entityTypeUniqueKey</b>	string	Unique 4-digit alphanumeric code that identifies the involved child object (child of associated project object definition). Corresponds to the TeamConnect Unique Code value.
<b>projectUniqueKey</b>	string	Project record unique key that the involved party should be associated with.
<b>contactUniqueKey</b>	string	Contact record unique key for the involved party.
<b>active</b>	boolean	True indicates the involved party is active.
<b>categories</b>	List<Category >	The categories to add or enable for the involved party record.

## Response

### insertInvolved Response Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	The unique key of the newly created involved party record.

## See Also:

[involvedUpdate](#) for more information about the properties you can update for the involved record

### 3.5.2.2.11.2 updateInvolved

Updates an involved record.

## Parameters

### updateInvolved Parameter Descriptions

Parameter Name	Data Type	Description
<b>involved</b>	<a href="#">involvedUpdate</a>	Data object

## Response

None



**See Also:**

[involvedUpdate](#) for more information about the properties you can update for the involved record

## 3.5.2.2.11.3 readInvolved

Gets requested properties for an existing involved record given its unique key.

**Parameters****readInvolved Parameter Descriptions**

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to read.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting project record.

**Response****readInvolved Response Descriptions**

Parameter Name	Data Type	Description
<b>involved</b>	<a href="#">involved</a>	An involved record. Only specified property values are returned with the record.

**See Also:**

[involved](#) for more information about the properties you can retrieve

## 3.5.2.2.11.4 readInvolvedsForProject

Gets requested properties for involved party records that are associated with a project record, given the unique key of the project record.

**Parameters****readInvolvedsForProject Parameter Descriptions**

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the project record whose associated involved party records to read.

<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting involved party records.
-------------------	--------------	--

## Response

### readInvolvedsForProject Response Descriptions

Parameter Name	Data Type	Description
<b>involved</b>	List< <a href="#">involved</a> >	A list of involved records. Only specified property values are returned with the records.

## See Also:

[involved](#) for more information about the properties you can retrieve

### 3.5.2.2.11.5 readInvolvedsByCriteria

Gets requested properties for involved party records given search criteria.

## Parameters

### readInvolvedsByCriteria Parameter Descriptions

Parameter Name	Data Type	Description
<b>involvedEntityTypeUniqueCode</b>	string	Unique code of the involved party object definition whose records to read. This should correspond to the involved party that is a child object of associated target project.
<b>criteria</b>	searchCriteria	Search criteria or conditions to search for.
<b>limit</b>	int	The maximum number of involved party records to return in the search results.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting involved party records.

## Response

### readInvolvedsByCriteria Response Descriptions

Parameter Name	Data Type	Description
<b>involved</b>	List< <a href="#">involved</a> >	A list of involved party records. Only specified property values are returned with the records.

**See Also:**

[involved](#) for more information about the properties you can retrieve

## 3.5.2.2.11.6 deleteInvolved

Deletes an involved part record given its unique key.

**Parameters**

deleteInvolved Parameter Description

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to read.

**Response**

None

## 3.5.2.2.12 TaskRepository Method Details

This section describes the TaskRepository methods, including parameters and return values. For the method to insert a new record, a list of required properties is provided.

## 3.5.2.2.12.1 insertTask

Inserts or creates a task record.

**Parameters**

insertTask Parameter Descriptions

Parameter Name	Data Type	Description
<b>task</b>	<a href="#">taskCreate</a>	Data object

**Required Properties from taskCreate**

The following table lists properties from the taskCreate data object that must be populated in order to create a task record. Also note that the same properties must be populated when you are updating a task record, using the taskUpdate data object.

#### taskCreate Required Properties

Property Name	Data Type	Description
shortDescription	string	A brief description of the task.
currentAssignee	<a href="#">taskAssigneeCreate</a>	The user assigned to the task.

## Response

#### insertTask Response Descriptions

Parameter Name	Data Type	Description
uniqueKey	string	The unique key of the newly created record.

## See Also:

[taskCreate](#) for more information about the properties you can set for the expense

### 3.5.2.2.12.2 updateTask

Update a task record.

When you send an updateTask request, certain task property values must be populated. For more information about the required properties when creating or updating tasks, see [Table 2.192 "taskCreate Required Properties"](#).

## Parameters

#### updateTask Parameter Descriptions

Parameter Name	Data Type	Description
task	<a href="#">taskUpdate</a>	Data object

## Response

There is no return value.

## See Also:

[taskUpdate](#) for more information about the properties you can change for tasks

## 3.5.2.2.12.3 readTask

Gets requested properties for an existing task given its unique key.

## Parameters

### readTask Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to read.
<b>properties</b>	List<String >	A list of String values that identify the property names whose values to return with the resulting task record.

## Response

### readTask Response Descriptions

Parameter Name	Data Type	Description
<b>task</b>	<a href="#">task</a>	A task record. Only specified property values are returned with the record.

## See Also:

[task](#) for more information about the properties you can retrieve for the task

## 3.5.2.2.12.4 readTasksByCriteria

Retrieves a list of tasks based on specified search criteria. If the criteria is empty, then the search will retrieve all tasks. The specified limit takes precedence over the current system setting for the maximum number of search results. For a list of available properties, see the type class, task.

## Parameters

### readTasksByCriteria Parameter Descriptions

Parameter Name	Data Type	Description
<b>criteria</b>	searchCriteria	Search criteria or conditions to search for.
<b>limit</b>	int	A numeric value for the maximum number of records to return among search results.

<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting expense records.
-------------------	--------------	---

## Response

### readTasksByCriteria Response Descriptions

Parameter Name	Data Type	Description
<b>tasks</b>	List< <a href="#">task</a> >	A list of task objects. Only specified property values are returned with the record.

## See Also:

[task](#) for more information about the properties you can retrieve for the task

### 3.5.2.2.12.5 postTask

Posts a given task.

## Parameters

### postTask Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to post.

## Response

There is no return value.

### 3.5.2.2.12.6 voidTask

Voids a given task.

## Parameters

### voidTask Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to void.

## Response

There is no return value.

### 3.5.2.2.12.7 reassign

Reassigns a given task to a different user.

## Parameters

### reassign Parameter Descriptions

Parameter Name	Data Type	Description
<b>taskUniqueKey</b>	string	Unique key of the task to reassign.
<b>userUniqueKey</b>	string	Unique key of the user to set as the new task assignee.

## Response

There is no return value.

### 3.5.2.2.12.8 deleteTask

Deletes the task specified by the unique key.

## Parameters

### deleteTask Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to delete.

## Response

There is no return value.

### 3.5.2.2.12.9 readRecentlyViewedTasks

Read recently viewed tasks.

## Parameters

### readRecentlyViewedTasks Parameter Descriptions

Parameter Name	Data Type	Description
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting task records.

## Response

### readRecentlyViewedTasks Response Descriptions

Parameter Name	Data Type	Description
<b>tasks</b>	List<task>	A list of task objects. Only specified property values are returned with the record.

### 3.5.2.2.13 UserAccountRepository Method Details

This section describes the UserAccountRepository methods, including parameters and return values. For the method to insert a new record, a list of required properties is provided.

#### 3.5.2.2.13.1 insertUserAccount

Inserts or creates a user account record.

## Parameters

### insertUserAccount Parameter Descriptions

Parameter Name	Data Type	Description
<b>userAccount</b>	<a href="#">userAccountCreate</a>	Data object

## Required Properties from userAccountCreate

The following table lists properties from the userAccountCreate data object that must be populated in order to create a user account record. Also note that the same properties must be populated when you are updating a user account record, using the userAccountUpdate data object.

### userAccountCreate Required Properties

Property Name	Data Type	Description
<b>username</b>	string	The user name (this will be used to log into the application UI). Maximum length: 50 characters.



<b>password</b>	string	The user's password. Maximum length: 250 characters
<b>contactUniqueKey</b>	string	The corresponding contact record's unique key. Each user must have an associated contact record.
<b>userType</b>	userType	<p>Although this property is not required to create a user account, it is highly recommended to populate it.</p> <ul style="list-style-type: none"> <li>• Super—Can access all public and private records. This setting overrides access restrictions based on record-level security settings. Recommended for system administrators and solution developers.</li> <li>• Normal—Can access all public records. Can access private records only if created by the user or if record-level security is granted to the user. Recommended for internal staff.</li> <li>• Limited-Privilege—Can access only records created by the user, records where the user is an assignee or attendee, and records with record-level security granted to the user. Recommended for Outside Counsel.</li> </ul> <p><b>Note:</b> Access to a record automatically grants access to its embedded records but not to its related records. For example, if a limited-privilege user has the right to access a dispute record, they cannot automatically access the dispute's related involved parties, tasks, histories, or expenses, but they can access its allegation (embedded object) records.</p>
<b>active</b>	boolean	<p>Although this property is not required to create a user account, it is highly recommended to populate it.</p> <p>Status of account activation (true indicates the account is active).</p>
<b>groupAccountAddUnique Keys</b>	List<String>	Although this property is not required to create a user account, it is highly recommended to populate it.

		List of unique keys for group records that the current user should be added to.
--	--	---

## Response

### insertUserAccount Response Descriptions

Parameter Name	Data Type	Description
uniqueKey	string	The unique key of the newly created user account record.

## See Also:

[userAccountCreate](#) for more information about the properties you can set for the user

[userType](#) for more information about the properties you can set for the user

### 3.5.2.2.13.2 updateUserAccount

Updates a user account record.

When you send an updateUserAccount request, certain contact property values must be populated. For more information, see [the userAccountCreate Required Properties table](#).

## Parameters

### updateUserAccount Parameter Descriptions

Parameter Name	Data Type	Description
userAccount	<a href="#">userAccountUpdate</a>	Data object

## Response

None

## See Also:

[userAccountUpdate](#) for more information about the properties you can set for the user

### 3.5.2.2.13.3 readUserAccount

Gets requested properties for an existing user account given its unique key.

## Parameters

**readUserAccount Parameter Descriptions**

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to read.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting user account record.

**Response****readUserAccount Response Descriptions**

Parameter Name	Data Type	Description
<b>userAccount</b>	<a href="#">userAccount</a>	A user account record. Only specified property values are returned with the record.

**See Also:**

[userAccount](#) for more information about the properties you can retrieve for the user

## 3.5.2.2.13.4 readUserAccountsByCriteria

Retrieves a set of user accounts based on specified search criteria. If the criteria is empty, then the search will retrieve all user accounts. The specified limit takes precedence over the current system setting for the maximum number of search results. For a list of available properties, see the type class, `userAccount`.

**Parameters****readUserAccountsByCriteria Parameter Descriptions**

Parameter Name	Data Type	Description
<b>criteria</b>	searchCriteria	Search criteria or conditions to search for.
<b>limit</b>	int	A numeric value for the maximum number of records to return among search results.
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting user account records.

## Response

### readUserAccountsByCriteria Response Descriptions

Parameter Name	Data Type	Description
<b>userAccounts</b>	List< <a href="#">userAccount</a> >	A list of user account objects. Only specified property values are returned with the record.

## See Also:

[userAccount](#) for more information about the properties you can retrieve

### 3.5.2.2.13.5 deleteUserAccount

Deletes the user account specified by the unique key.

## Parameters

### deleteUserAccount Parameter Descriptions

Parameter Name	Data Type	Description
<b>uniqueKey</b>	string	Unique key of the record to delete.

## Response

There is no return value.

### 3.5.2.2.13.6 readRecentlyViewedUserAccounts

Read recently viewed user accounts.

## Parameters

### readRecentlyViewedUserAccounts Parameter Descriptions

Parameter Name	Data Type	Description
<b>properties</b>	List<String>	A list of String values that identify the property names whose values to return with the resulting user account records.

## Response

### readRecentlyViewedUserAccounts Response Descriptions

Parameter Name	Data Type	Description
<b>userAccounts</b>	List<userAccount >	A list of user account objects. Only specified property values are returned with the record.

### 3.5.2.3 REST API HTTP Methods

You can use JSON formatted HTTP requests to create, read, update, or delete records from the TeamConnect server using the REST API methods. You must have the rights to view or make changes to these records.

Before you use the REST API HTTP methods, enter the following information:

- `http://<host>/<TeamConnectVersion>/webservice/rest` for the root of the API.
- `application/json` for the content-type and accept headers.
- HTTP basic authentication with TeamConnect credentials.

The HTTP methods are GET, POST, PUT, and DELETE. When you make a request using one of the HTTP methods, refer to the following tables.

#### HTTP Methods for Records

HTTP Method	Format	Description
<b>GET</b>	(nothing after GET)	Returns a list of record types.
	/<TypeOfRecord> <b>Example:</b> /tasks	Returns all the records of that type in the server.  To limit the number of results, enter a number in the <b>Number of records per result page</b> field on the <b>General</b> page of the <b>Preferences</b> in TeamConnect. To see additional pages of results, enter <code>?page=2</code> , or another number depending on the page, after /<TypeOfRecord>.  To search text or checkbox fields, enter the following: <code>?&lt;TreePosition&gt;.&lt;FieldName&gt;=&lt;value&gt;</code> after /<TypeOfRecord>, for example, ? <code>DISP_SUBP.DocketNoSU=1234</code>
	/<TypeOfRecord>/<ID> <b>Example:</b> /tasks/2	Returns information about the record.
<b>POST</b>	/<TypeOfRecord>	Creates a new record.  To determine the structure of the JSON object, refer to <a href="#">Determining the structure of a JSON object</a> .

	/<TypeOfRecord>/<ID>	Returns an error.
<b>PUT</b>	/<TypeOfRecord>	Returns an error.
	/<TypeOfRecord>/<ID>	Updates a record. To determine the structure of the JSON object, refer to <a href="#">Determining the structure of a JSON object</a> .
<b>DELETE</b>	/<TypeOfRecord>	Returns an error.
	/<TypeOfRecord>/<ID>	Deletes a record.

**HTTP Methods for Projects**

HTTP Method	Format	Description
<b>GET</b>	(nothing after GET)	Returns a list of projects and project codes.
	/projects/<ProjectCode> Example: /projects/DISP	Returns all the records of that project type in the server.
	/projects/<ProjectCode>/<ID> Example: /projects/DISP/3	Returns information about the project record.
<b>POST</b>	/projects/<ProjectCode>	Creates a new project record. To determine the structure of the JSON object, refer to <a href="#">Determining the structure of a JSON object</a> .
	/projects/<ProjectCode>/<ID>	Returns an error.
<b>PUT</b>	/projects/<ProjectCode>	Returns an error.
	/projects/<ProjectCode>/<ID>	Updates a project record. To determine the structure of the JSON object, refer to <a href="#">Determining the structure of a JSON object</a> .

<b>DELETE</b>	/projects/ <ProjectCode>	Returns an error.
	/projects/ <ProjectCode>/<ID>	Deletes a project record.

## Determining the structure of a JSON object

When you create or update a new record or project using the POST or PUT method, use the same format as a JSON object. Use the GET method with a record type and ID to return a formatted example.

The following structure is an example of a JSON object for /tasks:

```
{
  "shortDescription": "Created using the API",
  "currentAssignee": {
    "assignedOn": "Aug 15, 2012 8:15:00 AM", {
      "href": "/user/15"
    }
  }
}
```

The following structure is an example of a JSON object for /appointments:

```
{
  "subject": "Created using the API",
  "startOn": "Aug 20, 2012 4:31:00 PM",
  "endOn": "Aug 20, 2012 5:31:00 PM",
  "attendees": [ {
    "href": "/user/3"
  } ]
}
```

The following structure is an example of a JSON object for /DISP:

```
{
  "entityType": "DISP",
  "name": "Created using the API",
  "categories": {
    "DISP": {
      "categoryItem": "DISP",
      "customFields": {}
    }
  },
  "DISP_BANK": {
    "categoryItem": "DISP_BANK",
    "customFields": {}
  }
}
```

## Troubleshooting Error Messages

You receive an error message if you enter the following:

- An invalid username or password.
- Invalid parameter values for a record.
- An ID that does not exist.
- No ID when you enter a REST method that requires an ID (PUT and DELETE).
- An ID when the REST method does not require an ID (POST).

You might also receive an error message if you do not enter what the method requires. For example, if you are trying to update a task with the PUT method, but you do not enter new values, you receive an error message.

### 3.5.2.4 Abstract Classes

The following table summarizes abstract classes used in TeamConnect Web Services.

**Abstract Classes Summary**

Abstract Class Name	Description and Subclasses
<b>applicationEntityUpdate</b>	Provides a property, <code>uniqueKey</code> , with corresponding setter method. Use the <code>uniqueKey</code> setter method to identify which record to update.  Subclasses include: <ul style="list-style-type: none"> <li>• <a href="#">accountUpdate</a></li> <li>• <a href="#">appointmentUpdate</a></li> <li>• <a href="#">contactUpdate</a></li> <li>• <a href="#">documentUpdate</a></li> <li>• <a href="#">embeddedEntityUpdate</a></li> <li>• <a href="#">expenseUpdate</a></li> <li>• <a href="#">groupAccountUpdate</a></li> <li>• <a href="#">historyUpdate</a></li> <li>• <a href="#">invoiceUpdate</a></li> <li>• <a href="#">involvedUpdate</a></li> <li>• <a href="#">lineItemUpdate</a></li> <li>• <a href="#">projectUpdate</a></li> <li>• <a href="#">taskUpdate</a></li> <li>• <a href="#">userAccountUpdate</a></li> </ul>



	<p><b>Note:</b> The subclasses of <code>applicationEntityUpdate</code> specify additional properties with corresponding setter methods for updating record-specific properties.</p>
<p><b>lookupItem</b></p>	<p>Provides a group of properties and corresponding getter methods that you use to read values from TeamConnect system lookup tables, custom lookup tables, and the multi-currency lookup table.</p> <p>Subclasses include:</p> <ul style="list-style-type: none"> <li>• <a href="#">appointmentResourceType</a></li> <li>• <a href="#">contAddressType</a></li> <li>• <a href="#">countryItem</a></li> <li>• <a href="#">contEmailAddressType</a></li> <li>• <a href="#">contFaxNumberType</a></li> <li>• <a href="#">contInetAddressType</a></li> <li>• <a href="#">contPhoneNumberType</a></li> <li>• <a href="#">contRelationType</a></li> <li>• <a href="#">contSkillType</a></li> <li>• <a href="#">contTerritoryItem</a></li> <li>• <a href="#">currencyItem</a> (property of contact and invoice)</li> <li>• <a href="#">invoiceNonUSTaxType</a></li> <li>• <a href="#">lineItemAdjustmentReason</a></li> <li>• <a href="#">projRelationType</a></li> <li>• <a href="#">taskActivityItem</a></li> </ul> <p><b>Note:</b> The setter methods associated with <code>lookupItem</code> should not be used.</p> <p><b>Note:</b> The subclasses listed above have related <code>Create</code> and <code>Update</code> classes (for example, <code>contAddressCreate</code> and <code>contAddressUpdate</code>). Use the <code>setTypeUniqueKey</code> method, setting the value to the existing system lookup table item's <code>Tree Position</code>, to populate the corresponding type (for example <code>contAddressType</code>).</p>
<p><b>contact</b></p>	<p>Provides properties and corresponding getter methods for reading or searching contacts. When defining a contact repository search or</p>

	<p>read operation's return properties, refer to the property names defined in this object.</p> <p>Subclasses include:</p> <ul style="list-style-type: none"> <li>• <a href="#">Person</a></li> <li>• <a href="#">Company</a></li> </ul>
<b>contactCreate</b>	<p>Provides properties and corresponding setter methods to populate a contact's properties.</p> <p>Subclasses include:</p> <ul style="list-style-type: none"> <li>• <a href="#">PersonCreate</a></li> <li>• <a href="#">CompanyCreate</a></li> </ul>
<b>contactUpdate</b>	<p>Provides properties and corresponding setter methods to update or clear a contact's properties.</p> <p>Subclasses include:</p> <ul style="list-style-type: none"> <li>• <a href="#">PersonUpdate</a></li> <li>• <a href="#">CompanyUpdate</a></li> </ul>

### 3.5.2.5 Category and Custom Field Data Objects

The following table summarizes data objects used to add/remove categories to records or populate/update/clear custom field values in records using TeamConnect Web Services.

**Categories and Custom Field Data Objects**

<b>Data Object</b>	<b>Description</b>
<b>booleanCustomField</b>	Provides a method for identifying the existing boolean custom field (property) to update (populate). Also provides a method to set the value of a boolean custom field.
<b>dateTimeCustomField</b>	Provides a method for identifying the existing date/time custom field (property) to update (populate). Also provides a method to set the value of a date/time custom field.
<b>decimalCustomField</b>	Provides a method for identifying the existing decimal custom field (property) to update (populate). Also provides a method to set the value of a decimal custom field.
<b>involvedCustomField</b>	Provides a method for identifying the existing involved party custom field (property) to update (populate). Also provides a method to set the value of a involved party custom field.

<b>noteCustomField</b>	Provides a method for identifying the existing note custom field (property) to update (populate). Also provides a method to set the value of a note custom field.
<b>projectCustomField</b>	Provides a method for identifying the existing project custom field (property) to update (populate). Also provides a method to set the value of a project custom field.
<b>textCustomField</b>	Provides a method for identifying the existing text custom field (property) to update (populate). Also provides a method to set the value of a text custom field.

### 3.5.2.6 Data Objects Used in Search Criteria

There are a set of data objects that you use to define search criteria when you are searching records. This section provides a summary for data objects used in searches.

#### Data Objects Used in Search Criteria

Data Object	Description
<b>fieldSearchClause</b>	Subclass of searchCriteria. Provides methods to add field criteria to a search definition. Also provides a method to set the logic for combining multiple field criterion in a search.
<b>legacySearchFieldPathExpression</b>	Subclass of fieldPathExpression. Provides a method to set a searchKeyPath or define the TeamConnect field (Web Service property) whose value you will search by.
<b>booleanComparator</b>	Use this enumeration to set the method for comparing a boolean field value against existing values in the TeamConnect database.
<b>dateComparator</b>	Use this enumeration to set the method for comparing a date field value against existing values in the TeamConnect database.  Use this with both calendarDateFieldCriterion and dateFieldCriterion.
<b>decimalComparator</b>	Use this enumeration to set the method for comparing a decimal field value against existing values in the TeamConnect database.
<b>documentFolderComparator</b>	Use this enumeration to set the method for comparing a document folder field value against existing values in the TeamConnect database.

<b>enumerationComparator</b>	Use this enumeration to set the method for comparing an enumeration field value against existing values in the TeamConnect database.
<b>integerComparator</b>	Use this enumeration to set the method for comparing an integer field value against existing values in the TeamConnect database.
<b>objectComparator</b>	Use this enumeration to set the method for comparing an object field value against existing values in the TeamConnect database.
<b>relativeDateComparator</b>	Use this enumeration to set the method for comparing a date field value against existing values in the TeamConnect database when using the relativeDateFieldCriterion object.
<b>stringComparator</b>	Use this enumeration to set the method for comparing a string field value against existing values in the TeamConnect database.
<b>userComparator</b>	Use this enumeration to set the method for comparing a user field value against existing values in the TeamConnect database.
<b>booleanFieldCriterion</b>	Use this to define search criteria that filters search results according to a boolean type field value.
<b>calendarDateFieldCriterion</b>	Use this to define search criteria that filters search results according to a date (time zone dependent) field value.
<b>dateFieldCriterion</b>	Use this to define search criteria that filters search results according to a date (time zone independent) field value.
<b>decimalFieldCriterion</b>	Use this to define search criteria that filters search results according to a decimal field value.
<b>documentFolderFieldCriterion</b>	Use this to define search criteria that filters search results according to a document folder field value.
<b>enumerationFieldCriterion</b>	Use this to define search criteria that filters search results according to a field value of type, enumeration. For example, for searching documents by documentType (enumeration) property value.

<b>integerFieldCriterion</b>	Use this to define search criteria that filters search results according to a integer field value.
<b>objectFieldCriterion</b>	Use this to define search criteria that filters search results according to a field value of type, object.
<b>relativeDateFieldCriterion</b>	Use this to define search criteria that filters search results according to a relative date field value.
<b>stringFieldCriterion</b>	Use this to define search criteria that filters search results according to a string field value.
<b>userFieldCriterion</b>	Use this to define search criteria that filters search results according to a field value of type, user.

### 3.5.3 Frequently Asked Questions

This section provides tips for using Web Services.

#### 3.5.3.1 Batch Operations

When using Web Services, you need to make a unique repository request per record creation, record update, record read, records search, and record deletion operation.

Adding or updating certain record property values can be performed in a batch manner. For example:

- Contact addresses, email addresses, internet addresses
- Contact phone numbers, fax numbers
- Contact rates, relations, skills, territories
- Invoice adjustments
- Invoice line items, and line item adjustments
- Project assignees, relations
- Task assignees

#### 3.5.3.2 Operations Not Supported In Web Services

The following operations must be performed from the TeamConnect UI:

- Setting the default category for a record

When a record is created through Web Services, its default category is automatically set as the root category. To change a record's default category, use the TeamConnect UI.

- Creating, updating, reading, deleting object definitions
- Creating, updating, reading, deleting categories

- Creating or deleting custom fields
- Creating, updating, reading, deleting phases
- Creating, updating, reading, deleting phase transitions
- Creating, updating, reading, deleting rules

### 3.5.3.3 Cosmetic Data Objects

In the Web Service types.xsd file, you may see certain data objects that are used internally by Web Services but are not useful when you write a client program. As a result, you can ignore the following data objects (which may be auto-generated as data objects or classes when you run a SOAP toolkit against the Web Service source files):

- user
- userCreate
- userUpdate
- phaseIntervalCreate
- phaseIntervalUpdate
- involvedEntityType
- applicationEntityType
- projectEntityType
- applicationEntityType
- visibility
- directive
- fieldType
- systemTableCode

***Note:** If you want to work with the User record type, the following data objects are available: `userAccount`, `userAccountCreate`, and `userAccountUpdate`.*

## 3.5.4 Client Application Components (Java/Apache CXF)

This appendix contains sample code for the following components of a client application:

- [Client Proxy](#)
  - Web Service client interface classpath reference
  - Web Service URL
- [Security Headers](#) for system authentication
- [Client Application](#) that combines instantiation of the Client Proxy and insertion of the interceptor for Security Headers addition

**Note:** All samples were written in JAVA for use with the Apache CXF.

### 3.5.4.1 Client Proxy Sample Code

The following code sample illustrates concepts described in [Client Proxy](#) of the [Getting Started](#) section.

It is important to note that the `JaxWsProxyFactoryBean` or equivalent class should be provided by your SOAP toolkit to create a client proxy. In addition, note that you can use the `entityName` (object type name, such as `Contact`) to construct both the classpath to the corresponding Web Service client interface, and to construct the corresponding Web Service URL.

**Note:** In this sample, the *TeamConnect* version is presumed to be 3.2.

```
package com.mitratesoft.teamconnect;
import org.apache.commons.lang.StringUtils;
import org.apache.cxf.jaxws.JaxWsProxyFactoryBean;
/**
 * The purpose of this class is to create a Web Service client side proxy
 *
 * @author
 * @since 3.0
 */
public class WSClientFactory
{
    private final static String SERVER = "localhost";
    private final static String PORT = "8080";
    private final static String APP_NAME = "teamconnect-3.4SP1.0.0021";
    /**
     *
     * @param entityName
     * @return a service proxy for the specified entity
     */
    public Object createWSProxy(String entityName) {
        JaxWsProxyFactoryBean clientFactory = new JaxWsProxyFactoryBean();
        try {
            clientFactory.setServiceClass(getClassForEntity(entityName));
        } catch (ClassNotFoundException cnfe) {
            throw new IllegalArgumentException("No Web Services available for entity
            named: " + entityName);
        }
        clientFactory.setAddress(getWSAddress(entityName));
        return clientFactory.create();
    }
    private Class getClassForEntity(String entityName) throws
    ClassNotFoundException {
```

```

        return Class.forName("com.mitratesoft.teamconnect.webservice." +
            entityName.toLowerCase() + "repository." + entityName + "Repository");
    }

    private String getWSAddress(String entityName) {
        return "http://" + SERVER + ":" + PORT + "/" + APP_NAME + "/webservice/" +
            StringUtils.capitalize(entityName) + "Repository";
    }
}

```

### 3.5.4.2 Security Headers Sample Code

The following code sample illustrates concepts described in [SOAP Message Security Header](#) of the [Getting Started](#) section.

It is important to know that the TeamConnect user name and password need to be stored as String variables. Also note that in the resulting SOAP request message, the password will be transferred in cleartext format (unencrypted).

```

package com.mitratesoft.teamconnect;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import javax.security.auth.callback.Callback;
import javax.security.auth.callback.CallbackHandler;
import javax.security.auth.callback.UnsupportedCallbackException;
import org.apache.cxf.endpoint.Client;
import org.apache.cxf.endpoint.Endpoint;
import org.apache.cxf.frontend.ClientProxy;
import org.apache.cxf.ws.security.wss4j.WSS4JOutInterceptor;
import org.apache.ws.security.WSConstants;
import org.apache.ws.security.WSPasswordCallback;
import org.apache.ws.security.handler.WSHandlerConstants;
public class WSSecurityHeadersHelper
{
    private final static String USER = "system";
    private final static String PASSWORD = "s";
    public static void applySecurityHeaders(final Object sei) {
        Client cxfClient = ClientProxy.getClient(sei);
        Endpoint cxfEndpoint = cxfClient.getEndpoint();
        Map<String, Object> outProps = new HashMap<String, Object>();
        outProps.put(WSHandlerConstants.ACTION, WSHandlerConstants.USERNAME_TOKEN);
        outProps.put(WSHandlerConstants.USER, USER);
        outProps.put(WSHandlerConstants.PASSWORD_TYPE, WSConstants.PW_TEXT);
        outProps.put(WSHandlerConstants.PW_CALLBACK_CLASS,
            WSPasswordCallback.class.getName());
        WSS4JOutInterceptor wssOut = new WSS4JOutInterceptor(outProps);
        cxfEndpoint.getOutInterceptors().add(wssOut);
    }
}

```



```
public static class ClientPasswordCallback implements CallbackHandler
{
    public void handle(Callback[] callbacks) throws IOException,
        UnsupportedCallbackException {
        WSPasswordCallback pc = (WSPasswordCallback) callbacks[0];
        pc.setPassword(PASSWORD);
    }
}
```

### 3.5.4.3 Client Application Sample Code

The following code sample illustrates concepts described in [Putting Components Together in a Client Application](#) of the [Getting Started](#) section. The code is an excerpt from a client application that creates a client proxy for the contactRepository.

It is important to note that when you instantiate the client proxy using the createWSPProxy method, you would type cast the resulting object according to the entityName you had passed in. For example, if you passed in the entityName, Contact, then you'd type cast the returned object as ContactRepository. For the entityName, Document, then you'd type cast the returned object as DocumentRepository.

It is also important to note that you add the interceptor for the message security headers only one time to the client proxy. Afterward, when messages are generated for the client application requests, the corresponding security headers will be automatically inserted in each message. See the code like: WSSecurityHeadersHelper.applySecurityHeaders(contactRepository)

```
package com.mitratesoft.teamconnect;
import java.util.Date;
import java.util.List;
import com.mitratesoft.teamconnect.webservice.contactrepository.ContactRepository;
import com.mitratesoft.teamconnect.webservice.type.Category;
import com.mitratesoft.teamconnect.webservice.type.ContAddressCreate;
import com.mitratesoft.teamconnect.webservice.type.ContAddressUpdate;
import com.mitratesoft.teamconnect.webservice.type.Contact;
import com.mitratesoft.teamconnect.webservice.type.FieldSearchClause;
import com.mitratesoft.teamconnect.webservice.type.LegacySearchFieldPathExpression;
import com.mitratesoft.teamconnect.webservice.type.LogicOperator;
import com.mitratesoft.teamconnect.webservice.type.Person;
import com.mitratesoft.teamconnect.webservice.type.PersonCreate;
import com.mitratesoft.teamconnect.webservice.type.PersonUpdate;
import com.mitratesoft.teamconnect.webservice.type.StringComparator;
import com.mitratesoft.teamconnect.webservice.type.StringFieldCriterion;
import com.mitratesoft.teamconnect.webservice.type.TextCustomField;
public class WSContactClient extends AbstractWSClient
{
    private ContactRepository contactRepository;
    public WSContactClient() {
        contactRepository = (ContactRepository)createWSPProxy("Contact");
    }
}
```

```
        WSSecurityHeadersHelper.applySecurityHeaders(contactRepository);
        fieldNames.add("firstName");
        fieldNames.add("addresses.uniqueKey");
        fieldNames.add("addresses.type.name");
        fieldNames.add("addresses.city");
        fieldNames.add("addresses.state");
        fieldNames.add("addresses.countryCode.name");
        fieldNames.add("categories");
    }
    public static void main(String[] args) throws Exception {
        WSContactClient contactClient = new WSContactClient();
        String uniqueKey = contactClient.insertContact();
        Person person = contactClient.readContactByKey(uniqueKey);
        contactClient.updateContact(uniqueKey);
        person = contactClient.readContactByKey(uniqueKey);
        List<Contact> contacts =
            contactClient.readContactsByFirstName(person.getFirstName());
        contacts = contactClient.readContactsByPrimaryLanguage("English");
        contacts = contactClient.readContactsByPrimaryLanguage("French");
        contactClient.deletePerson(uniqueKey);
        person = contactClient.readContactByKey(uniqueKey);
        System.out.println("done");
    }
    private String insertContact() throws Exception {
        String uniqueKey = contactRepository.insertContact(createPerson());
        return uniqueKey;
    }
    private Person readContactByKey(String uniqueKey) throws Exception
    {
        Person person = (Person)contactRepository.readContact(uniqueKey,
            fieldNames);
        return person;
    }
    private PersonCreate createPerson() {
        String now = "" + new Date().getTime();
        PersonCreate person = new PersonCreate();
        person.setFirstName("first name " + now);
        person.getAddressCreates().add(createContAddress());
        ContAddressCreate anotherAddress = createContAddress();
        anotherAddress.setCity("San Jose");
        person.getAddressCreates().add(anotherAddress);
        Category cat = new Category();
        cat.setUniqueKey("CONT");
        TextCustomField primaryLanguage = new TextCustomField();
        primaryLanguage.setFieldName("primaryLanguage");
    }
}
```

```
        primaryLanguage.setValue("English");
        cat.getTextCustomFields().add(primaryLanguage);
        person.getCategories().add(cat);
        return person;
    }
    private ContAddressCreate createContAddress() {
        ContAddressCreate address = new ContAddressCreate();
        address.setTypeUniqueKey("ADDR_HOME");
        address.setCity("Los Angeles");
        address.setState("CA");
        address.setCountryCodeUniqueKey("COUN_0002");
        return address;
    }
}
```

### 3.5.5 Code Samples (Java/Apache CXF)

This chapter provides additional code samples (snippets) for using Web Services when writing a JAVA client program and using the Apache CXF SOAP toolkit. Samples for the following Web Services are provided:

- ContactRepository
- AccountRepository
- AppointmentRepository
- DocumentRepository
- ExpenseRepository
- GroupAccountRepository
- HistoryRepository
- InvoiceRepository
- ProjectRepository
- InvolvedRepository
- TaskRepository
- UserAccountRepository

Each TeamConnect Web Service's or repository's sample code is provided by repository name in alphabetic order for ease of navigation. However the object types can be grouped conceptually like the following:

- System Objects—AccountRepository, AppointmentRepository, ContactRepository, ExpenseRepository, InvoiceRepository, TaskRepository
  - DocumentRepository (this type of record can be stand-alone but is commonly associated with another record)

- HistoryRepository (this type of record is always associated with another record, for capturing information about changes to the record information)
- Custom Objects—ProjectRepository
  - InvolvedRepository (although a system object, this type of record is commonly associated with a project record)
- Account Administration Objects—GroupAccountRepository, UserAccountRepository

Basic overview descriptions for common functions like inserting, updating, reading, searching, and deleting records are in the chapter, [Getting Started](#) under [Common Functions](#).

**Note:** For JAVA code samples provided in this guide, JAX-WS and JAXB methods were used for JAVA XML Binding.

### 3.5.5.1 Contacts

The following .JAVA code samples show how you can use TeamConnect® Enterprise Web Services for functions specific to contact records.

In the code samples for creating person and company contacts, the required fields are indicated in comments.

#### 3.5.5.1.1 Creating Contacts

When creating a contact record, you need to further specify the type of contact you are creating. This is done through the contact argument. To specify a person type contact, pass in a contact created through the PersonCreate object. To specify a company type contact, pass in a contact created through the CompanyCreate object. When you instantiate the PersonCreate or CompanyCreate objects, they are used to store contact record properties. Both the PersonCreate and CompanyCreate objects extend ContactCreate, an abstract class.

#### **Code Snippet for creating a contact of type person and populating a property from a system lookup table**

```
protected ContactRepository contactRepository;
PersonCreate person = new PersonCreate();
//firstName is a required field
person.setFirstName("John");
//lastName is a required field
person.setLastName("Doe");
ContAddressCreate address = new ContAddressCreate();
address.setTypeUniqueKey("HOME");
//where the typeUniqueKey value, "HOME", is an existing item's Tree Position value
under the system lookup table, Address Type
address.setStreet("2468 Washington Ave.");
address.setCity("Los Angeles");
address.setState("CA");
address.setPostalCode("90026");
```

```
person.getAddressCreates().add(address);
String uniqueKey = contactRepository.insertContact(person);
//when you create a contact, the record's unique key is returned
//alternatively you could create a company contact like
contactRepository.insertContact(company)
//where company would be an instance of the CreateCompany object
```

### Code Snippet for creating a contact of type company

```
protected ContactRepository contactRepository;
CompanyCreate company = new CompanyCreate();
//name is a required field
company.setName("ACME Inc.");
//taxNumber is used to identify vendor contacts if you are using CSM/Collaborati
company.setTaxNumber("0112498531");
String uniqueKey = contactRepository.insertContact(company);
//when you create a contact, the record's unique key is returned
```

### Code Snippet for adding categories and custom fields to a person (contact)

```
protected ContactRepository contactRepository;
PersonCreate person = createPerson();
String personUniqueKey = contactRepository.insertContact(person);
private PersonCreate createPerson() {
    String firstName = "first" + new Date().getTime();
    PersonCreate person = new PersonCreate();
    person.setFirstName(firstName);
    person.setBirthdate(Calendar.getInstance().getTime());
    ContAddressCreate address = new ContAddressCreate();
    address.setTypeUniqueKey("ADDR_HOME");
    address.setCity("Portland");
    address.setState("OR");
    address.setCountryCodeUniqueKey("COUN_0002");
    person.getAddressCreates().add(address);
    Category cat = new Category();
    //the next two lines identify a category and then add the category to the
    contact record
    cat.setUniqueKey("CONT_EXTE");
    person.getCategories().add(cat);
    Category cat2 = new Category();
    person.getCategories().add(cat2);
    cat2.setUniqueKey("CONT");
    //the next four lines create a placeholder for an existing Boolean custom
    field, identify the known custom field name (synched), set the custom field
    value, and add the custom field placeholder to its parent category
    BooleanCustomField synched = new BooleanCustomField();
    synched.setFieldName("synched");
}
```

```

    synched.setValue(true);
    cat2.getBooleanCustomFields().add(synched);
    TextCustomField primaryLanguage = new TextCustomField();
    primaryLanguage.setFieldName("primaryLanguage");
    primaryLanguage.setValue("Englishe US");
    cat2.getTextCustomFields().add(primaryLanguage);
    return person;
}

```

### 3.5.5.1.2 Updating Contacts

When updating contacts, use the `PersonUpdate` or `CompanyUpdate` objects as parameters to the `ContactRepository` `updateContact` method.

Some data associated with a contact record is stored through separate objects. For example, a contact's rates are added through the `ContRateCreate` object. In addition, contact rates are updated through the `ContRateUpdate` object. The same concepts apply to a contact's mailing addresses (`ContAddressCreate`), email addresses (`ContEmailAddressCreate`), internet addresses (`ContInetAddressCreate`), phone numbers (`ContPhoneNumberCreate`), fax numbers (`ContFaxNumberCreate`), relations (`ContRelationCreate`), skills (`ContSkillCreate`), and territories (`ContTerritoryCreate`).

To delete certain data associated with a contact record (for example, rates, mailing addresses, email addresses, internet addresses, phone numbers, fax numbers, relations, skills, territories), use the following:

```
PersonUpdate.getRateDeleteUniqueKeys().add(newItem);
```

where `newItem` is the unique key of the rate to delete. Note that you need to repeat the line above for each rate to delete.

Also see the code snippets under [Creating Contacts](#) for comments about required contact fields.

#### Code Snippet for updating existing properties

```

protected ContactRepository contactRepository;
//for this example, it is assumed that you already know the target contact record's
unique key (where uniqueKey is a String variable)
PersonUpdate personUpdate = new PersonUpdate();
personUpdate.setUniqueKey(uniqueKey);
personUpdate.setLastName("Doe II");
contactRepository.updateContact(personUpdate);

```

#### Code Snippet for clearing existing property values

```

protected ContactRepository contactRepository;
//for this example, it is assumed that you already know the target contact record's
unique key (where uniqueKey is a String variable)
PersonUpdate personUpdate = new PersonUpdate();
personUpdate.setUniqueKey(uniqueKey);
personUpdate.getClearedProps().addAll(getPropertiesToClear());
private List<String> getPropertiesToClear() {

```

```
List<String> properties = new List<String>();
properties.add("firstName");
properties.add("lastName");
properties.add("address.type");
properties.add("address.state");
return properties;
}
```

### Code Snippet for adding contact rates

```
protected ContactRepository contactRepository;
//for this example, it is assumed that you already know the target contact record's
unique key (where uniqueKey is a String variable)
PersonUpdate personUpdate = new PersonUpdate();
personUpdate.setUniqueKey(uniqueKey);
ContrRateCreate rate = new ContrRateCreate();
Date today = new Date();
rate.setFirstEffectiveDate(today);
rate.setLastEffectiveDate(new Date(today.getTime() + 100000));
rate.setRate(new BigDecimal(20));
rate.setTaskCategoryCode("TASK_DEFA");
personUpdate.getRateCreates().add(rate);
contactRepository.updateContact(personUpdate);
```

### Code Snippet for updating a contact's custom field values

```
protected ContactRepository contactRepository;
//for this example, it is assumed that you already know the target contact record's
unique key (where uniqueKey is a String variable)
PersonUpdate personUpdate = new PersonUpdate();
personUpdate.setUniqueKey(uniqueKey);
Category cat = new Category();
personUpdate.getCategories().add(cat);
//The category should already exist. You should get its unique key (tree position)
from TeamConnect.
cat.setUniqueKey("CONT_EXTE");
TextCustomField primaryLanguage = new TextCustomField();
cat.getTextCustomFields().add(primaryLanguage);
primaryLanguage.setFieldName("primaryLanguage");
primaryLanguage.setValue("Chinese PRC");
contactRepository.updateContact(personUpdate);
```

### 3.5.5.1.3 Reading Contacts

When reading a contact record, identify the target record by unique key. Use the `ContactRepository` `readContact` method.

#### Code Snippet

```
//for this example, it is assumed that you already know the target contact record's
unique key (where uniqueKey is a String variable)
protected ContactRepository contactRepository;
public void readPersonContact() throws Exception {
    Person readPerson = (Person) contactRepository.readContact(uniqueKey,
        getPropertiesToRead());
}
private List<String> getPropertiesToRead() {
    List<String> properties = new List<String>();
    properties.add("firstName");
    properties.add("lastName");
    properties.add("address.type");
    properties.add("address.state");
    return properties;
}
```

### 3.5.5.1.4 Searching for Contacts

When searching for contacts, use the `ContactRepository` `readContactsByCriteria` method. Use the `Person`, `Company`, and `Contact` data objects to reference the field names you can use in the search criteria and include in the properties list for values to return with the search results (contact records). From the search results, the numeric limit value you set determines how many contact records will be returned. Similar to when you are reading a contact record, you must specify the names of property values to return with search results.

For more general information about how to define search criteria, see [Searching Records](#).

#### Code Snippet for readContactsByCriteria

```
//for this example, it is assumed that at least one contact record that meets the
search criterion already exists
protected ContactRepository contactRepository;
public void searchPersonContacts() throws Exception {
    StringFieldCriterion fieldCriterion = new StringFieldCriterion();
    fieldCriterion.setComparator(StringComparator.EQUALS_ENFORCE_CASE);
    LegacySearchFieldPathExpression fieldPathExpression = new
    LegacySearchFieldPathExpression();
    fieldPathExpression.setSearchKeyPath("firstName");
    fieldCriterion.setFieldPath(fieldPathExpression);
    fieldCriterion.getValue().add("John");
    FieldSearchClause searchCriteria = new FieldSearchClause();
    searchCriteria.setOperator(LogicOperator.AND);
    searchCriteria.getCriteria().add(fieldCriterion);
}
```



```
List<Contact> contacts =
    contactRepository.readContactsByCriteria(searchCriteria, 100,
        getPropertiesToRead());
}
private List<String> getPropertiesToRead() {
    List<String> properties = new List<String>();
    properties.add("firstName");
    properties.add("lastName");
    properties.add("addresses.type");
    properties.add("addresses.state");
    return properties;
}
```

### Code Snippet for readContactsByCriteria (specify return properties of contact property of type object)

```
//for this example, it is assumed that at least one contact record that meets the
search criterion already exists
//the last three properties defined in the properties list illustrate how you
specify contact search return property values where the contact property is an
object
protected ContactRepository contactRepository;
public void searchPersonContacts() throws Exception {
    StringFieldCriterion fieldCriterion = new StringFieldCriterion();
    fieldCriterion.setComparator(StringComparator.EQUALS_ENFORCE_CASE);
    LegacySearchFieldPathExpression fieldPathExpression = new
    LegacySearchFieldPathExpression();
    fieldPathExpression.setSearchKeyPath("firstName");
    fieldCriterion.setFieldPath(fieldPathExpression);
    fieldCriterion.getValue().add("John");
    FieldSearchClause searchCriteria = new FieldSearchClause();
    searchCriteria.setOperator(LogicOperator.AND);
    searchCriteria.getCriteria().add(fieldCriterion);
    List<Contact> contacts =
    contactRepository.readContactsByCriteria(searchCriteria, 100,
        getPropertiesToRead());
}
private List<String> getPropertiesToRead() {
    List<String> properties = new List<String>();
    properties.add("firstName");
    properties.add("lastName");
    properties.add("primaryPhoneNumber.number");
    //where number is a property of the contact property, primaryPhoneNumber (type
    object)
    properties.add("primaryEmailAddress.address");
    //where address is a property of the contact property, primaryEmailAddress
    (type object)
    properties.add("defaultRates.rate");
}
```

```

    //where rate is a property of the contact property, defaultRates (type object)
    return properties;
}

```

### Code Snippet for readContactsByCriteria (search by custom field value)

```

//for this example, it is assumed that at least one contact record that meets the
search criterion already exists
//also assume that a category, Employee (with unique code, EMPL) has been defined
for the Contact object definition
//also assume that a custom string field, DLNumb (driver's license number) has been
define for the Contact category, Employee
protected ContactRepository contactRepository;
public void searchPersonContacts() throws Exception {
    StringFieldCriterion fieldCriterion = new StringFieldCriterion();
    fieldCriterion.setComparator(StringComparator.EQUALS_ENFORCE_CASE);
    LegacySearchFieldPathExpression fieldPathExpression = new
    LegacySearchFieldPathExpression();
    fieldPathExpression.setSearchKeyPath("CONT_EMPL__DLNumb");
    //note there are two underscores between "EMPL" and "DLNumb" in the
    searchKeyPath above
    fieldCriterion.setFieldPath(fieldPathExpression);
    fieldCriterion.getValue().add("B301792Z");
    FieldSearchClause searchCriteria = new FieldSearchClause();
    searchCriteria.setOperator(LogicOperator.AND);
    searchCriteria.getCriteria().add(fieldCriterion);
    List<Contact> contacts =
    contactRepository.readContactsByCriteria(searchCriteria, 100,
    getPropertiesToRead());
}
private List<String> getPropertiesToRead() {
    List<String> properties = new List<String>();
    properties.add("firstName");
    properties.add("lastName");
    properties.add("addresses.type");
    properties.add("addresses.state");
    return properties;
}

```

#### 3.5.5.1.5 Deleting Contacts

When deleting a contact record, use the ContactRepository deleteContact method. Identify the target record to delete by its unique key.

### Code Snippet

```

//for this example, it is assumed that you already know the target contact record's
unique key (where uniqueKey is a String variable)

```

```
protected ContactRepository contactRepository;
contactRepository.deleteContact(uniqueKey);
```

### 3.5.5.2 Accounts

The following .JAVA code samples show how you can use TeamConnect® Enterprise Web Services for functions specific to account records.

#### 3.5.5.2.1 Creating accounts

The following sample creates an account record. In the sample, required fields are commented. During account creation, it is optional to set posting criteria or definitions on which types of financial records can post to the account (for example, tasks, expenses, invoices).

#### Code Snippet

```
Date date = new Date();
protected AccountRepository accountRepository;
AccountCreate account = createAccount();
String accountUniqueKey = accountRepository.insertAccount(account);
private String createAccount() throws Exception {
    AccountCreate account = new AccountCreate();
    //name is a required field
    account.setName("2009-Q1");
    //startOn date is a required field
    account.setStartOn(01-JAN-09);
    //endOn date is a required field
    account.setEndOn(31-MAR-09);
    //allocationLimit is a required field
    account.setAllocationLimit(25000.00);
    //(optional) set the account type as BUDGET or RESERVE
    account.setType(AccountType.BUDGET);
    //(optional) set the account overdraft type
    account.setAccountOverdraftType(AccountOverdraftType.ALLOW_NEGATIVE);
    //(optional) if this is a child account, set the parent account
    account.setParentAccountUniqueKey("ACCT_4141");
    returns account;
}
```

#### Code Snippet for setting account posting criteria

```
//the following sample code would be used as an excerpt from the above sample for
creating an account; it is presented separately just to simplify the process for
setting account posting criteria
//the following section sets posting criteria that allows tasks of a certain
category to post to this account
account.setAllowTask(true);
account.setTaskCategoryUniqueKey(taskCategoryUniqueKey);
```

```
account.setTaskPercent(100);
//the following section sets posting criteria that allows expenses of a certain
category to post to this account
account.setAllowExpense(true);
account.setExpenseCategoryUniqueKey(expenseCategoryUniqueKey);
account.setExpensePercent(100);
//the following section sets posting criteria that allows invoice expense line
items of a certain category to post to this account
account.setAllowInvoiceExpense(true);
account.setInvoiceExpenseCategoryUniqueKey(invoiceExpCategoryUniqueKey);
account.setInvoiceExpensePercent(100);
//the following section sets posting criteria that allows invoice fee line items of
a certain category to post to this account
account.setAllowInvoiceTask(true);
account.setInvoiceTaskCategoryUniqueKey(invoiceTaskCategoryUniqueKey);
account.setInvoiceTaskPercent(100);
```

#### 3.5.5.2.2 Updating accounts

The following sample updates an existing account record.

##### **Code Snippet**

```
Date date = new Date();
protected AccountRepository accountRepository;
//for this example, it is assumed that you already know the target account record's
unique key (where uniqueKey is a String variable)
AccountUpdate account = new AccountUpdate();
account.setUniqueKey(uniqueKey);
account.setName("Budget 2009 Q2");
account.setAllocated(15000.00);
account.setAllocationLimit(20000.00);
account.setUsed(5000.00);
account.setAvailable(10000.00);
account.setStartOn(date);
account.setAccountOverdraftType(AccountOverdraftType.ALLOW_NEGATIVE);
account.setType(AccountType.BUDGET);
account.setActive(true);
//setting the autoPost property to true indicates that financial transactions can
post to this account;
account.setAutoPost(true);
account.setParentAccountUniqueKey("ACCT_1519");
accountRepository.updateAccount(account);
```

### 3.5.5.2.3 Reading accounts

The following sample reads an existing account record.

#### Code Snippet

```
//for this example, it is assumed that you already know the target account record's
unique key (where uniqueKey is a String variable)
protected AccountRepository accountRepository;
public void readAccount() throws Exception {
    Account account = (Account) accountRepository.readAccount(uniqueKey,
        getPropertiesToRead());
}
private List<String> getPropertiesToRead() {
    List<String> properties = new List<String>();
    properties.add("createdBy");
    properties.add("createdOn");
    properties.add("modifiedBy");
    properties.add("modifiedOn");
    properties.add("name");
    properties.add("allocated");
    properties.add("allocationLimit");
    properties.add("used");
    properties.add("available");
    properties.add("startOn");
    properties.add("endOn");
    properties.add("expensePercent");
    properties.add("invoiceExpensePercent");
    properties.add("invoiceTaskPercent");
    properties.add("taskPercent");
    properties.add("allowExpense");
    properties.add("allowTask");
    properties.add("allowInvoiceExpense");
    properties.add("allowInvoiceTask");
    properties.add("accountInvolvedType");
    properties.add("accountVendorType");
    properties.add("accountProjectType");
    properties.add("accountOverdraftType");
    properties.add("type");
    properties.add("active");
    properties.add("autoPost");
    properties.add("categories");
    properties.add("expenseCategory");
    properties.add("taskCategory");
    properties.add("projectCategory");
    properties.add("invoiceExpenseCategory");
```

```
        properties.add("invoiceTaskCategory");
        properties.add("involved");
        properties.add("project");
        properties.add("vendor");
        properties.add("parentAccount");
        return properties;
    }
}
```

#### 3.5.5.2.4 Reading child accounts

The following sample reads child accounts of a parent account record. Identify the parent account by its unique key. In the sample, a complete list of account properties is provided but in your program, you can remove unnecessary properties to improve performance. Only the listed property values will be returned with the child accounts in the search results.

#### Code Snippet

```
//for this example, it is assumed that you already know the target parent account
record's unique key (where uniqueKey is a String variable)
protected AccountRepository accountRepository;
public void readChildAccounts() throws Exception {
    List<Account> accounts = (List<Account>)
        accountRepository.readChildAccounts(uniqueKey, getPropertiesToRead());
}
private List<String> getPropertiesToRead() {
    List<String> properties = new List<String>();
    properties.add("createdBy");
    properties.add("createdOn");
    properties.add("modifiedBy");
    properties.add("modifiedOn");
    properties.add("name");
    properties.add("allocated");
    properties.add("allocationLimit");
    properties.add("used");
    properties.add("available");
    properties.add("startOn");
    properties.add("endOn");
    properties.add("expensePercent");
    properties.add("invoiceExpensePercent");
    properties.add("invoiceTaskPercent");
    properties.add("taskPercent");
    properties.add("allowExpense");
    properties.add("allowTask");
    properties.add("allowInvoiceExpense");
    properties.add("allowInvoiceTask");
    properties.add("accountInvolvedType");
    properties.add("accountVendorType");
}
```

```
properties.add("accountProjectType");
properties.add("accountOverdraftType");
properties.add("type");
properties.add("active");
properties.add("autoPost");
properties.add("categories");
properties.add("expenseCategory");
properties.add("taskCategory");
properties.add("projectCategory");
properties.add("invoiceExpenseCategory");
properties.add("invoiceTaskCategory");
properties.add("involved");
properties.add("project");
properties.add("vendor");
properties.add("parentAccount");
return properties;
}
```

#### 3.5.5.2.5 Searching accounts

The following sample searches for account records where the name contains "2009".

#### Code Snippet

```
//for this example, it is assumed that at least one account record that meets the
search criterion already exists
protected AccountRepository accountRepository;
public void searchAccounts() throws Exception {
    StringFieldCriterion fieldCriterion = new StringFieldCriterion();
    fieldCriterion.setComparator(StringComparator.CONTAINS);
    LegacySearchFieldPathExpression fieldPathExpression = new
    LegacySearchFieldPathExpression();
    fieldPathExpression.setSearchKeyPath("name");
    fieldCriterion.setFieldPath(fieldPathExpression);
    fieldCriterion.getValue().add("2009");
    FieldSearchClause searchCriteria = new FieldSearchClause();
    searchCriteria.setOperator(LogicOperator.AND);
    searchCriteria.getCriteria().add(fieldCriterion);
    List<Account> accounts =
    accountRepository.readAccountsByCriteria(searchCriteria, 100,
    getPropertiesToRead());
}
private List<String> getPropertiesToRead() {
    List<String> properties = new List<String>();
    properties.add("createdBy");
    properties.add("createdOn");
    properties.add("modifiedBy");
}
```

```
properties.add("modifiedOn");
properties.add("name");
properties.add("allocated");
properties.add("allocationLimit");
properties.add("used");
properties.add("available");
properties.add("startOn");
properties.add("endOn");
properties.add("expensePercent");
properties.add("invoiceExpensePercent");
properties.add("invoiceTaskPercent");
properties.add("taskPercent");
properties.add("allowExpense");
properties.add("allowTask");
properties.add("allowInvoiceExpense");
properties.add("allowInvoiceTask");
properties.add("accountInvolvedType");
properties.add("accountVendorType");
properties.add("accountProjectType");
properties.add("accountOverdraftType");
properties.add("type");
properties.add("active");
properties.add("autoPost");
properties.add("categories");
properties.add("expenseCategory");
properties.add("taskCategory");
properties.add("projectCategory");
properties.add("invoiceExpenseCategory");
properties.add("invoiceTaskCategory");
properties.add("involved");
properties.add("project");
properties.add("vendor");
properties.add("parentAccount");
return properties;
}
```

#### 3.5.5.2.6 Activating accounts

The following sample activates an existing account record.

#### Code Snippet

```
// this sample assumes an account already exists with the unique key (String) value
equal to uniqueKey
protected AccountRepository accountRepository;
accountRepository.activateAccount(uniqueKey);
```



### 3.5.5.2.7 Deactivating accounts

The following sample deactivates an existing active account record.

#### Code Snippet

```
// this sample assumes an active account already exists with the unique key
(String) value equal to uniqueKey
protected AccountRepository accountRepository;
accountRepository.deactivateAccount(uniqueKey);
```

### 3.5.5.2.8 Allocating money to accounts

The following sample allocates money to an existing active account record.

#### Code Snippet

```
// this sample assumes an active account already exists with the unique key
(String) value equal to uniqueKey
protected AccountRepository accountRepository;
//the allocated variable defines the monetary amount to allocate to the account
BigDecimal allocated = new BigDecimal("1000.000000000000");
//the third parameter of the allocateMoney method is a String description of the
fund allocation transaction
accountRepository.allocateMoney(uniqueKey, allocated, "test money allocation
through Web Services");
```

### 3.5.5.2.9 Transferring money between accounts

The following sample transfers funds from a parent account to one of its child accounts. The transferMoney method can also transfer funds from a child account to its parent account. For the sample below, it is assumed that the following already exist:

- a parent account which is active and has money already allocated to it, where the record's unique key is a String equal to parentUniqueKey
- a child account for the parent which is active, where the record's unique key is a String equal to childUniqueKey

#### Code Snippet

```
protected AccountRepository accountRepository;
//the third parameter is the monetary amount to transfer between the parent and
child account; the fourth parameter is a String description of the transfer
transaction
accountRepository.transferMoney(parentUniqueKey, childUniqueKey, new
BigDecimal("3333.00"), "money transferred through Web Services");
```

#### 3.5.5.2.10 Withdrawing money from accounts

The following sample withdraws funds from an account. For the sample, it is assumed that an active account already exists.

##### Code Snippet

```
// this sample assumes an active account already exists with the unique key
(String) value equal to uniqueKey
protected AccountRepository accountRepository;
// Withdraw money from the account
BigDecimal used = new BigDecimal("100.000000000000");
//the third parameter is a String description of the withdrawal transaction
accountRepository.withdrawMoney(uniqueKey, used, "test allocating");
```

#### 3.5.5.2.11 Deleting accounts

The following sample deletes an existing account record.

##### Code Snippet

```
//for this example, it is assumed that you already know the target account record's
unique key (where uniqueKey is a String variable)
protected AccountRepository accountRepository;
accountRepository.deleteAccount(uniqueKey);
```

### 3.5.5.3 Appointments

The following .JAVA code samples show how you can use TeamConnect® Enterprise Web Services for functions related to appointment records.

#### 3.5.5.3.1 Creating appointments

The following sample creates an appointment record.

##### Code Snippet

```
protected AppointmentRepository appointmentRepository;
private String createAppointment() throws Exception {
    Date date = new Date();
    AppointmentCreate appointment = new AppointmentCreate();
    appointment.setSubject("Quarterly Meeting - Q2 2009");
    appointment.setLocation("Build A-5");
    appointment.setAssignedOn(date);
    appointment.setStartOn(date);
    appointment.setEndOn(date);
    appointment.setAllDay(true);
    appointment.setProjectUniqueKey("DISP_0003");
    appointment.getAttendeeCreates().add(attendees);
}
```

```
        appointment.getResourceCreates().add(resources);
        appointment.getCategories().add(categories);
        return appointmentRepository.insertAppointment(appointment);
    }
    private List<AttendeeCreate> attendeeCreate() {
        List<AttendeeCreate> attendees = new List<AttendeeCreate>();
        AttendeeCreate attendee1 = new AttendeeCreate();
        attendee1.setUserUniqueKey("USER_1005");
        attendee1.setAttendanceType(AttendanceType.WILL_ATTEND);
        AttendeeCreate attendee2 = new AttendeeCreate();
        attendee2.setUserUniqueKey("USER_1013");
        attendee2.setAttendanceType(AttendanceType.TENTATIVE);
        attendees.add(attendee1);
        attendees.add(attendee2);
        return attendees;
    }
    private List<AppointmentResourceCreate> appointmentResourceCreate()
    {
        List<AppointmentResourceCreate> appointmentResources = new
        List<AppointmentResourceCreate>();
        AppointmentResourceCreate apptResource = new AppointmentResourceCreate();
        //for this sample, it is assumed that for the Resource Type system lookup
        table, there is a table item with the unique key (or Tree position),
        "CRMA" (for Conference Room A)
        apptResource.setTypeUniqueKey("CRMA");
        appointmentResources.add(apptResource);
        return appointmentResources;
    }
    private List<Category> categoryAdd() {
        List<Category> categories = new ListArray<Category>();
        Category cat = new Category();
        //the next two lines identify a category and then add the category to the
        appointment record
        cat.setUniqueKey("APPT_TRIA");
        categories.add(cat);
        return categories;
    }
}
```

### 3.5.5.3.2 Updating appointments

The following sample updates an appointment record with given unique key.

#### Code Snippet

```
//for this sample, it is assumed that an appointment record already exists with a
unique key String equal to uniqueKey
protected AppointmentRepository appointmentRepository;
```

```
public void test_updateAppointment() throws Exception {
    Date date = new Date();
    AppointmentUpdate appointment = new AppointmentUpdate();
    //use the setUniqueKey method to identify the target task record to update by
    its unique key
    appointment.setUserUniqueKey("user_1003");
    appointment.setAssignedOn(date);
    appointment.setStartOn(date);
    appointment.setEndOn(date);
    appointment.setAllDay(true);
    appointment.setProjectUniqueKey("DISP_0003");
    appointment.getAttendeeUpdates().add(attendees);
    appointmentRepository.updateAppointment(appointment);
}

private List<AttendeeUpdate> attendeeUpdate() {
    List<AttendeeUpdate> attendees = new List<AttendeeUpdate>();
    AttendeeUpdate attendee1 = new AttendeeUpdate();
    //identify which attendee to update by the record's unique key
    attendee1.setUniqueKey("ATTN_1001");
    attendee1.setUserUniqueKey("USER_1005");
    attendee1.setAttendanceType(AttendanceType.WILL_NOT_ATTEND);
    attendees.add(attendee1);
    return attendees;
}
```

### 3.5.5.3.3 Reading appointments

The following sample reads an appointment record and returns the specified property values.

#### Code Snippet

```
protected AppointmentRepository appointmentRepository;
private List<String> getPropertiesToRead() {
    List<String> props = new List<String>();
    //the list of all available properties displays below but to increase
    efficiency in your searches, you can omit unnecessary properties from your
    application and those values will not be returned
    props.add("uniqueKey");
    props.add("version");
    props.add("subject");
    props.add("categories");
    props.add("createdOn");
    props.add("createdBy");
    props.add("modifiedBy");
    props.add("modifiedOn");
    props.add("location");
}
```

```
        props.add("startOn");
        props.add("endOn");
        props.add("allDay");
        props.add("project");
        props.add("attendees");
        props.add("resources");
        props.add("categories");
        return props;
    }
    private Appointment readAppointment() throws Exception {
        Appointment readAppointment = appointmentRepository.readAppointment(uniqueKey,
            getPropertiesToRead());
        return readAppointment;
    }
}
```

#### 3.5.5.3.4 Searching appointments

The following sample searches for one or more appointment records and returns the specified property values. For this sample, it is assumed there is one or more appointments already existing with a subject String field value equal to subject.

#### Code Snippet

```
protected AppointmentRepository appointmentRepository;
private List<Appointment> test_readAppointmentsByCriteria() throws
Exception {
    // Criterion for appointment searching
    StringFieldCriterion fieldCriterion = new StringFieldCriterion();
    fieldCriterion.setComparator(StringComparator.EQUALS_ENFORCE_CASE);
    LegacySearchFieldPathExpression fieldPathExpression = new
    LegacySearchFieldPathExpression();
    fieldPathExpression.setSearchKeyPath("subject");
    fieldCriterion.setFieldPath(fieldPathExpression);
    fieldCriterion.getValue().add("Client meeting");
    FieldSearchClause searchCriteria = new FieldSearchClause();
    searchCriteria.setOperator(LogicOperator.AND);
    searchCriteria.getCriteria().add(fieldCriterion);
    appointments =
    appointmentRepository.readAppointmentsByCriteria(searchCriteria, 100,
    getPropertiesToRead());
}
private List<String> getPropertiesToRead() {
    List<String> props = new List<String>();
    //the list of all available properties displays below but to increase
    efficiency in your searches, you can omit unnecessary properties from your
    application and those values will not be returned
    props.add("uniqueKey");
    props.add("version");
}
```

```
        props.add("subject");
        props.add("categories");
        props.add("createdOn");
        props.add("createdBy");
        props.add("modifiedBy");
        props.add("modifiedOn");
        props.add("location");
        props.add("startOn");
        props.add("endOn");
        props.add("allDay");
        props.add("project");
        props.add("attendees");
        props.add("resources");
        props.add("categories");
        return props;
    }
```

#### 3.5.5.3.5 Deleting appointments

The following sample deletes an existing appointment.

##### Code Snippet

```
//for this sample it is assumed that a appointment record exists with a unique key
String equal to uniqueKey
protected AppointmentRepository appointmentRepository;
public void test_deleteAppointment() throws Exception {
    appointmentRepository.deleteAppointment(uniqueKey);
}
```

#### 3.5.5.4 Document Records

The following .JAVA code samples show how you can use TeamConnect® Enterprise Web Services for functions related to document records.

##### 3.5.5.4.1 Creating document records

There are four types of document records: file, folder, hyperlink, shortcut. The following samples create a file. Separate ContactRepository methods are provided to create folders and shortcuts.

##### Code Snippet for creating a document file

```
Date date = new Date();
protected DocumentRepository documentRepository;
String uniqueKey = documentRepository.insertDocument(document);
private void createDocument() {
    DocumentCreate document = new DocumentCreate();
    document.setName("test_document.txt");
}
```

```
document.setContent("test document content".getBytes());
document.setType(DocumentType.FILE);
document.setParentFolderUniqueKey("DOCU_11");
document.setContentTypeUniqueKey("Text");
return document;
}
```

#### 3.5.5.4.2 Updating documents

The following samples show how to update documents for the available types (file, folder, hyperlink, shortcut).

##### Code Snippet for updating a document file

```
Date date = new Date();
protected DocumentRepository documentRepository;
documentRepository.updateDocument(document);
private void updateDocument() {
    DocumentUpdate document = new DocumentUpdate();
    document.setUniqueKey("DOCU_2010");
    document.setContent(byte[] value);
    document.setName("receipt012109");
    return document;
}
```

##### Code Snippet for updating a document folder

```
Date date = new Date();
protected DocumentRepository documentRepository;
documentRepository.updateDocument(document);
private void updateDocument() {
    DocumentUpdate document = new DocumentUpdate();
    document.setUniqueKey("DOCU_2011");
    document.setName("Correspondence_Jan09");
    document.setParentFolderUniqueKey("DOCU_3005");
    return document;
}
```

##### Code Snippet for updating a document hyperlink

```
Date date = new Date();
protected DocumentRepository documentRepository;
documentRepository.updateDocument(document);
private void updateDocument() {
    DocumentUpdate document = new DocumentUpdate();
    document.setUniqueKey("DOCU_2011");
    document.setName("BusinessWeek_URL");
    document.setUrl("http://businessweek.com");
}
```

```
        return document;
    }
}
```

### Code Snippet for updating a document shortcut

```
Date date = new Date();
protected DocumentRepository documentRepository;
documentRepository.updateDocument(document);
private void updateDocument() {
    DocumentUpdate document = new DocumentUpdate();
    document.setUniqueKey("DOCU_2010");
    document.setName("shortcutToWorksheet");
    return document;
}
```

#### 3.5.5.4.3 Reading documents

The following samples show how to read documents. Note that certain properties you can request to be returned are only relevant to particular document types (for example, the url property is only relevant to the document type, hyperlink). Comments are provided for properties that are document type-specific.

### Code Snippet

```
private List<String> getPropertiesToRead() {
    List<String> props = new List<String>();
    //the list of all available properties displays below but to increase
    efficiency in your searches, you can omit unnecessary properties from your
    application and those values will not be returned
    props.add("version");
    props.add("subject");
    props.add("categories");
    props.add("createdOn");
    props.add("createdBy");
    props.add("modifiedBy");
    props.add("modifiedOn");
    props.add("name");
    props.add("author");
    props.add("authorDate");
    props.add("size");
    //the contentType is only relevant to a document of type FILE
    props.add("contentType");
    //the contentType is only relevant to a document of type FILE
    props.add("content");
    props.add("type");
    props.add("parentFolder");
    props.add("checkedOutBy");
    props.add("checkedOutOn");
}
```



```
        props.add("checkedInBy");
        props.add("checkedInOn");
        props.add("versionText");
        //the url is only relevant to a document of type HYPERLINK
        props.add("url");
        //the contentType is only relevant to a document of type SHORTCUT
        props.add("shortcutToDocument");
        return props;
    }
    private Document readDocument() throws Exception {
        Document readDocument = documentRepository.readDocument(uniqueKey,
            getPropertiesToRead());
        return readDocument;
    }
}
```

#### 3.5.5.4.4 Reading child documents for a document

Performs one of the following:

- Gets child documents (files) for a given document folder.
- Gets previous versions of a document (files) for a given document file.

For both operations, documents are returned with only specified property values. For a list of available properties, see the type class, Document.

#### Code Snippet

```
private List<String> getPropertiesToRead() {
    List<String> props = new List<String>();
    //the list of all available properties displays below but to increase
    //efficiency in your searches, you can omit unnecessary properties from your
    //application and those values will not be returned
    props.add("version");
    props.add("subject");
    props.add("categories");
    props.add("createdOn");
    props.add("createdBy");
    props.add("modifiedBy");
    props.add("modifiedOn");
    props.add("name");
    props.add("author");
    props.add("authorDate");
    props.add("size");
    //the contentType is only relevant to a document of type FILE
    props.add("contentType");
    //the contentType is only relevant to a document of type FILE
    props.add("content");
}
```

```
        props.add("type");
        props.add("parentFolder");
        props.add("checkedOutBy");
        props.add("checkedOutOn");
        props.add("checkedInBy");
        props.add("checkedInOn");
        props.add("versionText");
        //the url is only relevant to a document of type HYPERLINK
        props.add("url");
        //the contentType is only relevant to a document of type SHORTCUT
        props.add("shortcutToDocument");
        return props;
    }
    private Document readChildDocument() throws Exception {
        //the first parameter, parentUniqueKey, identifies the unique key of the parent
        folder whose child documents' properties to return; where child documents would
        include files, subfolders, hyperlinks, and shortcuts included in the parent
        folder recursively
        List<Document> readChildDocument =
        documentRepository.readChildDocuments("DOCU_2534",
        getPropertiesToRead());
        return readChildDocument;
    }
}
```

#### 3.5.5.4.5 Reading documents by path

Retrieves a Document from the repository based on a folder path.

#### Code Snippet

```
private List<String> getPropertiesToRead() {
    List<String> props = new List<String>();
    //the list of all available properties displays below but to increase
    efficiency in your searches, you can omit unnecessary properties from your
    application and those values will not be returned
    props.add("version");
    props.add("subject");
    props.add("categories");
    props.add("createdOn");
    props.add("createdBy");
    props.add("modifiedBy");
    props.add("modifiedOn");
    props.add("name");
    props.add("author");
    props.add("authorDate");
    props.add("size");
    //the contentType is only relevant to a document of type FILE
}
```

```
        props.add("contentType");
        //the contentType is only relevant to a document of type FILE
        props.add("content");
        props.add("type");
        props.add("parentFolder");
        props.add("checkedOutBy");
        props.add("checkedOutOn");
        props.add("checkedInBy");
        props.add("checkedInOn");
        props.add("versionText");
        //the url is only relevant to a document of type HYPERLINK
        props.add("url");
        //the contentType is only relevant to a document of type SHORTCUT
        props.add("shortcutToDocument");
        return props;
    }
}

private Document readDocumentByPath() throws Exception {
    //the first parameter, path, identifies the document folder path to
    //the target document whose specified properties to return
    Document document =
        documentRepository.readDocumentByPath("TopLevel_Users_john",
        getPropertiesToRead());
    return document;
}
```

#### 3.5.5.4.6 Reading the child document of a parent folder

Retrieves a document from the repository given the file name and the unique ID of its parent folder.

#### Code Snippet

```
private List<String> getPropertiesToRead() {
    List<String> props = new List<String>();
    //the list of all available properties displays below but to increase
    //efficiency in your searches, you can omit unnecessary properties from your
    //application and those values will not be returned
    props.add("version");
    props.add("subject");
    props.add("categories");
    props.add("createdOn");
    props.add("createdBy");
    props.add("modifiedBy");
    props.add("modifiedOn");
    props.add("name");
    props.add("author");
    props.add("authorDate");
}
```

```
        props.add("size");
        //the contentType is only relevant to a document of type FILE
        props.add("contentType");
        //the contentType is only relevant to a document of type FILE
        props.add("content");
        props.add("type");
        props.add("parentFolder");
        props.add("checkedOutBy");
        props.add("checkedOutOn");
        props.add("checkedInBy");
        props.add("checkedInOn");
        props.add("versionText");
        //the url is only relevant to a document of type HYPERLINK
        props.add("url");
        //the contentType is only relevant to a document of type SHORTCUT
        props.add("shortcutToDocument");
        return props;
    }

    private Document readChildDocumentForName() throws Exception {
        //the first parameter, parentUniqueKey, identifies the unique key of the parent
        folder whose child documents' properties to return; where child documents would
        include files, subfolders, hyperlinks, and shortcuts included in the parent
        folder recursively

        //the second parameter identifies the target document file name, including file
        extension

        List<Document> readChildDocumentForName =
        documentRepository.readChildDocuments("DOCU_2789",
        "invoice101009_4009_attachment.doc", getPropertiesToRead());
        return readChildDocument;
    }
}
```

#### 3.5.5.4.7 Reading the Document Folder for a User

Retrieves a document folder from the repository given the unique key of the folder's owner (user).

#### Code Snippet

```
private List<String> getPropertiesToRead() {
    List<String> props = new List<String>();
    //the list of all available properties displays below but to increase
    efficiency in your searches, you can omit unnecessary properties from your
    application and those values will not be returned
    props.add("version");
    props.add("subject");
    props.add("categories");
    props.add("createdOn");
    props.add("createdBy");
    props.add("modifiedBy");
}
```

```
        props.add("modifiedOn");
        props.add("name");
        props.add("author");
        props.add("authorDate");
        props.add("size");
        props.add("content");
        props.add("type");
        props.add("parentFolder");
        props.add("checkedOutBy");
        props.add("checkedOutOn");
        props.add("checkedInBy");
        props.add("checkedInOn");
        props.add("versionText");
        return props;
    }

    private Document readDocumentFolderForDocumentOwner() throws Exception {
        //the first parameter, uniqueKey, identifies the unique key of the user whose
        //TeamConnect Documents area user folder to return (for example, the path to the
        //user folder for Al would look like Top Level\Users\al)
        document readDocumentFolderForDocumentOwner =
            documentRepository.readDocumentFolderForDocumentOwner("USER_1223",
                getPropertiesToRead());
        return readUserFolder;
    }
}
```

#### 3.5.5.4.8 Searching documents

The following samples show how to search documents by name, parent folder, date last modified, and author.

##### Code Snippet

```
\\for this sample it is assumed that a Date variable, date, exists representing a
past date
protected DocumentRepository documentRepository;
StringFieldCriterion fieldCriterion1 = new StringFieldCriterion();
fieldCriterion1.setComparator(StringComparator.CONTAINS);
LegacySearchFieldPathExpression fieldPathExpression1 = new
LegacySearchFieldPathExpression();
fieldPathExpression1.setSearchKeyPath("name");
fieldCriterion1.setFieldPath(fieldPathExpression);
fieldCriterion1.getValue().add("Invoice_10508");
StringFieldCriterion fieldCriterion2 = new StringFieldCriterion();
fieldCriterion2.setComparator(StringComparator.EQUALS);
LegacySearchFieldPathExpression fieldPathExpression2 = new
LegacySearchFieldPathExpression();
fieldPathExpression2.setSearchKeyPath("parentFolder.uniqueKey");
fieldCriterion2.setFieldPath(fieldPathExpression);
```

```
fieldCriterion2.getValue().add("DOCU_1099");
DateFieldCriterion fieldCriterion3 = new DateFieldCriterion();
fieldCriterion3.setComparator(DateComparator.EQUALS);
LegacySearchFieldPathExpression fieldPathExpression3 = new
LegacySearchFieldPathExpression();
fieldPathExpression3.setSearchKeyPath("modifiedOn");
fieldCriterion3.setFieldPath(fieldPathExpression);
fieldCriterion3.getValue().add(date);
StringFieldCriterion fieldCriterion4 = new StringFieldCriterion();
fieldCriterion4.setComparator(StringComparator.EQUALS);
LegacySearchFieldPathExpression fieldPathExpression4 = new
LegacySearchFieldPathExpression();
fieldPathExpression4.setSearchKeyPath("author.uniqueKey");
fieldCriterion4.setFieldPath(fieldPathExpression);
fieldCriterion4.getValue().add("CONT_1015");
FieldSearchClause searchCriteria = new FieldSearchClause();
searchCriteria.setOperator(LogicOperator.AND);
searchCriteria.getCriteria().add(fieldCriterion1);
searchCriteria.getCriteria().add(fieldCriterion2);
searchCriteria.getCriteria().add(fieldCriterion3);
searchCriteria.getCriteria().add(fieldCriterion4);
List<Document> searchDocument =
documentRepository.readDocumentsByCriteria(searchCriteria, 100,
getPropertiesToRead());
private List<String> getPropertiesToRead() {
    List<String> props = new List<String>();
    //the list of all available properties displays below but to increase
    efficiency in your searches, you can omit unnecessary properties from your
    application and those values will not be returned
    props.add("version");
    props.add("subject");
    props.add("categories");
    props.add("createdOn");
    props.add("createdBy");
    props.add("modifiedBy");
    props.add("modifiedOn");
    props.add("name");
    props.add("author");
    props.add("authorDate");
    props.add("size");
    //the contentType is only relevant to a document of type FILE
    props.add("contentType");
    //the contentType is only relevant to a document of type FILE
    props.add("content");
    props.add("type");
    props.add("parentFolder");
```

```
    props.add("checkedOutBy");
    props.add("checkedOutOn");
    props.add("checkedInBy");
    props.add("checkedInOn");
    props.add("versionText");
    //the url is only relevant to a document of type HYPERLINK
    props.add("url");
    //the contentType is only relevant to a document of type SHORTCUT
    props.add("shortcutToDocument");
    return props;
}
```

#### 3.5.5.4.9 Creating document shortcuts

Creates a shortcut or link to a document (file, folder, or existing shortcut) with a given unique key. The `parentFolderUniqueKey` defines the folder where the shortcut will be created. The unique key of the resulting shortcut (type of document record) is returned.

##### Code Snippet

```
protected DocumentRepository documentRepository;
//the first parameter, documentUniqueKey, identifies an existing document to create
a link to
//the second parameter, parentFolderUniqueKey, identifies the folder where the
shortcut will be created
String uniqueKey = documentRepository.createShortcut("DOCU_1451", "DOCU_1771");
```

#### 3.5.5.4.10 Creating subfolders

Creates a sub-folder with given name in given parent folder. The unique key of the resulting folder (type of document record) is returned.

##### Code Snippet

```
protected DocumentRepository documentRepository;
//the first parameter, name, sets the subfolder name
//the second parameter, parentUniqueKey, identifies the folder where the folder
will be created
String uniqueKey = documentRepository.createSubFolder("Child Folder", "DOCU_1010");
```

#### 3.5.5.4.11 Creating hyperlinks with the default Document category

Creates a hyperlink and sets its category to the default Document category.

##### Code Snippet

```
protected DocumentRepository documentRepository;
//the first parameter, parentFolderUniqueKey, identifies the folder where the
hyperlink will be created name, sets the subfolder name
//the second parameter, url, sets the website URL target for the hyperlink
```

```
//the third parameter, name, sets the hyperlink name that displays in the
application UI
//the fourth parameter, contactUniqueKey, identifies the hyperlink record author
String uniqueKey = documentRepository.createHyperlinkDefaultCategory("DOCU_1753",
"http://www.mitrates.com", "Mitrates URL", "CONT_1005");
```

#### 3.5.5.4.12 Creating hyperlinks

Creates a hyperlink.

##### Code Snippet

```
protected DocumentRepository documentRepository;
//the first parameter, parentFolderUniqueKey, identifies the folder where the
hyperlink will be created name, sets the subfolder name
//the second parameter, url, sets the website URL target for the hyperlink
//the third parameter, name, sets the hyperlink name that displays in the
application UI
//the fourth parameter, documentCategoryUniqueKey, identifies the document category
to set as the hyperlink category
//the fifth parameter, contactUniqueKey, identifies the hyperlink record author
String uniqueKey = documentRepository.createHyperlinkDefaultCategory("DOCU_1753",
"http://www.mitrates.com", "Mitrates URL", "DOCU_1855", "CONT_1005");
```

#### 3.5.5.4.13 Copying documents

Copies a document to a new parent folder.

##### Code Snippet

```
protected DocumentRepository documentRepository;
//the first parameter, documentUniqueKey, identifies an existing document to copy
//the second parameter, parentFolderUniqueKey, identifies the folder where the
document copy will be created
documentRepository.copyDocument("DOCU_1700", "DOCU_1811");
```

#### 3.5.5.4.14 Moving documents

Moves a document to a new parent folder.

##### Code Snippet

```
protected DocumentRepository documentRepository;
//the first parameter, documentUniqueKey, identifies an existing document to move
//the second parameter, newParentFolderUniqueKey, identifies the folder where the
document will be moved
documentRepository.moveDocument("DOCU_1900", "DOCU_1024");
```



#### 3.5.5.4.15 Checking out documents

Attempts to check out a document with a given unique key.

##### Code Snippet

```
protected DocumentRepository documentRepository;
//the first parameter, documentUniqueKey, identifies an existing document to check
out
documentRepository.checkOut("DOCU_2900");
```

#### 3.5.5.4.16 Undoing a document checkout

Cancels a checkout operation on a document with a given unique key.

##### Code Snippet

```
protected DocumentRepository documentRepository;
//the parameter, documentUniqueKey, identifies an existing document to undo the
checkout operation for
documentRepository.undoCheckOut("DOCU_2900");
```

#### 3.5.5.4.17 Checking in documents

Attempts to check in a document (of type file) with a given unique key.

##### Code Snippet

```
byte[] docContent = "test document content".getBytes();
protected DocumentRepository documentRepository;
//the first parameter, documentUniqueKey, identifies an existing document to check
in a new version for
//the second parameter, content, contains the updated file content to check in
//the third parameter, version, identifies the numeric version of the file you are
checking in
documentRepository.checkIn("DOCU_2901", docContent, "1.3");
```

#### 3.5.5.4.18 Reverting a document to a previous version

Attempts to revert a given document to a given old version.

##### Code Snippet

```
protected DocumentRepository documentRepository;
//the first parameter, documentUniqueKey, identifies the version of the document
file (for example the current version) to replace
//the second parameter, oldKey, identifies the version of the document file to
revert to (sets this file to the current version)
documentRepository.revert("DOCU_2902", "DOCU_2892");
```

#### 3.5.5.4.19 Deleting documents

The following sample deletes a document. You must get the unique key value for record to delete.

##### Code Snippet

```
//for this sample it is assumed that a document exists with the unique key equal to  
a String, uniqueKey  
protected DocumentRepository documentRepository;  
documentRepository.deleteDocument(uniqueKey);
```

#### 3.5.5.5 Expenses

The following .JAVA code samples show how you can use TeamConnect® Enterprise Web Services for functions related to expense records.

##### 3.5.5.5.1 Creating expenses

The following sample creates an expense record.

##### Code Snippet

```
protected ExpenseRepository expenseRepository;  
private String createExpense() throws Exception {  
    Date date = new Date();  
    ExpenseCreate expense = new ExpenseCreate();  
    expense.setContactUniqueKey("CONT_1003");  
    //expenseDate is a required property  
    expense.setExpenseDate(date);  
    //expensedByUniqueKey is a required property  
    expense.setExpensedByUniqueKey("EXPE_1212");  
    //shortDescription is a required property  
    expense.setShortDescription(String value);  
    expense.setProjectUniqueKey(String value);  
    expense.setQuantity(BigDecimal value);  
    expense.setUnitPrice(BigDecimal value);  
    return expenseRepository.insertExpense(expense);  
}
```

##### 3.5.5.5.2 Updating expenses

The following sample updates an expense record's expenseDate, projectUniqueKey, quantity, totalAmount, unitPrice values.

##### Code Snippet

```
//for this sample, it is assumed that an expense record already  
exists with a unique key String equal to uniqueKey  
protected ExpenseRepository expenseRepository;
```

```
public void test_updateExpense() throws Exception {
    Date date = new Date();
    ExpenseUpdate expenseUpdate = new ExpenseUpdate();
    //use the setUniqueKey method to identify the target expense record to update
    by its unique key
    expenseUpdate.setUniqueKey(uniqueKey);
    expenseUpdate.setExpenseDate(date);
    expenseUpdate.setProjectUniqueKey("DISP_2003");
    expenseUpdate.setQuantity(15.0);
    expenseUpdate.setUnitPrice(2.0);
    //Note that the totalAmount property is calculated by the product of the
    quantity and unitPrice property values. In the expense update sample provided,
    the totalAmount property will automatically be updated to 30.0
    expenseRepository.updateExpense(expenseUpdate);
}
```

### 3.5.5.5.3 Reading expenses

The following sample reads an expense record and returns the specified property values.

#### Code Snippet

```
protected ExpenseRepository expenseRepository;
private List<String> getPropertiesToRead() {
    List<String> props = new List<String>();
    //the list of all available properties displays below but to ncrease efficiency
    in your searches, you can omit unnecessary properties from your application and
    those values will not be returned
    props.add("shortDescription");
    props.add("categories");
    props.add("createdOn");
    props.add("modifiedBy");
    props.add("createdBy");
    props.add("modifiedOn");
    props.add("version");
    props.add("contact");
    props.add("expenseDate");
    props.add("expensedBy");
    props.add("project");
    props.add("quantity");
    props.add("unitPrice");
    props.add("totalAmount");
    return props;
}
private Expense readExpense() throws Exception {
    Expense readExpense = expenseRepository.readExpense(uniqueKey,
        getPropertiesToRead());
}
```

```
        return readExpense;
    }
}
```

#### 3.5.5.5.4 Searching expenses

The following sample searches for one or more expense records and returns the specified property values. For this sample, it is assumed there are one or more expenses already existing with a subject String field value equal to subject.

#### Code Snippet

```
protected ExpenseRepository expenseRepository;
private List<Expense> test_readExpensesByCriteria() throws Exception
{
    // Criterion for expense searching
    StringFieldCriterion fieldCriterion = new StringFieldCriterion();
    fieldCriterion.setComparator(StringComparator.EQUALS_ENFORCE_CASE);
    LegacySearchFieldPathExpression fieldPathExpression = new
    LegacySearchFieldPathExpression();
    fieldPathExpression.setSearchKeyPath("shortDescription");
    fieldCriterion.setFieldPath(fieldPathExpression);
    fieldCriterion.getValue().add(subject);
    FieldSearchClause searchCriteria = new FieldSearchClause();
    searchCriteria.setOperator(LogicOperator.AND);
    searchCriteria.getCriteria().add(fieldCriterion);
    expenses = expenseRepository.readExpensesByCriteria(searchCriteria, 100,
    getPropertiesToRead());
}
private List<String> getPropertiesToRead() {
    List<String> props = new List<String>();
    //the list of all available properties displays below but to increase
    efficiency in your searches, you can omit unnecessary properties from your
    application and those values will not be returned
    props.add("shortDescription");
    props.add("uniqueKey");
    props.add("categories");
    props.add("createdOn");
    props.add("modifiedBy");
    props.add("createdBy");
    props.add("modifiedOn");
    props.add("version");
    props.add("contact");
    props.add("expenseDate");
    props.add("expensedBy");
    props.add("project");
    props.add("quantity");
    props.add("unitPrice");
}
```

```
        props.add("totalAmount");
        return props;
    }
```

#### 3.5.5.5.5 Posting expenses

The following sample posts an expense with given unique key.

##### Code Snippet

```
String expenseUniqueKey = "EXPE_1001";
protected ExpenseRepository expenseRepository;
expenseRepository.postExpense(expenseUniqueKey);
```

#### 3.5.5.5.6 Voiding expenses

The following sample voids a expense with given unique key.

##### Code Snippet

```
String expenseUniqueKey = "EXPE_1001";
protected ExpenseRepository expenseRepository;
expenseRepository.voidExpense(expenseUniqueKey);
```

#### 3.5.5.5.7 Deleting expenses

The following sample deletes an existing expense.

##### Code Snippet

```
//for this sample, it is assumed that an expense record already exists with a
unique key String equal to uniqueKey
protected ExpenseRepository expenseRepository;
public void test_deleteExpense() throws Exception {
    expenseRepository.deleteExpense(uniqueKey);
}
```

### 3.5.5.6 Group Accounts

The following .JAVA code samples show how you can use the Group Account Web Service to manage group account records. From the Group Account Web Services, you can create group accounts and set the group account name, description, and add users to the group. Group Account Web Services do not support all functionality available from the TeamConnect application, such as setting group record rights, setting group category rights, setting group tool rights, and setting default object views.

### 3.5.5.6.1 Creating group accounts

The following sample creates a group account record. Also see the section about [User Accounts](#).

#### Code Snippet

```
// this sample assumes at least one user account record already exists with the
unique key (String) value equal to userUniqueKey
protected GroupAccountRepository groupAccountRepository;
private String insertGroupAccount() throws Exception {
    String groupAccountName = "administrators";
    GroupAccountCreate groupAccount = new GroupAccountCreate();
    //uniqueName is a required field
    groupAccount.setUniqueName(groupAccountName);
    //displayName is a required field
    groupAccount.setDisplayName(groupAccountName);
    //add users to the group
    groupAccount.getUserAccountUniqueKeys().add(userUniqueKeys);
    String uniqueKey = groupAccountRepository.insertGroupAccount(groupAccount);
    return uniqueKey;
}
private List<String> userAccountUniqueKeys() {
    List<String> userUniqueKeys = new List<String>();
    userUniqueKeys.add("USER_1005");
    userUniqueKeys.add("USER_1009");
    userUniqueKeys.add("USER_1041");
    return userUniqueKeys;
}
```

### 3.5.5.6.2 Updating group accounts

The following sample updates an existing group account record by changing the display name, changing the description, and removing an existing group member (user account).

#### Code Snippet

```
// this sample assumes a group account record already exists with the unique key
(String) value equal to uniqueKey
protected GroupAccountRepository groupAccountRepository;
public void test_updateGroupAccount() throws Exception {
    GroupAccountUpdate groupUpdate = new GroupAccountUpdate();
    //set the uniqueKey with the String value of the target group record to
    identify which record you are updating
    groupUpdate.setUniqueKey(uniqueKey);
    //update the existing display name
    groupUpdate.setDisplayName("BusinessAdministrators");
    //update the existing group description
```

```
groupUpdate.setDescription("Administrators who manage Admin Settings and
TeamConnect user/group accounts");
//remove a user from the group
groupUpdate.getUserAccountDeleteUniqueKeys().add(userUniqueKeys);
groupAccountRepository.updateGroupAccount(groupUpdate);
}
private List<String> userAccountUniqueKeys() {
    List<String> userUniqueKeys = new List<String>();
    userUniqueKeys.add("USER_1005");
    return userUniqueKeys;
}
```

#### 3.5.5.6.3 Reading group accounts

The following sample reads the specified properties of a group account record. All available properties of a group account are listed but you can comment out or remove unnecessary properties in your application to improve performance.

#### Code Snippet

```
//define the list of group account record properties to return with the
readGroupAccount results
protected GroupAccountRepository groupAccountRepository;
private List<String> getProperties() {
    List<String> properties = new List<String>();
    properties.add("uniqueName");
    properties.add("displayName");
    properties.add("description");
    properties.add("createdBy");
    properties.add("createdOn");
    properties.add("modifiedBy");
    properties.add("modifiedOn");
    properties.add("users");
    return properties;
}
public GroupAccount test_readGroupAccount() throws Exception {
    //it is assumed that a group account record exists with the unique key equal to
    String, uniqueKey
    GroupAccount searchedGroup = groupAccountRepository.readGroupAccount(uniqueKey,
    getProperties());
    return searchedGroup;
}
```

#### 3.5.5.6.4 Searching group accounts

The following sample searches groups by given search criteria and returns the specified properties of the resulting group account records.

##### Code Snippet

```
protected GroupAccountRepository groupAccountRepository;
private List<GroupAccount> test_readGroupAccountsByCriteria() throws
Exception {
    // Criterion for group account searching
    UserFieldCriterion fieldCriterion = new UserFieldCriterion();
    fieldCriterion.setComparator(UserComparator.ONE_OF);
    LegacySearchFieldPathExpression fieldPathExpression = new
LegacySearchFieldPathExpression();
    fieldPathExpression.setSearchKeyPath("createdBy");
    fieldCriterion.setFieldPath(fieldPathExpression);
    fieldCriterion.getValues().add("JohnDoe");
    FieldSearchClause searchCriteria = new FieldSearchClause();
    searchCriteria.setOperator(LogicOperator.AND);
    searchCriteria.getCriteria().add(fieldCriterion);
    groups = groupAccountRepository.readGroupAccountsByCriteria(searchCriteria,
100, getPropertiesToRead());
}
private List<String> getProperties() {
    List<String> properties = new List<String>();
    properties.add("uniqueName");
    properties.add("displayName");
    properties.add("description");
    properties.add("createdBy");
    properties.add("createdOn");
    properties.add("modifiedBy");
    properties.add("modifiedOn");
    properties.add("users");
    return properties;
}
```

#### 3.5.5.6.5 Deleting group accounts

The following sample deletes a group account record with a given unique key.

*Note: Deleting a group removes users from the group; however the user accounts are not deleted from the system. Also see the Administrator help, [Deleting a Group](#), topic for Scenarios that [Prevent Deletion](#).*

##### Code Snippet



```
//for this sample it is assumed that a group record exists with a unique key String
equal to uniqueKey
protected GroupAccountRepository groupAccountRepository;
public void test_deleteGroupAccount() throws Exception {
    groupAccountRepository.deleteGroupAccount(uniqueKey);
}
```

### 3.5.5.7 History Records

The following .JAVA code samples show how you can use TeamConnect® Enterprise Web Services for functions related to history records.

#### 3.5.5.7.1 Creating history records

The following sample creates a history record.

##### Code Snippet

```
Date date = new Date();
protected HistoryRepository historyRepository;
String uniqueKey = historyRepository.insertHistory(history);
private void createHistory() {
    HistoryCreate history = new HistoryCreate();
    //archivedOn is a required field
    history.setArchivedOn(date);
    //shortDescription is a required field
    history.setShortDescription("Subject - Update contact phone number");
    //text is an additional field that stores more information than the
    shortDescription; text allows up to 20000 characters
    history.setText("Additional information about the contact record update");
    Category cat = new Category();
    //the next two lines identify a category and then add the category to the
    history record
    cat.setUniqueKey("HIST_STAT");
    history.getCategories().add(cat);
    //use the parentObject property to associate the history with an existing
    record by unique key, for example, to describe a contact record that you have
    recently modified
    history.setParentObject("CONT_1234");
    return history;
}
```

#### 3.5.5.7.2 Updating history records

The following sample updates a history record. Prerequisite to working with history records, you must also get the unique key value for the history record to update. If the history record is associated with another record, you need to get that record's unique key

### Code Snippet

```
Date date = new Date();
protected HistoryRepository historyRepository;
historyRepository.updateHistory(history);
private void updateHistory() {
    HistoryUpdate history = new HistoryUpdate();
    //use setUniqueKey to identify the target history record to update
    history.setUniqueKey("HIST_0011");
    history.setShortDescription("Subject - Update contact phone number");
    history.setText("Additional information about the contact record update");
    history.setArchivedOn(date);
    Category cat = new Category();
    //the next two lines identify a category and then add the category to the
    history record
    cat.setUniqueKey("HIST_MANA");
    history.getCategories().add(cat);
    //use the parentObject property to associate the history with an existing
    record by unique key, for example, to describe a contact record that you have
    recently modified
    history.setParentObject("CONT_1234");
    return history;
}
```

#### 3.5.5.7.3 Reading history records

The following sample reads a history record. You must get the unique key value for the history record to update.

### Code Snippet

```
//for this sample, it is assumed that the unique key of the target
History record to read is a String, "HIST_0011"
protected HistoryRepository historyRepository;
History readHistory = historyRepository.readHistory("HIST_0011",
getPropertiesToRead());
private List<String> getPropertiesToRead() {
    List<String> properties= new List<String>();
    properties.add("version");
    properties.add("createdBy");
    properties.add("createdOn");
    properties.add("modifiedBy");
    properties.add("modifiedOn");
    properties.add("archivedOn");
    properties.add("text");
    properties.add("shortDescription");
    properties.add("parentObject");
}
```

```
        properties.add("categories");
        return properties;
    }
```

#### 3.5.5.7.4 Searching history records

The following sample searches for history records by short description (records that contain "updated contact record" in the Description) and date (records created after 12/01/2008).

#### Code Snippet

```
protected HistoryRepository historyRepository;
StringFieldCriterion fieldCriterion1 = new StringFieldCriterion();
fieldCriterion1.setComparator(StringComparator.CONTAINS);
LegacySearchFieldPathExpression fieldPathExpression1 = new
LegacySearchFieldPathExpression();
fieldPathExpression1.setSearchKeyPath("shortDescription");
fieldCriterion1.setFieldPath(fieldPathExpression);
fieldCriterion1.getValue().add("updated contact record");
DateFieldCriterion fieldCriterion2 = new DateFieldCriterion();
fieldCriterion2.setComparator(StringComparator.AFTER);
LegacySearchFieldPathExpression fieldPathExpression2 = new
LegacySearchFieldPathExpression();
fieldPathExpression2.setSearchKeyPath("createdOn");
fieldCriterion2.setFieldPath(fieldPathExpression);
fieldCriterion2.getValue().add(01-DEC-08);
FieldSearchClause searchCriteria = new FieldSearchClause();
searchCriteria.setOperator(LogicOperator.AND);
searchCriteria.getCriteria().add(fieldCriterion1);
searchCriteria.getCriteria().add(fieldCriterion2);
List<History> searchHistory =
historyRepository.readHistoriesByCriteria(searchCriteria, 100,
getPropertiesToRead());
private List<String> getPropertiesToRead() {
    List<String> properties = new List<String>();
    properties.add("version");
    properties.add("createdBy");
    properties.add("createdOn");
    properties.add("modifiedBy");
    properties.add("modifiedOn");
    properties.add("archivedOn");
    properties.add("text");
    properties.add("shortDescription");
    properties.add("parentObject");
    properties.add("categories");
    return properties;
}
```

#### 3.5.5.7.5 Deleting history records

The following sample deletes a history record. You must get the unique key value for record to delete.

#### Code Snippet

```
//for this sample it is assumed that a history record exists with the unique key  
equal to a String, uniqueKey  
protected HistoryRepository historyRepository;  
historyRepository.deleteHistory(uniqueKey);
```

#### 3.5.5.8 Invoices and Line Items

The following .JAVA code samples show how you can use TeamConnect® Enterprise Web Services for functions specific to invoice and line item records, such as adding line items to an invoice, posting, adjusting an invoice summary, and adjusting invoice line items.

##### 3.5.5.8.1 Creating an Invoice with Line Items

The following example shows how to create an invoice and add an expense and fee line item to the invoice. When creating an invoice, you must populate a few required properties using the following methods (from the type class, InvoiceCreate): setNumberString, setInvoiceDate, and setVendorUniqueKey.

It is most practical to add one or more line items to an invoice. There are two types of line items you can add: expense or fee line items. For expense line items, you must populate a few required properties using the following methods (from the type class, LineItemCreate): setItemNumber, setServiceDate, setProjectUniqueKey, setOriginalRate, setOriginalQuantity, setOriginalDiscount, setType, and getCategories. For fee line items, you must populate a few required properties using the following methods (from the type class, LineItemCreate): setItemNumber, setServiceDate, setProjectUniqueKey, setOriginalRate, setOriginalQuantity, setOriginalDiscount, setType, getCategories, setTimekeeperUniqueKey, and setActivityUniqueKey.

Before starting, get the following:

- For expense line items, get the corresponding Line Item Expense Category Code(s)
- For fee line items, get the corresponding Line Item Fee Category Code(s) and Task Activity Lookup Table Activity Code(s)

When adding an expense line item to an invoice, use the Category object to specify the corresponding expense category. As described in the category example for contacts, you must already know the target expense category's unique key.

#### Code Snippet

```
protected InvoiceRepository invoice = new InvoiceRepository();  
InvoiceCreate invoice = createInvoice();  
invoice.insertInvoice(invoice);  
private InvoiceCreate createInvoice() throws Exception {  
    Date date = new Date();  
    long time = date.getTime();
```

```
InvoiceCreate invoice = new InvoiceCreate();
//numberString is a required field
invoice.setNumberString("invoice Web Service test " + time);
//vendorUniqueKey is a required field
invoice.setVendorUniqueKey("CONT_8094");
//invoiceDate is a required field
invoice.setInvoiceDate(date);
invoice.setComment("comment");
invoice.setPeriodStartDate(date);
invoice.setPeriodEndDate(new Date(time + 1000000));
invoice.setSubmittedElectronically(true);
invoice.setReceivedDate(date);
invoice.setTaxRate(new BigDecimal("8"));
invoice.setWarnings("invoice warning");
//Note that it is practical although not required to add one or more line items
to an invoice
invoice.getLineItemCreates().addAll(createLineItems());
return invoice;
}
private List<LineItemCreate> createLineItems() throws Exception {
    List<LineItemCreate> lineItems = new List<LineItemCreate>();
    //create expense line item
    LineItemCreate lineItem1 = new LineItemCreate();
    //itemNumber is a required field
    lineItem1.setItemNumber(1);
    //for expense line items, originalQuantity is a required field that corresponds
to a quantity of expense items being charged
    lineItem1.setOriginalQuantity(new BigDecimal("2"));
    //for expense line items, originalRate is a required field that corresponds to
the per unit cost of expense item
    lineItem1.setOriginalRate(new BigDecimal("50"));
    //for expense line items, type (LineItemType) is a required field and would
always be EXPENSE
    lineItem1.setType(LineItemType.EXPENSE);
    lineItem1.setServiceDate(date);
    //category for each line item is a required field
    Category cat1 = new Category();
    cat1.setUniqueKey("EXPE");
    lineItem1.getCategories().add(cat1);
    lineItems.add(lineItem1);
    //create fee line item
    LineItemCreate lineItem2 = new LineItemCreate();
    lineItem2.setItemNumber(2);
    //for Fee line items, the originalQuantity is a required field and will
correspond to the hours worked by the timekeeper
    lineItem2.setOriginalQuantity(new BigDecimal("2"));
```

```

//for Fee line items, the originalRate is a required field and will correspond
to the associated timekeeper's rate
lineItem2.setOriginalRate(new BigDecimal("50"));
//for fee line items, type (LineItemType) is a required field and would always
be FEE
lineItem2.setType(LineItemType.FEE);
lineItem2.setServiceDate(date);
Category cat2 = new Category();
cat2.setUniqueKey("TASK");
lineItem2.getCategories().add(cat2);
//for fee line items, the activityUniqueKey is a required field; the
ActivityUniqueKey property needs to be constructed by "ACTI_"+<4 character Task
Activity Lookup Table Activity Code>; see the example below where the Activity
Code is A101
lineItem2.setActivityUniqueKey("ACTI_A101");
lineItems.add(lineItem2);
return lineItems;
}

```

#### 3.5.5.8.2 Updating and Adjusting an Invoice

The following example shows how to update an invoice. You can edit the properties of the invoice as well as adjust the invoice summary or adjust the invoice's line items when you perform an invoice update.

The section of the example that shows how to adjust an invoice summary does so by changing the invoice total to a specified amount. It is assumed that there is an existing invoice that will be adjusted. To access an existing invoice, use the InvoiceRepository's updateInvoice method. To create the invoice summary adjustment, use the type class, InvoiceAdjustmentCreate. Note that there are other ways to adjust an invoice summary and you can refer to the User Guide for more general information.

When creating an invoice summary adjustment, you must use the following set methods to populate required fields: setAdjustingUserUniqueKey(String value), setAdjustmentDate(Date value), setAdjustmentTarget(InvoiceAdjustmentTarget value), setAdjustmentValue(BigDecimal value), setAdjustmentMethod(AdjustmentMethod value).

For more information on adjusting line items, see [Adjusting Invoice Line Items](#).

#### Code Snippet

```

//for this example, it is assumed that you already know the target
invoice record's unique key (where uniqueKey is a String variable)
protected InvoiceRepository invoice = new InvoiceRepository();
InvoiceUpdate invoice = updateInvoice();
invoice.updateInvoice(invoice);
private InvoiceUpdate updateInvoice() throws Exception {
    Date date = new Date();
    String uniqueKey = "inv5010897";
    InvoiceUpdate invoice = new InvoiceUpdate();
}

```

```

//next line used to ID the invoice to update
invoice.setUniqueKey(uniqueKey);
//these 3 lines update invoice properties
invoice.setNumberString("inv100-2053");
invoice.setInvoiceDate(date);
invoice.setVendorUniqueKey("ven2345");
//these lines are used to adjust the invoice header/summary
invoice.getAdjustmentUpdates().add(createInvoiceAdjustment());
return invoice;
}
private InvoiceAdjustmentCreate createInvoiceAdjustment() throws Exception {
    String userUniqueKey = "user101";
    InvoiceAdjustmentCreate invoiceAdjustment = new InvoiceAdjustmentCreate();
    invoiceAdjustment.setAdjustingUserUniqueKey(userUniqueKey);
    invoiceAdjustment.setAdjustmentDate(date);
    invoiceAdjustment.setAdjustmentTarget(InvoiceAdjustmentTarget.TOTAL_INVOICE);
    invoiceAdjustment.setAdjustmentValue(2005.17);
    invoiceAdjustment.setAdjustmentMethod(AdjustmentMethod.NEW_AMOUNT);
    invoiceAdjustment.setInHouseComments("Rejected invoice because the timekeeper
rate exceeds our agreement. Adjusting the total per acceptable rates.");
    invoiceAdjustment.setCommentsToVendor("Rejected invoice because the timekeeper
rate exceeds our agreement. Adjusting the total per acceptable rates.");
    return invoiceAdjustment;
}

```

### 3.5.5.8.3 Adjusting Invoice Line Items

The following example shows how to adjust an invoice line item by changing the line item rate by a specified amount. In the example, it is assumed there is an existing invoice with one or more line items. To access an existing invoice, use the InvoiceRepository's updateInvoice method. To perform the line item adjustment, use the type class, LineItemAdjustmentUpdate.

*Note: You can also adjust line items at the time you add line items to an invoice. In this case, you can use either the InvoiceRepository's insertInvoice or updateInvoice method. Since you are adjusting the line item and creating it at the same time, use the type class, LineItemAdjustmentCreate.*

When performing a line item adjustment, you must use the following set methods to populate required fields: .

#### **Code Snippet for adjusting invoice line items on invoice update**

```

protected InvoiceRepository invoice = new InvoiceRepository();
InvoiceUpdate invoice = updateInvoice();
invoice.updateInvoice(invoice);
private InvoiceUpdate updateInvoice() throws Exception {

```

```
Date date = new Date();
InvoiceUpdate invoice = new InvoiceUpdate();
//these 3 lines are used to ID the invoice to update
invoice.setNumberString("inv100-2053");
invoice.setInvoiceDate(date);
invoice.setVendorUniqueKey("ven2345");
//these lines are used to adjust an invoice line item
invoice.getLineItemCreates().add(createLineItemAdjustment());
return invoice;
}
private LineItemAdjustmentCreate createLineItemAdjustment() throws Exception {
    String userUniqueKey = "user_101";
    LineItemAdjustmentCreate lineItemAdjustment = new LineItemAdjustmentCreate();
    lineItemAdjustment.setAdjustingUserUniqueKey(userUniqueKey);
    lineItemAdjustment.setAdjustmentDate(date);
    lineItemAdjustment.setAdjustmentTarget(LineItemAdjustmentTarget.RATE);
    lineItemAdjustment.setAdjustmentValue(45.5);
    lineItemAdjustment.setAdjustmentMethod(AdjustmentMethod.REDUCE_BY_AMOUNT);
    lineItemAdjustment.setInHouseComments("Rejected invoice because the timekeeper
rate exceeds our agreement. Adjusting the rate down.");
    lineItemAdjustment.setCommentsToVendor("Rejected invoice because the timekeeper
rate exceeds our agreement. Adjusting the line item rate.");
    return lineItemAdjustment;
}
```

#### 3.5.5.8.4 Reading an Invoice

The following sample reads an invoice record. You must get the unique key value for the invoice record to update.

#### Code Snippet

```
//for this sample, it is assumed that the unique key of the target Invoice record
to read is a String, "INVC_0011"
protected InvoiceRepository invoiceRepository;
Invoice invoice = invoiceRepository.readInvoice("INVC_0011",
getPropertiesToRead());
private List<String> getPropertiesToRead() {
    List<String> properties = new List<String>();
    properties.add("version");
    properties.add("createdBy");
    properties.add("createdOn");
    properties.add("modifiedBy");
    properties.add("modifiedOn");
    properties.add("numberString");
    properties.add("invoiceDate");
    properties.add("receivedDate");
}
```



```
properties.add("periodStartDate");
properties.add("periodEndDate");
properties.add("submittedElectronically");
properties.add("postingStatus");
properties.add("vendor");
properties.add("submittedTotal");
properties.add("currency");
properties.add("comment");
properties.add("warnings");
properties.add("taxRate");
properties.add("adjustments");
properties.add("lineItems");
properties.add("originalExpenseTotal");
properties.add("originalFeeTotal");
properties.add("originalInvoiceTotal");
properties.add("expenseDiscountTotal");
properties.add("feeDiscountTotal");
properties.add("invoiceDiscountTotal");
properties.add("expenseAdjustmentTotal");
properties.add("feeAdjustmentTotal");
properties.add("invoiceAdjustmentTotal");
properties.add("expenseTaxTotal");
properties.add("feeTaxTotal");
properties.add("invoiceTaxTotal");
properties.add("netExpenseTotal");
properties.add("netFeeTotal");
properties.add("netInvoiceTotal");
properties.add("categories");
properties.add("note");
return properties;
}
```

#### 3.5.5.8.5 Searching for Invoices

The following sample searches invoices by given search criteria (a particular invoice vendor) and returns the specified properties of the resulting invoice records.

#### Code Snippet

```
protected InvoiceRepository invoiceRepository;
private List<Invoice> test_readInvoicesByCriteria() throws Exception
{
    // Criterion for invoice searching
    StringFieldCriterion fieldCriterion = new StringFieldCriterion();
    fieldCriterion.setComparator(UserComparator.CONTAINS);
}
```

```
LegacySearchFieldPathExpression fieldPathExpression = new
LegacySearchFieldPathExpression();
fieldPathExpression.setSearchKeyPath("vendor");
fieldCriterion.setFieldPath(fieldPathExpression);
fieldCriterion.getValue().add("Rubidoux Accounting");
FieldSearchClause searchCriteria = new FieldSearchClause();
searchCriteria.setOperator(LogicOperator.AND);
searchCriteria.getCriteria().add(fieldCriterion);
invoices = invoiceRepository.readInvoicesByCriteria(searchCriteria, 100,
getPropertiesToRead());
}
private List<String> getPropertiesToRead() {
List<String> properties = new List<String>();
properties.add("version");
properties.add("createdBy");
properties.add("createdOn");
properties.add("modifiedBy");
properties.add("modifiedOn");
properties.add("numberString");
properties.add("invoiceDate");
properties.add("receivedDate");
properties.add("periodStartDate");
properties.add("periodEndDate");
properties.add("submittedElectronically");
properties.add("postingStatus");
properties.add("vendor");
properties.add("submittedTotal");
properties.add("currency");
properties.add("comment");
properties.add("warnings");
properties.add("taxRate");
properties.add("adjustments");
properties.add("lineItems");
properties.add("originalExpenseTotal");
properties.add("originalFeeTotal");
properties.add("originalInvoiceTotal");
properties.add("expenseDiscountTotal");
properties.add("feeDiscountTotal");
properties.add("invoiceDiscountTotal");
properties.add("expenseAdjustmentTotal");
properties.add("feeAdjustmentTotal");
properties.add("invoiceAdjustmentTotal");
properties.add("expenseTaxTotal");
properties.add("feeTaxTotal");
properties.add("invoiceTaxTotal");
```

```
properties.add("netExpenseTotal");
properties.add("netFeeTotal");
properties.add("netInvoiceTotal");
properties.add("categories");
properties.add("note");
return properties;
}
```

#### 3.5.5.8.6 Posting an Invoice

The following example shows how to post an invoice. Before starting, you must get the unique key of the target invoice record.

##### Code Snippet

```
String invoiceUniqueKey = "inv_1001";
protected InvoiceRepository invoiceRepository;
invoiceRepository.postInvoice(invoiceUniqueKey);
```

#### 3.5.5.8.7 Voiding an Invoice

The following example shows how to void an invoice. Before starting, you must get the unique key of the target invoice record.

##### Code Snippet

```
String invoiceUniqueKey = "inv_1001";
protected InvoiceRepository invoiceRepository;
invoiceRepository.voidInvoice(invoiceUniqueKey);
```

#### 3.5.5.8.8 Searching for Line Items

When searching for or filtering line items that are associated with an invoice, use the Invoice Repository's `readInvoicesByCriteria` method. In the definition of the `fieldPathExpression.setSearchKeyPath`, use the properties defined in the type class, `LineItem`, to identify which line item fields to use as search criteria.

The following example filters an invoice's line items based on associated project, fee line items, and a given timekeeper.

##### Code Snippet

```
//for this example, it is assumed that at least one invoice line item record that
meets the search criterion already exists
protected InvoiceRepository invoiceRepository;
public void searchInvoices() throws Exception {
    ObjectFieldCriterion fieldCriterion1 = new ObjectFieldCriterion();
    fieldCriterion1.setComparator(ObjectComparator.EQUALS);
    LegacySearchFieldPathExpression fieldPathExpression1 = new
    LegacySearchFieldPathExpression();
```

```
fieldPathExpression1.setSearchKeyPath("lineItemList.project.primaryKey");
fieldCriterion1.setFieldPath(fieldPathExpression1);
fieldCriterion.setValue("DISP_1007");
FieldSearchClause searchCriteria = new FieldSearchClause();
searchCriteria.setOperator(LogicOperator.AND);
searchCriteria.getCriteria().add(fieldCriterion1);
//the following section defines a second fieldCriterion item for filtering fee
line items
StringFieldCriterion fieldCriterion2 = new StringFieldCriterion();
fieldCriterion.setComparator(StringComparator.EQUALS_ENFORCE_CASE);
LegacySearchFieldPathExpression fieldPathExpression2 = new
LegacySearchFieldPathExpression();
fieldPathExpression2.setSearchKeyPath("lineItem.type");
fieldCriterion2.setFieldPath(fieldPathExpression2);
fieldCriterion.setValue("FEE");
searchCriteria.getCriteria().add(fieldCriterion2);
//the following section defines a third fieldCriterion item for filtering line
items associated with a given timekeeper
StringFieldCriterion fieldCriterion3 = new StringFieldCriterion();
fieldCriterion.setComparator(StringComparator.EQUALS_ENFORCE_CASE);
LegacySearchFieldPathExpression fieldPathExpression3 = new
LegacySearchFieldPathExpression();
fieldPathExpression3.setSearchKeyPath("lineItem.timekeeper.uniqueKey");
fieldCriterion3.setFieldPath(fieldPathExpression3);
fieldCriterion.setValue("CONT_1010");
searchCriteria.getCriteria().add(fieldCriterion3);
List<Invoice> invoices =
invoiceRepository.readInvoicesByCriteria(searchCriteria, 100,
getPropertiesToRead());
}
private List<String> getPropertiesToRead() {
List<String> properties = new List<String>();
properties.add("numberString");
properties.add("invoiceDate");
properties.add("vendor");
properties.add("lineItems.version");
properties.add("lineItems.itemNumber");
properties.add("lineItems.serviceDate");
properties.add("lineItems.version");
properties.add("lineItems.originalRate");
properties.add("lineItems.originalQuantity");
properties.add("lineItems.originalDiscount");
properties.add("lineItems.originalTotal");
properties.add("lineItems.adjustedTotal");
return properties;
}
```

### 3.5.5.8.9 Deleting an Invoice

The following sample deletes an invoice record. You must get the unique key value for record to delete.

#### Code Snippet

```
//for this sample it is assumed that an invoice record exists with the unique key
equal to a String, uniqueKey
protected InvoiceRepository invoiceRepository;
invoiceRepository.deleteInvoice(uniqueKey);
```

### 3.5.5.9 Projects (Matters)

The following .JAVA code samples show how you can use TeamConnect® Enterprise Web Services for functions specific to project or custom object records, such as changing a project record's phase. Prerequisite to working with project records, the Custom Object Definition must already exist in TeamConnect . In addition, you need to get the following information before writing an application using Web Services:

- The project's Unique Code
- The unique keys of the project's phases
- The defined phase transitions for the project

Also see [Working with Categories](#) and [Working with Custom Fields](#) for additional information to gather before writing an application.

#### 3.5.5.9.1 Creating a Project With Assignees and a Custom Field

Creating a project record is done using the ProjectRepository object. For example, use the ProjectRepository's insertProject (ProjectCreate project) method. The resulting record's unique key is returned. The insertProject call will result in a SOAP/HTTP request and requires authentication.

When creating a project you can also add one or more assignees. From the type class, ProjectCreate, you need to use the getAssigneeCreates() method (see the example below for how to add items to the resulting list of ProjectAssigneeCreate).

Before you can start, you must get the unique key for the target assignee role and the unique key for the user to add as assignee either from the TeamConnect Designer GUI or from the database.

**Note:** After you understand how to add assignees to a project, you can apply a similar concept to the procedure for adding relations to a project.

#### Code Snippet

```
//in the sample below, basic require fields are commented but it is recommended to
verify whether additional required fields have been defined from the TeamConnect
custom object definition or from the responsible solution developer
protected ProjectRepository projectRepository;
ProjectCreate project = createProject();
projectRepository.insertProject(project);
```

```
private ProjectCreate createProject() throws Exception {
    Date date = new Date();
    //the EntityTypeUniqueKey is an object's 4-character unique code; look this
    value up either in the Designer area Object Definitions list or from the
    database
    String Entity_Type_Unique_Key = "DISP";
    ProjectCreate project = new ProjectCreate();
    //entityTypeUniqueKey is a required field
    project.setEntityTypeUniqueKey(Entity_Type_Unique_Key);
    //name is a required field
    project.setName("Small Claims Forster vs. Gloucester");
    //idNumber is a required field
    project.setIdNumber("Dispute_1001");
    project.setClosedOn(date);
    //the next three lines add a category to the project
    Category cat = new Category();
    cat.setUniqueKey("DISP EXTE");
    project.getCategories().add(cat);
    //the next four lines create a placeholder for an existing Text custom field,
    identify the known custom field name (statusSummary), set the custom field
    value, and add the custom field placeholder to its parent category
    TextCustomField statusSummary = new TextCustomField();
    statusSummary.setFieldName("statusSummary");
    statusSummary.setValue("Dispute has been resolved. Settlement reached.");
    cat.getTextCustomFields().add(statusSummary);
    project.getAssigneeCreates().addAll(getAssigneeCreates());
    return project;
}

public List<ProjectAssigneeCreate> getAssigneeCreates() {
    Date date = new Date();
    List<ProjectAssigneeCreate> assignees = new List<ProjectAssigneeCreate>();
    ProjectAssigneeCreate assignee1 = new ProjectAssigneeCreate();
    assignee1.setRoleUniqueKey("RPRT");
    assignee1.setAssignedOn(date);
    assignee1.setUnassignedOn(date);
    assignee1.setUserUniqueKey("USER9614");
    assignee1.setActive(True);
    assignees.add (assignee1);
    ProjectAssigneeCreate assignee2 = new ProjectAssigneeCreate();
    assignee2.setRoleUniqueKey("RPRT");
    assignee2.setAssignedOn(date);
    assignee2.setUnassignedOn(date);
    assignee2.setUserUniqueKey("USER9630");
    assignee2.setActive(False);
    assignees.add (assignee2);
    return assignees;
}
```

```
}
```

### 3.5.5.9.2 Updating a Project by Adding an Embedded Object Record

Updating a project is similar to updating other record types, such as contacts. You need to use the ProjectRepository's updateProject method. In addition, from the type class, ProjectUpdate, use provided methods to identify the target record and update its property values. Before you can start, you need to know the target project record's unique key.

You can add one or more embedded object records either at the time you update or create a project record. The example below illustrates how to add an embedded object record (Damage) to an existing project record (Dispute).

**Note:** Adding an embedded object record to a project is analogous to adding an assignee or relation to a project.

#### Code Snippet

```
protected ProjectRepository projectRepository;
ProjectUpdate project = updateProject();
projectRepository.updateProject(project);
private ProjectUpdate updateProject() throws Exception {
    String date = String.valueOf(new Date().getTime());
    //the EntityTypeUniqueKey is an object's 4-character unique code; look this
    value up either in the Designer area Object Definitions list or from the
    database
    String Entity_Type_Unique_Key = "DISP";
    ProjectUpdate project = new ProjectUpdate();
    //the next line identifies which record to update by its unique key
    project.setUniqueKey("DISP_2468");
    project.setEntityTypeUniqueKey(Entity_Type_Unique_Key);
    project.setName("Small Claims Forster vs. Gloucster");
    project.setClosedOn(date);
    project.setIdNumber("Dispute_1001");
    //the next three lines add a category to the project
    Category cat = new Category();
    //the category must already be created in TeamConnect ; you need to get the
    unique key from the TeamConnect GUI or database
    cat.setUniqueKey("DISP EXTE");
    project.getCategories().add(cat);
    project.getEmbeddedEntityCreates().add(getEmbeddedEntityCreates());
    return project;
}
public EmbeddedEntityCreate getEmbeddedEntityCreates() {
    Date date = new Date();
    // List<EmbeddedEntityCreate> embeddedEntities = new
    List<EmbeddedEntityCreate>();
    EmbeddedEntityCreate embeddedEntity = new EmbeddedEntityCreate();
```

```
embeddedEntity.setName("Damage_1002");
//the embedded object or project object definition must already exist in
TeamConnect ; you need to get the embedded object's 4-character unique key
(Unique Code) from the TeamConnect GUI or database
embeddedEntity.setEntityTypeUniqueKey("DAMA");
embeddedEntity.setContactUniqueKey("cont_1004");
embeddedEntity.setIdNumber("Damage_1002");
Category catEmbedded = new Category();
//the embedded object's category must already exist in TeamConnect; you need to
get the corresponding category's unique key from the TeamConnect GUI or
database
catEmbedded.setUniqueKey("DAMG");
embeddedEntity.getCategories().add(catEmbedded);
return embeddedEntity;
}
```

### 3.5.5.9.3 Changing a Project's Phase

When changing a project's phase, you must have already gathered the record's unique key as well as the unique key of the target phase to transition to. The ProjectRepository changePhase method will be validated against existing Phase Transition definitions for the project object definition.

#### Code Snippet

```
proj.Repository.changePhase("DISP_1004", "OPNN_OPEN");
```

### 3.5.5.9.4 Reading Projects

Reading a project record is similar to reading a contact record. A main difference would be when you request the categories property in the resulting project record's returned property values. By default, the returned list of categories also stores all property values associated with all project categories.

#### Code Snippet

```
//for this example, it is assumed that you already know the target project record's
unique key (where uniqueKey is a String variable)
protected ProjectRepository projectRepository;
public void readProject() throws Exception {
    Project project = (Project)
    projectRepository.readProject(uniqueKey, getPropertiesToRead());
}
private List<String> getPropertiesToRead() {
    List<String> properties = new List<String>();
    properties.add("name");
    properties.add("idNumber");
    properties.add("currentPhase.storedValue");
    properties.add("assignees");
    properties.add("categories");
}
```



```
        return properties;
    }
}
```

#### 3.5.5.9.5 Reading Child Projects for a Record

The sample below illustrates how you can read child projects of a project record by the child project type (project's 4-character unique code).

##### Code Snippet

```
// in this sample, it is assumed there is an existing parent project record with a
unique key String, parentKey
// it is also assumed that there are one or more child project records (where the
custom object definition's unique code is "CLAI")
protected ProjectRepository projectRepository;
public void readChildProjectsForEntityType() throws Exception {
    // in the list of child Projects returned, only the name properties will be
    returned
    List<Project> children =
    projectRepository.readChildProjectsForEntityType(parentKey, "CLAI",
    Arrays.asList("name"));
}
}
```

#### 3.5.5.9.6 Searching for Projects

Searching for projects is similar to searching for other object types. The following example illustrates how to search among projects for the following criteria: a given project phase, and given main assignee (by their user name).

##### Code Snippet

```
protected ProjectRepository projectRepository;
public void searchProjects() throws Exception {
    StringFieldCriterion fieldCriterion1 = new StringFieldCriterion();
    fieldCriterion.setComparator(StringComparator.EQUALS_ENFORCE_CASE);
    LegacySearchFieldPathExpression fieldPathExpression1 = new
    LegacySearchFieldPathExpression();
    fieldPathExpression1.setSearchKeyPath("currentPhase.storedValue");
    fieldCriterion1.setFieldPath(fieldPathExpression1);
    //need clarification if unique key or name
    //if unique key, then filtering for projects that are in phase, Closed
    fieldCriterion1.getValues().add("CLOS");
    FieldSearchClause searchCriteria = new FieldSearchClause();
    searchCriteria.setOperator(LogicOperator.AND);
    searchCriteria.getCriteria().add(fieldCriterion1);
    //the following section defines a second fieldCriterion item for
    StringFieldCriterion fieldCriterion2 = new StringFieldCriterion();
    fieldCriterion2.setComparator(StringComparator.EQUALS);
}
```

```
LegacySearchFieldPathExpression fieldPathExpression2 = new
LegacySearchFieldPathExpression();
fieldPathExpression2.setSearchKeyPath("mainAssignee.user.username");
fieldCriterion2.setFieldPath(fieldPathExpression2);
fieldCriterion.getValue().add("JohnDoe");
searchCriteria.getCriteria().add(fieldCriterion2);
List<Project> projects =
projectRepository.readProjectsByCriteria(searchCriteria, 100,
getPropertiesToRead());
}
private List<String> getPropertiesToRead() {
List<String> properties = new List<String>();
properties.add("name");
properties.add("idNumber");
return properties;
}
```

#### 3.5.5.9.7 Deleting a Project

The following sample deletes a project record. You must get the unique key value for record to delete.

#### Code Snippet

```
//for this sample it is assumed that a project record exists with the unique key
equal to a String, uniqueKey
protected ProjectRepository projectRepository;
projectRepository.deleteProject(uniqueKey);
```

#### 3.5.5.10 Involved Records

An involved is a unique record typically associated with a matter or project.

##### 3.5.5.10.1 Inserting Involved records

Inserts or creates an involved party record (associated with a project/matter). Often, an involved will be created and added to a project record at the same time as the project creation. This example assumes a project has already been created.

#### Code Snippet

```
protected InvolvedRepository involvedRepository;
String uniqueKey = Repository.insertInvolved(involved);
private void createInvolved() {
InvolvedCreate involved = new InvolvedCreate();
//the following are required fields
//in the following line, "DINV" is the unique code of the Involved object
definition, which is a child of the related Dispute (project) object definition
involved.setEntityTypeUniqueKey("DINV");
```

```
//in the following line, "DISP_2008" is the existing dispute record unique key
involved.setProjectUniqueKey("DISP_2008");
//in the following line, "CONT_5497" corresponds to the unique key of the
contact record for the person you are adding as the involved to a project
involved.setContactUniqueKey("CONT_5497");
involved.setActive(true);
Category cat = new Category();
//the next two lines identify a category and then add the category to the
involved record; this category corresponds to the involved party's default role
cat.setUniqueKey("DINV_MNGR");
involved.getCategories().add(cat);
return involved;
}
```

#### 3.5.5.10.2 Updating Involved records

Updating an involved is similar to updating other record types, such as contacts, except that you will be updating only the involved record's categories, custom field values, or clearing existing property values. You need to use the `InvolvedRepository`'s `updateInvolved` method. Before you can start, you need to know the target record's unique key.

#### Code Snippet

```
private InvolvedRepository involvedRepository;
Date date = new Date();
InvolvedUpdate involved = new InvolvedUpdate();
Category cat = new Category();
//the next two lines identify a category and then add the category to the involved
record
cat.setUniqueKey("INPA_ARBI");
involved.getCategories().add(cat);
Category cat2 = new Category();
involved.getCategories().add(cat2);
cat2.setUniqueKey("INPA_MEDI");
//the next four lines create a placeholder for an existing DateTime custom field,
identify the known custom field name (InactiveDateDI), set the custom field value,
and add the custom field placeholder to its parent category
DateTimeCustomField dateTime1 = new DateTimeCustomField();
dateTime1.setFieldName("InactiveDateDI");
dateTime1.setValue(date);
cat2.getDateTimeCustomFields().add(dateTime1);
DateTimeCustomField dateTime2 = new DateTimeCustomField();
dateTime2.setFieldName("InvolvementDateDI");
dateTime2.setValue(date);
cat2.getDateTimeCustomFields().add(dateTime2);
involvedRepository.updateInvolved(involved);
```

### 3.5.5.10.3 Reading Involved records

Reading an involved record is similar to reading a contact record.

#### Code Snippet

```
//for this sample, it is assumed there exists an involved record with the unique
key equal to a String, invUniqueKey
private InvolvedRepository involvedRepository;
private List<String> getPropertiesToRead() {
    List<String> props = new List<String>();
    props.add("uniqueKey");
    props.add("contact");
    props.add("project");
    props.add("project.name");
    props.add("contact.uniqueKey");
    return props;
}
public void test_readInvolved throws Exception {
    // Search the involved record
    Involved involved = involvedRepository.readInvolved(invUniqueKey,
getPropertiesToRead());
}
```

### 3.5.5.10.4 Reading Involved Parties for a Project Record

The following sample gets requested properties for involved party records that are associated with a project record, given the unique key of the project record.

#### Code Snippet

```
readInvolvedsForProject
private InvolvedRepository involvedRepository;
private List<String> getPropertiesToRead() {
    List<String> props = new List<String>();
    props.add("uniqueKey");
    props.add("contact");
    props.add("project");
    props.add("project.name");
    props.add("contact.uniqueKey");
    return props;
}
public void test_readInvolvedsForProject throws Exception {
    // Search the involved records for a project
    //in the following operation, "DISP_5941" is the unique key of the project
record whose involved party records will be returned
    List<Involved> involvedParties =
involvedRepository.readInvolvedsForProject("DISP_5941",
```

```
        getPropertiesToRead());  
    }
```

### 3.5.5.10.5 Searching Involved Party Records

The following sample searches involved party records by entity type and contact last name.

#### Code Snippet

```
//for this sample, it is assumed there exists an involved record  
with the unique key equal to a String, invUniqueKey  
private InvolvedRepository involvedRepository;  
private List<String> getPropertiesToRead() {  
    List<String> props = new List<String>();  
    props.add("uniqueKey");  
    props.add("person.lastName");  
    props.add("person.firstName");  
    props.add("project");  
    props.add("project.name");  
    props.add("contact.uniqueKey");  
    return props;  
}  
  
public void searchInvolveds() throws Exception {  
    StringFieldCriterion fieldCriterion1 = new StringFieldCriterion();  
    fieldCriterion.setComparator(StringComparator.EQUALS_ENFORCE_CASE);  
    LegacySearchFieldPathExpression fieldPathExpression1 = new  
    LegacySearchFieldPathExpression();  
    fieldPathExpression1.setSearchKeyPath("entityType.uniqueCode");  
    fieldCriterion1.setFieldPath(fieldPathExpression1);  
    //filtering for involved party records defined by the Involved object  
    definition with the TeamConnect unique code, "DINV"  
    fieldCriterion.getValue().add("DINV");  
    FieldSearchClause searchCriteria = new FieldSearchClause();  
    searchCriteria.setOperator(LogicOperator.AND);  
    searchCriteria.getCriteria().add(fieldCriterion1);  
    //the following section defines a second fieldCriterion item for  
    StringFieldCriterion fieldCriterion2 = new StringFieldCriterion();  
    fieldCriterion.setComparator(StringComparator.EQUALS);  
    LegacySearchFieldPathExpression fieldPathExpression2 = new  
    LegacySearchFieldPathExpression();  
    fieldPathExpression2.setSearchKeyPath("contact.person.lastName");  
    fieldCriterion2.setFieldPath(fieldPathExpression2);  
    fieldCriterion.getValue().add("Doe");  
    searchCriteria.getCriteria().add(fieldCriterion2);  
    public void test_readInvolvedsByCriteria throws Exception {  
        // Search the involved records
```

```
    Involved involved = involvedRepository.readInvolvedsByCriteria(searchCriteria,
    100,
    getPropertiesToRead());
}
```

#### 3.5.5.10.6 Deleting an Involved

The following sample deletes an involved party record. You must get the unique key value for record to delete.

#### Code Snippet

```
//for this sample it is assumed that an involved party record exists with the
unique key equal to a String, uniqueKey
protected InvolvedRepository involvedRepository;
involvedRepository.deleteInvolved(uniqueKey);
```

#### 3.5.5.11 Tasks

The following .JAVA code samples show how you can use TeamConnect® Enterprise Web Services for functions related to task records.

##### 3.5.5.11.1 Creating tasks

The following sample creates a task record. Prerequisite to working with task records, you must also get the unique key value for an existing TeamConnect user record to add as a task assignee.

#### Code Snippet

```
protected TaskRepository taskRepository;
private String createTask() throws Exception {
    Date date = new Date();
    TaskAssigneeCreate taskAssignee = new TaskAssigneeCreate();
    taskAssignee.setUserUniqueKey("user_1003");
    taskAssignee.setAssignedOn(date);
    TaskCreate task = new TaskCreate();
    //the shortDescription field is required
    task.setShortDescription("subject");
    task.setActualHours(new BigDecimal("8.00"));
    task.setCurrentAssignee(taskAssignee);
    task.setPriority(TaskPriority.HIGH);
    task.setEstimatedHours(new BigDecimal("9.00"));
    task.setStartDate(date);
    task.setNote("A_task_note_" + date.getTime());
    task.setRateAmount(new BigDecimal("50"));
    return taskRepository.insertTask(task);
}
```

### 3.5.5.11.2 Updating tasks

The following sample updates a task record with the actual hours to complete the task, a newly calculated total amount, the date the task was completed, and percent completion value.

#### Code Snippet

```
//for this sample, it is assumed that a task record already exists with a unique
key String equal to uniqueKey
protected TaskRepository taskRepository;
public void test_updateTask() throws Exception {
    Date date = new Date();
    TaskUpdate taskUpdate = new TaskUpdate();
    //use the setUniqueKey method to identify the target task record to update by its
    unique key
    taskUpdate.setUniqueKey(uniqueKey);
    taskUpdate.setActualHours(new BigDecimal("15.00"));
    taskUpdate.setRateAmount(new BigDecimal("50"));
    taskUpdate.setCompletedDate(date);
    taskUpdate.setCompletedPercent(100.00);
    taskRepository.updateTask(taskUpdate);
}
```

### 3.5.5.11.3 Reading tasks

The following sample reads a task record and returns the specified property values.

#### Code Snippet

```
protected TaskRepository taskRepository;
private List<String> getPropertiesToRead() {
    List<String> props = new List<String>();
    //the list of all available properties displays below but to increase
    efficiency in your searches, you can omit unnecessary properties from your
    application and those values will not be returned
    props.add("shortDescription");
    props.add("uniqueKey");
    props.add("currentAssignee");
    props.add("categories");
    props.add("createdOn");
    props.add("modifiedBy");
    props.add("version");
    props.add("modifiedOn");
    props.add("contact");
    props.add("project");
    props.add("activityItem");
    props.add("priority");
    props.add("workStatus");
}
```

```
        props.add("postingStatus");
        props.add("dueDate");
        props.add("startDate");
        props.add("completedDate");
        props.add("completedPercent");
        props.add("actualHours");
        props.add("estimatedHours");
        props.add("rateAmount");
        props.add("totalAmount");
        props.add("note");
        return props;
    }
    private Task readTask() throws Exception {
        Task readTask = taskRepository.readTask(uniqueKey, getPropertiesToRead());
        return readTask;
    }
}
```

#### 3.5.5.11.4 Searching tasks

The following sample searches for one or more task records and returns the specified property values. For this sample, it is assumed there is one or more tasks already existing with a subject String field value equal to subject.

#### Code Snippet

```
protected TaskRepository taskRepository;
private List<Task> test_readTasksByCriteria() throws Exception {
    // Criterion for task searching
    StringFieldCriterion fieldCriterion = new StringFieldCriterion();
    fieldCriterion.setComparator(StringComparator.EQUALS_ENFORCE_CASE);
    LegacySearchFieldPathExpression fieldPathExpression = new
    LegacySearchFieldPathExpression();
    fieldPathExpression.setSearchKeyPath("shortDescription");
    fieldCriterion.setFieldPath(fieldPathExpression);
    fieldCriterion.getValue().add(subject);
    FieldSearchClause searchCriteria = new FieldSearchClause();
    searchCriteria.setOperator(LogicOperator.AND);
    searchCriteria.getCriteria().add(fieldCriterion);
    tasks = taskRepository.readTasksByCriteria(searchCriteria, 100,
    getPropertiesToRead());
}
private List<String> getPropertiesToRead() {
    List<String> props = new List<String>();
    //the list of all available properties displays below but to increase
    efficiency in your searches, you can omit unnecessary properties from your
    application and those values will not be returned
    props.add("shortDescription");
}
```



```
        props.add("uniqueKey");
        props.add("currentAssignee");
        props.add("categories");
        props.add("createdOn");
        props.add("modifiedBy");
        props.add("version");
        props.add("modifiedOn");
        props.add("contact");
        props.add("project");
        props.add("activityItem");
        props.add("priority");
        props.add("workStatus");
        props.add("postingStatus");
        props.add("dueDate");
        props.add("startDate");
        props.add("completedDate");
        props.add("completedPercent");
        props.add("actualHours");
        props.add("estimatedHours");
        props.add("rateAmount");
        props.add("totalAmount");
        props.add("note");
        return props;
    }
```

#### 3.5.5.11.5 Posting tasks

The following sample posts a task with given unique key.

##### **Code Snippet**

```
String taskUniqueKey = "TASK_1001";
protected TaskRepository taskRepository;
taskRepository.postTask(taskUniqueKey);
```

#### 3.5.5.11.6 Voiding tasks

The following sample voids a task with given unique key.

##### **Code Snippet**

```
String taskUniqueKey = "TASK_1001";
protected TaskRepository taskRepository;
taskRepository.voidTask(taskUniqueKey);
```

### 3.5.5.11.7 Reassigning tasks

The following sample reassigns a task from a current assignee (user) to a different assignee.

#### Code Snippet

```
//for this sample, it is assumed that a task record already exists with a unique
key String equal to uniqueKey
protected TaskRepository taskRepository;
private test_reassign() throws Exception {
    //reassign a task identified by its unique key to a new assignee, identified by
    his corresponding user unique key ("user_1005")
    taskRepository.reassign(uniqueKey, "user_1005");
}
```

### 3.5.5.11.8 Deleting tasks

The following sample deletes an existing task.

#### Code Snippet

```
//for this sample it is assumed that a task record exists with a unique key String
equal to uniqueKey
protected TaskRepository taskRepository;
public void test_deleteTask() throws Exception {
    taskRepository.deleteTask(uniqueKey);
}
```

## 3.5.5.12 User Accounts

The following .JAVA code samples show how you can use TeamConnect® Enterprise Web Services for functions related to user records.

### 3.5.5.12.1 Creating user accounts

The following sample creates a user account record. Prerequisite to working with user records, a corresponding contact (type person) record must already exist in TeamConnect . In addition, you need to get the following information before writing an application using Web Services: the unique key of the related contact record.

#### Code Snippet

```
// this sample assumes a contact record (type person) already exists with the
unique key (String) value equal to contactUniqueKey
protected UserAccountRepository userAccountRepository;
private String insertUserAccount() throws Exception {
    String userAccountName = "John_Doe";
    UserAccountCreate userAccount = new UserAccountCreate();
    //username is a required field
    userAccount.setUsername(userAccountName);
}
```

```
//password is a required field
userAccount.setPassword("p@ssw0rd");
//contactUniqueKey is a required field
userAccount.setContactUniqueKey(contactUniqueKey);
userAccount.setShortDescription("new user hired 10/01/2008, business
administrator");
userAccount.setUserType(UserType.NORMAL);
userAccount.setActive(true);
String uniqueKey = userAccountRepository.insertUserAccount(userAccount);
return uniqueKey;
}
```

#### 3.5.5.12.2 Updating user accounts

The following sample updates an existing user account record by changing the password, changing the user type, and inactivating the user account.

**Note:** *In actual business usage, if you are inactivating a user account, it is unlikely you would need to change other properties but the samples are provided for you to use as necessary.*

#### Code Snippet

```
// this sample assumes a user record already exists with the unique key (String)
value equal to uniqueKey
protected UserAccountRepository userAccountRepository;
public void test_updateUserAccount() throws Exception {
    UserAccountUpdate userUpdate = new UserAccountUpdate();
    //set the uniqueKey with the String value of the target user record to identify
    which record you are updating
    userUpdate.setUniqueKey(uniqueKey);
    //update the existing password
    userUpdate.setPassword("newP@ssw0rd");
    //update the existing userType
    userUpdate.setUserType(UserType.LIMITED);
    //inactive the user account
    userUpdate.setActive(false);
    userAccountRepository.updateUserAccount(userUpdate);
}
```

#### 3.5.5.12.3 Reading user accounts

The following sample reads the specified properties of a user account record.

#### Code Snippet

```
//define the list of user record properties to return with the readUserAccount
results
protected UserAccountRepository userAccountRepository;
```

```
private List<String> getProperties() {
    List<String> properties = new List<String>();
    properties.add("username");
    properties.add("active");
    properties.add("password");
    properties.add("contact");
    properties.add("shortDescription");
    properties.add("uniqueKey");
    properties.add("userType");
    return properties;
}

public UserAccount test_readUserAccount() throws Exception {
    //it is assumed that a user record exists with the unique key equal to String,
    uniqueKey
    UserAccount searchedUser = userAccountRepository.readUserAccount(uniqueKey,
    getProperties());
    return searchedUser;
}
```

#### 3.5.5.12.4 Searching user accounts

The following sample searches users by given search criteria and returns the specified properties of the resulting user account records.

#### Code Snippet

```
protected UserAccountRepository userAccountRepository;
private List<UserAccount> test_readUserAccountsByCriteria() throws
Exception {
    // Criterion for user account searching
    StringFieldCriterion fieldCriterion = new StringFieldCriterion();
    fieldCriterion.setComparator(StringComparator.EQUALS_ENFORCE_CASE);
    LegacySearchFieldPathExpression fieldPathExpression = new
    LegacySearchFieldPathExpression();
    fieldPathExpression.setSearchKeyPath("username");
    fieldCriterion.setFieldPath(fieldPathExpression);
    fieldCriterion.getValue().add("JohnDoe");
    FieldSearchClause searchCriteria = new FieldSearchClause();
    searchCriteria.setOperator(LogicOperator.AND);
    searchCriteria.getCriteria().add(fieldCriterion);
    users = userAccountRepository.readUserAccountsByCriteria(searchCriteria, 100,
    getPropertiesToRead());
}

private List<String> getProperties() {
    List<String> properties = new List<String>();
    properties.add("username");
    properties.add("active");
```

```
properties.add("password");
properties.add("contact");
properties.add("shortDescription");
properties.add("uniqueKey");
properties.add("userType");
return properties;
}
```

#### 3.5.5.12.5 Deleting user accounts

The following sample deletes a user account record with a given unique key.

#### Code Snippet

```
//for this sample it is assumed that a user record exists with a unique key String
equal to uniqueKey
protected UserAccountRepository userAccountRepository;
public void test_deleteUserAccount() throws Exception {
    userAccountRepository.deleteUserAccount(uniqueKey);
}
```

## 3.5.6 Glossary

### A

#### ***authentication***

In the TeamConnect Web Service environment, authentication is the process by which a client program establishes trusted status with the TeamConnect system (database). This is accomplished by providing a valid TeamConnect user name and password in the security header section of each Web Service request.

For more information, see [SOAP Message Security Header](#).

### H

#### ***HTTP (HyperText Transfer Protocol)***

Network communications protocol used between Web Service clients and servers. In the TeamConnect Web Service environment, a developer programs in his or her desired programming language. Then he or she uses a 3rd party SOAP toolkit to transform the client program into packaged Web Service requests in SOAP messages transferred over HTTP.

### O

#### ***object***

In the TeamConnect Web Service environment, an object may refer to the following: an instance of a class provided in the Web Service client API, a TeamConnect system object or custom object.

## R

### ***record***

In the TeamConnect Web Service environment, a record refers to an instance of a TeamConnect system object or custom object in the TeamConnect database.

### ***repository***

In the TeamConnect Web Service environment, each entity has a corresponding repository class that supports operations for working with records of that entity type. For example, the ContactRepository provides methods for inserting a contact, updating a contact, reading a contact, searching contacts, deleting a contact from a TeamConnect system.

### ***request***

In the TeamConnect Web Service environment, each client program's repository method call will be transformed by the 3rd party SOAP toolkit into a request to the corresponding Web Service.

## S

### ***SOAP (Simple Object Access Protocol)***

Protocol for exchanging Web Service requests (from the client to server) and responses (from the server to the client). In the TeamConnect Web Service environment, a 3rd party SOAP toolkit will transform the client program to corresponding SOAP messages. On the server, a corresponding framework will transform the SOAP message to the appropriate programming language (and transform server responses from a programming language to a SOAP message).

Note: Mitrtech does not provide a 3rd party SOAP toolkit. You need to download one from the internet.

## W

### ***Web Service***

Web application programming interface (API) hosted on your TeamConnect server. In the TeamConnect Web Service environment, the Web Service API supports most operations that a TeamConnect end-user could perform in the user interface. Some operations that a TeamConnect administrator could perform in the user interface are also supported.

## X

### ***XML (extensible markup language)***

In the TeamConnect Web Service environment, source files (.WSDL, .XSD) use XML format. In addition, generated SOAP messages use XML format. Generally developers do not need to create files directly in XML format. Developers should be expected to code in their desired programming language and use a 3rd party SOAP toolkit to auto-generate the corresponding SOAP (XML) messages.

# Index

## - . -

.css 545  
.lss 545

## - 4 -

4.0.8 1169  
4.2 511

## - A -

accessibility 429  
    assistive technology 430  
    enabling 429  
    keyboard shortcuts 431  
accessibility mode 49  
account rights 645  
    additional 646  
    basic 645  
AccountRepository 1935  
    activateAccount 1965  
    allocateMoney 1966  
    deactivateAccount 1966  
    deleteAccount 1967  
    insertAccount 1962  
    method details 1962  
    readAccount 1963  
    readAccountsByCriteria 1964  
    readChildAccounts 1965  
    readRecentlyViewedAccounts 1968  
    transferMoney 1966  
    updateAccount 1963  
    withdrawMoney 1967  
accounts 239  
    activating 250  
    adding funds 250  
    best practices 462  
    categories 247  
    child 249  
    creating 240  
    deactivating 250

    editing 240  
    General page 241  
    history 265  
    parent 248  
    posting criteria 253  
    relationships 248  
    rights 463  
    subtracting funds 250  
    transactions 264  
    transferring funds 251  
    viewing 240  
accounts object model 1535  
    EAcctDetail 1556  
    EAcctGroupAccess 1560  
    EAcctUserAccess 1557  
    JTranDetail 1551  
    RTransaction 1554  
    TAccount 1535  
Action buttons 551  
actions 1146  
adding documents 365  
address book rights 652  
    basic 652  
address books 147  
    adding contacts 151  
    contacts 149  
    creating 149  
    private 152  
    public 152  
    removing contacts 152  
    subscribing 148  
adjusting invoices  
    line item history 345  
    line items 322  
    new amounts 344  
    summaries 315  
    summary history 345  
admin setting rights 696  
Admin Settings 538, 545  
    About page 604  
    Approval page 598  
    Calendar page 567  
    Connections page 602  
    Documents page 569  
    Email page 590  
    general 459  
    General page 541  
    general settings 539

- Admin Settings 538, 545
  - History page 603
  - Invoices page 596
  - localization 555
  - Maintenance page 600
  - Notifications page 592
  - regions 555
  - Regions page 555
  - Reporting page 576
  - Search Results page 573
  - Security page 577
  - User Interface page 552
- administrator tools 614
  - Scheduled Actions Monitor 615
- administrators 458
  - accounts 461
  - system 538
- AFA rules 1756
  - creating pages 1760
  - updating 1760
- All tab 64
- appeal 288
- appeals 288
- appeals invoices 288
- appender rights 714
- appenders 783
  - audit 797
  - audit loggers 802
  - clearing logs 785
  - default 784
  - file layouts 789
  - files 786
  - HTML layout 792
  - loggers 799
  - pattern layout 789
  - rolling over logs 786
  - SMTP 793
  - socket 794
  - system 796
  - system loggers 799
  - viewing logs 784
  - XML layout 792
- appointment calendar 49
- appointment rights 648
  - additional 650
  - basic 648
- AppointmentRepository 1937
  - deleteAppointment 1971
  - insertAppointment 1968
  - method details 1968
  - readAppointment 1970
  - readAppointmentsByCriteria 1971
  - readRecentlyViewedAppointment 1972
  - updateAppointment 1970
- appointments 212
  - attendees 222
  - categories 221
  - creating 217
  - editing 217
  - General page 217
  - history 223
  - opening 216
  - outlook 216
  - resources 223
  - Scheduler 219
- appointments object model 1516
  - EApptDetail 1528
  - EApptGroupAccess 1532
  - EApptUserAccess 1530
  - JApptAttendee 1524
  - JApptResource 1526
  - LApptArealtem 1523
  - LApptResourceType 1527
  - TAppointment 1516
- approval 428, 1146
  - approval admin settings 598
  - approval history 389
  - approval processes
    - email notifications 394
    - requests 395
  - approval requests 395
    - approving 403
    - canceling 399
    - completed 398
    - managing 401
    - pending approvals 401
    - reassigning 422
    - rejecting 403
    - sending for review 409
    - viewing 396
  - approval rules 1111
  - approval schedule 389
  - approvals 392
  - approvers 1146
  - approving requests 403



approving requests 403  
     through email 408  
     within records 406  
 assignee roles 851  
     adding 851  
     deleting 854  
     editing 853  
 assistive technology 430  
     JAWS 430  
 Audit Logging page 609  
     clearing 611  
     turning off 613  
     turning on 613  
 audit rules 1114  
     default description 1116  
     history 1117  
     tracking 1117  
     triggers 1115  
     updates 1115  
 authenticating users 36  
 authentication 619  
     custom plug-in 635  
     LDAP plug-in 625  
     login URLs 619  
     NTLM plug-in 620  
 automatically created structure 722

## - B -

Backward Compatibility Settings 742  
 batch entries 95  
     deleting 96  
     editing 96  
     primary entries 95  
 blocks 922  
     clearing cache 951  
     creating 923  
     creating XML files 926  
     defining 938  
     reports 932  
     sample XML file 935  
     tags 927  
     troubleshooting 942  
     uploading XML files 938  
 budget requests 347  
     adjusting 350  
     canceling 352

deleting 352  
 editing 350  
 handling 353  
 submitting 350  
 vendors 350  
 building reports 186  
     Chart page 202  
     Data page 187  
     data series 187  
     General page 186  
     Schedule page 208  
     scheduling 208  
     Security page 210  
 business object reports  
     running 56  
 BusinessObjects 540

## - C -

cached reports 180  
 calendar 212  
     viewing 213  
 categories 863  
     activating 873  
     adding 871  
     custom object 870  
     deleting 874  
     editing 871  
     end-user interface 867  
     inactivating 871  
     system object 869  
     tree position 868  
     using 864  
     viewing 865  
 categories for API 1847  
     adding categories 1848  
     checking for categories 1850  
     default categories 1850  
     deleting categories 1848  
     returning categories 1849  
 category rights 471  
     prerequisites for all 473  
     prerequisites for individuals 474  
 charts 202  
 child accounts 249  
 clearing cache 951  
 client application components 2045

- client application components 2045
  - client proxy 2043
  - security headers 2044
- collections 65
  - subscribing 67
- color profile 545, 545
- color profile css 545
- combining data series 198
  - best practices 201
  - join conditions 199
  - results display 199
  - sample 200
- common API functions 1857
  - current user 1861
  - system user 1859
- common object model 1665
  - JNote 1665
  - UGrupMember 1666
  - WObjdDetailField 1667
  - WObjdProjectInfo 1671
  - YDetailLookupItem 1676
  - YDetailLookupTable 1678
  - YGroup 1679
  - YRecentlyViewedRecord 1682
  - YUser 1683
  - YUserSubscribedCollection 1689
- conditional expressions 1187
- conditions 1187
  - creating 1188
  - logic 1196
  - operators 1190
  - right arguments 1190
  - routes 1196
- connections admin settings 601
- contact rights 652
  - additional 654
  - basic 652
- contact sweeper 759
  - custom tools 781
  - duplicates 766
  - editing 762
  - fixing 764
  - generating 760
  - logging 766
  - prerequisites 759
  - references 765
  - uploading 764
- ContactRepository 1939
  - deleteContact 1976
  - insertContact 1972
  - method details 1972
  - readContact 1974
  - readContactsByCriteria 1975
  - ReadRecentlyViewedContacts 1976
  - updateContact 1974
- contacts 127
  - address books 139
  - card view 129
  - categories 138
  - company 145
  - creating 130
  - details 136
  - duplicates 135
  - employees 145
  - general information 136
  - involvement 146
  - list view 129
  - rates 139
  - related 143
  - skills 144
  - territories 144
- contacts object model 1464, 1506
  - EContDetail 1510
  - EContGroupAccess 1514
  - EContUserAccess 1511
  - JContAddress 1482
  - JContDefaultRate 1498
  - JContEmail 1491
  - JContFax 1489
  - JContInetAddress 1493
  - JContPhone 1486
  - JContRate 1499
  - JContRelation 1501
  - JContSkill 1495
  - JContTerritory 1504
  - JDistMember 1508
  - LContAddressType 1485
  - LContEmailType 1492
  - LContFaxType 1490
  - LContInetAddressType 1494
  - LContPhoneType 1488
  - LContRelationType 1503
  - LContSkillType 1496
  - LContTerritoryType 1505
  - LCountryItem 1484

- contacts object model 1464, 1506
  - TContact 1464
- converting EasyDocs RTF files to templates 1343
- creating EasyDocs RTF file 1304
  - auto-mapping system fields 1313
  - filter@ merge fields 1318
  - if@ merge fields 1319
  - inserting filter@ merge fields 1318
  - inserting loop@ merge fields 1317
  - loop@ merge fields 1315
  - mapped custom fields 1314
  - mapped system fields 1311
  - merge field codes 1311
  - merge fields 1309
  - not using filter@ 1318
  - not using loop@ 1317
- creating records 84
  - batch entries 95
  - categories 97
  - copying 101
  - fields 89
  - importing data 105
  - saving 99
  - wizards 85
- creating wizards 1070
  - actions 1084
  - automated actions 1085
  - General tab 1072
  - initial actions 1086
  - Page Components tab 1076
  - page order 1101
  - page transition rules 1097
  - Page Transitions tab 1094
  - Pages tab 1074
  - parameters 1081
  - simple actions 1087
  - Wizards tab 1071
- CSM
  - admin setup 1754
  - designers 1750
  - first synchronization 1753
  - invoices 1753
  - new vendors 1753
  - prerequisites 1750
  - process setup 1752
  - restoring data 1764
  - setup 1750
  - uninstalling 1762
  - validation rules 1754
- CSM rules 1760
  - AFAs 1756
  - invoices 1754
  - rate request approval 1755
  - setting screen 1755
- custom action rules 1112
- custom authentication plug-in 635
  - default method 636
  - message screens 636
  - overriding authentication 636
- custom blocks 1804
  - sample 1806
- custom code 1197
  - rule component files 1197
  - rules 1200
  - start up classes 1199
  - wizard page actions 1203
  - wizard page transitions 1202
- custom designs
  - Design Import Tool 1211
  - Design Snapshot Tool 1209
  - migrating 1209
  - requirements 1209
  - troubleshooting 1228
- Custom field 511
- custom field and groups 511
- custom field rights 471
  - prerequisites for all 473
  - prerequisites for individuals 474
- custom field types 903
  - custom objects 907
  - Involved 906
  - lists 906
- custom fields 894
  - creating 920
  - Data Warehouse requirements 899
  - name requirements 899
  - required fields 902
  - tab 897
  - types 903
  - where to use 896
- custom fields for API 1850
  - getting values 1852
  - requirements 1851
  - setting values 1852
- custom hover 961

custom hover over 961  
 custom logo 545  
 custom lookup table rights 701  
 custom lookup tables 885
 

- activating 891
- adding 887
- deleting 888
- deleting items 892
- editing 889
- inactivating 889
- items 888
- viewing 886

 custom object fields 907
 

- data types 913
- Object Navigator 916
- qualifier syntax 913
- qualifiers 911
- search views 910

 custom objects 805
 

- assignee roles 851
- creating 822
- deleting 863
- embedded 812
- general information 824
- Involved 805
- naming patterns 834
- notifications 855
- phase transitions 846
- phases 842
- unique IDs 829

 custom pages 892
 

- blocks 922
- custom fields 894
- object views 943
- process 894
- views 893

 custom pages in API 1803
 

- blocks 1804
- screens 1804
- tools 1810

 custom record rights 681
 

- involved parties 685
- matters 681
- milestones 689
- projects 681

 custom scheduled actions 1813
 

- adding to TeamConnect 1813
- sample 1814

custom screens
 

- API sample 1806

 custom screens in API 1804  
 custom search 70
 

- additional fields 75
- executing 77
- operators 72
- results 76
- saving 77
- search criteria 70

 custom tool rights 698  
 custom tools 781
 

- API sample 1811
- configuring 782
- in API 1810
- uploading 782

 custom views 893, 961  
 customization 738
 

- custom messages 747
- Data Warehouse 748
- design 738
- sequence 744
- Setup 739
- SQL Server 748
- system settings 749
- user interface 739

## - D -

data series 187
 

- combining 198
- creating 190
- data 191
- filtering 193
- results display 196

 date format 45  
 design 738  
 design documents 715
 

- automatically created folders 722
- directory structure 715

 Design Import Tool 1211
 

- design upgrades 1212

 Design Snapshot Tool 1209
 

- creating snapshot 1210
- reverting 1211
- upgrade file 1211

 document admin settings 567

- document admin settings 567
  - file systems 568
  - version control 568
  - WebDAV 568
- document directory structure 715
  - Attachments folder 717
  - HTTPRoot folder 717
  - System folder 718
  - top level folder 716
  - Users folder 722
- document folders
  - creating 372
  - deleting 387
  - exporting 372
- Document Generator 1232
  - content 1253
  - headers 1238
  - user input 1250
- document rights 657
  - additional 658
  - basic 657
- document template tags 1256
  - tc:block 1279
  - tc:compute 1280
  - tc:conditional 1274
  - tc:data 1259
  - tc:date 1272
  - tc:detail 1264
  - tc:file 1273
  - tc:filter 1284
  - tc:input 1287
  - tc:loop 1266
  - tc:operand 1281
  - tc:search 1282
  - tc:user 1272
- document templates
  - basics 1233
  - completing 1296
  - content 1253
  - converting templates to XML 1297
  - creating headers 1297
  - creating tags 1296
  - creating user input fields 1250
  - displaying entry fields 1252
  - Document Generator 1232
  - document templates 1237
  - filter pages 1238
  - filtering sub-objects 1362
  - headers 1238
  - main filters 1246
  - nested tags 1288
  - nesting tags 1353
  - non-related object filters 1245
  - object attribute names 1291
  - record filters 1242
  - related object filters 1241
  - retrieving data from child objects 1359
  - retrieving data from contact fields 1355
  - retrieving data from custom fields 1358
  - retrieving data from Involved custom fields 1359
  - retrieving data from related objects 1357
  - samples 1353
  - sub-filters 1246
  - sub-object filters 1242
  - tags 1256
  - tc:conditional samples 1363
  - testing 1299
  - troubleshooting 1347
  - uploading 1299
  - user input 1250
  - writing text 1254
- DocumentRepository 1945
  - checkIn 1986
  - checkOut 1986
  - copyDocument 1988
  - createHyperlink 1980
  - createHyperlinkDefaultCategory 1980
  - createShortcut 1979
  - createSubFolder 1979
  - deleteDocument 1989
  - insertDocument 1977
  - method details 1976
  - moveDocument 1988
  - readChildDocuments 1984
  - readChildDocumentForName 1984
  - readDocument 1982
  - readDocumentFolderForDocumentOwner 1985
  - readDocumentsByCriteria 1982
  - readDocumentsByPath 1983
  - readFolderHierarchy 1990
  - ReadRecentlyViewedDocuments 1989
  - revet 1987
  - setDocumentAsRecentlyViewed 1989
  - undoCheckOut 1987
  - updateDocument 1981
- documents 355

- documents 355
    - adding 365
    - canceling 371
    - categories 386
    - checking in 370
    - checking out 370
    - collections 364
    - copying 373
    - deleting 387
    - downloading 379
    - editing 369
    - folders 372
    - General page 383
    - generating 380
    - history 386
    - managing 364
    - moving 376
    - printing 386
    - reverting 385
    - searching 358
    - shortcuts 377
    - versions 385
    - viewing 360
    - Web page links 378
    - Windows Explorer 382
  - documents object model 1649
    - EDocuDetail 1659
    - EDocuGroupAccess 1663
    - EDocuUserAccess 1660
    - JDocuContent 1657
    - TDocument 1649
    - YDocuContentType 1657
  - documents templates
    - inserting tags 1292
    - preparing RTF files 1292
  - drill-down reports 179
- E -**
- EAcctDetail 1556
  - EAcctGroupAccess 1560
  - EAcctUserAccess 1557
  - EApptDetail 1528
  - EApptGroupAccess 1532
  - EApptUserAccess 1530
  - EasyDocs 1300
    - converting RTF files to templates 1343
    - creating RTF file 1304
    - Data Mapping Tool 1320
    - deleting template files 1347
    - Document Templates folder 1320
    - functions 1301
    - group rights 1304
    - mapping RTF merge fields 1325
    - modifying data mapping 1346
    - modifying RTF files 1345
    - process 1302
      - step 1 1304
      - step 2 1324
      - step 3 1325
      - step 4 1343
      - step 5 1344
    - template files 1345
    - testing templates 1344
    - uploading RTF file 1324
  - EContDetail 1510
  - EContGroupAccess 1514
  - EContUserAccess 1511
  - editing documents 369
    - canceling 371
    - checking in 370
    - checking out 370
  - Editing items pending approval 1169
  - editing multiple records 86
  - editing records 86
    - locking 88
    - reassigning 104
  - EDocuDetail 1659
  - EDocuGroupAccess 1663
  - EDocuUserAccess 1660
  - EExpeDetail 1568
  - EExpeGroupAccess 1572
  - EExpeUserAccess 1570
  - EHistDetail 1642
  - EHistGroupAccess 1647
  - EHistUserAccess 1645
  - EInvcDetail 1625
  - EInvcGroupAccess 1635
  - EInvcUserAccess 1632
  - EInvDetail 1442
  - EInvGroupAccess 1446
  - EInvUserAccess 1444
  - ELitmDetail 1626
  - ELitmGroupAccess 1628

- ELitmUserAccess 1630
  - email admin settings 584
    - email notifications 585
    - IMAP server 586
  - email notifications 50
  - embedded objects 812
    - converting to child 861
    - creating 857
    - defining 860
    - displaying 856
    - list displays 855
  - EMileDetail 1457
  - EMileGroupAccess 1461
  - EMileUserAccess 1458
  - enable email approvals
    - email 392
  - enumerations 1786
  - EProjDetail 1428
  - EProjGroupAccess 1432
  - EProjUserAccess 1429
  - ETaskDetail 1585
  - ETaskGroupAccess 1589
  - ETaskUserAccess 1586
  - exchange rates 353
  - executing search 77
  - Expense Entry Tool 275
    - adding 275
    - copying 277
    - deleting 278
    - editing 278
    - posting 278
  - expense rights 664
    - additional 665
    - basic 664
  - Expense Unit Prices 279
  - ExpenseRepository 1948
    - deleteExpense 1994
    - insertExpense 1991
    - method details 1990
    - postExpense 1993
    - readExpense 1992
    - readExpensesByCriteria 1993
    - readRecentlyViewedExpenses 1994
    - updateExpense 1991
    - voidExpense 1994
  - expenses 265
    - creating 267
    - editing 267
  - Expense Entry Tool 275
  - General page 268
  - history 275
  - posting 274
  - preparing 270
  - transactions 274
  - unit prices 279
  - viewing 267
  - voiding 274
  - expenses object model 1562
    - EExpeDetail 1568
    - EExpeGroupAccess 1572
    - EExpeUserAccess 1570
    - TExpense 1563
  - Export List 1215
  - exporting reports 186
  - exporting search 78
- F -**
- field value types 1053
    - memo 1053
    - number 1053
    - text 1053
  - file names 1709
    - base classes 1712
    - batch display files 1712
    - custom Java blocks 1712
    - parts 1709
    - utility classes 1711
  - finance 238
    - accounts 239
    - budget requests 347
    - expenses 265
    - invoices 281
  - Finance tab 238
  - forward 427
  - frequently asked questions
    - address books for administrators 640
    - address books for users 443
    - Admin Settings 641
    - budgets 641
    - contacts 443
    - CSM 1779
    - from administrators 638
    - from users 441
    - groups 638

frequently asked questions  
 licenses 641  
 questions 445  
 records 441  
 reports 443  
 users 638  
 web services 2041

## - G -

general admin settings 539  
 getting started  
 administrators 457  
 users 36  
 global navigation 1039  
 Tab bar 1040  
 Tab bar contents 1041  
 global navigation bar 64  
 global navigation toolbar rights 699  
 global search 69  
 glossary  
 for administrators 724  
 for users 446  
 group category rights 507  
 group custom field rights 507  
 group history 515  
 group object views 514  
 group record rights 692  
 additional 693  
 basic 693  
 Group Rights 511  
 group system rights 485  
 assigning disputes 506  
 assigning global admin 495  
 assigning global setup 497  
 assigning global users 487  
 assigning projects example 505  
 contact categories 509  
 contact fields 509  
 contacts example 504  
 creating contacts example 503  
 deleting contacts example 504  
 editing contacts example 503  
 global admin rights 496  
 global setup rights 498  
 global user results 489  
 viewing contacts example 502

group tool rights 512  
 GroupAccountRepository 1949  
 deleteGroupAccount 1998  
 insertGroupAccount 1995  
 method details 1995  
 readGroupAccount 1996  
 readGroupAccountsByCriteria 1997  
 readRecentlyViewedGroupAccounts 1998  
 updateGroupAccount 1996  
 groups 511  
 accounts 477  
 category rights 507  
 creating 482  
 custom field rights 507  
 deleting 515  
 editing 482  
 history 515  
 members 484  
 object views 514  
 pages 458  
 preparing 738  
 rights 477  
 searching 481  
 system rights 485  
 templates 484  
 tool rights 512  
 viewing 482  
 working with 481

## - H -

Hierarchy 511  
 Hierarchy Rights 511  
 histories object model 1637  
 EHistDtail 1642  
 EHistGroupAccess 1647  
 EHistUserAccess 1645  
 THistory 1638  
 history admin settings 603  
 recording 603  
 history rights 667  
 additional 668  
 basic 667  
 HistoryRepository 1950  
 deleteHistory 2002  
 insertHistory 1999  
 method details 1999  
 readHistoriesByCriteria 2001



- HistoryRepository 1950
  - readHistory 2000
  - readRecentlyViewedHistories 2002
  - updateHistory 2000
- Hold 428
- hold conditions 428
- home page rights 699
- home page settings
  - modifying 58
- home pages 52
  - adding portal panes 1037
  - basics 997
  - content 1023
  - Content tab 1035
  - creating 1024
  - customization 1021
  - customizing 1026
  - editing 1029
  - layout 1029
  - master 1027
  - modifying 1028
  - settings 1022
  - summary 996
  - synchronizing 61
  - synchronizing customizations 1029
  - working with 53
- hover feature 43
- hover over 961
- | -**
- IMAP email settings 586
  - configuring 588
  - email clients 590
  - email rights 590
  - SSL 587
  - starting 589
  - stopping 589
- importing data 108
  - data file 106
- involved party rights
  - basic 686
- inserting new color profile 545
- invoice admin settings 596
  - decimal places 598
- invoice line items
  - adding 301
  - adjusting 322
  - copying 305
  - deleting 308
  - editing 307
  - searching 286
- invoice rights 670
  - additional 671
  - basic 670
- invoice validation rules 1754
  - adjusting 1766
  - do not allow multi-matter invoices 1768
  - expense charges exceed acceptable unit costs 1767
  - fee charges exceed agreed rates 1766
  - invoice contains duplicate expense 1775
  - invoice contains duplicate timekeeper charge 1773
  - invoice currency doesn't match vendor currency 1778
  - invoice is too old 1772
  - invoice line item description contains unauthorized keywords 1777
  - invoice service/expense charges are too old 1771
  - timekeeper billed too much time 1769
- invoice versions 292
  - viewing 292
- InvoiceRepository 1951
  - adjustInvoiceHeader 2008
  - deleteInvoice 2008
  - insertInvoice 2003
  - method details 2002
  - postInvoice 2007
  - readActiveApprovals 2009
  - readCompletedApprovals 2010
  - readInvoice 2005
  - readInvoiceApprovalsPendingOnPost 2010
  - readInvoicesByCriteria 2006
  - readRecentlyViewedInvoices 2008
  - updateInvoice 2005
  - voicemail 2007
- invoices 281
  - adjusting 315
  - approving 336
  - creating 295
  - currencies 293
  - differences 290
  - duplicates 292

invoices 281  
   editing 305  
   history 345  
   new amounts 344  
   posting 313  
   preparing 309  
   rejecting 337  
   reviewing 340  
   sending for review 338  
   Summary Breakdowns page 341  
   timekeeper rates 301  
   transactions 346  
   versions 292  
   viewing 284  
   voiding 340  
   with workflow 294  
   without workflow 294  
   workflow 346  
 invoices object model 1591  
   EInvDetail 1625  
   EInvGroupAccess 1635  
   EInvUserAccess 1632  
   ELitmDetail 1626  
   ELitmGroupAccess 1628  
   ELitmUserAccess 1630  
   JInvCategoryAdjustment 1605  
   JInvHeaderAdjustment 1607  
   JInvLineItem 1610  
   JInvNonUSTax 1618  
   JInvTimekeeperAdjustment 1621  
   JLitmAdjustment 1622  
   LInvNonUSTaxType 1619  
   TInvoice 1591  
   YCurrencyInfo 1636  
 Involved objects 805  
   creating 827  
 involved parties 165  
   accounts 168  
   adding 165  
   editing 165  
   related 168  
   roles 167  
   status 167  
 involved party rights 685  
   additional 687  
 InvolvedRepository 1958  
   deleteInvolved 2023  
   insertInvolved 2019

  method details 2019  
   readInvolved 2021  
   readInvolvedsByCriteria 2022  
   readInvolvedsForProject 2021  
   updateInvolved 2020  
 item selections 552

## - J -

JApptAttendee 1524  
 JApptResource 1526  
 Java samples 1881  
   automated actions 1891  
   automated qualifiers 1883  
   categories 1881  
   child projects 1881  
   parameterized actions 1899  
   related records 1881  
   rule actions 1891  
   searching 1909  
   Wizard page actions 1904  
 JContAddresss 1482  
 JContDefaultRate 1498  
 JContEmail 1491  
 JContFax 1489  
 JContInetAddress 1493  
 JContPhone 1486  
 JContRate 1499  
 JContRelation 1501  
 JContSkill 1495  
 JContTerritory 1504  
 JDistMember 1508  
 JDocuContent 1657  
 JInvCategoryAdjustment 1605  
 JInvHeaderAdjustment 1607  
 JInvLineItem 1610  
 JInvNonUSTax 1618  
 JInvTimekeeperAdjustment 1621  
 JInvRelation 1441  
 JLitmAdjustment 1622  
 JNote 1665  
 JProjAssignee 1416  
 JProjPhase 1420  
 JProjRelation 1425  
 JTaskAssignee 1584  
 JTranDetail 1551

**- K -**

keyboard shortcuts 431  
 additional 433  
 common 431  
 TeamConnect-specific 434

**- L -**

LAptArealtem 1523  
 LAptResourceType 1527  
 LContAddressType 1485  
 LContEmailType 1492  
 LContFaxType 1490  
 LContInetAddressType 1494  
 LContPhoneType 1488  
 LContRelationType 1503  
 LContSkillType 1496  
 LContTerritoryType 1505  
 LCountryItem 1484  
 LDAP authentication plug-in 625  
 error pages 626  
 installing files 630  
 parameters 626  
 prerequisites 625  
 sample configuration file 634  
 sample error file 633  
 sample files 631  
 sample login file 631  
 sample restore file 633  
 less file 545  
 license information 606  
 schemes 606  
 style 607  
 uploading file 607  
 LInvcNonUSTaxType 1619  
 list fields for API 1852  
 getting values 1857  
 setting values 1857  
 Locale Manager Tool 558  
 localization 563  
 custom blocks 1206  
 custom messages 1206  
 customizing 1203  
 exporting data 563  
 i18n keys 1204

importing data 566  
 portal panes 1206  
 reports 1205  
 rules 1206  
 saved searches 1207  
 Unique Key fields 1205  
 upgrade considerations 1207  
 wizards 1207  
 localizing 210  
 log in 37  
 log out 41  
 log rights 696  
 logging 608  
 appenders 608  
 default folders 610  
 logging in 37  
 logging out 41  
 logging using API 1878  
 logger levels 1880  
 logging messages 1879  
 login 37  
 login URLs 619  
 logout 41  
 lookup table rights 700  
 lookup tables 863  
 activating 884  
 adding 881  
 custom 885  
 deleting 885  
 editing 882  
 inactivating 882  
 system 876  
 types 875  
 using 875  
 lookup tables for API 1852  
 custom lookup table tree positions 1856  
 system lookup table tree positions 1854  
 LookupTableSource 1955  
 method details 2011  
 readSystemLookupTable 2011  
 LProjAssigneeType 1418  
 LProjRelationType 1426  
 LTaskActivityItem 1582

**- M -**

maintenance admin settings 600

- mapping EasyDocs RTF merge fields 1325
    - converting to XML files 1342
    - current date 1340
    - current user 1340
    - custom field mapping 1329
    - filter@ merge field 1333
    - filter@ to related objects 1334
    - filter@ to sub-objects 1334
    - if@ merge field mapping 1337
    - loop@ merge field 1330
    - loop@ to related objects 1330
    - loop@ to sub-objects 1330
    - nested within filter@ 1335
    - nested within loop@ 1331
    - opening RTF file 1325
    - system field mapping 1326
      - to a file 1341
      - to XML files 1342
    - user input screens 1343
  - mass editing records 86
  - matter management
    - adding CSC mapping files 1748
    - adding CT mapping files 1747
    - assignee roles 1717
    - categories 1715
    - CSC mapping files 1748
    - CSM configuration 1740
    - CT mapping files 1747
    - custom fields 1718
    - custom screens 1720
    - custom tools 1729
    - customizing 1708
    - E-Billing Roles tab 1741
    - embedded objects 1714
    - groups 1742
    - lookup tables 1719
    - object definitions 1713
    - payment settings 1742
    - phase transitions 1716
    - phases 1716
    - reports 1723
    - rules 1721
    - SOP 1746
    - Time Entry Tool 1730
    - tool rights 1739
    - users 1742
    - vendor settings 1741
    - wizards 1724
  - matter rights 681
    - additional 682
    - basic 681
    - involved parties 685
    - milestones 689
  - milestone rights 689
    - additional 690
    - basic 690
  - milestones 169
    - appointments 171
    - categories 171
    - creating 170
    - editing 170
    - tasks 172
    - viewing 169
  - miscellaneous setting rights 702
  - mobile 436
    - contacting a vendor 440
    - invoice adjustments 438
    - invoice details 438
    - invoices 437
    - pending approval 437
  - model classes 1784
  - moving requests 427
  - moving requests forward 427
  - multi-column sort 80
  - multi-currency lookup table rights 701
  - My Approvals 401
  - My Requests 396
  - My Workflow Processes
    - Attention Required 414
    - Completed 419
    - In Progress 414
- N -
- nesting tags 1353
  - notification admin settings 592
    - system notifications 595
  - NTLM authentication configuration file
    - editing 623
    - installing 623
  - NTLM authentication plug-in 620
    - anonymous access 622
    - enabling 624
  - Internet information services 622
    - prerequisites 621

**- O -**

- object definition rights 704
- object definitions 814
  - custom objects 822
  - system objects 820
- object model 1393
  - accounts 1535
  - appointments 1516
  - audience 1394
  - columns 1397
  - common objects 1665
  - contacts 1464
  - conventions 1395
  - database table names 1403
  - documents 1649
  - enumerations 1395
  - expenses 1562
  - histories 1637
  - invoices 1591
  - legacy names 1394
  - list attributes 1403
  - lookup tables 1401
  - new names 1394
  - object names 1403
  - objects 1394
  - projects 1404
  - properties 1399
  - system objects 1401
  - table prefixes 1399
  - tasks 1574
  - terms 1396
  - XCT files 1403
- object model properties 1399
  - database table names 1403
  - list attributes 1403
  - lookup tables 1401
  - object names 1403
  - system objects 1401
  - table prefixes 1399
  - XCT files 1403
- object naming patterns 834
  - attribute format 838
  - auto naming 842
  - literals 840
  - object attributes 837
- object navigator 1377
  - attributes 1378
  - creating paths 1388
  - filtering related objects 1385
  - filtering sub-objects 1385
  - navigating objects 1390
  - paths 1380
  - starting tables 1383
  - window 1380
- object views 943
  - assigning 950
  - clearing cache 951
  - creating 948
  - defining 947
  - invoice line items 952
  - troubleshooting 950
- objects 803
  - creating 813
  - custom 805
  - default icons 821
  - definitions 814
  - embedded 812
  - Involved 805
  - related 805
  - sub-objects 811
  - system 803
- objects definitions
  - viewing 817
- online help 43

**- P -**

- page title 58
- parent accounts 248
- password
  - reset 49
- phase transitions 846
  - creating 849
  - deleting 850
  - editing 849
- phases 842
  - adding 844
  - deleting 845
  - editing 844
- portal pane rights 712
- portal panes 52, 999
  - action links 1008
  - adding to home pages 57

- portal panes 52, 999
  - basics 997
  - changes 1030
  - color 1032
  - contents 1032
  - creating 1020
  - creating from home page 1038
  - custom content 1017
  - customizing 1026
  - default 999
  - deleting 1033
  - editing 60
  - information 55
  - layout 58
  - modifying 1028
  - modifying search views 59
  - multiple contents 1019
  - object links 1009
  - opening 1000
  - options 1032
  - search views 1013
  - settings 1001
  - summary 996
  - synchronizing 61
  - synchronizing customizations 1029
  - templates 1027
  - text 1014
  - title 1032
  - Web URLs 1016
  - WebIntelligence URLs 1018
- portal panes contents 1004
- post commit rules 1118
- posting criteria 253
- posting expenses 274
  - preparing 270
- posting invoices 313
  - preparing 309
- pre-population rules 1110
- process manager activities 413
- processes 428
- project contacts 160
  - adding 161
  - primary assignee 163
  - removing 164
  - unassigning 163
  - viewing 161
- project rights 681
  - additional 682
  - basic 681
  - involved parties 685
  - milestones 689
- ProjectRepository 1955
  - changePhase 2016
  - deleteProject 2017
  - insertProject 2012
  - method details 2012
  - readChildProjectsForEntityType 2015
  - readProject 2014
  - readProjectEntityTypes 2017
  - readProjectIntegrationSearches 2018
  - readProjectsByCriteria 2014
  - readProjectsUsingSearch 2018
  - readRecentlyViewedProjects 2017
  - updateProject 2013
- projects 153
  - accounts 173
  - appointments 172
  - categories 159
  - child 157
  - contacts 160
  - creating 154
  - editing 158
  - embedded 157
  - expenses 173
  - involved parties 165
  - milestones 169
  - phases 160
  - related 164
  - tasks 172
  - types 157
- projects object model 1404
  - EInvDetail 1442
  - EInvGroupAccess 1446
  - EInvUserAccess 1444
  - EMileDetail 1457
  - EMileGroupAccess 1461
  - EMileUserAccess 1458
  - EProjDetail 1428
  - EProjGroupAccess 1432
  - EProjUserAccess 1429
  - JInvRelation 1441
  - JProjAssignee 1416
  - JProjPhase 1420
  - JProjRelation 1425
  - LProjAssigneeType 1418
  - LProjRelationType 1426

projects object model 1404  
 TInvolved 1434  
 TMilestone 1449  
 TProject 1404  
 WObjdPhaseTransition 1424  
 WObjdPhaseType 1421  
 public record  
 private record 122

## - Q -

qualifier items 1131  
 creating 1139  
 examples 1140  
 limitations 1143  
 operators 1133  
 right arguments 1135

## - R -

rate request approval rule 1755  
 reassign 1169  
 reassigning approval requests 422  
 Reassignment page 422  
 record history 122  
 categories 125  
 creating 125  
 opening 123  
 viewing 124  
 record locking 540  
 record security 118, 122  
 editing 121  
 viewing 120  
 records 61  
 batch entries 95  
 canceling edits 99  
 categories 97  
 check-box 89  
 collections 65  
 copying 101  
 creating 84  
 date field 89  
 deleting 114  
 documents 127  
 drop-down lists 91  
 editing 86  
 fields 89

history 122  
 importing data 105  
 list box fields 92  
 list fields 91  
 locking 88  
 memo text 93  
 multiple 86  
 number fields 93  
 options buttons 94  
 printing 117  
 reassigning 104  
 related 126  
 saving 99  
 searching 68  
 security 118  
 sorting 79  
 text fields 94  
 time zone fields 90  
 types 64  
 unsaved 100  
 viewing 82  
 wizards 85  
 workflow 127  
 region 45  
 region settings 555  
 rejecting requests 403  
 through email 408  
 within records 406  
 related objects 805  
 contact-centric 809  
 parent-child relationships 808  
 relationship types 806  
 related records for API 1843  
 custom fields 1844  
 getting 1844  
 primary keys 1844  
 report rights 675  
 report security 210  
 reporting admin settings 576  
 reports 174  
 appearance 180  
 building 186  
 cached 180  
 Chart page 202  
 dashboards 176  
 Data page 187  
 data series 187  
 drill-down 179

- reports 174
  - exporting 186
  - folders 177
  - General page 186
  - localizing 210
  - portal panes 176
  - running 178
  - Schedule page 208
  - Security page 210
  - types 182
  - viewing 175
- REST API 2033
- retrieving tc:loop data 1266
  - qualifier examples 1271
  - qualifier syntax 1269
  - qualifiers 1268
  - sub-objects 1267
- retrieving template data
  - from child objects 1359
  - from contact fields 1355
  - from custom fields 1358
  - from Involved custom fields 1359
  - from related objects 1357
- reviews 411
- rights 463, 511
  - accounts 645
  - address books 652
  - admin settings 696
  - administer custom tools 698
  - administrator types 465
  - administrators 692
  - appenders 714
  - appointments 648
  - categories 471
  - contacts 652
  - custom fields 471
  - custom lookup tables 701
  - custom record types 476
  - custom records 681
  - documents 657
  - end-user types 464
  - end-users 643
  - expenses 664
  - global navigation toolbar 699
  - group records 692
  - history 667
  - home pages 699
  - invoice 670
  - involved parties 685
  - logs 696
  - lookup tables 700
  - matters 681
  - milestones 689
  - miscellaneous settings 702
  - multi-currency lookup tables 701
  - object definitions 704
  - portal panes 712
  - projects 681
  - report 675
  - routes 712
  - solution developers 697
  - solution developers types 465
  - system record types 468
  - system records 645
  - system settings 714
  - task 676
  - TeamConnect areas 477
  - TeamConnect features 477
  - user records 694
- route rights 712
- routes 1144
  - accessing 1170
  - conditions 1196
  - customizing 1181
  - email notifications 1183
  - General tab 1171
  - stops 1171
- RTransaction 1554
- rule actions 1794
  - custom code 1197
  - parameters 1799
  - writing 1796
  - writing in Java 1797
- rule qualifers 1794
  - custom code 1197
  - parameters 1799
  - writing 1795
  - writing in Java 1797
- rule triggers 1118
  - events 1118
- rule types 1108
  - approval 1111
  - audit 1114
  - custom action 1112
  - post commit 1118
  - pre-population 1110



- rule types 1108
    - scheduled action 1113
    - security 1109
    - user invoked action 1113
    - validation 1110
  - rules 1107, 1146
    - approval 1111
    - audit 1114
    - automated actions 1794
    - automated qualifiers 1794
    - comoponents 1108
    - creating 1122
    - creating qualifiers 1139
    - custom action 1112
    - general information 1122
      - General tab 1126
    - post commit 1118
    - pre-population 1110
    - qualifier items 1131
    - qualifier logic 1137
    - Qualifier tab 1130
    - qualifiers 1129
    - scheduled action 1113
    - security 1109
    - specifications 1185
    - tab 1122
    - trigger events 1118
    - triggers 1118
    - types 1108
    - user invoked action 1113
    - validation 1110
  - rules in Java 1797
    - automated qualifier 1801
  - running reports 178
    - scheduling 208
- S -**
- saving 460
  - saving search 77
  - scheduled action rules 1113
  - Scheduled Actions Monitor 615
    - actions 617
    - activating 618
    - creating 616
    - deactivating 618
    - deleting 619
    - finding 615
    - Run Now button 619
    - running 619
    - tasks 616
  - scheduled reports 208
    - activating 209
    - deactivating 209
    - deleting 210
    - editing 208
    - run time 210
    - running 210
    - viewing 209
  - Scheduler 219
  - search views
    - creating 958
  - search 68
    - custom 70
    - exporting 78
    - global 69
    - modifying 69
    - modules 78
  - search criteria 70
  - search operators 72
  - search results 76
  - search results admin settings 571
    - maximum search results 572
  - search view filter 985
    - contacts 990
    - creating 991
    - filter tab 989
    - qualifier conditions 986
    - qualifier tips 993
    - qualifiers 985
    - section titles 992
  - search view results 978
    - object link types 983
    - object links 984
    - search example 982
    - spanned columns 982
  - search views 961
    - approvals 965
    - auto search 973
    - designing 954
    - document content 966
    - field content 967
    - filter display 985
    - full-text search 966
    - general information 968
    - global search 962

- search views 961
  - home pages 959
  - IMAP search 975
  - projects 977
  - requests 965
  - results display 978
  - screen 955
  - search modules 974
  - subscribing 972
  - templates 978
  - Third-party calendars 977
  - troubleshooting 996
  - unsubscribing 972
  - workflow processes 965
- searching using API 1867
  - executing 1877
  - field paths 1874
  - on fields 1873
  - search criteria 1869
  - search operators 1876
  - search parameters 1876
  - search views 1869
- security admin settings 577
  - user authentication 582
- security rights for API 1862
  - functional security 1863
  - record security 1862
  - system user 1859
- security rules 1109
- service classes 1786
  - actions 1789
  - dependencies 1790
  - other 1790
  - qualifiers 1789
  - ResourceService 1787
  - screens 1789
- settings 45
  - changing 45
  - other 52
  - personalizing 44
  - search result 48
- Setup 739
  - Backward Compatibility Settings page 742
  - designer menu 740
  - overview 739
- single sign-on 36
- single-column sort 80
- sorting
  - multi-column 80
  - single-column 80
- sorting records 79
- ssearch view results
  - display columns 984
- stop members 1175
  - groups 1176
  - user paths 1177
  - users with roles 1178
- stops 1171
  - customizing 1181
- submitting reviews
  - through My Approvals 411
  - through records 413
- sub-objects 811
  - embedded 812
- sub-objects for API 1845
  - contacts 1847
  - project assignees 1846
  - task assignees 1846
- System Logging page 609
  - clearing 611
- system lookup tables 876
  - contact address 880
  - structure 879
- system objects 803
  - configuring 820
  - default icons 821
  - general information 820
- system record rights 645
  - accounts 645
  - address books 652
  - appointments 648
  - contacts 652
  - documents 657
  - expenses 664
  - history 667
  - invoice 670
  - prerequisites 470
  - related records 470
  - report 675
  - task 676
- system setting rights 714
- system settings 749
  - accessing 749
  - appenders 783
  - contact sweeper 759

- system settings 749
    - duplicate checking 767
    - duplicate contact manager 758
    - invoice duplicates 753
    - invoices 752
    - line items 753
    - line items behavior 753
    - miscellaneous 750
    - non-US taxes 752
    - preventing duplicates 767
    - rule execution 749
    - XML Worksheet tool 780
  - system views 893
  - system settings
    - line item adjustments 755
- T -**
- Tab bar
    - appearance 1040
    - contents 1041
  - TAccount 1535
  - tag attributes 1366
    - all field types 1368
    - custom object fields 1376
    - date fields 1374
    - field labels 1367
    - Involved fields 1376
    - list fields 1374
    - memo text fields 1373
    - number fields 1370
    - text fields 1372
  - TAppointment 1516
  - task entry tool 234
    - creating 234
    - deleting 237
    - editing 236
    - posting 237
  - task rights 676
    - additional 677
    - basic 676
  - TaskRepository 1959
    - deleteTask 2027
    - insertTask 2023
    - method details 2023
    - postTask 2026
    - readRecentlyViewedTasks 2027
    - readTask 2025
    - readTasks 2025
    - reassign 2027
    - updateTask 2024
    - voidTask 2026
  - tasks 223
    - assigning 231
    - creating 227
    - editing 227
    - General page 225
    - history 234
    - posting 232
    - task entry tool 234
    - transactions 233
    - viewing 225
    - voiding 233
    - wizards 226
  - tasks object model 1574
    - ETaskDetail 1585
    - ETaskGroupAccess 1589
    - ETaskUserAccess 1586
    - JTaskAssignee 1584
    - LTaskActivityItem 1582
    - TTask 1574
  - tc:conditional 1274
    - categories 1279
    - comparing custom field values 1277
    - comparing sub-objects 1278
    - custom fields 1363
    - meeting conditions 1365
    - object attribute values 1278
    - samples 1363
    - sub-objects 1364
    - testing 1364
  - TContact 1464
  - TDocument 1649
  - TeamConnect API 1782
    - account samples 1817
    - accounts 1817
    - actions 1794
    - appointment samples 1819
    - appointments 1819
    - blocks 1804
    - categories 1847
    - common functions 1857
    - contact samples 1822
    - contacts 1821
    - custom fields 1850
    - custom objects 1840

- TeamConnect API 1782
  - custom settings 1865
  - development environment 1783
  - document samples 1824
  - documents 1824
  - enumerations 1786
  - executing search 1877
  - expense samples 1827
  - expenses 1827
  - getting started 1790
  - group samples 1829
  - groups 1829
  - history 1831
  - history samples 1831
  - internationalization 1866
  - invoice samples 1833
  - invoices 1832
  - Involved parties 1842
  - Involved party samples 1842
  - Java samples 1881
  - list fields 1852
  - logging 1878
  - lookup tables 1852
  - model classes 1784
  - new records 1815
  - object definitions 1815
  - objects 1815
  - packages 1784
  - pages 1803
  - policy 1781
  - project samples 1841
  - projects 1840
  - qualifiers 1794
  - related records 1843
  - ResourceService 1787
  - rules 1794
  - scheduled actions 1813
  - screens 1804
  - search criteria 1869
  - search parameters 1876
  - searching 1867
  - security rights 1862
  - service classes 1786
  - sub-objects 1845
  - system objects 1817
  - system settings 1864
  - task samples 1835
  - tasks 1835
  - tools 1810
  - user samples 1838
  - user settings 1864
  - users 1838
- TeamConnect area rights 679
  - preferences 680
  - release edit locks 680
  - scheduled actions 680
- TeamConnect customization 738
- TeamConnect Enterprise
  - administrators 457
  - API 1780
  - designers 737
  - developers 1780
  - users 36
  - web services 1912
- TeamConnect feature rights 679
  - preferences 680
  - release edit locks 680
  - scheduled actions 680
- TeamConnect logs 608
  - clearing 611
  - system logging levels 611
- TeamConnect Mobile 436
- TeamConnect summary 604
  - available updates 605
  - license information 606
  - support options 606
  - versions 604
- templates 1042
  - adding fields 1050
  - assignees 1058
  - attendees 1058
  - attribute values 1055
  - categories 1057
  - creating 1048
  - defining fields 1050
  - deleting sub-objects 1060
  - document folders 1046
  - features 1045
  - field values 1053
  - formula values 1055
  - general information 1045
  - General tab 1045
  - literal values 1054
  - planning 1043
  - Records tab 1047
  - related objects 1060

- templates 1042
    - sub-objects 1056
  - terminology
    - administrators 461
  - testing EasyDocs templates 1344
  - TExpense 1563
  - THistory 1638
  - Time Entry Tool Settings 1730
    - administrator 1737
    - default fields 1732
    - editing 1730
    - time periods 1734
    - timekeepers 1735
  - time format 45
  - time interval 49
  - TInvoice 1591
  - TInvolved 1434
  - TMilestone 1449
  - top level folder
    - Attachments 717
    - HTTPRoot 717
    - System 718
    - Users 722
  - TProject 1404
  - troubleshooting
    - CSM 1779
    - for administrators 641
    - for users 445
  - troubleshooting custom designs 1228
    - BUILD FAILED message 1232
    - design import restrictions 1229
    - DMT script errors 1232
    - error messages 1230
    - general steps 1230
    - logging 1230
    - no design components 1229
    - performance issues 1232
    - warnings 1228
  - TTask 1574
  - types of projects 157
    - child 157
    - embedded 157
- U -
- UGrupMember 1666
  - understanding 463
  - unique IDs 829
    - auto numbering patterns 831
    - object attributes 833
    - project link display 833
  - unsaved records 100
  - upgrade considerations for localization 1207
    - best practices 1208
    - custom blocks 1207
    - date formatting 1208
    - parameters 1208
  - uploading EasyDocs RTF file 1324
  - user category rights 534
  - user history 537
  - user interface 42
  - user interface admin settings 544
    - Action buttons 551
    - custom logo 545
    - item selections 552
    - user selections options 551
  - user involved action rules 1113
  - user record rights 694
    - additional 695
    - basic 695
  - user selection options 551
  - user system rights 527
    - assigning global admin 532
    - assigning global setup 533
    - assigning global users 531
  - user tool rights 536
  - UserAccountRepository 1961
    - deleteUserAccount 2032
    - insertUserAccount 2028
    - method details 2028
    - readRecentlyViewedUserAccounts 2032
    - readUserAccount 2030
    - readUserAccountsByCriteria 2031
    - updateUserAccount 2030
  - users
    - accounts 516
    - activating 524
    - category rights 534
    - creating 519
    - deleting 537
    - editing 519
    - General page 538
    - history 537
    - inactivating 524

## users

- limited-privilege 524
- membership 526
- passwords 525
- preferences 525
- preparing 738
- rights 516
- searching 517
- system rights 527
- templates 523
- tool rights 536
- unlocking accounts 525
- viewing 518
- working with 516

**- V -**

- validation rules 1110
- voiding expenses 274

**- W -**

- web browser 44
- web browser buttons 44
- web service API reference 1934
  - AccountRepository 1935
  - AppointmentRepository 1937
  - ContactRepository 1939
  - details 1962
  - DocumentRepository 1945
  - ExpenseRepository 1948
  - GroupAccountRepository 1949
  - HistoryRepository 1950
  - InvoiceRepository 1951
  - InvolvedRepository 1958
  - LookupTableSource 1955
  - ProjectRepository 1955
  - summary 1934
  - TaskRepository 1959
  - UserAccountRepository 1961
- web service applications 1925
  - categories 1930
  - client proxy 1925
  - common functions 1927
  - components 1927
  - creating records 1928
  - custom fields 1932
  - deleting records 1933

- reading records 1928
- searching records 1928
- SOAP security header 1926
- updating records 1928
- URLs 1925
- web service code samples 2047
  - accounts 2055
  - appointments 2062
  - contacts 2048
  - documents 2066
  - expenses 2078
  - group accounts 2081
  - history 2085
  - invoices 2088
  - Involved parties 2102
  - line items 2088
  - projects 2097
  - tasks 2106
  - user accounts 2110
- web services 1912
  - abstract classes 2036
  - API reference 1934
  - batch operations 2041
  - category data objects 2038
  - client application components 2042
  - client source-code 1921
  - code samples 2047
  - cosmetic data objects 2042
  - creating applications 1925
  - custom field data objects 2038
  - data types 1919
  - error handling 1920
  - getting started 1912
  - glossary 2113
  - key concepts 1918
  - non-supported operations 2041
  - overview 1912
  - record reading 1919
  - record updating 1919
  - repository classes 1918
  - request formats 1919
  - requirements 1920
  - response formats 1919
  - REST API 2033
  - search criteria data objects 2039
  - supported operations 1914
  - unique keys 1920
- web services for accounts 2055

- web services for accounts 2055
  - activating 2060
  - allocating money 2061
  - creating 2055
  - deactivating 2061
  - deleting 2062
  - reading 2057
  - reading child accounts 2058
  - searching 2059
  - transferring money 2061
  - updating 2056
  - withdrawing money 2062
- web services for appointments 2062
  - creating 2062
  - deleting 2066
  - reading 2064
  - searching 2065
  - updating 2063
- web services for contacts 2048
  - creating 2048
  - deleting 2054
  - reading 2052
  - searching 2052
  - updating 2050
- web services for documents 2066
  - checking in 2077
  - checking out 2077
  - copying 2076
  - creating 2066
  - creating hyperlinks 2076
  - creating hyperlinks with default category 2075
  - creating shortcuts 2075
  - creating subfolders 2075
  - deleting 2078
  - moving 2076
  - reading 2068
  - reading by path 2070
  - reading child documents 2069
  - reading documents of parent folder 2071
  - reading folder for user 2072
  - reverting to previous version 2077
  - searching 2073
  - undoing checkout 2077
  - updating 2067
- web services for expenses 2078
  - creating 2078
  - deleting 2081
  - posting 2081
- reading 2079
- searching 2080
- updating 2078
- voiding 2081
- web services for group accounts 2081
  - creating 2082
  - deleting 2084
  - reading 2083
  - searching 2084
  - updating 2082
- web services for history 2085
  - creating 2085
  - deleting 2088
  - reading 2086
  - searching 2087
  - updating 2085
- web services for invoices 2088
  - adjusting 2090
  - creating 2088
  - deleting 2097
  - posting 2095
  - reading 2092
  - searching 2093
  - updating 2090
  - voiding 2095
- web services for Involved parties 2102
  - deleting 2106
  - inserting 2102
  - reading 2104
  - reading for projects 2104
  - searching 2105
  - updating 2103
- web services for line items 2088
  - adjusting 2091
  - creating 2088
  - searching 2095
- web services for projects 2097
  - changing phases 2100
  - creating 2097
  - deleting 2102
  - reading 2100
  - reading child projects 2101
  - searching 2101
  - updating with embedded records 2099
- web services for tasks 2106
  - creating 2106
  - deleting 2110
  - posting 2109

web services for tasks 2106  
     reading 2107  
     reassigning 2110  
     searching 2108  
     updating 2107  
     voiding 2109  
 web services for user accounts 2110  
     creating 2110  
     deleting 2113  
     reading 2111  
     searching 2112  
     updating 2111  
 wizards 1063  
     actions 1084  
     automated actions 1085  
     creating 1070  
     designing 1063  
     execution 1107  
     General tab 1072  
     initial actions 1086  
     layout 1068  
     Page Components tab 1076  
     page order 1101  
     page transition rules 1097  
     page transitions 1069  
     Page Transitions tab 1094  
     pages 1074  
     parameters 1081  
     requirements 1065  
     simple actions 1087  
     specifications 1065  
     starting 1107  
     tab 1071  
     templates 1067  
     testing 1102  
     troubleshooting 1102  
 Wizards tab 1071  
 WObjdPhaseTransition 1424  
 WObjdPhaseType 1421  
 WObjdProjectInfo 1671  
 WObjfDetailField 1667  
 workflow 388, 427, 428, 1146, 1169  
 workflow processes  
     email approvals 392  
     issues 414  
     monitoring 414  
     reassigning 421  
     returning requests 420

    reviewing 419  
     workflow reviews 409

## - Y -

ycontactgroup  
     Y\_Contact\_Group 1506  
 YCurrencyInfo 1636  
 YDetailLookupItem 1676  
 YDetailLookupTable 1678  
 YDocuContentType 1657  
 YGroup 1679  
 YRecentlyViewedRecord 1682  
 YUser 1683