MITRATECH

TeamConnect® Enterprise 5.0

Installation Guide

TeamConnect® Enterprise 5.0 Installation Guide

Document ID: tce_5_0_ig_user_2, published on 12/2/2016

Disclaimer of Warranty

Mitratech Holdings, Inc. (Mitratech) 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.

Mitratech 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 Mitratech product documentation. For further information regarding third-party non-standard or non-default functionality, please contact Mitratech 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 Mitratech.

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

If you need support for TeamConnect™ Enterprise 5.0, please contact the Mitratech support team by sending an email to: support@mitratech.com. For more information about Mitratech, visit our web site: http://www.mitratech.com.

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

NOTE: Throughout Mitratech 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

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

1 Installation Help

Welcome to the *TeamConnect® Enterprise Installation Help*.

Installation Requirements	TeamConnect Installer
TeamConnect Deployment	Business Objects
Verification	Upgrading TeamConnect
Installer Checklist	Installing Elasticsearch

1.1 Installation Overview

TeamConnect installation should be completed in the following order:

1. Download Java 1.7 or 1.8 and ensure the proper requirements are met for application servers, web servers, or browsers as detailed in Installation Requirements. Please note the additional comments on Application and Web Servers for Tomcat users.
2. Set up the server that will be used with Elasticsearch.
3. Download Elasticsearch (see Elasticsearch Installation for more details). **Only Elasticsearch 1.7.4 is officially supported by Mitratech.**
4. Configure Elasticsearch for your designated server. For more information, see Elasticsearch Configuration.
5. Put in a Support ticket for the 5.0 Installer with support@mitratech.com.
6. Run the TeamConnect installer. Configure the teamconnect.properties file in the installer (or manually, if desired) for Elasticsearch.

For information on upgrading an existing TeamConnect 4.x to TeamConnect 4.2, please see Upgrade Considerations.

1.2 Installation Requirements

Before you begin to install TeamConnect, please ensure that your system meets the requirements listed in this section.

Note: *If you add any TeamConnect plug-ins, refer to the corresponding documentation for additional requirements.*

Note: *If you use a custom logo on your TeamConnect web pages, be sure that it does not exceed 50 pixels in height.*

1.2.1 Database Requirements

For **Oracle**, the login that you provide for TeamConnect's use must have the following database privileges:

- **CONNECT** and **CTXAPP** roles
- **CREATE TABLE** system privilege
- **CREATE VIEW** system privilege
- **CREATE PROCEDURE** system privilege
- **CREATE SEQUENCE** system privilege
- **CREATE TRIGGER** system privilege
- **CREATE MATERIALIZED VIEW** system privilege
- User quota "**Unlimited**", value -1, unit MBytes

For Data Warehouse you need additional rights, which are listed in the Data Warehouse Release Notes.

For SQL Server, the login that you provide for TeamConnect's use must have the following roles:

- **public** role
- **db_owner** role

Your SQL Server DBA must also alter the database configuration as follows:

- ALTER DATABASE <dbname> SET ALLOW_SNAPSHOT_ISOLATION ON
- ALTER DATABASE <dbname> SET READ_COMMITTED_SNAPSHOT ON

1.2.1.1 Localization

If you plan to localize TeamConnect for multiple languages, your Oracle database configuration should include:

- **NLS_NCHAR_CHARACTERSET** parameter set to **AL16UTF16**. (This is the default.)

- All text columns set to **NVARCHAR2** datatype. This is done automatically for newly created databases; if you are upgrading from an older version, you may need to perform manual tasks to make your database compatible. See Upgrade Considerations for more information.

1.2.1.2 Supported Databases

TeamConnect 5.0 supports, and is certified on, the following database versions:

Supported Versions	Certified On
Oracle 12c R1	12.1.0.1.0
Oracle 11g R2	11.2.0.3.0
SQL Server 2014 R2	12.0.5000.0
SQL Server 2012	11.0.6020.0

The database server should be installed and configured on a hardware platform and operating system certified by your database server vendor.

TeamConnect requires at least 500 MB for its tablespace, and requires a larger tablespace when your data is added or you add custom elements to your design.

(Oracle only) For future expansion, enable auto extend for the data file.

(SQL Server only) For future expansion, ensure that "Enable Autogrowth" is checked for the database.

TeamConnect features full-text search capabilities (outside of Elasticsearch's Global Search), but those capabilities are constrained by the search capabilities of your database server software. If you store a document with a specific file type that your database server does not support for its full-text search feature, the content of that document is skipped during the TeamConnect full-text search. Please review the documentation supplied with your database server's version to determine which file types are supported.

1.2.1.3 Estimating Your Actual Database Size

Each customer's design is unique; thus the number of custom fields, field types, documents stored and the numbers of rows of data for each object type will be unique as well. You must also take into account the amount of data and documents that will be migrated from legacy applications into TeamConnect via your initial data conversions. Finally, the number of users and the volume of new records they will be entering yearly is important. Using this information, and any historic information you have regarding year to year data growth from the legacy systems and expected increases in user community etc., is important to be able to estimate the initial database size and yearly database growth.

For these reasons, estimating the size of the TeamConnect database is not a simple task. We recommend that you conduct a database sizing exercise as part of the initial TeamConnect implementation plan. This will ensure that you are able to provision the correct amount of initial

database space and accommodate the yearly growth. Current TeamConnect customers have database sizes varying from less than 5 gigabytes of data to over a terabyte.

1.2.2 Application and Web Servers

TeamConnect 5.0 supports, and is certified on, the following application servers. The servers listed below support JDK 1.8 and JDK 1.7.

Comprehensive details on Java installation can be found at the Java 1.8 download site here.

Supported Versions	Certified On	Drivers
Oracle Weblogic 11g		• For SQL Server databases, use WebLogic administration tools to create a JDBC data source. Use driver "Oracle's MS SQL Server Driver (Type 4) Versions: 7.0 and later • For Oracle databases, create a JDBC data source that uses driver "Oracle's Driver (Thin) Versions: 9.0.1 or later
WebSphere 8.5.5	8.5.5.10	• For SQL Server databases, use Integrated Solutions Console to create a JDBC data source that uses driver "WebSphere embedded ConnectJDBC driver for MS SQL Server". • For Oracle databases, create a JDBC data source that uses driver "Oracle JDBC Driver" • For either database, set implementation type to "Connection pool data source" • WebSphere/IBM users need to download the proprietary Java 8 bundle: IBM SDK, Java Technology Edition, Version 8 ***Note:*** *The Web Services Feature Pack for WebSphere is not supported.*
Tomcat 8.5 **Tomcat 7.0**	8.5.5	• For Oracle use ojdbc6.jar • For SQL Server, use a version 4 driver

		• For Tomcat 8.5, remove JSTL 1.1.2 and replace with 1.2 **Important Note for Tomcat 8.5 Servers** **Important: For Tomcat 8.5, please add the following for Connector in the server.xml file (found in root/conf):** compression="on" compressableMimeType="text/html,text/xml,text/plain,text/css,application/javascript" **TeamConnect's web.xml file (found in the WEB-INF folder) must have the following commented out:** -<filter-mapping> <filter-name>compressionFilter</filtername> -mapping> -<filter-mapping> <filter-name>compressionFilter</filter-name> <url-pattern>*.js</url-pattern> </filter-mapping> -<filter-mapping> <filter-name>compressionFilter</filter-name> <url-pattern>*.css</url-pattern> </filter-mapping> -<filter-mapping> <!-- compress the gwt .cache.html files --> <filter-name>compressionFilter</filter-name> <url-pattern>*.html</url-pattern> </filter-mapping>

TeamConnect supports the hardware platforms and operating systems that are certified by your application server software vendor.

To provide TeamConnect with sufficient Java heap space to operate optimally, you must ensure that the server on which you are running TeamConnect has enough memory and CPU capacity available to support the usage pattern and the number of users.

The required Java minimum heap size setting is at least 2 GB for TeamConnect. Depending on your environment and usage, you may need to increase your minimum heap size. It is recommended that you use a 64-bit operating system on the TeamConnect application server. It is important that you monitor JVM heap utilization during the test cycle as well as perform a load test to validate that the settings you are planning to use in production can handle the expected user scenarios and load. There are other environmental and configuration considerations that should be taken into account when setting up TeamConnect for optimal performance. For more information, please refer to the performance tuning documentation provided by your application server vendor and database vendor.

Consult the application server documentation for a list of compatible web servers. For details regarding application and web server installation, see your vendors' documentation at the corresponding URLs:

- **Apache Tomcat**—tomcat.apache.org
- **Oracle WebLogic**—www.oracle.com
- **IBM WebSphere**—pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp
- **Microsoft Internet Information Server (IIS)**—www.microsoft.com

It is recommended that you configure your web server to "keep alive" the TeamConnect session. This is particularly recommended if you are using a single sign-on method of authentication. "Keep alive" improves the performance of TeamConnect.

1.2.2.1 Testing Your Environment's Support for TeamConnect Features

TeamConnect uses network features that require support from the database, application, and web servers on which it is deployed. Three test applications are provided with TeamConnect that can quickly confirm that your environment supports these features.

All three test applications are run by issuing Java commands from the command line, using **tcutility.jar**, a file that is found in the **utilities/test** subfolder of your installation directory. In some cases, you need to use two machines when testing - one as a server and one as a client.

Before testing, copy tcutility.jar to a local directory on the test machines, then start a command-prompt session.

For Windows execute the following command:

```
set CLASSPATH=<location>/tcutility.jar
```

For UNIX:

```
Export CLASSPATH = $CLASSPATH:<location>/tcutility.jar
```

1.2.2.1.1 JDBC Latency Test

This application measures database server response time.

To test SQL Server, execute this command. Type the entire command on a single line:

```
java -classpath tcutility.jar;sqljdbc.jar
teamconnect.jdbc.JdbcLatencyTest arg0 arg1 arg2 arg3 arg4 arg5
```

To test Oracle:

```
java -classpath tcutility.jar;ojdbc5.jar
teamconnect.jdbc.JdbcLatencyTestarg0 arg1 arg2 arg3 arg4 arg5
```

The argument definitions are:

- 0: JDBC URL (e.g., `jdbc:sqlserver://10.0.0.127;databaseName=TEST_DB` or `jdbc:oracle:thin:@10.0.0.127:1521:test_db`
- 1: class name for database driver (e.g. `oracle.jdbc.OracleDriver` or `com.microsoft.sqlserver.jdbc.SQLServerDriver`)
- 2: Database username
- 3: Database password
- 4: Number of iterations in the test
- 5: Text of the SQL query (e.g. select * from Y_USER)

The output from the app states the time, in microseconds, required to execute this simple query. Expected acceptable range for this query is 0.6 to 0.8 seconds. Longer times may indicate a database issue that should be resolved before installation.

1.2.2.1.2 UDP Test

This application tests clustered server environments. Both UDP and RMI (described below) must work properly for TeamConnect to run on clustered servers.

Login to Machine1 and run these commands:

```
set CLASSPATH=tcutility.jar
java -classpath tcutility.jar teamconnect.udp.UDPServer 239.192.16.15 6018 5
```

The argument values shown are examples only. The argument definitions are:

- arg0: UDP address
- arg1: UDP port
- arg2: Time to live (number of hops)

The application responds with message "waiting..."

Login to Machine2 and run these commands:

```
set CLASSPATH=tcutility.jar
java -classpath tcutility.jar teamconnect.udp.UDPClient 239.192.16.15 6018 Hello 5
5
```

The argument values shown are examples only. The argument definitions are:

- arg0: UDP address
- arg1: UDP port
- arg2: Message string to be sent
- arg3: Message count
- arg4: Time to live (number of hops)

The application will respond with messages showing the test string and the status of the send request. These messages will repeat for as many times as are specified in the arg3 value. The server application on Machine1 should display messages, one for each send request, containing the text of the sent string.

Repeat the process by terminating the server application on Machine1, then changing Machine2 to Server and Machine1 to Client.

1.2.2.1.3 RMI Test

This application tests clustered server environments. Both UDP (described above) and RMI must work properly for TeamConnect to run on clustered servers.

Note: *The RMI registry should not already be running when you begin this test. It is important that the CLASSPATH value be set before the registry is started.*

Login to Machine1 and run these commands:

```
set CLASSPATH=tcutility.jar
rmiregistry <port>
java -classpath tcutility.jar teamconnect.rmi.Server <port>
```

The application responds with "Server ready".

Login to Machine2 and run these commands:

```
set CLASSPATH=tcutility.jar
java -classpath tcutility.jar teamconnect.rmi.Client <IP addr of RMI server> <port>
```

The application responds with the message "Hello, World" and with the run time that was required to execute the test.

Repeat the process by terminating the server application on Machine1, then changing Machine2 to Server and Machine1 to Client, then rerunning the test.

1.2.2.2 Web Servers

TeamConnect 5.0 only supports IIS Web Server 8.

For the Apache web server, it may be necessary to change a default timeout setting to prevent the server from resubmitting a HTTP request when the original request does not return for a long time. The setting, `WLIOTimeoutSecs`, has a default value of 300 seconds. Shown below is an example for the WebLogic plug-in module, from file `weblogc.conf` file in the `%APACHE_HOME%/conf` folder:

```
<Location /Filenet_Demo>
    SetHandler weblogic-handler
    WebLogicPort 4050
```

```
    WLIOTimeoutSecs 300
</Location>
```

You should be able to configure this setting in other web server plug-ins, too.

If your requests may take longer than 300 seconds to execute, increase the value from 300 to a more appropriate value.

1.2.2.3 Web Proxy Settings

To provide access to TeamConnect via your organization's URL (for example, `http://www.yourcompany.com/TeamConnect/`), set up the appropriate plug-in for your company's web server to proxy (redirect) the HTTP requests to the application server. You must specify the proxy URL (`/TeamConnect` in this example) and the hostname and port used by your application server.

Each web server will have a different procedure and syntax for configuring the plug-in. Refer to your web server's documentation for details.

1.2.3 Web Browsers

The TeamConnect 5.0 application supports, and is certified for use with, the following web browsers:

Supported Versions
Safari 9.1 (Mac only)
Internet Explorer 11 (with Windows 8.1)
Microsoft Edge
Firefox (current)
Chrome (current)

TeamConnect is best viewed with a browser width minimum of 1024px.

Note: *Safari has an issue when running on 32-bit Mac 10.6.4. Other Mac operating systems, including 64-bit Mac OS, do not have this issue.*

Important: *The browser options must be set to their default settings. TeamConnect is not certified for use with non-default browser settings.*

It is recommended that client workstations do not use JavaScript debugging when accessing TeamConnect or other Mitratech applications.

Using the Share URL Feature with a Browser

For Internet Explorer only, you can configure your browser to use the existing TeamConnect session when you click on a Share URL that you obtain from an email or other source. Clicking on the URL link will position your current TeamConnect session to the record referenced by the Share URL. To enable this capability, you must configure IE as follows: Go to menu item **Tools -> Internet Options -> Advanced**. Check the **Reuse windows for launching shortcuts** checkbox. Save.

Other web browsers do not support the ability to reuse existing TeamConnect sessions. Clicking on a Share URL will open a second TeamConnect session and invalidate the existing session.

1.2.4 Elasticsearch Setup

Elasticsearch setup for TeamConnect should proceed through the following steps in order. It is important to note that users must read the Installation Overview and install Java 1.7.

Installation Overview
Installation
Configuration
TeamConnect Properties Configuration
Best Practices

For more detailed information on how Elasticsearch works, clusters and nodes, and how the global search indexing functions outside of TeamConnect, please see the Elasticsearch help found here. The latest support matrix for Elasticsearch compatibility can be found here.

1.2.4.1 Elasticsearch Installation

The following instructions detail a RedHat installation via the Elasticsearch file. If your system is unable to perform this type of installation, please contact Mitratech Support at support@mitratech.com.

1. Download the RPM file from https://www.elastic.co/downloads/past-releases/elasticsearch-1-7-4. **TeamConnect is only certified against Elasticsearch 1.7.4. Later 1.7.x patches and updates will be supported. No updates outside of the 1.7.x line will be supported.**
2. yum install the elasticsearch.xxxxx.rpm or rpm –Uvh elasticsearch.xxxxx.rpm

3. chkconfig –add elasticsearch to have Elasticsearch start on boot.

For the following steps, the paths listed below should be accurate to your installation. If not, find the elasticsearch/bin folder and substitute that path in the directions below.

1. Run /usr/share/elasticsearch/bin/plugin install elasticsearch/elasticsearch-mapper-attachments/2.7.0

This plug-in provides the 'attachment' field type to Elasticsearch, thereby allowing it to index to content of documents like PDF files and Microsoft Word files.

2. Run /usr/share/elasticsearch/bin/plugin install royrusso/elasticsearch-HQ

This plug-in provides monitoring, management, and querying web interface for ElasticSearch instances and clusters.

Note: *Monitoring the cluster is done on port 9200 by default, i.e., servername:9200/_plugin/hq. The firewall may need to be manually opened.*

3. Configure properties as detailed in Elasticsearch Configuration.

4. Start elasticsearch with: /etc/init.d/elasticsearch start after configuration is set.

For Windows

1. Download the Elasticsearch zip file from the following website:

   https://www.elastic.co/downloads/elasticsearch

Note: If you are looking for an older version of ES, look here: https://www.elastic.co/downloads/past-releases

2. Unzip the file in your directory of choice. Having a dedicated dev folder not in Program Files is usually a good choice. (e.g. C:\dev\Apache\Elasticsearch)

3. Navigate to your Elasticsearch /bin either through the command line using Windows Explorer and typing cmd in the top bar.

   The directory should be similar to C:\dev\app\Apache\Elasticsearch\elasticsearch-1.7.1\bin

4. Install the Mapper Attachments Plugin

This plugin is used to parse documents that are sent over from TeamConnect. It must be installed for documents to index properly. https://github.com/elastic/elasticsearch-mapper-attachments

5. Run the following command from the /bin:

plugin -install elasticsearch/elasticsearch-mapper-attachments/2.7.0

Note: The version number at the end corresponds to the main version of your ES install. If ES is 1.7.x, the plugin is 2.7.0. You can see the available versions here: https://github.com/elastic/elasticsearch-mapper-attachments

5. Install the ElasticHQ Plugin

This plugin is used to monitor Elasticsearch. http://www.elastichq.org/index.html

6. Run the following command from the /bin:

plugin -install royrusso/elasticsearch-HQ

You can monitor the cluster by pointing your browser to: domain:port/_plugin/HQ/

7. Configure properties as detailed in Elasticsearch Configuration.

Running Elasticsearch on Windows is fairly simple, just navigate to the bin directory of your Elasticsearch installation and start the elasticsearch.bat file.

Elasticsearch picks up your java path from your environment variables, so you'll have to add the JAVA_HOME variable pointed to java 1.7.

In order to set the heap size for ES, you must define an environment variable for it.

1. From the Start menu or Windows Explorer, right click on Computer and select Properties"
2. From the left nav, select "Advanced System Settings".
3. In System Properties, select "Environment Variables".
4. Under "System variables", add a new variable with:

name: ES_HEAP_SIZE

value: 1g

5. Select "OK" and "apply"

1.2.4.2 Elasticsearch Configuration

The following configuration details cover a basic, functional installation of Elasticsearch. Some cases may require more robust configurations; users should contact Mitratech Support if unable to proceed through the following configuration.

The configuration file is the same for both Windows and Linux.

Many of these fields can be filled in during the installer; however, viewing the configuration file in a text editor will provide better visibility and custom configuration options.

1. Open the elasticsearch.yml file located in elasticsearch-1.7.4/config . This is the config file for Elasticsearch.

To access the configuration file on Linux, use a text editor such as vim or nano. For Windows users, simply open the config file with your text editor of choice.

The following properties **must be uncommented** and set:

cluster.name: Example Cluster

This is the name of the cluster that TeamConnect will look for:

discovery.zen.ping.multicast.enabled: false

Make sure that this is set to false; otherwise, Elasticsearch will use all nodes in the cluster for discovery which is bad practice.

Since multicast is disabled, ensure that you provide the entry points into the cluster by specifying the server locations here. The example below shows the configuration with two nodes. If you only have one server, simply put the one server without duplication or a "hostname2:port".

discovery.zen.ping.unicast.hosts: ["hostname1:port", "hostname2:port"]

This setting disables dynamic mapping so that Elasticsearch doesn't try to infer what the mapping for an object should look like. This may need to be manually added.

index.mapper.dynamic: false

Optional Properties

The following properties are commented out and set to default values (defaults listed with properties below). If you would like to enter custom values for these properties, simply comment them back in and replace the default value.

network.bind_host: 0.0.0.0

http.port: 9200

transport.tcp.port: 9300

Optional Properties Regarding Nodes

The following properties are optional, but may be useful for instances with multiple nodes:

This is the name of this specific node. If it is not set, Elasticsearch will simply choose from a list of names from the Marvel Universe.

node.name: Node1

This allows the node to be master eligible. The default is true. For further detail, please see the Master Node sections below.

node.master: true

This allows the node to store data. The default is true.

node.data: true

Set this to true ***if*** the server is in a Linux environment. It locks the memory for Elasticsearch so that the JVM does not start swapping.

bootstrap.mlockall: true

Master Nodes

Master nodes are nodes that are in charge of maintaining the state of the cluster. All nodes within the cluster report to the master node.

There can only be 1 Master Node, but there can be multiple Master Eligible Nodes that can take it's place were something to go wrong.

There is a known issue with having more than 1 Master Eligible Node called the Split Brain. The scenario plays out as follows:

There are 2 Master Eligible Nodes in the cluster.

A node loses communication (does not crash).

The lost node now thinks that it's in a cluster with no Master, so it elects itself as Master.

The communication is regained between the nodes, and there are now 2 Master Nodes.

Data is sent to one node for indexing, and search requests are sent to another node that does not hold the recently indexed information. This causes corruption of data.

In order to remedy this, Elasticsearch has a setting called discovery.zen.minimum_master_nodes. This allows you to set the minimum number of Master Eligible Nodes that need to be present for a Master Node to be elected. The idea is that if you have 3 Master Eligible Nodes, you can set this setting to "2". If one node gets lost, the cluster will still be up and running because it has 2 Master Eligible Nodes. The one node that lost communication will try to elect itself as master but won't be able to because it needs at least one more Master Eligible Node in the cluster to become Master.

A general rule of thumb is to have this setting set to (number of master-eligible nodes / 2) + 1.

This setting is useless if you have 2 Master Eligible Nodes in the cluster. Setting it to 2 means that if one node goes down, the entire cluster is inoperable. Setting it to 1 does not protect against split brain.

Dedicated Master Nodes

If the cluster becomes too large, then it becomes difficult for a data/master node combo to maintain the state of the cluster and perform the regular work of a data node. In these cases, it becomes useful to have Dedicated Master Nodes.

A Dedicated Master Node is a node that has node.data: false & node.master: true. Since a master node is only in charge of maintaining the state of the cluster, it is fairly lightweight; thus, it can be allocated less memory than a normal node. This reduces the risk of the Master Node crashing and making the cluster inoperable.

Because there is already a Dedicated Master Node, other nodes in the cluster can also be relieved of their burden as Master Eligible Nodes (i.e. node.data: true & node.master: false).

A good configuration for larger clusters is to have the proper number of Master Eligible Nodes that are Dedicated Masters, and an equal (or more) amount of data nodes underneath them with the

Master Eligible Nodes being the entry point into the cluster (discovery). An example configuration would be:

- 3 Master Eligible Nodes that are Dedicated Masters with discovery.zen.ping.unicast.hosts pointing to them.
- 6 Data nodes.
- discovery.zen.minimum_master_nodes: 2

1.2.4.3 TeamConnect Properties Configuration

Details for optimizing and configuring Global Search for TeamConnect can be found below. This information can be entered automatically into the TeamConnect properties file during the installer or manually entered into the teamconnect.properties file after installation.

Configuring the TeamConnect Properties File

Elasticsearch (Global search) parameters are able to be edited in the TeamConnect properties file found in the WEB-INF folder.

The server address, name of the cluster, and indexing frequency are **required** for search to function. The name of the cluster can be found in the Elasticsearch configuration file referenced here. The indexing frequency refers to how often (in seconds) TeamConnect will look for changes to existing, enabled indexed items and automatically update them. In the image below, the index will be scanned for modifications, additions, and entry removal every ten seconds.

The number of primary shards defaults to a value of '5'. Entering in any invalid character here will thereby result in 5 primary shards.

The number of replica shards defaults to a value of '1'. Entering in any invalid character here will thereby result in 1 replica shard.

Details on shards and replicas from Elasticsearch documentation:
An index can potentially store a large amount of data that can exceed the hardware limits of a single node. For example, a single index of a billion documents taking up 1TB of disk space may not fit on the disk of a single node or may be too slow to serve search requests from a single node alone.

To solve this problem, Elasticsearch provides the ability to subdivide your index into multiple pieces called shards. When you create an index, you can simply define the number of shards that you want. Each shard is in itself a fully-functional and independent "index" that can be hosted on any node in the cluster.

Note: *The concurrent requests and bulk size for flushing requests are not required and do not have default values.*

```
##============================================================================
## Elasticsearch Parameters
##============================================================================

## The location(s) of the Elasticsearch server(s) (hostname:port). If there are multiple
## servers, separate them using a comma. (e.g. hostname1:port, hostname2:port, hostname3:port)
elasticsearch.server.location=localhost:9300

## The name of the Elasticsearch cluster.
elasticsearch.server.cluster.name=Elasticsearch_1.7.4

## The frequency used for indexing of records in seconds.
elasticsearch.index.frequency=10

## The number of primary shards for the index.
elasticsearch.index.shards.primary=5

## The number of replica shards for the index.
elasticsearch.index.shards.replica=5

## The number of concurrent requests to be used for indexing.
elasticsearch.client.bulkProcessor.concurrentRequests=

## The bulk size for flushing requests to Elasticsearch in MB.
elasticsearch.client.bulkProcessor.bulkSize=

##============================================================================
```

1.2.4.4 Elasticsearch Best Practices

It is recommended to set the ES_HEAP_SIZE variable for Elasticsearch's allocated memory. This is because the ES_HEAP_SIZE variable sets both the -Xmx and -Xms variables to that value. This prevents the heap from resizing at runtime which is a very costly process.

Never allocate all available memory to Elasticsearch. A best practice for this is to only ever allocate 50% of your available memory to Elasticsearch. The other half should be reserved for Lucene caching, which uses ANY free memory on the machine. It loads segments (inverted indices) into memory for faster searching, so definitely keep that in mind when calculating Elasticsearch memory requirements.

- If multicast is disabled (which is suggested), it is a good practice to make the Master Eligible Nodes the discovery nodes.

- For TeamConnect instances that are 100GB, it usually suffices to have a single node cluster where the node acts as both a master and data node.

- For instances that require more nodes, or if the client wishes to utilize Elasticsearch's shard replication for backup and high availability, then a larger cluster is usually preferred.

In order to utilize shard replication, you simply need to start up another node within the same cluster (don't forget to specify the entry point into the cluster for unicast).

Please read the Master Eligible Nodes details here for more information.

1.2.4.5 Elasticsearch FAQ

Question	Answer
Is the full-text search in TeamConnect the same as Elasticsearch?	No. Elasticsearch provides global searching based on an index stored on a dedicated server. Full text search can still be achieved directly from your database, though it may not be as robust as the new global search.
How many shards should I use?	The default value of 5 primary shards should be more than sufficient for almost all clients.
Can Elasticsearch work with clustered instances? What infrastructure is needed on premise?	Yes. On-premise clients will need one instance of TeamConnect Enterprise (clustered or not) and one instance of Elasticsearch.
What hardware requirements are suggested for Elasticsearch?	The suggested Elasticsearch hardware requirements are flexible depending on each use case. General requirements include: 8 GB RAM (most configurations can make do with 4 GB RAM)
Are there words which Elasticsearch will not search on?	Elasticsearch/Lucene has the following words filtered out of searches: "a", "an", "and", "are", "as", "at", "be", "but", "by", "for", "if", "in", "into", "is", "it", "no", "not", "of", "on", "or", "such", "that", "the", "their", "then", "there", "these", "they", "this", "to", "was", "will", "with"

1.2.5 Business Objects Support

TeamConnect 5.0 is certified with Business Objects Enterprise XI 4.0.

Business Objects Single Sign-on (BOSS) is also certified for the above version. You can deploy BOSS on Apache Tomcat, Oracle WebLogic, or IBM WebSphere. Deploying BOSS XI and Configuring URLs specifies which versions of these application servers are fully supported for BOSS.

Note: *If you are using Business Objects, ensure that the database server for TeamConnect and the computer that runs Business Objects both use the same character set.*

Note: *TeamConnect 5.0 also has a native, feature-rich reporting capability. For more information, see Reports.*

1.2.6 Reporting

The native TeamConnect reporting feature requires that client computers have Adobe Flash version 10.1 or higher installed. TeamConnect version 5.0 was certified with version 11.4.

For exporting report results to PDF format, the client computer must have Adobe Acrobat Reader version 9.4.0 or higher installed.

1.2.7 Licensing

TeamConnect 5.0 provides a licensing file that allows active users to log-in and use TeamConnect. The number of users per license is based on your agreement with Mitratech.

Important: *If you are upgrading from TeamConnect 4.0 or later to TeamConnect 5.0, you have a current license file and do not need to upload/activate it again.*

Once you install TeamConnect, contact Mitratech Support for your company's licensing file. The license file must be uploaded before users can log in.

Before you Begin

Contact Mitratech Support to obtain the license file and save it to an accessible location.

To upload and activate the license file

1. Log in to TeamConnect.
2. Click the **Admin** tab, and the 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.

Expired License

If your license expires, please contact Mitratech Support for a new license. Details about expired licenses can be found in the logs.

For details on licensing concepts and how to upload a license, refer to System Administration for more information.

1.2.8 XML Layer Support

The XML Layer tools and documentation contained in this version are for backward compatibility usage only. This is to support existing XML Layer integrations developed by existing clients in TeamConnect 2.x who are now upgrading to version 5.0. The XML Layer is not supported for new clients or recommended for new integrations by existing clients who have upgraded to version 5.0. For new integration development in version 5.0 please refer to the TeamConnect Web Services integration layer.

1.2.9 TeamConnect Installer Requirements

The TeamConnect Installer works in a Microsoft Windows environment. Once the TeamConnect Installer is run, you must deploy the resulting WAR or EAR file to the application server.

Before running the TeamConnect Installer:

- The TeamConnect tablespace and appropriate users should already exist.
- Your machine must be connected to the internet.
- Your machine must have one of the Java environments listed in Application and Web Servers.

The TeamConnect Installer may require in excess of 2 GB of Java heap space, particularly when running against large databases. It is recommended that you use a machine with at least 3 GB of available heap space to run the installer.

1.2.10 Upgrade Considerations

The current upgrade paths supported are from: TeamConnect 4.2.

If you are using a version of TeamConnect prior to TeamConnect 4.2, please see the *TeamConnect 4.2 Installation Guide* for information on this release.

1.2.10.1 Optional Script to Support Data Warehouse and Localization

When upgrading from a version prior to 3.3 SP1, the database schema is altered, but not to the extent necessary to completely support localization. If your organization is upgrading and wishes to support localization, you must run an optional database script after the upgrade process is complete.

Even if your organization does not expect to use localization, you need to run this script if:

- You use Data Warehouse, and
- You expect to add new custom object types to your application design in the future.

That's because some Data Warehouse tables related to custom objects have datatypes that are compatible with localization, but not compatible with the datatypes that existed prior to version TC3.3 SP1. Running the script will resolve this incompatibility. You must drop and recreate the Data Warehouse after running the script.

This script requires exclusive use of your database while it is running. It increases database size by 2% to 4% in internal testing done by Mitratech. Nearly all VARCHAR columns have their datatypes changed to NVARCHAR2 (Oracle) or NVARCHAR (SQL Server.)

Contact Mitratech Support to obtain this script.

1.2.10.2 Upgrading to Legal and Financial Management

For more complex installations across several modules, contact Mitratech documentation at documentation@mitratech.com for the Module Upgrade Detailed Matrix.

Use this page if you have a version of Legal Matter Management prior to 5.0 and are upgrading to the following versions:

- TeamConnect Enterprise 5.0
- TeamConnect Legal Matter Management 5.0
- TeamConnect Financial Management 5.0

Following these steps to upgrade:

1. Upgrade to TeamConnect Enterprise 5.0.

   Note: *You may run into validation errors that you need to correct before continuing.*

2. Restart your application server.
3. Upgrade to Legal Matter Management 5.0.

   After correcting conflicts, re-run the Upgrade Toolkit.

   Note: *If the tool fails, review the error log to identify the necessary fix and report errors to Mitratech support.*

4. Restart your application server.
5. Install Financial Management 5.0 and any other modules from the **About** page. Each time you install a new product, restart you application server.
6. Configure Financial Management.

7. Proceed upgrading as desired.

1.2.10.3 Validate Your Data Before Running the Installer

When you are upgrading an existing TeamConnect database, particularly one that was once used with a TeamConnect 2.x version, the database may contain duplicate detail records that would interfere with the upgrade to version 5.0. Such duplicates may have arisen from earlier conversions that disabled validation rules while the conversion was running.

To check for this possibility, you must run a validation SQL script, which will provide information about whether duplicates exist. If they do exist, you must run a second small SQL script, doing a separate run for each duplicate record that must be removed. Follow these steps:

1. In your installation media, locate the validation script that is appropriate to your database server:
   - `\validation\TC33_Upgrade_Data_Validation\MSSQL\MSSQL_TC33_upgrade_data_validation.sql`
   - `\validation\TC33_Upgrade_Data_Validation\Oracle\Oracle_TC33_upgrade_data_validation.sql`
2. Run this SQL script against your TeamConnect database and observe the results. If the result set is empty, you have no duplicate detail records and you do not need to perform the rest of these steps.

   For Oracle, use SQL*Plus to run this script, and look for the result set in file TC33_Data_Validation_report.log, in the same folder as the script file. For SQL Server, the result set is returned on-screen, although you can direct the result set to a file if you wish, depending upon the query tool that you use.
3. If the validation script returns results, you must manually remove the duplicate records. You will need to supply the values of four parameters, which are found in the validation script's result set:
   - TABLE_NAME
   - MASTER_PRIMARY_KEY AND
   - FIELD_PRIMARY_KEY
   - DETAIL_PK_to_keep
4. If you are using Oracle, skip to step 6. If you are using Microsoft SQL Server, execute SQL script

   `\validation\TC33_Upgrade_Data_Validation\MSSQL\MSSQL_Parametrized_To_Remove_Duplicate_Details.sql`

   This script will create a procedure named **remove_duplicate_detail_pro**, which you will execute repeatedly in the next step.
5. For each row in the result set of step 2, execute this procedure, supplying four parameters as shown in this example:

   ```
   exec remove_duplicate_detail_pro 'E_CONT_DETAIL_TEXT_VALUE ', 703 , 604 , 502 ;
   ```
6. If you are using Oracle, for each row in the result set of step 2, run script

```
\validation\TC33_Upgrade_Data_Validation\Oracle
\Oracle_Parametrized_To_Remove_Duplicate_Details.sql
```

supplying the four parameters in the syntax shown by the example in step 5.

7. After you have run the procedure for each row in the validation script's result set, manually drop table **BAD_DATA_RPT**, then do a final check by running the validation script from step 1 again. This time, there should be no rows in the result set.

1.2.10.4 Data Warehouse

If you use **Data Warehouse**, which has a denormalized database containing data from the TeamConnect production database, consider dropping the data from that denormalized database and repopulating it after you upgrade. In some cases, upgrade scripts for TeamConnect may repair incorrect data in the production database but the repaired data is not automatically moved to the denormalized database. Clearing and repopulating the denormalized database resolves this potential issue.

Note: *Only Data Warehouse version 5.0 will work with TeamConnect version 5.0. Earlier versions of Data Warehouse must be upgraded to 4.0.*

1.3 Running the Installer

The TeamConnect Installer is a wizard that completes many steps of your installation or upgrade process for you. These are the main tasks completed by the TeamConnect Installer:

1. **Preparation**—In this section, specify where the TeamConnect installation files should be copied. The Installer will detect whether the local PC has enough disk space for the unzipped installation files (including the application, utilities, and documentation files).
2. **Extracting Files**—In this section, the Installer copies TeamConnect installation files to a local folder.
3. **Configuration**—In this section, the Installer allows screen-based editing of the TeamConnect application configuration information. Ultimately this configuration information is stored in the teamconnect.properties file. If you have already run the Installer and have a local TeamConnect application file or teamconnect.properties file, you can use the Installer to update the configuration information from those sources. The Installer packages the latest configuration information (in the teamconnect.properties file) into a new TeamConnect application file(s).

   Elasticsearch parameters are also set via the configuration options in the installer. Users will be able to defer their configuration until after installation; in this case, the parameters will need to be put in manually. For help, see Manually Configuring Search Parameters.
4. **Create/Upgrade Database**—In this section, the Installer either creates a new TeamConnect database schema or upgrades an existing database schema. Refer to the Release Notes for upgrade requirements.

TeamConnect Data Warehouse has its own separate installer programs, which are not discussed here.

Available Updates

TeamConnect Legal Matter Management, Service of Process (SOP), and Collaborati Spend Management (CSM) are installed through the **Available Updates** feature in the administrative interface of TeamConnect, which will be accessible to you after you install TeamConnect version 5.0. Detailed installation of CSM is not discussed here, but there are some important dependencies and cautions about CSM installation in the CSM installation and admin guides.

After your TeamConnect installation is complete, you may wish to consider installing optional features, as described in Web Folders.

1.3.1 Before You Begin

For customers with older versions of TeamConnect, this chapter includes procedures for upgrading from an older version of TeamConnect to TeamConnect 5.0. Only the versions in the list below are eligible for upgrading. The installer will recognize that you have one of these versions and will run the appropriate procedures.

Important: *This assumes that you have modules to upgrade along with TeamConnect Enterprise. Instances that solely seek to upgrade TeamConnect Enterprise can be upgraded from any previous TCE version after TCE 3.3.*

- TeamConnect 4.2+

If you are using a version of TeamConnect older than TeamConnect 4.2, you must first separately upgrade to 4.2 before running the tasks in this chapter. Instructions for upgrading to 4.2 can be found in the *TeamConnect 4.2 Installation Guide* and *Release Notes.*

Important: *When performing an upgrade, there are several important tasks that must be completed before running the installer. Refer to Upgrade Considerations. Regardless of your previous version, you must back up your database before beginning the upgrade.*

1.3.1.1 TeamConnect Installer

Please note the following before running the TeamConnect Installer:

- Review the requirements for running the TeamConnect Installer in Installation Requirements. The TeamConnect Installer must be run from a Windows machine.
    - If you are deploying TeamConnect to a Windows-based application server, you may choose to run the TeamConnect Installer directly on the server. If you run the TeamConnect Installer on a different machine, you must transfer the output of the TeamConnect Installer program (either a TeamConnect .war or TeamConnect .ear file, and TeamConnect Online Help .war file) to your application server.
    - If you are deploying TeamConnect to a Unix or Linux-based server, first run the Installer on a Windows machine. When finished, you must transfer the output of the Installer program (either a TeamConnect .war or TeamConnect .ear file, and TeamConnect Online Help .war file) to your TeamConnect application server.
- Make sure that your PATH environment variable includes the **bin** directory of a valid Java Development Kit (JDK).
- The instructions for running the TeamConnect Installer are based on the TeamConnect Installer Form. Print this form and collect the required system data before actually running the TeamConnect Installer.

- Some parts of the TeamConnect Installer, particularly those that run database scripts, may take several minutes to complete. In this version, database scripts related to Account records are complex, so if your instance has many Account records, the installer may run noticeably more slowly than usual. No progress messages appear during this time. Do not presume that the TeamConnect Installer is "hung up" unless it has been running for at least thirty minutes. Check the database server to see whether database activity is still being logged against the installer.
- If you are upgrading an existing TeamConnect database on Oracle, the upgrade includes a script that re-indexes Document records. If your database contains a very large number of Document records, this script may require several hours of runtime. You can postpone the running of this script, allow the installer to finish, then run the script later, manually.

If You are Using Oracle

An optional script is available named `ORACLE_upgrade_2_FullTextSearch_Reindexing.sql`. The purpose of this script is to resolve an issue with full-text search in Document records when the search string includes a hyphen. Until the script is run, such searches may return incorrect results.

1.3.1.2 Manually Configuring Search Parameters

Details for optimizing and configuring Global Search for TeamConnect can be found below.

Configuring the TeamConnect Properties File

Elasticsearch (Global search) parameters are able to be edited in the TeamConnect properties file found in the WEB-INF folder.

The server address, name of the cluster, and indexing frequency are **required** for search to function.

The number of primary shards defaults to a value of '5'. Entering in any invalid character here will thereby result in 5 primary shards.

The number of replica shards defaults to a value of '1'. Entering in any invalid character here will thereby result in 1 replica shard per primary shard.

Note: *The concurrent requests and bulk size for flushing requests are not required and do not have default values.*

```
##============================================================================
## Elasticsearch Parameters
##============================================================================

## The location(s) of the Elasticsearch server(s) (hostname:port). If there are multiple
## servers, separate them using a comma. (e.g. hostname1:port, hostname2:port, hostname3:port)
elasticsearch.server.location=localhost:9300

## The name of the Elasticsearch cluster.
elasticsearch.server.cluster.name=Elasticsearch_1.7.4

## The frequency used for indexing of records in seconds.
elasticsearch.index.frequency=10

## The number of primary shards for the index.
elasticsearch.index.shards.primary=5

## The number of replica shards for the index.
elasticsearch.index.shards.replica=5

## The number of concurrent requests to be used for indexing.
elasticsearch.client.bulkProcessor.concurrentRequests=

## The bulk size for flushing requests to Elasticsearch in MB.
elasticsearch.client.bulkProcessor.bulkSize=

##----------------------------------------------------------------------------
```

TeamConnect Properties

1.3.2 Installing TeamConnect

To run the installer, start program **setup.exe** and follow the prompts. For help with the Installer, click the Help button on any Installer screen or see the sections below.

WebDAV and WebLogic

If you use a WebLogic application server, it is important to edit the server's **config.xml** file to disable basic authentication. If basic authentication remains enabled, it interferes with WebDAV access and Screen Designer. A WebLogic login dialog pops up before the actual TeamConnect login dialog. End users confuse this WebLogic dialog with the TeamConnect one and enter TeamConnect credentials into it, which causes errors.

Disabling basic authentication prevents this first dialog from appearing and the end user sees only the TeamConnect login dialog.

Introduction

The TeamConnect Enterprise installer can be run repeatedly. On the very first run of the installer, you must choose "Extract files to install or upgrade TeamConnect Enterprise." On any subsequent runs of the installer, you can choose "Continue using previously extracted files." Normally you would want to use the previously extracted files, because they contain information (such as configuration values) that is customized to your installation. However, you may still choose "Extract files to install or upgrade TeamConnect Enterprise." If you do, you can run the installer as if you were running it for the first time.

Configuration

On this screen, you tell the installer how to modify TeamConnect Enterprise's configuration files. Your choice will vary depending on whether you have an existing version of TeamConnect Enterprise already installed, and whether that version is the current version or an older one. If there is no existing version of TeamConnect Enterprise, choose "Manually Configure TeamConnect Enterprise" and fill out the configuration information in screens that appear later. If there is an older version of TeamConnect Enterprise installed, and you want to use the configuration information that is contained in that older version, you have two choices. You can choose "Import Settings (from .war archive)" or "Import Settings (from .xml or .properties files)", depending on which kind of file is available from the older version. The installer will read those files and will extract their configuration information. That information will appear in a series of screens that appear later, and you have the opportunity to accept the existing values or to edit those values. If there is an older version of TeamConnect Enterprise installed, and you don't want to use the configuration information that is contained in that older version, choose "Manually Configure TeamConnect Enterprise", then fill out the configuration information in screens that appear later. If there is a current version of TeamConnect Enterprise installed, and you are simply re-running the installer, your choice depends on what you did during the earlier run of the installer.

If you already filled out configuration information in the earlier run, and you don't need to change anything, choose "Skip this step".

If you did fill out configuration information in the earlier run, but you now want to change some of the information, choose"Manually Configure TeamConnect Enterprise", then fill out the configuration information in screens that appear later.

If you did not fill out configuration information in the earlier run , choose "Manually Configure TeamConnect Enterprise", then fill out the configuration information in screens that appear later.

After making this selection, you will be prompted to select your application server. For users not using WebLogic, select others.

Select your database type on the following screen (SQL Server or Oracle) and whether or not you would like to use data source. If you choose YES, the database connection mechanism will be defined by the application server. You will then add the data source name, ensuring that it matches the one listed in the application server.

Create/Upgrade Database

Completed

1.3.3 Running Database Drop Scripts

The TeamConnect installation media includes scripts for deleting a TeamConnect database instance.

Caution: *These procedures are for reference only and should only be performed if you made a mistake with your initial TeamConnect database schema creation while running the Installer and you want to delete it.*

To delete a TeamConnect database on Oracle:

1. Open and log into your database application, using SQL*Plus.
2. From your TeamConnect installation path (**C:/Program Files/Mitratech/TeamConnect/database/drop/oracle** directory, where **/Program Files/Mitratech** might be different if you specified a different installation folder during the Installer), run the following script:

   TC_drop_tables_oracle.sql

To delete a TeamConnect database on SQL Server:

1. Open and log in to your database application, using the vendor's dynamic SQL tool.
2. From your TeamConnect installation path (**C:/Program Files/Mitratech/TeamConnect/database/drop/mssql** directory, where **/Program Files/Mitratech** might be different if you specified a different installation folder during the Installer), run the following script:

   TC_drop_tables_sqlserver.sql

1.3.4 Troubleshooting

- If you quit the TeamConnect Installer during the installation procedure, and database schema creation scripts started but did not complete, database changes will not be backed out. If you rerun the Installer and select the Reinstall option, you will be able to drop any existing database tables and rerun the create scripts on a clean database.
- If an error occurs during the TeamConnect installation process, an error message will be displayed. For additional details, click **Detail**. To quit the installation, click **Abort** and a summary of performed actions should display with a link to the installation error log. The log can be sent to Mitratech Support for troubleshooting assistance.

Note: *The TeamConnect Installer repackages the TeamConnect .war and .ear file with your latest* ***teamconnect.properties*** *configuration settings but you need to manually transfer the .war/.ear file to your application server and deploy the file.*

1.4 Deploying

This chapter details the deployment steps for TeamConnect 5.0.

Important: *Each release of TeamConnect includes some deployable files that are named differently from one version to the next, since the version number is included as part of the file name. You can always determine the exact file names for your version by examining the file*

***TCE_readme_installer_contents.txt**, which is in the root directory of the installation media. Refer to that file for the exact names of the TeamConnect .war file, the TeamConnect .ear file, and the TeamConnect online help .war file. Those files will be referred to in this publication by their generic names, not their explicit version-dependent names.*

In order to proceed with deployment, you should have already run the Installer, as described in Running the Installer. The Installer should have output a .war and .ear file, which you will now deploy on your application server.

***Note:** .war Files can only be deployed on Apache Tomcat. Other application server platforms require you to deploy .ear files.*

When you are finished with the steps in this chapter, see Verifying Installation to verify that you have installed TeamConnect successfully.

These instructions assume that your application server is successfully installed. For all details regarding the application server installation, please refer to the vendor's documentation.

1.4.1 Special Considerations for Microsoft Outlook

If you are deploying on WebLogic, and you plan to use Microsoft Outlook to create TeamConnect Appointment records, you must include an additional argument in the command that you use to start the server that is hosting TeamConnect. The following default argument is in teamconnect.properties:

```
-
Djavax.xml.soap.MessageFactory=com.sun.xml.internal.messaging.saaj.soap.ver1_1.SOAP
MessageFactory1_1Impl
```

If you have a version of Weblogic earlier than 10.3.6, use the following argument:

```
-
Djavax.xml.soap.MessageFactory=com.sun.xml.messaging.saaj.soap.ver1_1.SOAPMessageFa
ctory1_1Impl
```

1.4.2 Creating and Configuring Your Domain (WebLogic only)

This section describes how to set up your WebLogic domain to deploy TeamConnect.

***Important:** If you are not deploying TeamConnect on WebLogic, skip this section and go to Preparing for Deployment.*

The WebLogic server configuration consists of domains, each having one administrative server running the administrative console, dynamically distributing TeamConnect application information to multiple managed servers. All application management occurs on the administrative server. To install TeamConnect, you must create the domain, configure the administrative server, and deploy TeamConnect to the managed servers.

All WebLogic server activities take place within a domain, for example, the Mitratech domain. You may deploy TeamConnect on a single managed server or on a cluster of managed servers. You can deploy TeamConnect to the administrative server, but it is not recommended, except as a testing

environment where application performance is not a factor. This document assumes that you are not deploying TeamConnect to an administrative server.

WebLogic comes integrated with its own web server, which is suitable for serving TeamConnect pages or load balancing a cluster. However, if you wish to access TeamConnect through your company's primary web server, you must configure a plug-in for that web server. For details, see Web Proxy Settings.

Pre-installation Requirements

Aside from the general system requirements listed in Installation Requirements, you should ensure that:

- All of the machines on which you run WebLogic servers (administrative or managed) are using a version of WebLogic that is specified in Application and Web Servers.
- You should use the Sun JVM for the Production Mode.
- Your **PATH** environment variable should be set up properly on each of your servers.
- Licensing must be correctly configured on your WebLogic installation.
- Modify the instructions in this document to replace the **WL_HOME** variable with the appropriate directory.
- If you have an Oracle database, modify the instructions in this document to replace the **ORACLE_HOME** variable with the appropriate directory.
- To avoid problems displaying special characters such as the Euro currency symbol and British pound currency symbol, be sure that this environment variable is set prior to the command line that starts the WebLogic application server, or add this option to the server start arguments in Managed Server:

  ```
  set JAVA_OPTIONS=-Dfile.encoding=UTF-8
  ```
- Be sure to modify the instructions according to your operating system where necessary.

Recommended Values for Domain Configuration

You can create domains using the Domain Configuration Wizard that walks you through the steps and provides the necessary instructions. In this documentation, we only provide the recommended values that you may use when creating a Mitratech domain (you may use another name for your domain, if you wish):

Recommended Values for Admin Server Configuration

Field	Value
Server Name	TCAdmin
Server Listen Address	Leave this field blank.
Server Listen Port	7001

Server SSL Listen Port	7002

Recommended Values for Managed Server Configuration

Field	Value
Server Name	TCManaged
Listen Address	Enter the hostname or IP address of the machine on which you run this server.
Listen Port	4001
SSL Listen Port	4002

After you finish creating and configuring a domain for use with TeamConnect, you must install the TeamConnect .ear file. See Preparing for Deployment.

1.4.3 Preparing for Deployment

Before deploying TeamConnect, there are several steps you must take to prepare it for deployment. You need to install a TeamConnect .ear file, assemble the modules of the application, set deployment descriptor properties, and generate code for deployment.

Important: *You must allocate no less than 2048 MB of memory for the application server.*

Before you begin, please note the following:

- This document assumes that the environment variable for **ORACLE_HOME** has already been configured on your system, or that you are aware of its actual directory location so that you may make the appropriate substitutions in the instructions.
- Make sure to modify the instructions according to your operating system where necessary.

To prepare the TeamConnect application for deployment:

1. Before installing TeamConnect, please ensure that your system meets the minimum system requirements for TeamConnect. For details, see Installation Requirements.
2. Create a directory on your local machine and name it TeamConnect.
3. Copy the TeamConnect .war or TeamConnect .ear file from the **C:/Program Files/Mitratech/TeamConnect/programs/teamconnect** directory (from the computer where you ran the TeamConnect installer, where **C:/Program Files/Mitratech** might vary if you specified a different installation path) to the application server /TeamConnect directory (or location from which to deploy the TeamConnect application).

If you set your **PATH** environment variable to include the Java **bin** directory, you are able to run executables from the JDK without using the entire path.

1.4.4 Configuring Clustered Servers

TeamConnect's clustering feature allows transparent failover and recovery for end users. In the event of a node's failure, each session on that node is transferred to another node, preserving even the uncommitted edits in the session's browser. In the case of a recovered session, all work is saved except the uncommitted edits in the browser of the expired session.

Each application server vendor implements clustering and cache synchronization in a different way. This section does not discuss details of a specific vendor's implementation, but describes properties of TeamConnect that are common across all implementations.

Configure your application servers for session replication across all nodes.

All nodes must run exactly the same version of TeamConnect.

Note: *Clustered servers are not supported with JBoss.*

1.4.4.1 Editing teamconnect.properties

The TeamConnect installer makes many changes to teamconnect.properties when it runs but to configure clustering you will need to make additional manual edits. Each node in a cluster runs an identical copy of TeamConnect, so the properties file needs to be edited only once per cluster.

Here are the properties that require editing, with their definitions:

Clustering Properties

Property Name	Definition
sync.enabled	Must be **YES** to enable clustering
sync.channel	Use a unique character string of your choice. The purpose of this property is to avoid the situation where multiple clusters of application servers are running on the same network, and one cluster attempts to synchronize cache with a different cluster Each cluster should have a different value in this property.
sync.rmi.port	Enter a port number that will be used by all nodes in the cluster. The RMI registry is started in-process by TeamConnect.
sync.multicastGroupAddress	Optional. Default is 226.10.12.64
sync.multicastPort	Optional. Default is 3121

1.4.4.2 Special Considerations for IMAP

If you are using IMAP to integrate with an Exchange server, you must configure TCP/IP load balancing. IMAP does not use the failover or recovery capabilities of the main TeamConnect clustering feature.

1.4.5 Configuring Database Connection

The following section describes two ways of connecting TeamConnect to a database:

- Using a Data Source
- Using an EclipseLink Connection Pool

Using a Data Source

This is the recommended method of connecting to the database if you are using a JBoss, WebSphere, or WebLogic application server. If you are using a Tomcat application server, see Using an EclipseLink Connection Pool.

Once the data source is set up, you only need to know the name of the data source in order to deploy TeamConnect, rather than all of the database connection parameters.

To use a data source as your method of database connection for TeamConnect, you must first configure a JDBC data source in your application server. For details about how to configure a data source, consult your application server documentation, and see driver information in Application and Web Servers.

The following table provides the recommended settings for configuring your data source in WebLogic. For WebSphere settings, see the WebSphere Data Source Configuration Settings table.

WebLogic Data Source Configuration Settings

Setting	Recommended values
Database Driver	See driver information in Application and Web Servers.
Data source name	`TeamConnect` The data source name defaults to TeamConnect.

JNDI name	`jdbc/DATASOURCE`
	where `DATASOURCE` is the name of the data source. For example:
	`jdbc/TeamConnect`

The following table provides the recommended settings for configuring your data source in WebSphere.

WebSphere Data Source Configuration Settings

Setting	Recommended values

Database Driver	See driver information in Application and Web Servers.
Data source name	`TeamConnect` The data source name defaults to TeamConnect.

Using an EclipseLink Connection Pool

Connection pools can be used with the Tomcat and JBoss application servers. You can use EclipseLink's internal connection pool to connect TeamConnect directly to the database. When using this method, you must know your database connection parameters and write them on your Installer Form. The Installer, which runs prior to deployment, places those parameters into the TeamConnect properties file. You must ensure that your database server's configuration matches the parameters that were entered into the Installer earlier.

Depending on your database server, you must also put a JDBC driver file in the **/lib** directory for Tomcat or JBoss. See Application and Web Servers for driver information.

After you have configured the database connection, the TeamConnect application is ready to be installed on an application server by using the administrative console.

1.4.6 Configuring the Java Virtual Machine

TeamConnect runs much more effectively when the application server's Java Virtual Machine (JVM) has been configured properly. A common way to do this is by specifying command-line arguments in the script file that starts the application server itself, but there are other methods of changing the JVM configuration that differ by brand of server. See the instructions for Sun JVM and IBM JVM to learn how to change the configurations.

1.4.6.1 Sun JVM

Note: *For the JBoss application server, see Deploying TeamConnect .ear on JBoss. Use those configuration values instead of the ones in this section.*

Command-line arguments specify the memory allocation for the server.

`-Xms???m` determines the minimum amount of memory the server reserves. If that amount is not available, the application server does not start.

`-Xmx???m` determines the maximum amount of memory the server reserves, if available.

Important: *Mitratech recommends setting the maximum and minimum memory arguments to the same number, for example: -Xms2048m -Xmx2048m*

1. Insert the numeric amounts to allocate as much memory as you can for the application server. Make sure that it is at least -Xms2048m -Xmx2048m.
2. Directly after the above recommended memory flags, add the following parameters:

```
-XX:PermSize=256m
-XX:MaxPermSize=256m
```

3. Ensure that the final string looks like the following (all within one line):

```
-Xms2048m -Xmx2048m -XX:PermSize=256m
-XX:MaxPermSize=256m -XX:NewSize=100m
```

Note: *These are recommended values for 32-bit operating systems. If you are using a 64-bit operating system, you can alter these values to match your configuration.*

Optional Recommended Arguments

There are some optional, recommended arguments that can help tune performance. To apply them, append the following arguments to the ones listed above. All of the arguments should be listed on a single line:

```
-server -XX:+HeapDumpOnOutOfMemoryError
-XX:+UseConcMarkSweepGC
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000
-verbose:gc
-Dcollaborati.connection.readTimeout=<number of minutes>
```

Most of these advanced Java Virtual Machine arguments relate to garbage collection in memory. The last argument shown relates to Collaborati. Default value is 60 minutes and it is rare that you would need to change it. If you anticipate dealing with unusually large volumes of data through Collaborati, you should set this argument to a larger value, to avoid timeouts during synchronization. An example of large volumes would be timekeeper counts of 2000 or more.

For more details on performance tuning your Java environment and garbage collection, see your Sun documentation.

1.4.6.2 IBM JVM

Heap size should be set to 4000 (MB) for both initial and maximum size, presuming a load of 800 users. If your load is smaller you may be able to reduce the heap size proportionately.

Optional Recommended Arguments

There is a recommended argument that can help tune performance. Specify the following argument in the command line:

```
-Xgcpolicy:gencon -Xmn1000m
```

If you are running a heap size different than 4000 then you should adjust -Xmn correspondingly to stay at 25% of the heap size.

Another argument relates to Collaborati Spend Management (CSM). The syntax of the argument is:

```
-Dcollaborati.connection.readTimeout=<number of minutes>
```

Default value is 60 minutes and it is rare that you would need to change it. If you anticipate dealing with unusually large volumes of data through CSM, you should set this argument to a larger value, to avoid timeouts during synchronization. An example of large volumes would be timekeeper counts of 2000 or more.

1.4.7 Deploying TeamConnect

This section includes instructions for deploying TeamConnect on each of the supported application servers. Deployment should always incorporate a full undeployment of the existing TeamConnect instance and a deployment of the new instance. Do not use your application server's update facility. Also, the web server cache should be cleared after the new instance is deployed.

Refer to the procedure for your application server:

- Deploying the TeamConnect .war on Apache Tomcat
- Deploying TeamConnect .ear on WebSphere
- Deploying TeamConnect .ear on WebLogic

1.4.7.1 Deploying the TeamConnect .war on Apache Tomcat

To set the JVM configuration, refer to Sun JVM or IBM JVM, depending on which JVM your application server is using. Construct a string of JVM arguments with the values recommended in those sections.

To deploy TeamConnect on Apache Tomcat

1. On the Tomcat manager web page, in the Deploy section, browse to and select the TeamConnect .war file.
2. Click **Deploy**.

   A message appears near the top of the screen, indicating that the application has been deployed.

   In the list of applications (or contexts), in the **Running** column, the value shown should be **true** for the TeamConnect application.

   Also, the TeamConnect context root should appear in the **Display Name** column.
3. Deploy the TeamConnect online help either on your web server or your application server.

TeamConnect should be running and available for login. For details, see Verifying Installation.

1.4.7.2 Deploying TeamConnect .ear on WebSphere

TeamConnect should be installed by using the WebSphere administrative console and the related WebSphere documentation.

You must first adjust the configuration file for correct database functionality in WebSphere:

1. For TeamConnect to connect to Data Warehouse, remove the following file from the TeamConnect .war file:

   WEB-INF\lib\validation-api-1.0.0.GA.jar

2. Make an adjustment to file **teamconnect.properties**. Extract the TeamConnect .war file from the TeamConnect .ear file, explode the .war file, and locate teamconnect.properties. Edit the file to remove this line:

   ```
   ## appserver.datasourcePrefix=java:comp/env/
   ```

3. Repackage the teamconnect.properties file into the .war file. Repackage the .war file into the TeamConnect .ear file.

4. Install the TeamConnect application as WebSphere advises, then configure the application as described below before starting the application.

To set the JVM configuration:

1. Refer to IBM JVM. Construct a string of JVM arguments with the values recommended in that section.

2. Set the recommended options using the WebSphere administrative console.

1.4.7.2.1 Other Configuration Changes

You must ensure that the class-loading policy used by WebSphere is "parent last". Use WebSphere Console to set this policy for the TeamConnect module that you have deployed.

You can improve the performance of TeamConnect by adding the optional command line argument `-Xgcpolicy:optavgpause` to the **Generic JVM arguments** field on the **Java Virtual Machine** page of the WebSphere Application Server settings. For more details on tuning your Java environment, see your vendor's documentation.

You can include optional JSP engine parameter **keepgenerated** with value **true** in your module's configuration. This retains Java files after the translator is finished, which can aid in debugging. Refer to WebSphere documentation for details on this and other details on tuning your Java environment.

SSL and Collaborati Spend Management

If you use the Collaborati Spend Management feature, and your application server or web server uses Secure Sockets Layer (SSL), you must perform some additional configuration steps on your WebSphere application server.

1. In the `utilities\config\ssl` subfolder of your installation media, locate files `versign.intermediate.cer` and `verisignroot.cer`. Copy those files to a folder that is accessible to the WebSphere Console.

2. Use the WebSphere Console to add two new entries to the list of Signer Certificates in the default node truststore. Add each of the files named above, and use data type "Base64-encoded ASCII data".

Restart Required

After performing any of the WebSphere configuration changes described above, you must restart the application server for the changes to take effect.

1.4.7.3 Deploying TeamConnect .ear on JBoss

To set the JVM configuration

1. Set Java options prior to starting the application server. Use the **EXPORT** command or some equivalent method to set the options (all on one command line):

   ```
   set JAVA_OPTS="-Xms2048m -Xmx2048m -XX:MaxPermSize=512m -XX:
   +UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled -XX:
   +HeapDumpOnOutOfMemoryError -Dsun.rmi.dgc.client.gcInterval=3600000 -
   Dsun.rmi.dgc.server.gcInterval=3600000" -
   Dcollaborati.connection.readTimeout=<number of minutes>
   ```

   The last option relates to Collaborati. Default value is 60 minutes and it is rare that you would need to change it. If you anticipate dealing with unusually large volumes of data through Collaborati, you should set this option to a larger value, to avoid timeouts during synchronization. An example of large volumes would be timekeeper counts of 2000 or more.

2. Use JDK version 1.7 or higher.
3. Use the default Java security policy.

To deploy the application:

1. Make an adjustment to file **teamconnect.properties**. Extract the TeamConnect .war file from the TeamConnect .ear file, explode the .war file, and locate teamconnect.properties. Edit the file to remove this line:

   ```
   ## appserver.datasourcePrefix=java:comp/env/
   ```

2. Repackage the teamconnect.properties file into the .war file. Repackage the .war file into the TeamConnect .ear file.
3. Copy the TeamConnect .ear file into the **/deploy** folder of the production server.

1.4.7.4 Deploying TeamConnect .ear on WebLogic

When deploying TeamConnect, you must start and access the administrative console. This involves first starting the WebLogic administrative server, then opening the administrative console. Make sure you know how to do that, as well as how to start and shut down managed servers. If necessary, refer to the WebLogic documentation.

Important: *WebLogic is case-sensitive. Use the correct case when you type file names, directory names, or other information.*

To set the JVM configuration:

1. Refer to Sun JVM. Construct a string of JVM arguments with the values recommended in those sections.
2. Use the administrative console to place those values into WebLogic's options.

To deploy TeamConnect

1. Start the WebLogic Server administration console for the domain in which you are working.
2. Follow WebLogic documentation instructions for deploying a new application.

Optional Recommended Arguments

There are some optional, recommended arguments that can help tune performance. To apply them, append the following arguments to the ones already existing in Managed Server or Node Manager:

```
-server -XX:+HeapDumpOnOutOfMemoryError -XX:+UseConcMarkSweepGC -
Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -
verbose:gc
```

Most of these advanced Java Virtual Machine arguments relate to garbage collection in memory. For more details on performance tuning your Java environment and garbage collection, see your Sun documentation.

1.4.7.5 Configuring Web Servers

The WebLogic Server software suite includes a web server, suitable for processing HTTP requests for TeamConnect pages. However, if you would like your users to be able to access TeamConnect through your company's official URL (for example, http://www.yourcompany.com/TeamConnect/), you need to set up a plug-in for your company's web server to proxy the appropriate HTTP requests to WebLogic.

Also, if you have set up a cluster of managed servers, you must set up a proxy to route the client requests to the appropriate managed servers. If you do not set up a cluster proxy, you cannot take advantage of the benefits of clustering (load balancing, failover, and so on).

Oracle provides plug-ins for Microsoft IIS, as well as Apache and Sun Java System Web Server, which enable the web server to proxy HTTP requests to WebLogic. Many configuration options for these plug-ins can be found in the WebLogic Server *Administration Guide*.

1.4.8 Troubleshooting WebLogic Installations

- (Windows only) If you did not define your WebLogic servers as Windows services when you ran the Domain Configuration Wizard, you can still set them to start at boot time by using the **WL_HOME/user_projects/domains/Mitratech/installService.cmd** utility.
- If, when attempting to start the WebLogic server, you encounter a java.lang.OutOfMemory exception, the Java environment does not have enough memory for the WebLogic server.
- If you are having problems connecting to the TeamConnect database, check the following:
    - Your TeamConnect database must have been fully configured prior to TeamConnect deployment.
    - If you are using an Oracle database, ensure that your Oracle client is installed and properly configured to connect to your database.
    - For Solaris/Linux: ensure that **ORACLE_HOME/bin** is in your **LD_LIBRARY_PATH** environment variable. For more information, see the Oracle documentation.

- For Windows: ensure that **ORACLE_HOME/bin** is in your **PATH** environment variable. For more information, see the Oracle documentation.

1.5 Verifying Installation

If you are doing an upgrade installation, user accounts from the previous version are still available so you can skip this section.

To verify that TeamConnect has been properly installed, you must also perform the initial TeamConnect application configuration steps. To do so, you must do the following:

1. Login to the system as user **TeamConnectAdmin**.
2. If this is a new installation, change the default **TeamConnectAdmin** password to a secure password.
3. In **Admin Settings**, click the **About** link and upload the license file.
4. Create a group account with the full rights of system administrators.
5. Create a contact user account, typically for your TeamConnect system administrator, and make that user a member of the group account that you just created.
6. (optional) Configure the TeamConnect Login Screen (for SSL only)
7. Log in with that user.

   (optional) Verify that the online help can be displayed.

Because the TeamConnect installation files may have been customized before you received them, certain screens may differ slightly from those shown in the following steps. However, if they differ significantly, or if described functionality is missing, carefully ensure that you have followed all of the installation instructions.

1.5.1 Logging in for the First Time

To log in to TeamConnect for the first time

1. Ensure that the TeamConnect servers are properly configured and running. For more information, see the previous chapters, if necessary.
2. Open an internet browser window and input the following address:

   `http://Hostname:port/TeamConnect/login`

   Replace `Hostname:port` with the IP address and port of the machine on which TeamConnect is deployed.

   If you have deployed TeamConnect to one managed server, this is the IP address and port of the managed server. If you have deployed TeamConnect to a cluster, this is the IP address and port of the administrative server on which the cluster is defined. If you have set up a proxy through an alternate web server, this is the IP address of the web server.

   The TeamConnect login screen should appear.

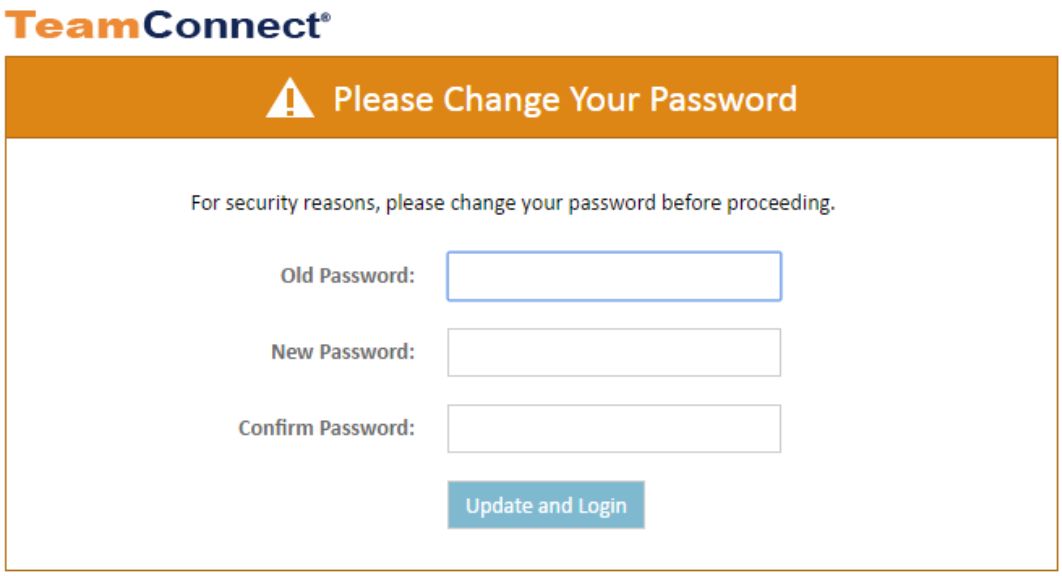
3. Log in as the following:

   User Name: `TeamConnectAdmin`

   Password: `PASSWORD` (case sensitive)

   Input these values into the **Username** and **Password** fields, and click Login.

   You are requested to change your password.



Change Your Password Window

4. Enter the appropriate values in the password fields and click **Save and log in**.

   This logs you in as the **TeamConnectAdmin** user. This user has limited rights within TeamConnect, and should be used only for licensing and to create new users.

   Logging in to TeamConnect for the first time takes longer than usual. TeamConnect generates its JSP interface files on the server at the first login. Subsequent login, regardless of the client, do not take as long.

To setup users

1. When you install a new instance of TeamConnect, only the **TeamConnectAdmin** user should log in. The first task for this user is to upload a licensing file into TeamConnect so that ordinary users can also log in.
   - Obtain the licensing file from your Mitratech Support representative.
   - After you have received the licensing file, go to **Admin**, **Admin Settings**, and **About**. There you must click a button, **Upload License**, that prompts you for the location of your license file and uploads it. Then you may begin using TeamConnect with all users.

     It is good practice not to use **TeamConnectAdmin** to log in on a regular basis, so these next steps will set up other usernames with administrator rights, and you can log in with those usernames.

2. Create a **Group** with comprehensive rights.

   All rights in TeamConnect are based on user groups. Your system administrator cannot acquire comprehensive rights unless they belong to a group with those rights.

   Refer to Creating or Editing Groups to learn how to create a new group, and how to assign rights to that group. Assign all possible rights to the new group. Save the Group record.

3. Create a **User** to represent the system administrator.

   Refer to Creating or Editing Users to learn how to create a new user account, and how to assign that user to the group that you created in the previous step. Save the User record.

   You must associate the User record with a **Contact** record. Since TeamConnect has just been installed, there aren't any Contact records in existence yet, and you will create a new Contact (with the system administrator's name) as part of the process of creating the user account.

4. Test your work by logging out of TeamConnect (using the Log Out link in the upper right of the page), then logging in with the username and password of the User account that you just created. This user should have access to all parts of TeamConnect, including the **Admin** tab and the **Setup** link, which are both visible on the user's home page.

1.5.2 Logging Options

Several options are available for logging in TeamConnect to alert you to potential problems and program errors.

- TeamConnect debug logging

  Set the location of the log files using the TeamConnect web application descriptor `app.logFolder`.

  In TeamConnect, you can control the level of detail in the logs using the settings on the **System Logging** screen in **Admin Settings**.

- General server logging

  Set the appropriate logging options on all of your application servers. For details, see the respective vendor documentation.

Ensure that all of these logging options are turned off when you have finished configuring the final TeamConnect implementation interface.

1.5.3 Troubleshooting

Issue: After deploying the new WAR file for online help, I do not see the help screens.

Resolution: If your application server or web server does not automatically clear cached JSP or JAR files on restart, you may need to delete those files manually and restart your application server.

1.6 Business Objects Single Sign-on

This chapter provides the general steps that you must take in order to successfully deploy BOSS on Apache Tomcat, WebLogic, or WebSphere.

Note: *BOSS requires separate purchase. Please contact your account manager for details.*

Disclaimer

These instructions are not entirely specific to Mitratech products and are provided for convenience only. Mitratech has validated these instructions, but does not claim responsibility for the accuracy or completeness of information about products from other vendors.

For more specific information on system requirements and instructions for deploying applications, please see the appropriate vendor documentation.

1.6.1 Deploying BOSS XI and Configuring URLs

This section describes the requirements for deploying **BOSSXI.war** for Business Objects Single Sign-on (BOSS), and how to verify that it is deployed successfully.

It also describes how to construct URLs to access the main Business Objects login page and to Business Objects documents (reports). These conventions can be used when defining URLs, for example, in a TeamConnect portal pane on a home page.

For more information about supported platforms for deploying a custom Business Objects application such as BOSS, see the Business Objects documentation.

For more information about how to deploy an application on your application server, see the vendor's documentation.

Support for Business Objects XI

Business Objects XI is supported for single sign-on (BOSS) with TeamConnect 5.0. BOSS has been certified using WebSphere, Apache Tomcat and WebLogic to deploy **BOSSXI.war** as an application.

Since BOSS is delivered as a J2EE application, you must deploy it as such on an application server. BOSS cannot be deployed on IIS.

Requirements for Deploying BOSSXI.war

- **BOSSXI.war** for Business Objects XI does not have to be deployed on a Business Objects node. You must simply deploy it on your application server.
- Both the Business Objects server and the application server must be on the same domain (for example, `.mitratech.com`).
- If BOSS and Business Objects are deployed on the same machine, then it is not required to have a domain name in the Cookie Domain field of Business Objects WebIntelligence Settings in TeamConnect's system settings screen. Also, URLs to BOSS do not need to include a fully qualified domain name. You can simply specify the machine name in the URL.

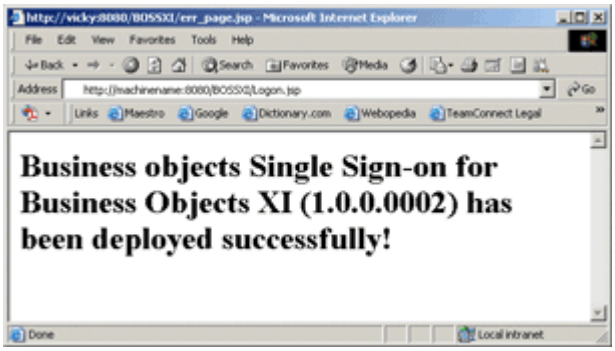
- From the **global.properties** file for Business Objects, you must set the trusted.auth.user.retrieval variable to WEB_SESSION.
- Once you have verified that BOSS has been deployed successfully, you must configure the Business Objects WebIntelligence Settings in TeamConnect's System Settings on the Admin menu. For details, see BusinessObjects WebIntelligence Single Sign-on, provided on your 5.0 installation media.

1.6.1.1 Verifying Successful BOSS Deployment

To verify whether **BOSSXI.war** is deployed successfully, open a browser window and type in the following URL. Replace the properties in brackets with the application's property values:

```
http://<appserver.host>:<appserver.port>/<jsp.app.name>/Logon.jsp
```

If you see the page below, then **BOSSXI.war** has deployed successfully.



BOSSXI.war Successful Deployment Page

1.6.1.2 Constructing URLs to Business Objects Pages

When providing URLs to end users so that they can access Business Objects documents through TeamConnect (for example, in a portal pane on their home page), you must construct the URLs as described in "OpenDocument Parameters" in the BusinessObjects Enterprise XI Release 2 .NET SDK or Java SDK. This information can be accessed at one of the following locations:

- (.NET SDK)

  ```
  http://devlibrary.businessobjects.com/BusinessObjectsXIR2/en/en/BOE_SDK/
  boesdk_dotNet_doc/doc/boesdk_net_doc/html/Report_Linking13.html
  ```

- (Java SDK)

  ```
  http://devlibrary.businessobjects.com/BusinessObjectsXIR2/en/en/BOE_SDK/
  boesdk_java_dg_doc/doc/boesdk_java_dg/Report_Linking13.html
  ```

The following examples can help you construct the proper URLs. Note that if BOSS is not deployed on a Business Objects node, then the <hostname> must include the fully qualified domain name of the server.

Caution: *Do not use a token in your URL. This will invalidate the link.*

For details about adding Business Objects URLs to TeamConnect portal panes, see Adding WebIntelligence URLs.

URLs When Business Objects Deployed on IIS

The following are examples of valid URL construction when the Business Objects server is deployed on IIS.

- URL to Business Objects server main login page:

```
http://<hostname:port>/businessobjects/Enterprise115/InfoView/main.aspx
```

- Example URL to Business Objects document:

```
http://<hostname:port>/businessobjects/enterprise115/InfoView/scripts/
opendocument.aspx?sType=wid&iDocId=2179
```

URLs When Business Objects Deployed as J2EE Application

The following are examples of valid URL construction when the Business Objects server is deployed on your application server.

- URL to Business Objects server main login page:

```
http://<hostname:port>/businessobjects/enterprise115/desktoplaunch/InfoView/
main/main.jsp
```

- Example URL to a Business Objects document:

```
http://<hostname:port>/businessobjects/enterprise115/desktoplaunch/opendoc/
openDocument.jsp?
sDocName=T21CUN03_Case_Aging_1st_party&sType=rpt&sKind=FullClient
```

1.7 TeamConnect Installer Form

This form should be printed and completed before starting a new TeamConnect installation or reconfiguring an existing TeamConnect application. Your entries on this form will help you answer prompts when the installer runs.

Some of the application parameters described below will already have suggested values. Those values will appear when you run the installer. You can edit them and replace them with your chosen values.

If you are reconfiguring an existing TeamConnect application, that application's values will appear, not the default suggested values.

Most of the parameters shown below have corresponding entries in the teamconnect.properties configuration file which is detailed in teamconnect.properties parameters for all application servers. However, there is not always a perfect correspondence between the installer form's values and the configuration file's values.

TeamConnect installation data

Application parameter	Description (Suggested value)	Value
Where Would You Like to Install	TeamConnect installation media will be copied to the default path or the path that you specify.	

Configuration Folder	(Not used in this version)	
Log Folder	(Optional) You can edit the default location of TeamConnect log files on your application server. For example, `./TeamConnect/logs`	
Designer Help URL	The location of the Designer help file. Use values like: **http://*hostName*/TeamConnectHelp/Designer/tc_designer_help.html** (where *hostName* should be replaced by a server name or IP address and might include port number)	
User Help URL	The location of the User and Admin combined help file. Use values like: **http://*hostname*/TeamConnectHelp/user/tc_user_help.html** (where *hostName* should be replaced by a server name or IP address and might include port number)	
jMonitor Config File	Specifies the location of the file that contains configuration information for the Performance Monitor (see Performance Monitor).	If you do not specify a location, the file will reside in the same folder as the Java Virtual Machine that is being used by TeamConnect.
Sequence pre-allocation size	Set this value to 500 always. Do not modify this value.	
Minimum number of connections	(If using a connection pool for database connection) The minimum number of connections the connection pool is initialized with. For example, 10	
Maximum number of connections	(If using a connection pool for database connection) The maximum number of connections the connection pool can grow to. For example, 10	

Minimum number of read connections	(If using a connection pool for database connection) The minimum number of connections the READ connection pool is initialized with. For example, 10	
Maximum number of read connections	(If using a connection pool for database connection) The maximum number of connections the READ connection pool can grow to. For example, 10	
Database Type	Select **Oracle** or **SQL Server**.	
Do you want to use data source	Select **Yes** to use data source for database connection	
Data Source Name	(If you are using data source) Type the data source name. For example, TeamConnect	
These parameters are only required if you are not using a data source.		
Database Name	(SQL Server only) The name of the database.	
The hostname and port number of your database, separated by colons. ***Note:*** *Only required if not using a data source.*	For example, **jdbc:sqlserver://** ***serverName*** **:*port*;databaseName=*dbName*** where *serverName*, *port*, and *dbName* should be changed appropriately.	
Username	The database user name.	
Password	The encrypted password string. This password needs to be encrypted using the encryption tool.	
Enable Cache Synchronization	Select **Yes** if you will Enable Cache Synchronization; run TeamConnect in a clustered server environment; or **No** if you will not run TeamConnect in a clustered server environment	

RMI Registry Port Number	(No value required unless you will enable Cache Synchronization; running TeamConnect in a clustered server environment) The port number used to create and launch TeamConnect's own internal RMI registry. If you deploy more than one instance of TeamConnect, each instance should use a different value for this parameter.	
ClusterName	(No value required unless you will enable Cache Synchronization; running TeamConnect in a clustered server environment) Type a unique name for your cluster of servers, to distinguish it from other clusters on the network.	
Oracle Database URL	(If you use an Oracle database) Type the database URL to connect to, to create a database schema and import TeamConnect components. The format should be: **jdbc:oracle:thin:@*serverName*:*port*:*dbName*** Where *serverName*, *port*, and *dbName* should be changed appropriately.	
SQL Server Database URL	(If you Microsoft SQL Server) Type the database URL to connect to, to import TeamConnect components. The format should be: **jdbc:sqlserver://*serverName*:*port*;databaseName=*dbName*** Where *serverName*, *port*, and *dbName* should be changed appropriately.	
Cache synchronization, multi-cast group address	If you use multi-cast to keep cache synchronized among a group of servers, you must specify the group address. If you provide no value, the default value of 226.10.12.64 is used.	
Cache synchronization, multi-cast port	If you use multi-cast to keep cache synchronized among a group of servers, you	

	must specify the port number used for multi-cast. If you provide no value, the default value of 3121 is used.	
The following parameters are for the Data Warehouse database, not the TeamConnect production database		
Database Type	Select **Oracle** or **SQL Server**.	
Do you want to use data source	Select **Yes** to use data source for database connection	
Data Source Name	(If you are using data source) Type the data source name. For example, DW	
DataWarehouse Name	(SQL Server only) The name of the database.	
The hostname and port number of your datawarehouse, separated by colons. ***Note:*** *Only required if not using a data source.*	For example, **jdbc:sqlserver://** ***serverName*** **:*port*;databaseName=*dbName*** where *serverName*, *port*, and *dbName* should be changed appropriately.	
Username	The database user name if not using data source.	
Password	The encrypted password string. This password needs to be encrypted using the encryption tool.	
Interval (seconds) at which WebLogic checks to see if JSP files have changed and need recompiling	(WebLogic only) Sets the interval, in seconds, at which WebLogic checks to see if JSP files have changed and need recompiling. If set to -1, page checking and recompiling is disabled (recommended for production mode). If set to 0, pages are checked on every request.	
Name of the session cookie	(WebLogic only) The name of the session cookie (for example, TCSESSIONID).	

IMAP Server Port	IMAP server port number. Required only if you choose to enable the IMAP server feature.	

1.8 Web Application Parameters

This appendix describes the web application parameters whose values you may need to modify before deploying TeamConnect 5.0.

This appendix contains the following sections:

- weblogic.xml Parameters
- teamconnect.properties parameters for all application servers
- Cache Synchronization Parameters
- Encryption Tool

1.8.1 weblogic.xml Parameters

If you are using WebLogic as your application server, there are certain values in the **weblogic.xml** file, located in the **/WEB-INF** directory of the TeamConnect **.war** file. These parameters apply only to WebLogic. You should not modify these values.

Parameters in weblogic.xml

Parameter	Description	Suggested value
pageCheckSeconds	Sets the interval, in seconds, at which WebLogic checks to see if JSP files have changed and need recompiling. If set to -1, page checking and recompiling is disabled (recommended for production mode). If set to 0, pages are checked on every request.	-1 (in production environment)

CookieName	Sets the name of the session cookie. If the session cookie has a unique name, this reduces some runtime behavior issues such as unexpected session time-outs. One way to establish a unique name is to derive the name of the cookie from the context path of the application: <ContextPath>SessionID. Note that this parameter is contained inside different elements than the other **weblogic.xml** parameters. Example:
	`<session-descriptor>`
	`<cookie-name>TCSESSIONID</cookie-name>`
	`</session-descriptor>`
		TCSESSIONID

container-descriptor	Sets the policy for checking for expired static resources. Values of -1 suppress checking, thus improving performance. Example:
	`<container-descriptor>`
	`<servlet-reload-check-secs>-1</servlet-reload-check-secs>`
	`<resource-reload-check-secs>-1</resource-reload-check-secs>`
	`<prefer-web-inf-classes>false</prefer-web-inf-classes>`
	`</container-descriptor>`
		-1

1.8.2 teamconnect.properties parameters for all application servers

The following table lists the configuration parameters for TeamConnect whose values you may need to modify when preparing TeamConnect for deployment. They are defined in the **teamconnect.properties** file.

Parameters in teamconnect.properties

Parameter	Description	Suggested values
Application parameters:		
app.runStartUp	Indicates whether TeamConnect is set to execute start up code. Enter YES to run start up code defined in TeamConnect's Documents area in the **/Root/System/StartUp** folder. Enter `NO` to disable the start up code. The default value is `YES`.	`YES` or `NO`
app.runStartUp.failOnError	Indicates whether TeamConnect will continue to execute start up code if it encounters an error or exception. Enter `YES` to stop the execution of the rest of the class files on error or exception. Enter `NO` to continue onto the next class file. The default value is `NO`.	`YES` or `NO`
app.logFolder	The directory where TeamConnect stores log files. Logging can be configured through **System Logging** on the **Admin** tab, **Logging** sub-tab, in the TeamConnect user interface.	`./TeamConnect/logs`

	***Tip:** If you are running multiple instances of TeamConnect on a single application server (for example, in a development environment), set different log folders for each of them so that you can more easily distinguish between the logs. Default value is ./logs.*	(use backslashes for Windows)
app.dontUseSecurityViews	This parameter influences how object security is checked when a search is executed. A value of `NO` means that object security is checked at the database level, using database views introduced in version 3.3. This approach reduces network traffic between application server and database server. A value of `YES` means that security-related database views are ignored and security checking takes place in the application's memory space. This is the better choice if your database server is not powerful, and if you have resources available on your application server. If you are encountering performance issues when using the security views, change this property to `YES`.	`NO` (default)
app.preLoadStores	This parameter influences how static resources are fetched into application memory. A value of `YES` causes all resources to be fetched at startup. This slows the startup process, but improves response time for all subsequent processes during the application session. A value of `NO` causes resources to be loaded only when required for the current process, which speeds startup but causes all other processes to run more slowly.	`YES` (default)
app.expiresHeader.timeInMinutes	Determines how long, in minutes, static resources may persist in cache before they are explicitly refreshed.	`360` (default)
db.product	The database being used to store TeamConnect data. Enter the appropriate software name.	Oracle or SQLServer
url.adminHelp	Specifies the TeamConnect online Designer help location.	See under **Description**

	http://<*hostname*>/TeamConnectHelp/DesignerHelp/tc_designer_help.html (where <hostname> can be web application server name or IP address and might include port number).	
url.userHelp	Specifies the TeamConnect online user help location. **http://<*hostname*>/TeamConnectHelp/user/tc_user_help.html** (where <hostname> can be web application server name or IP address and might include port number).	See under **Description**

javax.xml.soap.MessageFactory	Specifies the SOAP messaging implementation that will be used. This is required for using Web Services on WebLogic and WebSphere servers, and overrides the default SOAP implementation that those servers use. This parameter is ignored for Tomcat servers. The parameter has the following default value:
	
	`com.sun.xml.internal.messaging.saaj.soap.ver1_1.SOAPMessageFactory1_1Impl`
	
	If you have a version of Weblogic earlier than 10.3.6, use the following argument:
	
	`com.sun.xml.messaging.saaj.soap.ver1_1.SOAPMessageFactory1_1Impl`
		See under **Description**

jmonitor.configuration.file	Specifies the location of the file that contains configuration information for the Performance Monitor (see Performance Monitor). If you do not specify a location, the file will reside in the same folder as the Java Virtual Machine that is being used by TeamConnect. In environments with multiple TeamConnect installations, you should override the default with a location that is specific to the current TeamConnect installation.	See under **Description**
EclipseLink parameters:		
eclipselink.sequencePreallocationSize	The EclipseLink primary key sequence pre-allocation size. This value should not be changed.	500

eclipselink.minConnections	The minimum number of connections the EclipseLink connection pool is initialized with.	30
eclipselink.maxConnections	The maximum number of connections the EclipseLink connection pool can grow to.	30
eclipselink.minReadConnections	The minimum number of connections the EclipseLink READ connection pool is initialized with.	30
eclipselink.maxReadConnections	The maximum number of connections the EclipseLink READ connection pool can grow to.	30
eclipselink.bindParameters (Optional)	Whether to use bind variables in database queries. Affects performance. Default behavior is as shown in Suggested Values.	`YES` (for Oracle) or `NO` (for SQL Server)
Parameters for directdatabase connection (Only for Tomcat deployments):		
db.username	The username that TeamConnect should use to access the database.	
db.password	The password that TeamConnect should use to access the database. ***Important:*** *This password is encrypted automatically when the Installer runs. Future changes to the password must be manually encrypted using the encryption tool provided by Mitratech. For more details, see Encryption Tool.*	
db.databaseName	(SQL Server only) The name of the database.	
db.server	For SQL Server—Specify only the hostname and port number of your database, separated by colons. For Oracle Thin connection—Specify the hostname, port number, and the SID of your database, separated by colons.	
Data source parameters:		

app.useDataSource	Enter YES to use a data source (defined in the application server) as the database connection mechanism. If set to YES, all EclipseLink connection pool and WebLogic connection pool parameters are ignored. Enter NO to use a EclipseLink connection pool or a WebLogic connection pool. By default, the value is YES.	YES or NO
appserver.datasourceName	The name of the data source to use, if app.useDataSource is set to YES. If you do not provide a data source name, the name defaults to TeamConnect.	
Data Warehouse parameters:		
dwh.db.product	The database being used to store Data Warehouse data. Enter the appropriate software name.	Oracle or SQLServer
dwh.useDataSource	Enter YES to use a data source (defined in the application server) as the database connection mechanism. If set to YES, all EclipseLink connection pool and WebLogic connection pool parameters are ignored. Enter NO to use a EclipseLink connection pool or a WebLogic connection pool. By default, the value is NO.	YES or NO
dwh.datasourceName	The name of the data source to use, if dwh.useDataSource is set to YES. If you do not provide a data source name, the name defaults to TeamConnect.	
dwh.db.username	The username that TeamConnect should use to access the Data Warehouse database.	
dwh.db.password	The password that TeamConnect should use to access the Data Warehouse database. ***Important:*** *This password is encrypted automatically when the Installer runs. Future changes to the password must be manually encrypted using the*	

	encryption tool provided by Mitratech. For more details, see Encryption Tool.	
dwh.db.databaseName	(SQL Server only) The name of the database.	
dwh.db.server	The location of the database, in format **hostname:port:SID**.	
Global search parameters:		

globalSearchMenu.doNotShow	This property can be used to suppress the display of certain options in Global Search.
	If you do not want the "All" option to appear in the dropdown list of Global Search, enter the following property and value:
	`globalSearchMenu.doNotShow=All`
	If you want neither the "All" option nor the "All Projects" option to appear in the dropdown list of Global Search, enter the following property and value:
	`globalSearchMenu.doNotShow=AllAndAllProjects`	

Security enhancement parameters:		
app.disableSpringSecurityLogging	Indicates whether the spring security logger assumes the TeamConnect logging levels. By default, this parameter is not in the .properties file and is automatically set to false, which means that the spring security logger does not use the logging levels. If the logger levels are set to debug and you want the spring security logger to include debug messages, add this parameter to the file and set it to true. Note: If this parameter is set to true, the spring logger may log sensitive information, such a CSRF token and session IDs. Indicates whether the spring security logger assumes the TeamConnect logging levels. By default, this parameter is not in the .properties	

	file and is automatically set to false, which means that the spring security logger does not use the logging levels. If the logger levels are set to debug and you want the spring security logger to include debug messages, add this parameter to the file and set it to true. ***Note:*** *If this parameter is set to true, the spring logger may log sensitive information, such a CSRF token and session IDs.*	
app.setupAccessPort	Allows you to specify a valid port number that gives users access to the TeamConnect setup. Even if the user has the correct TeamConnect security rights and permissions, TeamConnect blocks users without this port from accessing the setup. From your web server, you can also specify the same port to block external traffic.	
encryption.iterations	Specifies the number of iterations that TeamConnect runs the password encryption algorithm. The default is 10000.	

1.8.3 Cache Synchronization Parameters

If you request cache synchronization during installation, the following parameters to **teamconnect.properties** are initialized. They can also be manually edited later.

teamconnect.properties Parameters for Cache Synchronization

Parameter	Description	Suggested or example value
sync.enabled	A YES or NO value that determines whether TeamConnect attempts to do cache synchronization.	`YES`
sync.channel	A unique name for your cluster so that it won't try to synchronize caches with any other clusters in your network.	`SpendManagement`
sync.rmi.port	The port number used to create and launch TeamConnect's own internal RMI registry. If you deploy more than one instance of TeamConnect, each instance should use a	`2258`

	different value for this parameter. Default is 2258.	
sync.multicastGroupAddress	(Optional) To keep cache synchronized among a group of servers, you must specify the group address. If you provide no value, the default value of 226.10.12.64 is used.	`226.10.12.64`
sync.multicastPort	(Optional) To keep cache synchronized among a group of servers, you must specify the port number used for multi-cast. If you provide no value, the default value of 3121 is used.	`3121`
sync.packetTimeToLive	(Optional) Multi-cast packets should expire after a certain number of routing hops, to avoid network congestion. Specify the number of hops. If you provide no value, the default of 16 is used.	`16`

1.8.4 Encryption Tool

When editing any passwords in the TeamConnect **teamconnect.properties** file--ONLY in this file, which is the web application descriptor, not elsewhere--you must encrypt the password using the TeamConnect Encryption Tool provided in your TeamConnect installation media.

Important: *Make sure to modify the following instructions according to your operating system.*

To encrypt a password

1. In a command-line window, navigate to the directory **/utilities/encrypt** on the TeamConnect installation media.
2. Make sure your Java bin path is set properly. For more information on setting this path, see your Java documentation.
3. Run the following command:

   ```
   java -jar encrypt.jar yourpassword
   ```

   Replace your password in the example above with the password you would like to encrypt.

   This command outputs an encrypted version of your password. The encrypted password is a string of capital letters. You may copy this password and paste it where necessary.

1.9 Performance Monitor

This appendix describes the Performance Monitor in TeamConnect and explains how to configure and view it.

This appendix contains the following sections:

1.9.1 Overview

The TeamConnect Performance Monitor silently captures data in the background on all TeamConnect requests. Any time a request exceeds a configurable threshold, the monitor log captures the data for that one request so that the information can be analyzed later by IT staff and/or Mitratech Support. Such analysis, and subsequent design changes, can improve application performance.

In contrast to general debug loggers in TeamConnect:

- Performance Monitor logs data for long-running requests only, so it doesn't create a strain on the system or on the person analyzing these logs.
- All data that is logged by Performance Monitor for a given request is guaranteed to be contiguous in the log file and does not overlap with any data from any other requests, which is another common difficulty when analyzing data from general debug loggers.

1.9.1.1 Resource requirements

The Performance Monitor is designed to be always on in production environments so that you can capture data related to performance issues whenever they happen. It has been load-tested and shown to incur less than 2% overhead on even highly loaded systems (using an 1800-user load test that drives an 8-core application server to 80% CPU utilization rates).

1.9.1.2 What information is captured?

For every long-running request, the Performance Monitor captures and logs HTTP request data, (limited) HTTP session data, JDBC call data and periodically gathered stack traces. The HTTP request and (limited) HTTP session data can be used to identify the user, the record, the search, the tool, etc. The JDBC call data, including timings, full SQL text and bind parameter values, can help isolate a variety of issues, including poorly performing queries, too many queries being executed, or too much data being returned. The periodically gathered stack traces can help identify performance issues in custom rules, custom Java blocks, or even core TeamConnect code.

You can configure a threshold, in milliseconds, for "stuck" requests. As soon as a request exceeds this stuck threshold, the monitor immediately logs all of the data it has gathered for this request so far to make sure this information is not lost (since normally long requests are only logged upon completion). Also, if any email addresses have been configured, the monitor will email an alert with the same data that it logged for the stuck request. If the request later completes, a second email is sent that the request has completed (become "unstuck").

The TeamConnect Performance Monitor doesn't capture any information related to hardware utilization. Hardware utilization data can be very valuable and can be captured by a variety of tools. However, often spikes in hardware utilization (e.g. CPU and memory) are caused by long-running requests and not vice versa. Of course, once CPU or memory reaches its limits, this can cause all requests to run slow, but often the key is to identify the initial long-running request(s) which triggered the CPU or memory spike in the first place.

1.9.2 Performance Monitor Configuration

The Performance Monitor is divided into a server agent, which gathers information about specific requests and determines when it should be logged, and a logger, which collects the information from the agent and structures it so that the log is easily readable.

The properties of these two components are configurable through the Performance Monitor user interface. Access this UI through your browser, using the same URL that you use to log in to your TeamConnect instance, but substituting **jmonitor** for **login**. For example:

```
http://myserver:port/instancename/jmonitor
```

The resulting page displays several buttons.

1. Click **Change Password** to set a new password.
2. Click **Update Configuration** to view and change the various configurable properties, which are described in the table below.

Configurable properties for Performance Monitor

Property	Definition
Enabled	This checkbox determines whether the server agent should collect performance data at all.
Threshold (milliseconds)	All requests exceeding this threshold are logged (along with all associated data) once they complete. Default=30000.
StuckThreshold (milliseconds)	As soon as a requests exceeds this threshold, it is logged immediately (along with all associated data gathered to that point). Also, if email is configured (see settings below), a "stuck" alert is emailed to the configured address(es) containing the same data that was written to the log file. If the request later completes, an additional email is sent that the request has completed (become "unstuck"). Default=600000.
Stack Trace Initial Delay(milliseconds)	Once a request exceeds this initial delay, stack traces for the thread processing the given request are gathered periodically (see stackTracePeriodMillis below). Default=10000.
Stack Trace Period (milliseconds)	Once a request exceeds an initial delay (see Stack Trace Initial Delay above), this is the interval at which successive stack traces for the thread processing the given request are gathered. Default=1000.

Maximum Trace Events per Request/Operation	If the number of stack traces for one request reaches this maximum number, no additional trace events are gathered Default=1000.
Log Archive Filename Pattern	This naming pattern controls how frequently the log file is rolled over. It can also control file compression if you add a standard compressed-file extension such as **.gz** or **.zip** to the end of the pattern. The pattern can contain a path specification, which is useful in separating the archive log files from the currently active log file (see next property.) Default="jmonitor%d{yyyy-MM-dd}.log, which specifies daily rollover and no compression.
Log Active Filename	This naming pattern can contain a path specification, which is useful in separating the archive log files from the currently active log file there is no default value. If you do not supply a value, the "Log Archive Filename Pattern" value will be used.
Log Max History	If you enter an integer here, then the number of archived log files will not exceed that integer. The oldest archives will be deleted as new ones are created. Default=0 (no limit on number of archived logs).
Email Host	SMTP host to send email through.
Email Smtp Port	The port used to communicate with the SMPTP host. Default=25.
Email Username	Username if SMTP host requires authentication.
Email Password	Password, if SMTP host requires authentication.
Email SLL	Set to this box to checked if the SMTP host requires an SSL connection.
Email From Address	Email address to use as the sender.
Email To Addresses	Comma-separated list of email addresses to send alerts.
Maximum Trace Elements emailed per Operation	Controls the size of the email messages by limiting how much trace event information is contained in an email.

1.9.2.1 Recovering a Lost Password

The value of the administrator password, used for logging in to Performance Monitor, can also be changed by editing the properties file and updating the password property with the new password in clear text, e.g., `adminPassword=mynewpassword`. Updating the properties file requires a restart, after which the monitor will encrypt the password and update the jmonitor.properties file with the encrypted value. Using this approach requires that you have access and authority to the jmonitor.properties file.

1.9.2.2 Important Information About Garbage Collection

Garbage collection is a common cause of performance issues which cannot be detected from the TeamConnect Performance Monitor logs. Garbage collection logging should be on in production environments and these logs should be analyzed in conjunction with the TeamConnect Performance Monitor logs. Mitratech typically recommends these GC logging settings for the Sun JVM:

```
-XX:+PrintGCTimeStamps -XX:+PrintGCDetails -XX:-TraceClassUnloading
```

Optionally include `-Xloggc:<file>` if the application server is not capturing stdout to a log file. The only downside to this option is that it overwrites <file> on every restart, which can erase valuable information.

Use this GC logging setting for the IBM JVM:

```
-verbose:gc
```

This setting is the same as enabling the **Verbose garbage collection** checkbox in WebSphere administration console.

1.9.2.3 Clustering

Support for clustering requires configuring Performance Monitor separately on each node. The monitor web interface is node-specific, so it will work best if you have direct access to address the monitor web interface on each node individually, as opposed to access through a front end load balancer.

1.9.3 Verifying Performance Monitor

The most straightforward way to verify that Performance Monitor is working is to cause some performance exceptions and observe that they are being logged correctly. Here is an example of how to do that.

- Start the Performance Monitor UI as described in Performance Monitor Configuration.
- Change property **Threshold** to an artificially low value such as 100.
- Log into TeamConnect to generate requests which exceed the temporary low threshold.
- Change **Threshold** back to the previous value (default is 30000) before the monitor log file starts to accumulate a large number of entries.
- View the monitor log file.
- Validate that some of the requests generated above were captured and logged since they should have exceeded the 100 millisecond threshold.

1.10 Upgrading from Previous Versions

This appendix describes issues that may arise when you upgrade from previous versions to TeamConnect 5.0.

This appendix contains the following sections:

Important: *All customers upgrading from a version prior to 3.3 SP1 will need to install a new license file in the first session with the new version. Obtain a new license file by contacting Mitratech Support. See Licensing for more information.*

1.10.1 Database Design

If your existing design contains custom fields of type Involved, and the names of those fields are longer than 26 characters, problems may arise when those fields are exported to the Data Warehouse. If you intend to use Data Warehouse, delay upgrading to the new version until you have altered those field names to 26 characters or less, and you have altered any views, rules, or other logic that refers to those fields.

1.10.2 Automatic Conversion Processes

When the installer detects an existing version 2.5 SP4 Update 1 installation, it runs multiple tools to transform expressions, tags, and other elements from the version 2.x style to the version 4.x style. This section focuses on conversion of custom screen tags.

1.10.2.1 Manual conversion

It is possible to run a conversion tool manually against a single XML file to get a report of possible errors in the file. **upgrade.bat**, found in the exploded contents of setup.exe, should be run from the command line, taking the XML path and file name as a command-line argument. Example:

```
upgrade.bat \work\MyCustomJavaBlock.xml
```

The file specified in the argument is converted and then overwritten with the conversion results.

The output from upgrade.bat can be found in files **ConversionLog.txt** and **ManualConversionLog.txt**, in the same folder.

Note: ManualConversionLog.txt *is only created when the tool encounters situations that are known to require manual intervention. Such situations are also listed in the console window as the job executes.*

1.10.2.2 Logging

The results of the conversion tools' work is logged in two files:

- **ConversionLog.txt** contains progress messages for the automated tools.
- **ManualConversionLog.txt** identifies situations that can't be handled by the automated tool and will require manual editing by a developer familiar with HTML and Javascript and, in some cases, Java.

These files are found in a subfolder beneath the installation media folder:

```
upgrade/tools/external/CustomScreenConversionTool/logs
```

It is essential that you read the logs after the installer has run, watching for possible conversion issues.

If a conversion situation is detected then you will find, near the beginning of the log file, the phrase "Custom screen conversion is needed". Additional messages regarding conversion will be found throughout the log file.

In the **ConversionLog.txt** excerpt below, a summary of successful conversion actions is presented:

```
06/11/2009 13:08:09 (266ms) --- --------Custom screen conversion-Started--------
06/11/2009 13:08:09 (376ms) --- Root/System/Object Definitions/Account
06/11/2009 13:08:09 (579ms) --- Root/System/Object Definitions/Advice & Counsel
06/11/2009 13:08:09 (719ms) --- WzCjbAdviceCounselSearchResultsSYS.xml
06/11/2009 13:08:10 (251ms) ---         Updated 1 tc:component
componentType:CLString tag
06/11/2009 13:08:10 (251ms) ---         Updated 1 HTML Tag  : link tag
06/11/2009 13:08:10 (251ms) ---         Updated 11 tc:CLTextField tag
06/11/2009 13:08:10 (251ms) ---         Updated 1 tc:useBlockTemplate tag
06/11/2009 13:08:10 (251ms) ---         Updated 1 tc:component
componentType:WORepetition tag
06/11/2009 13:08:10 (251ms) ---         Updated 2 tc:CLRadioButton tag
06/11/2009 13:08:17 (298ms) --- WzCjbAdviceCounselUseExistingSYS.xml
06/11/2009 13:08:17 (313ms) --- Converted Cjb expression cjb.acWasRelated() for
class WzCjbAdviceCounselSearchResultsSYS
06/11/2009 13:08:17 (329ms) --- Converted Cjb expression cjb.acWasRelated() for
class WzCjbAdviceCounselSearchResultsSYS
06/11/2009 13:08:17 (329ms) ---         Updated 4 tc:if tag
06/11/2009 13:08:17 (329ms) ---         Updated 1 HTML Tag  : link tag
06/11/2009 13:08:17 (329ms) ---         Updated 1 tc:useBlockTemplate tag
06/11/2009 13:08:17 (923ms) --- CjbAdviceCounsel.xml
06/11/2009 13:08:18 (016ms) ---         Updated 1 tc:if tag
06/11/2009 13:08:18 (016ms) ---         Updated 8 tc:label tag
06/11/2009 13:08:18 (016ms) ---         Updated 2 tc:component
componentType:CLLabel tag
06/11/2009 13:08:18 (016ms) ---         Updated 11 tc:field tag
06/11/2009 13:08:18 (016ms) ---         Updated 1 HTML Tag  : link tag
06/11/2009 13:08:18 (016ms) ---         Updated 1 tc:useBlockTemplate tag
```

```
06/11/2009 13:08:18 (016ms) ---         Updated 29 tc:component componentType:
WOImage tag
06/11/2009 13:08:18 (016ms) ---         Updated 2 tc:CLLabel tag
06/11/2009 13:08:18 (016ms) ---         Updated 1 HTML Tag  : script tag
06/11/2009 13:08:18 (719ms) --- Root/System/Object Definitions/Allegation
06/11/2009 13:08:18 (719ms) --- Root/System/Object Definitions/Appointment
06/11/2009 13:08:18 (766ms) --- Root/System/Object Definitions/Award
06/11/2009 13:08:18 (766ms) --- Root/System/Object Definitions/BDC Setting
06/11/2009 13:08:18 (782ms) --- Root/System/Object Definitions/Contact
```

Any problems in conversion will be highlighted by the word **ERROR** in uppercase. Search the log files for this word and note the preceding conversion action. It will require manual intervention by a TeamConnect professional.

Here is an example of **ManualConversionLog.txt**, showing some situations that can't be handled by the automated tool. Suggestions for handling some of these situations can be found in Manual Conversion Suggestions.

```
This file contains XML screens that could not be automatically converted to the 3.x
format due to the presence of tags or JavaScript code that need to be rewritten
using 3.x API.

Please refer to the 3.x developer guide for the details.

================================================

Filename: Root/System/Object Definitions/Advice & Counsel/Screens/
CjbAdviceCounsel.xml

Tag: <script ...>

No. of occurrences: 5

Instructions: HTML element names have changed in 3.x. Please rewrite the JavaScript
using the 3.x UI element names.

================================================

Filename: Root/System/Object Definitions/Cost Center/Screens/
CjbCTransactionHistorySYS.xml

Tag: <tc:component componentType=CLBatchDisplay ...>

No. of occurrences: 1

Instructions: Please use the 3.x batch display API to rewrite this tag, including
the corresponding CJB, NewRow and RegularRow XMLs.
```

1.10.2.3 Automatic Custom Screen Tag Conversion

The table below describes the custom screen tags that are handled automatically during the conversion process. A subsequent table describes other tags that will require manual conversions.

Even if a tag is listed in the table below, automatic conversion might not handle every attribute that is associated with the tag. Read the remarks for details. When an attribute is described as being "ignored" in the remarks, that means that the attribute will be dropped from the code as the code is converted to version 3.x syntax.

Custom screen tags that are automatically converted

Tag syntax in 2.x	Tag syntax in 3.x	Remarks

`<tc:useBlockTemplate blockTitle="{cjb.blockTitle}">`
text
`</tc:useBlockTemplate>`
	tc:blockTemplate	Only the blockTitle attribute is supported. Other attributes are ignored.

`<tc:component componentType="CLDropDownList">`
or
`<tc:CLDropDownList>`
	tc:select
	The following attributes are renamed:
	isAlwaysEditable—forceEditable
	isNotEditable—forceNotEditable
	nullValueDisplayString—nullValueDisplayName
	displayStringPath—displayNamePath
	sortKey—sortPath
	isMultiple—multiple
		These attributes are ignored and not converted:
		• doubleClick
		• onKeyDown
		• item
		• maxLength
		• isSubmittedOnChange
		• applet
		• notUseApplet
		• notShowRootApplet
		• entityName
		• forceSize
		• notLinkToUser
		• useDidChange
		• emptyValueString
		• allowEmptyList
		• disableInactiveItem

`<tc:component componentType="CLDateField">`
or
`<tc:CLDateField>`
	tc:date or tc:dateTime or tc:time
	The following attributes are renamed:
	isAlwaysEditable—forceEditable
	isNotEditable—forceNotEditable
	isTimeZoneIndependent—timeZoneIndependent
		Converted to tc:date unless attribute showTime is true, in which case the tag is converted to tc:dateTime. Regardless of the conditions above, if attribute doNotShowDate is `true`, the tag is converted to tc:time
		These attributes are ignored and not converted:

		• emptyValueString • disableChangeCheck • submitValue • useDidChange • dateFormatInput • dateFormatOutput • timeFormatInput • timeFormatOutput • timeUnderDate

`<tc:component componentType="CLTimeField">`
or
`<tc:CLTimeField>`
	tc:time	These attributes are ignored and not converted:
		• emptyValueString

`<tc:component componentType="CLCheckBox">`
or
`<tc:CLCheckBox>`
	tc:checkBox	These attributes are ignored and not converted:
		• disableChangeCheck
		• submitValue
		• useDidChange

`<tc:component componentType="CLString">`
or
`<tc:CLString>`
	tc:string **(3.x)**
	tc: out **(4.x)**
	The following attributes are renamed:
	displayStringPath—value
		The following attribute must be handled in a special way:
		The attribute value of **eoObject** must be applied to the **value** attribute by taking the displayStringPath then appending to the eoObject path. Thus, value = eoObject + "." + displayStringPath.)

`<tc:component componentType="CLLabel">`
or
`<tc:CLLabel>`
	tc:label
	The following attributes are renamed:
	isFieldRequired—required
		These attributes are ignored and not converted:
		• blockLabel

<tc:component componentType="CLString"> or <tc:CLString>	tc:out The following attribute is converted: dsiplayPath	All other attributes are ignored and not converted.
<tc:component componentType="CLTextArea"> or <tc:CLTextArea>	tc:note The following attributes are renamed: isFieldRequired—required	These attributes are ignored and not converted: • blockLabel
<tc:component componentType="CLTextField"> or <tc:CLTextField>	tc:text The following attributes are renamed: isAlwaysEditable—forceEditable isNotEditable—forceNotEditable maxlength—maxLength	This conversion will occur only if none of the following attributes are present in the tag: • format • maximumFractionDigits • minimumFractionDigits • allowNegativeValues • currencySymbol • showCurrencySymbol • These attributes are ignored and not converted: • disableChangeCheck • searchKey • useDidChange • emptyValueString • escapeXMLTags
<tc:component componentType="CLTextField"> or <tc:CLTextField>	tc:number The FORMAT attribute is converted; any trailing digit becomes the value of new attribute fractionDigits. These attributes are also converted: isAlwaysEditable—forceEditable	This conversion will occur only if at least one of the following attributes is present in the tag: • format

	isNotEditable—forceNotEditable maxlength—maxLength	• maximumFractionDigits • minimumFractionDigits • allowNegativeValues • currencySymbol • showCurrencySymbol These attributes are ignored and not converted: • disableChangeCheck • searchKey • useDidChange • emptyValueString • escapeXMLTags

`<tc:component componentType="CLCategoryTable">`
or
`<tc:CLCategoryTable>` or `<tc:component componentType="CLCategoryList">`
or
`<tc:CLCategoryList>`
	tc:categorySelect
	The following attributes are renamed:
	isAlwaysEditable—forceEditable
	isNotEditable—forceNotEditable
	nullValueDisplayString—nullValueDisplayName
	isMultiple—multiple
		These attributes are ignored and not converted:
		• item
		• maxLength
		• isSubmitedOnChange
		• applet
		• notUseApplet
		• notShowRootApplet
		• entityName
		• forceSize
		• notLinkToUser
		• useDidChange
		• emptyValueString
		• allowEmptyList
		• action
		• prepopulatedList

		• realEntityCode • showParent • subCategoryTreePosition • usePrePopulatedList

`<tc:component componentType="CLImage">`
or
`<tc:CLImage>`
Example:
`<tc:CLImageHyperlink href="${cjb.hyperlink}" tabIndex="2" imageFileName="customimages/spacer.gif" imageAltText="spacer" />`
	img
	Example:
	`<a href="#" onClick="${cjb.hyperlink}" tabIndex="2" ><img src="./customimages/spacer.gif" alt="spacer" /></a>`
		Version 3.x uses conventional HTML syntax for this situation, not a custom tag.

`<tc:component componentType="CLHiddenField">`
or
`<tc:CLHiddenField>`
	input	Version 3.x uses conventional HTML syntax for this situation, not a custom tag. The **name** attribute is converted from an EO reference to an `applicationEntity` reference.

`<tc:component componentType="CLHyperlink">`
or
`<tc:CLHyperlink>`
	link	

`<tc:component componentType="CLDetailTable">`
or
`<tc:CLDetailTable>`
	tc:customLookup
	These attributes are renamed:
	isAlwaysEditable—forceEditable
	isNotEditable—forceNotEditable
	nullValueDisplayString—nullValueDisplayName
	isMultiple—multiple
		The following attributes are ignored and not automatically converted:
		• displayStringPath
		• sortKey
		• doubleClick
		• onKeyDown
		• item

	detailLookupTableCode—tableCode	• maxLength • isSubmitedOnChange • applet • notUseApplet • notShowRootApplet • entityName • forceSize • notLinkToUser • useDidChange • emptyValueString • allowEmptyList • disableInactiveItem • displayRootNode • notUseApplet • prepopulatedList • useApplet • usePrepopulatedList

`<tc:component componentType="CLSystemTable">`
or
`<tc:CLSystemTable>`
	tc:systemLookup
	These attributes are renamed:
	isAlwaysEditable—forceEditable
	isNotEditable—forceNotEditable
	nullValueDisplayString—nullValueDisplayName
	isMultiple—multiple
	tableName—tableName
		The following attributes are ignored and not automatically converted:
		• displayStringPath
		• sortKey
		• doubleClick
		• onKeyDown
		• item
		• maxLength
		• isSubmitedOnChange
		• applet
		• notUseApplet
		• notShowRootApplet

		• entityName • forceSize • notLinkToUser • useDidChange • emptyValueString • allowEmptyList • disableInactiveItem • showAll • showAlmostAll • pleaseDontUseTree • showRoot • tnApplication • prepopulatedList • usePrepopulatedList
`<tc:registerBlockAction>`	tc:useClass	If two useClass tags in the same CJB reference the same class name, that is acceptable; there is no conflict.

`<tc:component componentType="CLObjectField">`
or
`<tc:CLObjectField>`
	tc:account (or tc:contact, tc:project, etc.) **(3.x)**
	tc:contact, tc:project, tc:user **(4.x)**
	The actual tag depends upon the value of attritube EntityName. The 3.x tag name should be the value of that attribute, minus the leading "T" character~.
	The following attributes are renamed:
	fieldName—name
	isAlwaysEditable—forceEditable
	isNotEditable—forceNotEditable
	nullValueDisplayString—nullValueDisplayName
	displayStringPath—displayNamePath
		The following attributes are ignored and not automatically converted:
		• allUsers
		• alt
		• alwaysUsePrimaryKey
		• applicationID
		• containerID
		• disableChangeCheck
		• disabled
		• display
		• displayString

	sortKey—sortPath isMultiple—multiple object—value	• displayStringFieldName • emptyValueString • entityName • forceSize • height • isFieldRequired • isPrintable • omitOpenIcon • onBlur • onChange • onClick • onFocus • onKeyDown • onSubmit • prepopulatedList • primaryKey • primaryKeyFieldName • qualifier • reloadPage • searchViewName • secured • sortKey • submitCommandArgs • tabIndex • type • useDidChange • usePrepopulatedList • width

`<tc:if negate="true" conditional="{cjb.isEditable}">;`
text
`</tc:if>`
	`<tc:if negate="true" test="{cjb.isEditable}">`
	text
	`</tc:if>`
		The tag is the same but the attribute name changes. If the conversion tool encounters a "1" or "0" as the value for **negate**, it will replace those values with "true" and "false" during conversion.

`<tc:component componentType="CLInternalTable">`
or
`<tc:CLInternalTable>`
	tc:enumeration
	The following attributes are renamed:
	table—enumeration
	isAlwaysEditable—forceEditable
	isNotEditable—forceNotEditable
	isMultiple—multiple
		enumeration is both a tag name and an attribute name. For each IID class value of **table** attribute in 2.x, the corresponding 3.x Enum type is substituted as the value in **enumeration** attribute. For example, **ZNContTypeIID** becomes **contactType**.
		The following attributes are ignored:
		• identifier
		• filterIID
		• disableInactiveItem
		• tabIndex
		• item
		• emptyValueString
		• listAsString
		• keyPath
		• useDidChange
		• notLinkToUser
		• forceSize
		• entityName
		• disabled
		• maxLength
		• isSubmitedOnChange
		• applet

		• notUseApplet • notShowRootApplet • doubleClick

Custom screen tags that are not automatically converted

Tag name	Remarks

`<tc:component componentType="CLBatchDisplay">`
or
`tc:CLBatchDisplay`
	Each instance of batch display must be manually converted by a TeamConnect professional.

1.10.3 Manual Conversion Suggestions

This section contains suggested approaches for manual editing of code that could not be handled by the automated conversion tool. Such code will be noted in the log file **ManualConversionLog.txt**, which is produced during the execution of the automated conversion tool.

1.10.3.1 JavaScript Conversions

DOM: Form name conversion

The document object model consisted of the following naming conventions:

2.x: `document.bodyform.EO__Name__STR` and `document.forms['bodyform'].EO__Name__STR`

3.x: `document.forms['mainForm'].EO__Name__STR`

which must be renamed to for 4.x

`document.forms['mainForm'].elements`

System fields name conversion

Binding to system fields and usage of system fields in Javascript will require edits. For example,

`EO__NumberString__STR`

must become

`enterpriseEntity.idNumber`

Also, you must check for changes in field names in the object model. For example the version 2.x field numberString became idNumber in later versions.

Custom fields name conversion

In 2.x custom field names were formatted as: `EO__Detail_{category}__{fieldType}_{fieldName}__{paramType}`

In 4.x these must be formatted as:

```
command_categories[category]_customFields[fieldName]_value
```

So in the example below, custom field name

```
EO__Detail_OPPS__DetailNumbValue_PlAcOCReserveChange__DBL
```

will become

```
command_categories[OPPS]_customFields[PlAcOCReserveChange]_value
```

Access to elements in a form, in Javascript, must be rewritten like this:

```
document.bodyform.EO__Name__STR.value
```

becomes

```
document.forms['mainForm'].elements['applicationEntity.name'].value
```

i.e. **document.bodyform** becomes **document.forms['mainForm']**

Another example:

```
document.forms['bodyform'].EO__Detail_LITP_OUTC__DetailTextValue_InvAccrual5__STR.v
alue
```

becomes

```
document.forms['mainForm'].elements['command_categories[LITP_OUTC]
_customFields[InvAccrual5]_value'].value
```

1.10.3.2 Common Javascript method conversions

Calls to Javascript methods in TeamConnect 2.x required referencing the "top" frame. However, in TeamConnect 3.x and 4.x, frames do not exist. Thus, all references to the "top" frame must be removed in calls on Javascript functions and variables.

Translating references to top.submitCommand()/invokePageDetailAction() in Tools

In 2.x and 3.x versions, you might see a reference to the `submitCommand(...)` and `invokePageDetailAction` functions, which allowed implementers the ability to call methods with or without parameters in page details, as shown in the following examples:

```
top.submitCommand('_self', 'anchor', 'PD', 'clickEntityCheck(*STR, *STR)',
'checkName', 'entityValue');
invokePageDetailAction(this, 'anchor', 'clickEntityCheck(*STR, *STR), 'checkName',
'entityValue');


top.submitCommand('_self', 'anchor', 'PD', 'generate()');
invokePageDetailAction(this, 'anchor', 'generate()');
```

While 3.x CJBs using the `invokePageDetailAction` continue to work, you can use the `invokeToolAction` function in any new 4.x CJBs.

For example, the following examples execute a 4.x tool action:

```
invokeToolAction('action')
invokeToolAction('actionWithArguments', arg1, arg2)
```

In addition to executing a 4.x tool action, you can also indicate the HTML element for the code to focus on once the page reloads, as shown in the following examples:

```
invokeToolActionAndAnchor('anchor', 'action')
invokeToolActionAndAnchor('anchor', 'actionWithArguments', arg1, arg2)
```

Translating references to top.submitCommand()/invokeBlockAction() in CJBs

In 2.x and 3.x versions, you might see a reference to the `invokeBlockAction` function, as shown in the following examples:

```
top.submitCommand('_self', 'anchor1', 'PD', 'invokeBlockAction(STR, STR)',
'invokeBlockActionCjb', 'argument');
invokeBlockAction(this, 'anchor1', 'invokeBlockActionCjb', 'argument');
```

While 3.x CJBs using the `invokeBlockAction` function continue to work, you can use the `invokeCustomAction` function in any new 4.x CJBs.

For example, the following examples execute a 4.x block action:

```
invokeCustomAction(this, 'cjb', 'action')
invokeCustomAction(this, 'cjb', 'actionWithArguments', arg1, arg2)
```

In addition to executing a 4.x block action, you can also indicate the HTML element for the code to focus on once the page reloads, as shown in the following examples:

```
invokeCustomActionAndAnchor(this, 'anchor', 'cjb', 'action')
invokeCustomActionAndAnchor(this, 'anchor', 'cjb', 'actionWithArguments', arg1,
arg2)
```

Remove occurrences of top.didChange and top.didChangeText()

All occurrences of top.didChange and top.didChangeText should be removed. In 2.x these are used to enable/disable buttons like **Save**, or **Save & Close**. In 3.x and 4.x these are no longer needed.

<tc:newRecordLink..> conversion review

The `tc:newRecordLink` tag is a 3.x and 4.x autoconversion of the 2.x Javascript method `openPageWithKeyAndArgs(...)`.

openPageWithKeyAndArgs in 2.x

For example:

```
<a onClick="top.openPageWithKeyAndArgs('NEW_PROJECT_WIZARD_OBJECT', '{ wizard =
ADVICE; application = ADCO; isFromMatter=1; embeddedObject=RADC; parentKey =
3}');">
    New Advice And Counsel
</a>
```

is converted by the automated conversion tool to `newRecordLink` tag:

```
<tc:newRecordLink entityCode="ADCO" entityToLink="${enterpriseEntity}"
pageArgs="{ isFromMatter=1; embeddedObject="RADC"; parentKey = 3}"
wizardUniqueKey="ADVICE">
    New Advice And Counsel
</tc:newRecordLink>
```

Here is the conversion of attributes from `openPageWithKeyAndArgs` to tc:`newRecordLink`:

```
{ wizard = ADVICE; application = ADCO; isFromMatter=1; embeddedObject=RADC;
parentKey = 3}
```

will become

```
pageArgs="{ isFromMatter=1; embeddedObject="RADC"; parentKey = 3}"
```

- `application` value is used as value for `entityCode` attribute.

  ```
  entityCode="ADCO"
  ```

- `wizard` value is used as value for `wizardUniqueKey` attribute.

  ```
  wizardUniqueKey="ADVICE"
  ```

- Other arguments become attribute values in pageArgs.
- New attribute `entityToLink` value should be set to ${`enterpriseEntity`}

Dynamic content conversion

In certain screens dynamic content is rendered by invocation of methods using tags.

The following code dynamically renders a call to `openPageWithKeyAndArgs`.

```
<tc:component componentType="WOString" value="${cjb.getWizardHref}" />
```

where `getWizardHref()` returns:

```
<a onClick="top.openPageWithKeyAndArgs('NEW_PROJECT_WIZARD_OBJECT', '{ wizard =
ADVICE; application = ADCO; isFromMatter=1; embeddedObject=RADC; parentKey =
3}');">
    New Advice And Counsel
</a>
```

In the previous example, `tc:component` can rewritten as the `tc:newRecordLink` tag in 3.x and 4.x.

Other Conversions

`<CLProjectTitle..>` conversion:

There is no equivalent tag for the 2.x `<CLProjectTitle..>` in 4.x. This needs to be manually handled in the screens.

An equivalent title can be displayed by writing a method in the CJB:

```
public String getProjectTitle(){
    return platform.getInternationalizationService().getObjectTitle(project);
}
```

...and then invoking that method in XML with the `tc:out` tag, like so:

```
<tc:out value="${cjb. projectTitle}"/>
```

1.10.4 Converting ExpressionBuilder References

When upgrading from TeamConnect 3.2 or earlier, searches that use `ExpressionBuilder` and directly reference custom fields will fail. As a result, you must rewrite these searches to use `SearchCriteria`. Use the ExpressionBuilderHighlighterTool to analyze customizations for references to the `ExpressionBuilder` class.

To detect existing references to `ExpressionBuilder` in your code:

1. Open the installation directory of TeamConnect on the application server.
2. Navigate to the following folder from the installation directory: `\upgrade\tools\external \ExpressionBuilderHighlighterTool`.
3. Open the `upgrade.properties` file.
4. Update the file to point to the TeamConnect database.
5. Run the `analyze.bat` script.

The `analyze.bat` script highlights all references.

1.11 Web Folders

TeamConnect includes a WebDAV server to enable management of TeamConnect documents through a compatible WebDAV client application. WebDAV is a standardized protocol for accessing and sharing files over a distributed environment. If you enable it, your users can access the TeamConnect Documents area through WebDAV clients. Mitratech supports two clients: Windows Explorer and the Mac OS WebDAV client. The WebDAV server is certified with the Windows Explorer client.

There is one place where access to the TeamConnect WebDAV server is client-specific. If the system setting "Enable WebDAV" check box is checked, then the **Documents** tab of any TeamConnect record will include a hyperlink, **Open Attachments Folder in Windows Explorer**. This same hyperlink will also appear in the Documents area of TeamConnect. Clicking this hyperlink opens a Windows Explorer Web Folder to the current record's (or current user's) documents folder. For this hyperlink to work, a Web Folder must be established on every client machine, pointing to the TeamConnect root documents directory. Any client machines that do not have this Web Folder established are not able to use the hyperlink to access WebDAV (though they may still access the WebDAV server through other clients).

Note: *Web Folders are a standard Windows feature; for more information on what Web Folders are and how they connect to servers, refer to the Microsoft Windows help.*

The following are the points that you should keep in mind when setting up Web Folders in TeamConnect:

- You can create Web Folders through the Add Network Place wizard. For specific steps and instructions, see the Microsoft Windows documentation.

- The URLs that you need to provide as the locations of the Web Folders are based on your TeamConnect URL:

  `http://Hostname:port/TeamConnect/login`

  Where you should make the following replacements:

  - Hostname:port—with the IP address and port of the machine on which TeamConnect is deployed.
  - login—with davroot for the **Root** folder in the TeamConnect Documents area.

  So the resulting URL for the Web Folder on each client machine is:

  `http://Hostname:port/TeamConnect/davroot`

  If a user needs to access their My Documents area through Web Folders, she should use URL `http://Hostname:port/TeamConnect/davroot/Users/<username>`

- Every time users access the Web Folder, they are prompted to enter their TeamConnect username and password.
- Only one TeamConnect user can access the Web Folder per Windows session. This means that if another TeamConnect user needs to access the Web Folder, Windows should be restarted on the client machine.

Index

- B -

- D -

- E -

- I -

- P -

- R -

- T -

- U -

- V -

- W -