Oracle® Business Transaction Management

Installation Guide

12.1.0.4

E47799-01

July 10, 2013

ORACLE®

Oracle Business Transaction Management Installation Guide, 12.1.0.4

E47799-01

Primary Author: Alan Davidson

Contents

Preface

Oracle Business Transaction Management Installation Guide explains how to install Business Transaction Management 12.1.0.4, including the central servers, monitors, and observers.

Audience

This document is intended for system administrators and others who want to install Business Transaction Management 12.1.0.4.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Introduction

This guide explains how to install Oracle Business Transaction Management. The information provided in this guide can help you install and configure a general purpose Business Transaction Management system suitable for experimenting with the system functions and use cases. The deployment configuration and resources required in a production environment can vary based on a variety of factors, such as anticipated throughput, message size and type, the number of applied monitoring policies, and so forth. An Oracle consultant can help you determine the appropriate configuration and resource requirements for your specific needs.

This chapter provides a high-level description of the product architecture, a description of the product packaging, and general deployment guidelines.

1.1 Architecture

At the highest level, Business Transaction Management consists of three types of components:

- **Central servers** – The central servers are application EAR files that you deploy to an application server. There are three central servers. You deploy each of these servers once, and for performance considerations you should deploy each to a separate WebLogic Server instance. You must not deploy any of the central servers to a WebLogic Server instance that hosts services or components you intend to monitor. The central servers are:
  - **Main Server** (btmMain.ear) – Contains all the central Business Transaction Management system services and user interface applications, including the *sphere*. The sphere is the Business Transaction Management component that manages the Business Transaction Management environment. In addition, btmMain.ear contains a subdeployment for the F5 intermediary.
  - **Performance Server** (btmPerformanceServer.ear) – Contains the service-level management components. Deploy btmPerformanceServer.ear on an application server other than where btmMain.ear or btmTransactionServer.ear are deployed.
  - **Transaction Server** (btmTransactionServer.ear) – Contains the transaction management components. Deploy btmTransactionServer.ear on an application server other than where btmMain.ear or btmPerformanceServer.ear are deployed.
- **Observers** – Observers are sets of libraries that you install into the application server that hosts the business applications you want to monitor. The observers monitor messages and calls between the components of your applications. Observers are capable of monitoring many types of components, and are classified

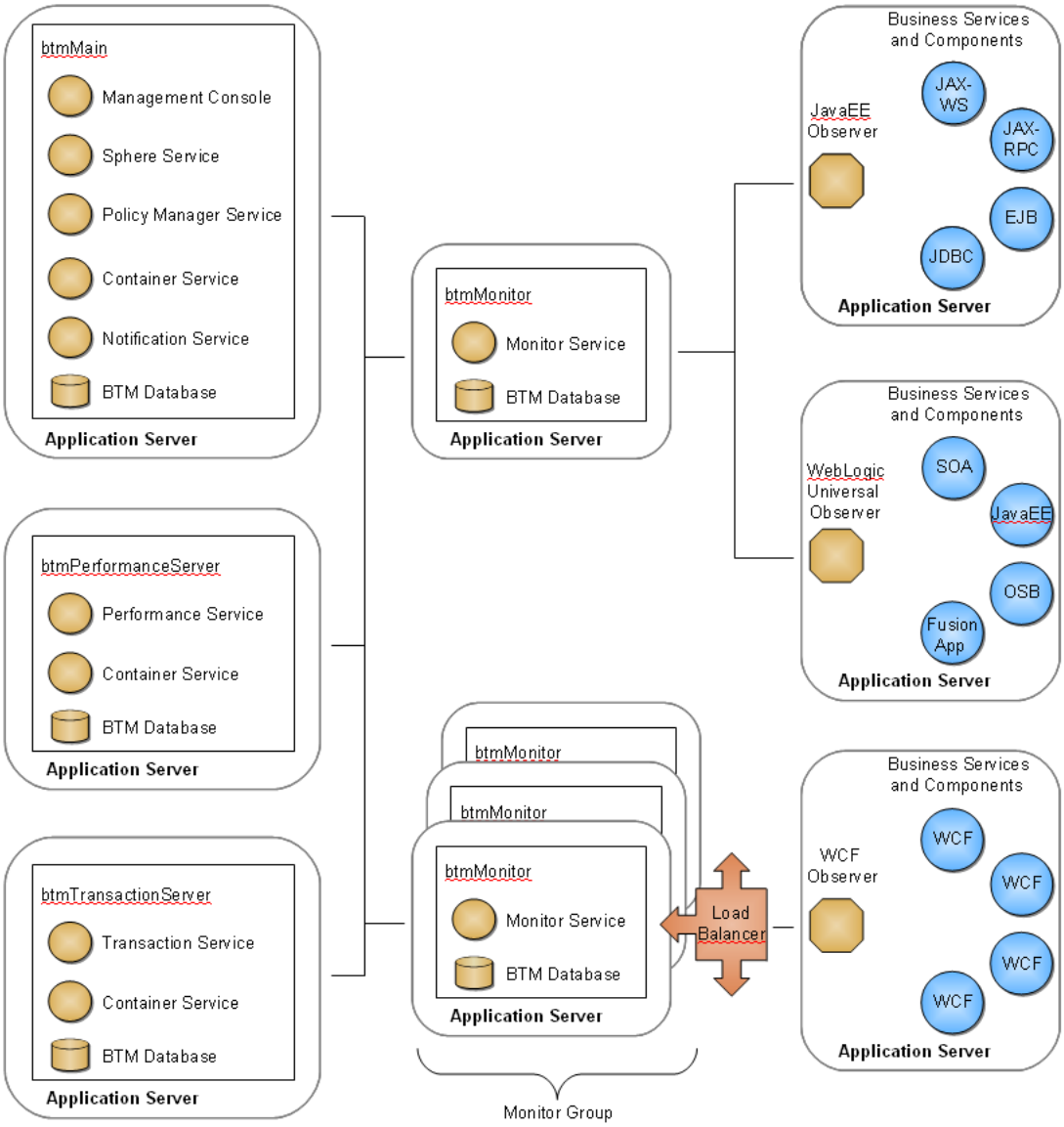
according to the types they monitor as well as the application server they are designed to run in.

- **Monitors** (btmMonitor.ear) – Monitors collect application performance and usage measurements from observers. The monitor is an application EAR file that you deploy to an application server. For large systems, you can deploy multiple monitors, either as singletons or replicates. For performance reasons, you should not deploy the monitor on an application server where the central servers are deployed.

Business Transaction Management also requires access to an Oracle RDBMS for storing performance measurements, logging messages, and maintaining the environment model and Business Transaction Management configuration.

The diagram below shows a typical distributed application environment, and the relationship of the Business Transaction Management components to that environment.

Figure 1–1 Deployment of Business Transaction Management components in a typical application environment



Business Transaction Management is designed for use in a distributed application environment in which the various Business Transaction Management components are deployed onto multiple machines and application servers.

Technically, you can install all the central servers into a single application server, but such a deployment scenario is not recommended for production environments. Installation in a single application server can be useful for demonstrations and for learning how to use the product, but this scenario might not scale successfully with a large number of business services or high volume of message traffic, just to name a few factors.

We recommend that you deploy each of the central servers to separate application servers. The Performance and Transaction components, in particular, typically perform a large amount of performance analysis computations. Dividing processes across application servers allows you to control memory and processor resources.

You should also deploy the monitor to an application server separate from the central servers. Depending on your monitoring requirements, you might need to deploy multiple monitors. You can deploy monitors either as singletons or as replicates behind a load balancer. For information about replicating the monitor, refer to Chapter 7, "Installing Monitors."

Observers must always be installed outside the application server hosting the central servers or monitors.

1.2 Packaging

Oracle distributes Business Transaction Management by way of ZIP files. The central servers and monitor are packaged together in BTM_Servers_*.zip (the * refers to the Business Transaction Management version number). The ZIP file's archive directory contains the central server and monitor deployments. Table 1–1 describes these deployments:

Table 1–1 The archives directory of BTM_Servers_*.zip contains the central servers and monitor in the following EAR files.

Deployment Name	Sub-deployments	Deployment strategy
btmMain.ear	btmcentral.war btmcontainer.war btmhelp.war btmui.war f5Intermediary.war	Deploy once per Business Transaction Management environment.
btmPerformanceServer.ear	btmcontainer.war btmperformance.war	Deploy once per Business Transaction Management environment on a separate application server from btmMain.ear and btmTransactionServer.ear.
btmTransactionServer.ear	btmcontainer.war btmtransaction.war	Deploy once per Business Transaction Management environment on a separate application server from btmMain.ear and btmPerformanceServer.ear.
btmMonitor.ear	btmmonitor.war	Deploy as many as needed on separate application servers from any of the central servers.

Observers are packaged in individual ZIP files according to platform and observer type. Table 1–2 lists the release 12.1.0.4 observers. The * in the ZIP file names refers to the observer version number, for example, 12.1.0.4.

Table 1–2 Observers distributed with release 12.1.0.4.

Observer ZIP File Name	Description
BTMObserver_Wls_10.3_Universal_*.zip	Contains the observer for JavaEE, Oracle SOA Suite, Oracle Service Bus 11gR1, and Oracle Fusion Applications (ADF-UI, ADF-BC and SOA deployments) running in a WebLogic 10.3 server.
BTMObserver_Wls_12_JavaEE_*.zip	Contains the observer for JavaEE running in a WebLogic 12 server.
BTMObserver_Wls_10.3_Osb11gR1_*.zip	Contains the observer for Oracle Service Bus 10gR3 on WebLogic 10.3. **Note:** Although the ZIP file name contains the string "Osb11gR1", this observer supports Oracle Service Bus 10gR3. This observer is not recommended for Oracle Service Bus 11gR1 (you should instead use BTMObserver_Wls_10.3_Universal_*.zip for that purpose).
BTMObserver_OEG_11.1.1.6_OEG_*.zip	Contains the observer for Oracle Enterprise Gateway 11.1.1.6.
BTMObserver_Iis_7.5_DotNet4_*.zip	Contains the observer for WCF 3.5 and 4.0 on Microsoft IIS version 7.5.

The observers listed in Table 1–3 and Table 1–4 are distributed as part of earlier releases. These observers are not included with, but are compatible with, release 12.1.0.4. You should install these observers if you require monitoring on their designated platforms. (The * in each ZIP file name refers to the observer version number, for example, 12.1.0.1.0.) Installation instructions for these observers are not included in the current release of this installation guide. You can find installation instructions for these observers in the installation guide for their designated release.

Table 1–3 Release 12.1.0.3 observers not included with, but compatible with, release 12.1.0.4.

Observer ZIP File Name	Description
BTMObserver_Oc4j_10.1_Soa10g_*.zip	Contains the observer for an Oracle Containers for Java (OC4J) instance running within Oracle SOA Suite 10g.

Table 1–4 Release 12.1.0.1 observers not included with, but compatible with, release 12.1.0.4*.*

Observer ZIP File Name	Description
BTMObserver_Wls_9.2_JavaEE_*.zip	Contains the observer for JavaEE on WebLogic 9.2.
BTMObserver_Was_6.1_JavaEE_*.zip	Contains the observer for JavaEE on WebSphere 6.1.
BTMObserver_Jboss_4.3_JavaEE_*.zip	Contains the observer for JavaEE on JBossEAP 4.3.

2

Upgrading Business Transaction Management

This chapter explains how to perform an in-place upgrade of Business Transaction Management from any version of release 11 or 12 to release 12.1.0.4. Performing an in-place upgrade means that you upgrade components by simply replacing them with new components, without editing configuration settings.

2.1 Overview

A complete upgrade of Business Transaction Management consists of upgrading these components:

- all three central servers—these are, the Main server (btmMain.ear), the Performance server (btmPerformanceServer.ear), and the Transaction server (btmTransactionServer.ear)
- all monitors (btmMonitor.ear)
- all observers

If any of these components are older than release 11, do not attempt to perform an upgrade using the instructions in this chapter. Instead, enter a service request at My Oracle Support (http://support.oracle.com) for assistance in upgrading your installation.

2.1.1 Ordering of Upgrade Steps

The order in which you upgrade the components of your Business Transaction Management system is important and depends on the release version of your installed components and on whether your observers are supported by the release 12.1.0.4 monitors. So, before proceeding further, complete these steps:

1. Take note of the release numbers of your central servers, monitors, and observers:

   To determine the release of your installed observer, open the observer's NanoAgent.log file and search for the line that displays the release number, for example:

   ```
   INFO: Release 11.1.0.4: build 25237 of b21 (11.1.0.4/147754) on 2011-05-13
   ```

   For information about locating the NanoAgent.log file, see Chapter 13, "Logging Observer Errors and Debugging Information."

   You can find the release number of your central servers and monitors in the README.txt file located in the nanoagent directory of the distribution ZIP file.

2. Determine whether your observers are supported by the release 12.1.0.4 monitors:

   If your observers are release 12.1.0.2 or newer, they are supported.

   If your observers are release 11.x, they are not supported.

   If your observers are release 12.x, but older than 12.1.0.2, you must check the Business Transaction Management (BTM) Certification Matrix. You can find this document online at `http://support.oracle.com`. If your observer is listed in this document, then it is supported by release 12.1.0.4. If it is not listed, it is not supported.

Read the following scenarios to determine how to order the steps of your upgrade procedure, and then refer to section 2.2, "Upgrading the Central Servers and Monitors" and section 2.3, "Upgrading Observers" for detailed explanations of the upgrade steps:

All components are current

If your central servers and monitors are at release 12.1.0.3 or newer and your observers are supported by release 12.1.0.4, then you can perform a rolling upgrade as follows:

1. **Central Servers** – Shut down all the central servers together, upgrade them, then restart them.
2. **Monitors** – Shut down the monitors, upgrade them, then restart them. You can upgrade monitors one at a time, or all together.
3. **Observers** – *(Optional)* Shut down the observers, upgrade them, then restart them. You can upgrade observers one at a time, or all together.

Central servers and monitors are old but observers are still supported

If your central servers and monitors are older than release 12.1.0.3 but your observers are supported by release 12.1.0.4, order your upgrade steps as follows:

1. **Central Server and Monitors** – Shut down all the central servers and monitors together, upgrade them, and then restart them.
2. **Observers** – *(Optional)* Shut down the observers, upgrade them, then restart them. You can upgrade observers one at a time, or all together.

Observers are no longer supported

If your observers are so old that they are not supported by release 12.1.0.4, then you can follow either of the following upgrade paths:

- Shut down all the central servers, monitors, and observers together, upgrade them, and then restart them.
- First, upgrade your observers to a release that is supported by release 12.1.0.4 but that is not newer than your monitors (observers must never be of a newer release than their associated monitors). Then, upgrade your central servers and monitors to release 12.1.0.4. Finally (and optionally), upgrade your observers to release 12.1.0.4.

> **Note:** In comparing version numbers between an observer and monitor, you can ignore digits following the final (fourth) point. For example, in “11.2.0.1.3”, you can ignore the “3”. Final digits refer to patches and are not important in determining compatibility between observers and monitors.

2.1.2 Importance of Upgrading Observers

After upgrading both the central servers and monitors, you should upgrade your observers, if possible. If your observers are older than release 12, you must upgrade them. If your observers are release 12 and still supported, then upgrading them is optional. Note that if you do not upgrade your observers, you cannot take advantage of new functionality that depends on upgraded observers.

Note: The upgrading of central servers older than release 12.1.0.3 adds an observer authentication field to new Observer Communication policies that you create. This observer authentication field is enabled by default. If you want to use observer authentication, you must also upgrade your observers. Observers from releases older than 12.1.0.3 do not have the capability to authenticate themselves. If you don't upgrade your observers to at least 12.1.0.3, then you must disable the observer authentication field in any new policies that you create or else the observer will not be able to connect with the monitor.

2.1.3 Resolving Post-Upgrade Issues (This is Important!)

After you finish upgrading your Business Transaction Management system, read the online help topic named **Functional Upgrade Issues**. This topic provides information about upgrade-related tasks you might want to perform and about changes in behavior you should expect to see in your upgraded system. You can locate this topic by first choosing **Help > Help** in the Management console. After the online help opens, navigate to **Overview of Oracle Business Transaction Management** in the **Contents** pane, and then scroll down to the **Functional Upgrade Issues** section.

The **Functional Upgrade Issues** topic addresses issues such as:

- Utilizing new probe types
- Fixing broken transactions
- Identifying and removing artifacts related to deprecated probes

2.2 Upgrading the Central Servers and Monitors

> **Note:** Before beginning the upgrade procedure, we suggest that you deactivate your RMI probe unless you require monitoring of RMI calls. Most applications utilize RMI by way of higher-level APIs, such as JAX-RPC, JAX-WS, EJB, and JMS. If your application utilizes RMI only by way of these higher-level APIs, you should deactivate the RMI probe. However, if your application calls RMI directly, you should leave it activated.
>
> To deactivate your RMI probe, first open your Observer Communication policy. The **Active Probes** section of the policy provides an **Enable Discovery** and **Monitor Upon Discovery** checkbox for each type of probe. Deselect both of these checkboxes for RMI. If you are using multiple Observer Communication policies, you should perform this task for each policy.
>
> After deactivating your RMI probe, you should unregister any RMI components that have been discovered. To unregister an RMI component, select the component (modeled as a service) in the Business Transaction Management Console and choose **Modify > Delete** *Your_Service* **Registration**, where *Your_Service* is the name of your RMI component. You can alternatively use the **unregister** CLI command. For more information, refer to the Business Transaction Management online help.

To upgrade your Business Transaction Management installation:

1. Back up your Business Transaction Management databases and configuration data.

   For information on how to perform this task, refer to the online help topic **Backing up and Restoring Business Transaction Management**. You can locate this topic by first choosing **Help > Help** in the Management console. After the online help opens, navigate to **Administration of Business Transaction Management > Backing up and Restoring Business Transaction Management** in the **Contents** pane.

   > **Note:** The central servers must be running in order for you to access the online help.

2. Back up the persistent storage directories for each of the central servers and monitors.

   For information on the locations of the persistent storage directories, refer to the online help topic **Persistent Data**. You can locate this topic by first choosing **Help > Help** in the Management console. After the online help opens, navigate to **Administration of Business Transaction Management > Persistent Data** in the **Contents** pane.

3. Ensure that all WebLogic domains in which the Business Transaction Management central servers and monitors are installed include the Java Required Files (JRF) template. (If you are upgrading a 12.1.0.3 installation, then you have already met this requirement.)

If any of these domains doesn't include the JRF template, extend the domain and add the template. You will get the following exception when you attempt to start the server if the JRF template is not included in the domain:

```
java.lang.ClassNotFoundException:
oracle.security.jps.wls.listeners.JpsApplicationLifecycleListener
```

Note: The JRF template is part of the Oracle Application Development Framework (ADF) runtime, which means that you must install the ADF runtime into your WebLogic installation before you can extend any domain with the JRF template. When installing the ADF runtime, take care to install the release version that matches your version of WebLogic. You can download the ADF runtime at:

```
http://www.oracle.com/technetwork/developer-tools/adf/downloads/index.html
```

---

4. Locate the distribution archive that contains the Business Transaction Management central servers and monitor and unzip it into a directory (referred to henceforth as *Install_Dir*).

   The distribution archive is named BTM_Servers_*.zip, where * is the Business Transaction Management version number.

5. *Optional security step for UNIX-like operating systems* – If you want to set permissions on the files that make up the distribution to the most restrictive level that still maintains functionality, complete this step:

   a. Locate setPermissions.sh at the top level of *Install_Dir*.

      This script contains commands for setting file permissions of all regular files to Owner – read/delete; all directories to Owner – read/execute/delete; and all scripts to Owner – read/execute/delete.

      Note: These permission levels are extremely restrictive, for example, only the owner can read the files.

      ---

   b. On a command line, at the top level of *Install_Dir*, run this command:

      ```
      source setPermissions.sh
      ```

      This command runs the commands in the script file and sets permissions for all files and directories in the expanded archive.

6. Upgrade the central server and monitor EAR files:

   If you are upgrading from release 12.1.0.3 or newer, you can either upgrade all of your central server and monitor EAR files together, or upgrade all of your central server EAR files and then upgrade your monitor EAR files one at a time.

   If you are upgrading from a release older than 12.1.0.3, you must upgrade all of your central server and monitor EAR files together.

   To upgrade the central server and monitor EAR files together:

   a. Shut down all of the central servers and monitors (btmMain.ear, btmPerformanceServer.ear, btmTransactionServer.ear, and btmMonitor.ear).

      It is essential that you shut them all down.

For information about shutting down Business Transaction Management components, see Chapter 12, "Starting and Shutting Down Business Transaction Management."

b. Using your application server's deployment tools, redeploy each of the central servers and monitors using your new EAR files located in *Install_Dir*\archives.

c. Restart the central servers and monitors.

When upgrading from a release older than 12.1.0.4, restarting the central servers triggers the in-place upgrade of Business Transaction Management data. During this process, the system might suffer from reduced performance and some of the data might be temporarily unavailable. This process should complete within 30 minutes. If you are simply patching a 12.1.0.4 installation, your system should start up at its normal speed.

For information about starting Business Transaction Management components, see Chapter 12, "Starting and Shutting Down Business Transaction Management."

d. Notify all Business Transaction Management users to flush their web browser caches.

The Management Console contains a number of Adobe Flash widgets. Web browsers normally cache these widgets and will continue to use the older cached widgets until you either flush the cache or restart your web browser.

To upgrade the central server EAR files followed by the monitor EAR files:

a. Shut down all of the central servers (btmMain.ear, btmPerformanceServer.ear, and btmTransactionServer.ear).

For information about shutting down Business Transaction Management components, see Chapter 12, "Starting and Shutting Down Business Transaction Management."

b. Using your application server's deployment tools, redeploy each of the central servers using your new EAR files located in *Install_Dir*\archives.

c. Restart the central servers.

When upgrading from a release older than 12.1.0.4, restarting the central servers triggers the in-place upgrade of Business Transaction Management data. During this process, the system might suffer from reduced performance and some of the data might be temporarily unavailable. This process should complete within 30 minutes. If you are simply patching a 12.1.0.4 installation, your system should start up at its normal speed.

For information about starting Business Transaction Management components, see Chapter 12, "Starting and Shutting Down Business Transaction Management."

d. Notify all Business Transaction Management users to flush their web browser caches.

The Management Console contains a number of Adobe Flash widgets. Web browsers normally cache these widgets and will continue to use the older cached widgets until you either flush the cache or restart your web browser.

e. Shut down your monitors (btmMonitor.ear), redeploy them using your new EAR file located in *Install_Dir*\archives, and restart them.

You can perform this step for all monitors at once, or perform it individually for each monitor.

After you are finished with upgrading the central servers and monitors, and the system has settled down, you can optionally upgrade your observers by following the instructions in the next section.

2.3 Upgrading Observers

You must upgrade the Business Transaction Management central servers and monitors, as described in the preceding section, before upgrading the observers.

> **Note:** This installation guide provides instructions for upgrading observers to versions that are included as part of release 12.1.0.4. If you want to upgrade an observer to a version from an earlier release, consult the installation guide from that earlier release. In particular, if you want to upgrade an observer for WebLogic 9.2, WebSphere, JBoss, or ASP.NET, consult the installation guide for release 12.1.0.2.2.

The procedure for upgrading observers is specific to the application server and observer type. Refer to the following sections for detailed instructions on upgrading observers:

- section 2.3.1, "Upgrading Observers on WebLogic 10.3"
- section 2.3.2, "Upgrading the Observer for WCF"
- section 2.3.3, "Upgrading the Observer for Oracle Enterprise Gateway"

> **Note:** For detailed information about a specific observer's compatibility and functionality, refer to the README.txt file located in the observer's nanoagent directory after you expand the observer ZIP file.

2.3.1 Upgrading Observers on WebLogic 10.3

This section describes how to upgrade an observer installed into a WebLogic 10.3 application server.

In releases previous to 12.1.0.3, a variety of observers were provided for installing into WebLogic 10.3 servers. Each type of observer contained a set of probes that gave it the ability to monitor a particular set of component types. For example, the JavaEE observer contained probes for monitoring JavaEE components and the SOA observer contained probes for monitoring SOA components. In the current release, only two observers are provided for installing into WebLogic 10.3 servers—the "universal" observer and the observer for Oracle Service Bus 10gR3. These observers are packaged in the following ZIP files:

- **BTMObserver_Wls_10.3_Universal_*.zip** – This ZIP file contains the universal observer. Use this ZIP file to upgrade the following observers:
  - JavaEE observer for WebLogic 10.3
  - Oracle Fusion Applications observer
  - Oracle SOA Suite observer
  - Oracle Service Bus 11gR1 observer

The universal observer is capable of monitoring JavaEE, Oracle SOA Suite, Oracle Service Bus 11gR1, and Oracle Fusion Applications (ADF-UI, ADF-BC and SOA deployments).

- **BTMObserver_Wls_10.3_Osb11gR1_*.zip** – Use this ZIP file to install the observer for Oracle Service Bus 10gR3 into a WebLogic 10.3 server. Note that although the ZIP file name contains the string "Osb11gR1", this observer supports Oracle Service Bus 10gR3. This observer is not recommended for Oracle Service Bus 11gR1 (you should instead use BTMObserver_Wls_10.3_Universal_*.zip for that purpose).

If you upgrade from a limited observer (such as a JavaEE observer) to the universal observer, your system will gain the ability to discover and monitor new types of components. Depending upon your system and your monitoring needs you might, or might not, want to monitor new component types. Before upgrading to the universal observer, you should review and adjust, if necessary, the active probes in your Observer Communication policy. For more information on this topic, read the online help topic named **Functional Upgrade Issues**. You can locate this topic by first choosing **Help > Help** in the Management console. After the online help opens, navigate to **Overview of Oracle Business Transaction Management** in the **Contents** pane and then scroll down to the **Functional Upgrade Issues** section.

Note: If you require an updated version of an observer that monitors only JavaEE, Oracle Fusion Applications, Oracle SOA Suite, or Oracle Service Bus, enter a service request at My Oracle Support (http://support.oracle.com) for assistance.

Separate procedures are provided for three different upgrade paths. The procedure you use depends on the type and release version of your installed observer and whether your WebLogic server is Node-Manager configured or script-configured. These are the procedures to choose from:

- section 2.3.1.1, "Upgrading 12.1.0.4, 12.1.0.3, or 12.1.0.2.2 Observers, the Observer for OSB 10gR3, or the 12.1.0.2.0 Observer for JavaEE"
- section 2.3.1.2, "Upgrading a 12.1.0.2.0 Observer for SOA or OSB 11gR1, or an 11 or 12.1.0.1 Observer for JavaEE on a Node Manager-Configured Server"
- section 2.3.1.3, "Upgrading a 12.1.0.2.0 Observer for SOA or OSB 11gR1, or an 11 or 12.1.0.1 Observer for JavaEE on a Script-Configured Server"

2.3.1.1 Upgrading 12.1.0.4, 12.1.0.3, or 12.1.0.2.2 Observers, the Observer for OSB 10gR3, or the 12.1.0.2.0 Observer for JavaEE

Use this procedure to upgrade the following observers on WebLogic servers:

- release 12.1.0.4 universal observer
- release 12.1.0.3 universal observer
- all release 12.1.0.2.2 observers
- release 12.1.0.2.0 observer for JavaEE
- the observer for Oracle Service Bus 10gR3

1. Locate the appropriate observer distribution ZIP file for your environment and monitoring needs.
2. Shut down the WebLogic application server in which the observer is installed.

3. Make a backup copy of the *WL_HOME*\nanoagent directory.

   The string *WL_HOME* refers to your WebLogic server's home directory, which is the weblogic92, wlserver_10.0, or wlserver_10.3 directory located in your WebLogic installation directory.

4. Delete the *WL_HOME*\nanoagent directory.

5. Unpack the observer distribution ZIP file into *WL_HOME*.

   Unpacking the ZIP file creates directories named nanoagent and security_add_ons. The nanoagent directory contains three subdirectories named bin, config, and lib.

6. Ensure that the user account running WebLogic has at least the following privileges:

   - read permission on the nanoagent/config and nanoagent/lib directories (on UNIX-like systems traverse permission is also required)
   - read permission on all JAR files in the lib directory

7. If you originally installed the observer by editing script files rather than by using the Node Manager, replace your new observer script file with the observer script file located in your backup copy of the nanoagent directory. If you use the Node Manager, you can skip this step.

   On Windows systems, the observer script file is located at:

   ```
   WL_HOME\nanoagent\bin\nanoEnvWeblogic.cmd
   ```

   On UNIX-like systems, the observer script file is located at:

   ```
   WL_HOME/nanoagent/bin/nanoEnvWeblogic.sh
   ```

8. Restart your WebLogic server.

2.3.1.2 Upgrading a 12.1.0.2.0 Observer for SOA or OSB 11gR1, or an 11 or 12.1.0.1 Observer for JavaEE on a Node Manager-Configured Server

Use this procedure to upgrade the following observers to the current release in a WebLogic 10.3 Node Manager-configured environment:

- release 12.1.0.2.0 observer for Oracle SOA Suite
- release 12.1.0.2.0 observer for Oracle Service Bus 11gR1
- release 11 or 12.1.0.1 observer for JavaEE

1. Locate the distribution ZIP file for the universal observer (BTMObserver_Wls_10.3_Universal_*.zip).

2. Open the WebLogic Administration Console (the default URL is http://*machine_name*:7001/console).

3. Remove *WL_HOME*/nanoagent/lib/bootstrap/ap-nano-bootstrap.jar from your managed server's classpath (this setting was required by the version of the observer you are upgrading from, but must be removed for the current release):

   a. Using the Domain Structure pane (on the left), navigate to **Environment > Servers**.

   b. In the **Servers** table, click your managed server.

   c. Display the **Configuration / Server Start** tab.

d. Click **Lock & Edit**.

> **Note:** These instructions assume you are operating in a production environment and that your WebLogic server's **Automatically Acquire Lock and Activate Changes** setting is therefore disabled. However, if this setting is enabled as it might be in a development environment, you do not have to click **Lock & Edit** in order to make changes and you do not have to activate changes after saving them.

e. Delete the entry from the **Class Path** field as follows:

For Windows systems, delete this string:

```
WL_HOME\nanoagent\lib\bootstrap\ap-nano-bootstrap.jar
```

For UNIX-like systems, delete this string:

```
WL_HOME/nanoagent/lib/bootstrap/ap-nano-bootstrap.jar
```

f. Keep this page open.

4. With the **Configuration / Server Start** tab still displayed, edit your WebLogic startup arguments as follows:

a. Remove JVM arguments from the **Arguments** field as follows (these settings were required by the previous release of the observer but must be removed for the current release):

For Windows systems, remove these arguments:

```
-Daspectwerkz.classloader.preprocessor=com.amberpoint.nanoagent.plugins.APA
spectPreProcessor -javaagent:
WL_HOME\nanoagent\lib\bootstrap\aspectwerkz-jdk5-2.0.jar
```

For UNIX-like systems, remove these arguments:

```
-Daspectwerkz.classloader.preprocessor=com.amberpoint.nanoagent.plugins.APA
spectPreProcessor -javaagent:
WL_HOME/nanoagent/lib/bootstrap/aspectwerkz-jdk5-2.0.jar
```

b. Configure the observer bootstrap module into your server by adding a JVM argument to the **Arguments** field as follows.

For Windows systems, add this argument:

```
-javaagent:WL_HOME\nanoagent\lib\bootstrap\ap-nano-bootstrap.jar
```

For UNIX-like systems, add this argument:

```
-javaagent:WL_HOME/nanoagent/lib/bootstrap/ap-nano-bootstrap.jar
```

5. Click **Save** and then click **Activate Changes**.

You should receive the following status:

"All changes have been activated. No restarts are necessary. Settings updated successfully."

6. Shut down the WebLogic managed server in which the observer is installed.

7. Delete the *WL_HOME*\nanoagent directory from your WebLogic managed server.

The string *WL_HOME* refers to the server's home directory, which is the wlserver_10.3 directory located in your server's installation directory. For the remainder of this procedure, replace the string *WL_HOME* with the actual path to the WebLogic home directory.

8. Unpack the observer distribution ZIP file into *WL_HOME*.

   Unpacking the ZIP file creates a directory named nanoagent that contains three subdirectories bin, config, and lib.

9. Ensure that the user account running the WebLogic server has at least the following privileges:

   - read permission on the nanoagent/config and nanoagent/lib directories (on UNIX-like systems traverse permission is also required)
   - read permission on all JAR files in the lib directory

10. If you inserted <filter> and <filter-mapping> elements into the web.xml files of your web applications during your original installation of the observer, you must now remove those elements.

    The elements to remove from the web.xml files are these:

```
<filter>
  <filter-name>ORACLE_BTM_WEB_APP_OBSERVER</filter-name>
  <filter-class>
   com.amberpoint.nanoagent.bootstrap.servlet.FilterHandlerBootstrap
  </filter-class>
</filter>
<filter-mapping>
  <filter-name>ORACLE_BTM_WEB_APP_OBSERVER</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

11. Restart your WebLogic server.

2.3.1.3 Upgrading a 12.1.0.2.0 Observer for SOA or OSB 11gR1, or an 11 or 12.1.0.1 Observer for JavaEE on a Script-Configured Server

Use this procedure to upgrade the following observers to the current release in a WebLogic 10.3 script-configured environment:

- release 12.1.0.2.0 observer for Oracle SOA Suite
- release 12.1.0.2.0 observer for Oracle Service Bus 11gR1
- release 11 or 12.1.0.1 observer for JavaEE

> **Note:** When you originally installed the observer, you either installed it into all servers defined in the WebLogic installation or into servers of specific domains. The term *global install* refers to installing the observer into all servers, and the term *domain install* refers to installing the observer into a specific domain.

In the following procedure, you will replace the nanoagent directory and its contents. If you installed the observer as a global install, the nanoagent directory is located inside your WebLogic server's home directory, which is the wlserver_10.3 directory located in your WebLogic installation directory.

If you installed the observer as a domain install, the nanoagent directory is located inside your domain directory. In this case, you need to repeat this procedure for each domain in which you installed the observer.

1. Locate the distribution ZIP file for the universal observer (BTMObserver_Wls_10.3_Universal_*.zip).
2. Shut down the WebLogic application server in which the observer is installed.
3. Make a backup copy of your observer script file.

   The observer script file is the nanoEnvWeblogic.cmd or nanoEnvWeblogic.sh file located inside the nanoagent/bin directory.
4. Delete the nanoagent directory.
5. Unpack the observer distribution ZIP file into the directory from which you deleted the nanoagent directory.

   Unpacking the ZIP file creates a directory named nanoagent that contains three subdirectories bin, config, and lib.
6. Ensure that the user account running WebLogic has at least the following privileges:
   - read permission on the nanoagent/config and nanoagent/lib directories (on UNIX-like systems traverse permission is also required)
   - read permission on all JAR files in the lib directory
7. Open your new observer script file for editing and open your backup copy so you can copy settings from it.
8. Ensure that the values of the NANOAGENT_HOME and NANOAGENT_CONFIGURATION_URL variables in your new observer configuration file match the values in your backup copy.
9. If you inserted <filter> and <filter-mapping> elements into the web.xml files of your web applications during your original installation of the observer, you must now remove those elements.

   The elements to remove from the web.xml files are these:

```
<filter>
  <filter-name>ORACLE_BTM_WEB_APP_OBSERVER</filter-name>
  <filter-class>
   com.amberpoint.nanoagent.bootstrap.servlet.FilterHandlerBootstrap
  </filter-class>
</filter>
<filter-mapping>
  <filter-name>ORACLE_BTM_WEB_APP_OBSERVER</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

10. Restart your WebLogic server.

2.3.2 Upgrading the Observer for WCF

This section explains how to upgrade an observer for WCF.

1. Unpack the observer distribution file (BTMObserver_Iis_7.5_DotNet4_*.zip) into a temporary directory (referred to henceforth as *observer_temp*).

   Unpacking the ZIP file creates a nanoagent directory containing two subdirectories—config and lib. The lib directory contains the observer DLL files.

2. Make a note of the version number of the new DLLs.

   To find the version number, open the Windows Properties dialog box for one of the DLL files and click the Version tab.

3. Use gacutil.exe or a Windows Explorer to copy all of the DLL files from *observer_temp*\nanoagent\lib to the global application cache (GAC; normally located at C:\WINDOWS\assembly).

4. Using a text editor, open the application configuration file that contains the observer configuration code that you added when you originally installed the observer.

   The file is either the machine.config, or a web.config file.

5. In the application configuration file, locate the two occurrences of the Version attribute that refer to the version number of the observer DLLs.

   The elements containing the Version attributes are inside a </behaviorExtensions> element and should look similar to this:

```
<add name="APEPInterceptor"
type="AmberPoint.NanoAgent.DotNet.Wcf.APEPBehaviorExtnElem,
AmberPoint.NanoAgentWCF, Version=64000.64000.25233.19024, Culture=neutral,
PublicKeyToken=d8685c0afbb35893"/>

<add name="APServiceInterceptor"
type="AmberPoint.NanoAgent.DotNet.Wcf.APServiceBehaviorExtnElem,
AmberPoint.NanoAgentWCF, Version=64000.64000.25233.19024, Culture=neutral,
PublicKeyToken=d8685c0afbb35893"/>
```

6. Edit the setting of the Version attribute so that it matches the version number of the new observer DLLs, and then save and close the file.

   Note: If you configured the observer into multiple web.config files on the machine, edit the Version attributes in each file.

7. *Optional* – Remove the old observer DLLs from the GAC (unless they are being used by another observer on the machine).

   The name of each observer DLL begins with the string "AmberPoint". To remove a DLL, right-click it and choose **Uninstall**.

   Note: The observer for ASP.NET uses many of the same DLLs as the observer for WCF. If you have the observer for ASP.NET installed on the machine, you must not remove the version of the DLLs that are being used by it.

2.3.3 Upgrading the Observer for Oracle Enterprise Gateway

This section explains how to upgrade an observer for Oracle Enterprise Gateway.

1. Shut down your Enterprise Gateway server.

2. Delete all of the observer-related JAR files from *OEG_HOME*/ext/lib, where *OEG_HOME* is your Enterprise Gateway server's home directory (the top-level installation directory).

The observer-related JAR files are those prefixed with “ap-” plus the following files:

```
orahttp_client_1.0.0.jar
orawsdl_1.0.0.jar
xstream-1.2.2.jar
```

3. Unpack the observer ZIP file (BTMObserver_OEG_11.1.1.6_OEG_*.zip) into a temporary directory, referred to henceforth as *observer_temp*.

   Unpacking the ZIP file creates three directories named config, lib, and scripts.

4. Copy all of the JAR files located in the lib directory to *OEG_HOME*/ext/lib.

5. Open *OEG_HOME*/system/conf/jvm.xml in a text editor and make the following changes inside the <JVMSettings> element:

   a. Locate the following line (it should be the 1st child of the <JVMSettings> element):

   ```
   <SystemProperty name="AP_NANO_CONFIG_URL" value="http://Host_Name:Port_
   Number/btmmonitor/agent/agent/"/>
   ```

   b. Add the following lines as the 2nd, 3rd, and 4th children of the <JVMSettings> element:

   ```
   <SystemProperty name="AP_NANO_HOME" value="$VDISTDIR/NanoAgent"/>
   <SystemProperty name="AP_NANO_LOG_BASEDIR" value="$VDISTDIR/NanoAgent"/>
   <SystemProperty name="AP_NANO_CLASSLOADER_BASEDIR"
   value="$VDISTDIR/ext/lib"/>
   ```

   c. *Optional* – If you want to target this observer by way of an observer configuration label, add the following line as the 5th child of the <JVMSettings> element:

   ```
   <SystemProperty name="ap.nano.config.label" value="My_Label_String"/>
   ```

   Replace *My_Label_String* with the string you want to use as a label for this observer. For more information about targeting observers refer to Section 7.5, "Applying an Observer Communication Policy."

6. Restart your Enterprise Gateway server.

3 Installation Overview

This chapter provides an overview of the entire Business Transaction Management installation procedure.

3.1 Installation Overview

1. Installation and initial configuration of the central servers:
   a. *Optional* – Configure security for the central servers. You can also perform the security configuration for the monitors and observers at this time, if you wish.

      See Chapter 4, "Configuring Security."

   b. Ensure that you have a supported web browser installed.

      See Section 5.1, "Web Browser Requirements."

   c. Ensure that observers have access to WSDL and schema resources.

      See Section 5.2, "Access to WSDL and Schema Resources."

   d. Configure the application servers that will host your central servers.

      See section 5.3, "Setting up your WebLogic Environment".

   e. Set up the Business Transaction Management databases.

      See Section 5.4, "Setting up Business Transaction Management Databases."

   f. *Optional* – Configure the persistent storage directories.

      See Section 6.2, "Configuring Persistent Storage Directories."

   g. Deploy the central servers.

      See Section 6.3, "Deploying the Central Servers."

   h. If needed, remap Business Transaction Management user roles in your application server.

      See Section 6.4, "Mapping Users to Roles."

   i. Perform initial configuration of Business Transaction Management.

      After deploying the central servers, you configure them using the browser-based configuration wizard (see Section 6.5, "Initial Configuration of Business Transaction Management.").

      Alternatively, you can use a command line script (see Chapter 14, "Scripted Configuration of Oracle Business Transaction Management."). For first-time configuration, however, we recommend that you use the browser-based

wizard. The wizard produces an XML output file that can be used with the command line script for subsequent configurations.

j. Configure the location of your Oracle Enterprise Manager server.

   See Section 6.6, "Configuring the Connection to Enterprise Manager."

2. Installation of monitors and configuration of the Observer Communication policy (see Chapter 7, "Installing Monitors."):

   a. *Optional* – Configure security for the monitors if you haven't already done so.

      See Chapter 4, "Configuring Security."

   b. Configure the application servers that will host your monitors.

      See section 5.3, "Setting up your WebLogic Environment".

   c. Deploy monitors.

   d. Configure monitor-observer communication by way of the Observer Communication policy.

3. Installation of observers (see Chapter 8, "Installing Observers Overview"):

   a. *Optional* – Configure security for the observers if you haven't already done so.

      See Chapter 4, "Configuring Security."

   b. Install the observer libraries.

   c. Configure the observers to locate a monitor.

4

Configuring Security

This chapter explains how to configure security for Business Transaction Management. You should read this chapter thoroughly, and, if you decide to configure security, you should perform the security configuration for the relevant execution environment before installing any Business Transaction Management components into it.

You can configure security for Business Transaction Management using the following mechanisms:

- Network-level security, in which you configure your application servers to communicate over HTTPS.

  For management components that communicate using HTTPS, this security mechanism provides one-way or mutual authentication, message integrity, and encryption of on-the-wire messaging data. Refer to section 4.2, "Setting up Network-Level Security" for details.

- Business Transaction Management's built-in security for HTTP(S) communications between Business Transaction Management components.

  This security mechanism provides trust between management components that communicate using HTTP(S) and encryption of sensitive elements of their messaging data prior to serialization (that is, before being sent over the wire or stored to disk). Refer to section 4.3, "Configuring the Assertion Secret and Encryption Key" for details.

- Business Transaction Management's built-in security for the transmission of observation messages from observers to monitors using Observation Protocol Secure (OPS). OPS is the Oracle proprietary Observation Protocol (OP) over a secure socket (SSL).

  This security mechanism provides authentication between an observer and monitor, message integrity, and encryption of observation messages sent over the wire. Server authentication (authentication of the monitor to the observer) is provided by SSL. Client authentication (authentication of the observer to the monitor) is provided by OPS. Refer to section 4.4, "Setting up a Secure Socket (SSL) for Observation Messages" for details.

4.1 Communication Protocols and Deployment Scenarios

Business Transaction Management is designed for use in a distributed environment. As such, some components might be deployed in remote data centers, in a DMZ, or in a network without access to a centralized database. The following table describes the type of access required between the various Business Transaction Management components and between those components and your business services.

Table 4–1 Business Transaction Management Communication Protocols

From this Component	Required Protocol	To this Component	Description	Index to Diagrams
Each central server (the Main, the Transaction, and the Performance server)	HTTP or HTTPS	Any other central server	The central servers must be able to communicate with each other. The usual deployment topology will place each of these in the same network zone.	A
Each central server	HTTP or HTTPS	All monitors	Each of the central servers must be able to communicate with all monitors. This includes monitors that are deployed outside of the central network zone.	B
Main server	TCP, HTTP or HTTPS	All monitored business services	The Main server should be able to communicate with the monitored business services to determine whether they are “alive” (up or down). Note that the precise protocol used to determine aliveness can be configured on a per-service basis. (For information on configuring this protocol, enter a service request at My Oracle Support.)	C
Each monitor	HTTP or HTTPS	All central servers	Monitors must be able to communicate with each of the central servers.	D
Each monitor	JDBC	A Message Log Database (messageLogDB)	Each monitor must be able to write to, and read from, its Message Log Database using JDBC. The Message Log Database can be a single database inside a trusted network that is used by all monitors. Alternatively, if you have monitors in a DMZ or in a network without access to a centralized database, you can deploy one or more databases for the remote monitors.	E
Each observer	TCP and HTTP or HTTPS	Associated monitor	Each observer requires two connections to its associated monitor. The observer uses an HTTP(S) connection to download its configuration from the monitor. To transmit observation messages to the monitor, the observer connects using Observation Protocol (OP) over TCP or Observation Protocol Secure (OPS) over a secure socket. For monitor groups, these connections are made to the load balancer that fronts the monitor group. The socket port number is configured by way of the Observer Communication policy.	F
Transaction server	JDBC	All Message Log Databases	Optional. The Transaction server does not require direct access to the Message Log Databases. However, by providing such access, you can improve the performance of message log queries. You should enable this communication channel whenever possible.	G
Main server	JDBC	Sphere database (sphereDB)	The Main server must be able to write to the Sphere database using JDBC.	H

Table 4–1 (Cont.) Business Transaction Management Communication Protocols

From this Component	Required Protocol	To this Component	Description	Index to Diagrams
Performance	JDBC	Measurement database (measurementDB)	The Performance server must be able to write to the Measurement database using JDBC.	I
Transaction	JDBC	Transaction database (transactionDB)	The Transaction server must be able to write to the Transaction database using JDBC.	J
Monitor (in a monitor group)	JDBC	Monitor group database	Each monitor in a particular monitor group must be able to write to the same monitor group database using JDBC.	K

The following diagrams illustrate the type of access required by Business Transaction Management components in a variety of deployment scenarios. Note that the circled letters in the diagrams cross reference Table 4–1, " Business Transaction Management Communication Protocols".

- Figure 4–1, "Deployed Business Transaction Management Components and Business Services"
- Figure 4–2, "Communication Connections Among the Central Servers and Between the Central Servers and Monitors"
- Figure 4–3, "Connections Between Business Transaction Management Components and Databases"
- Figure 4–4, "Communication Connections Between the Monitors and Observers"
- Figure 4–5, "Communication Connections Between the Main Server and Business Services"

Figure 4–1 Deployed Business Transaction Management Components and Business Services

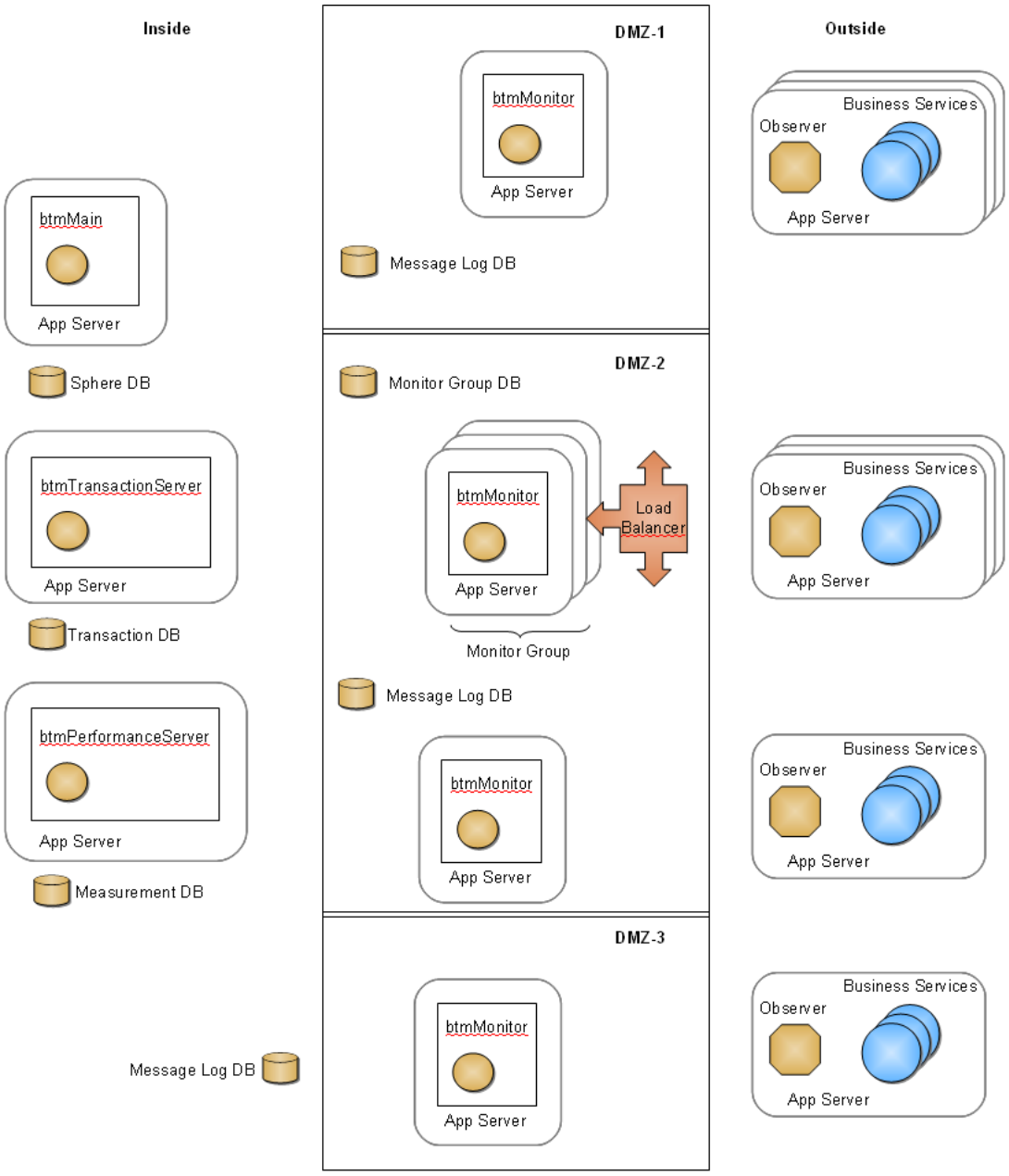


Figure 4–2 Communication Connections Among the Central Servers and Between the Central Servers and Monitors

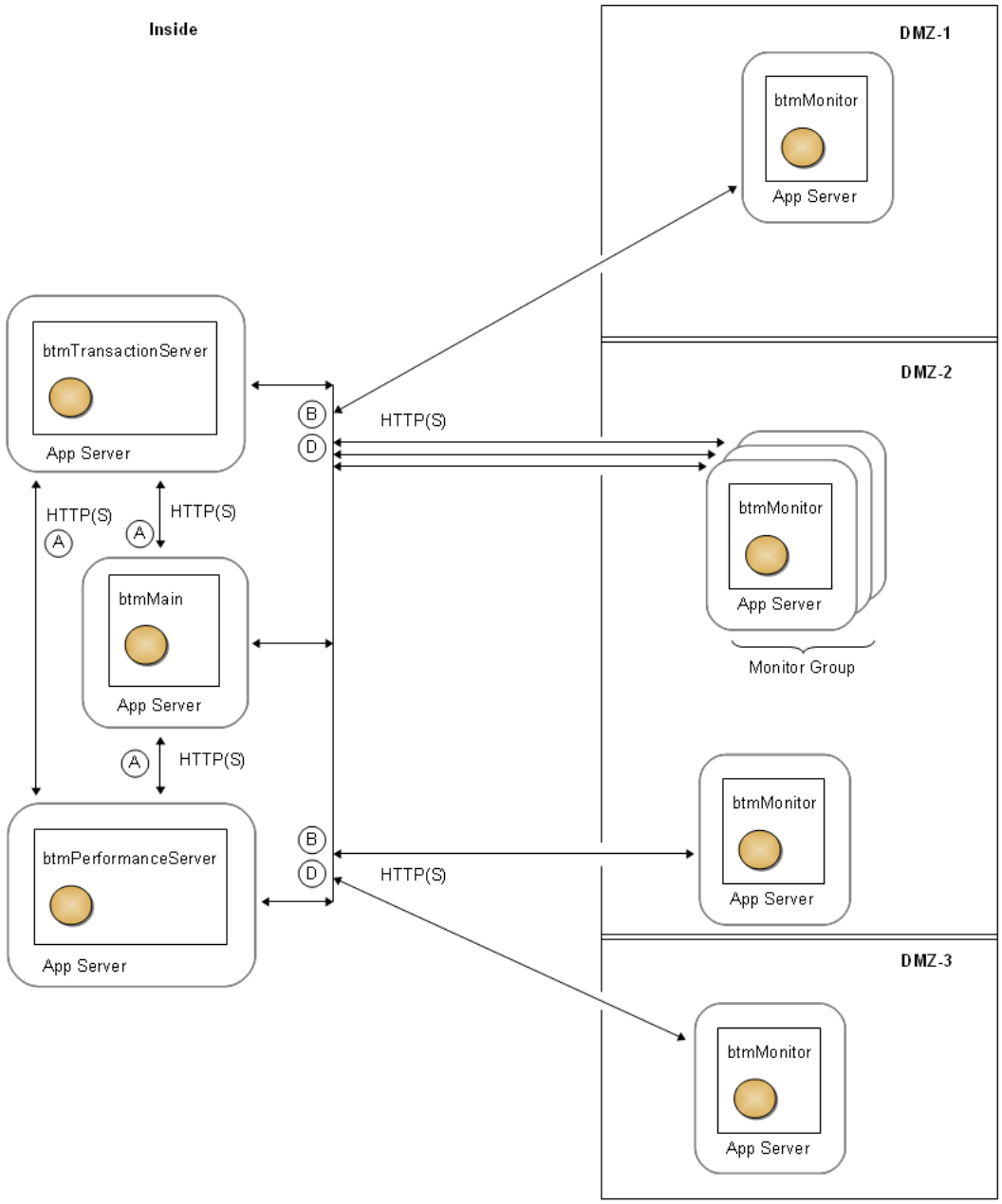


Figure 4–3 Connections Between Business Transaction Management Components and Databases

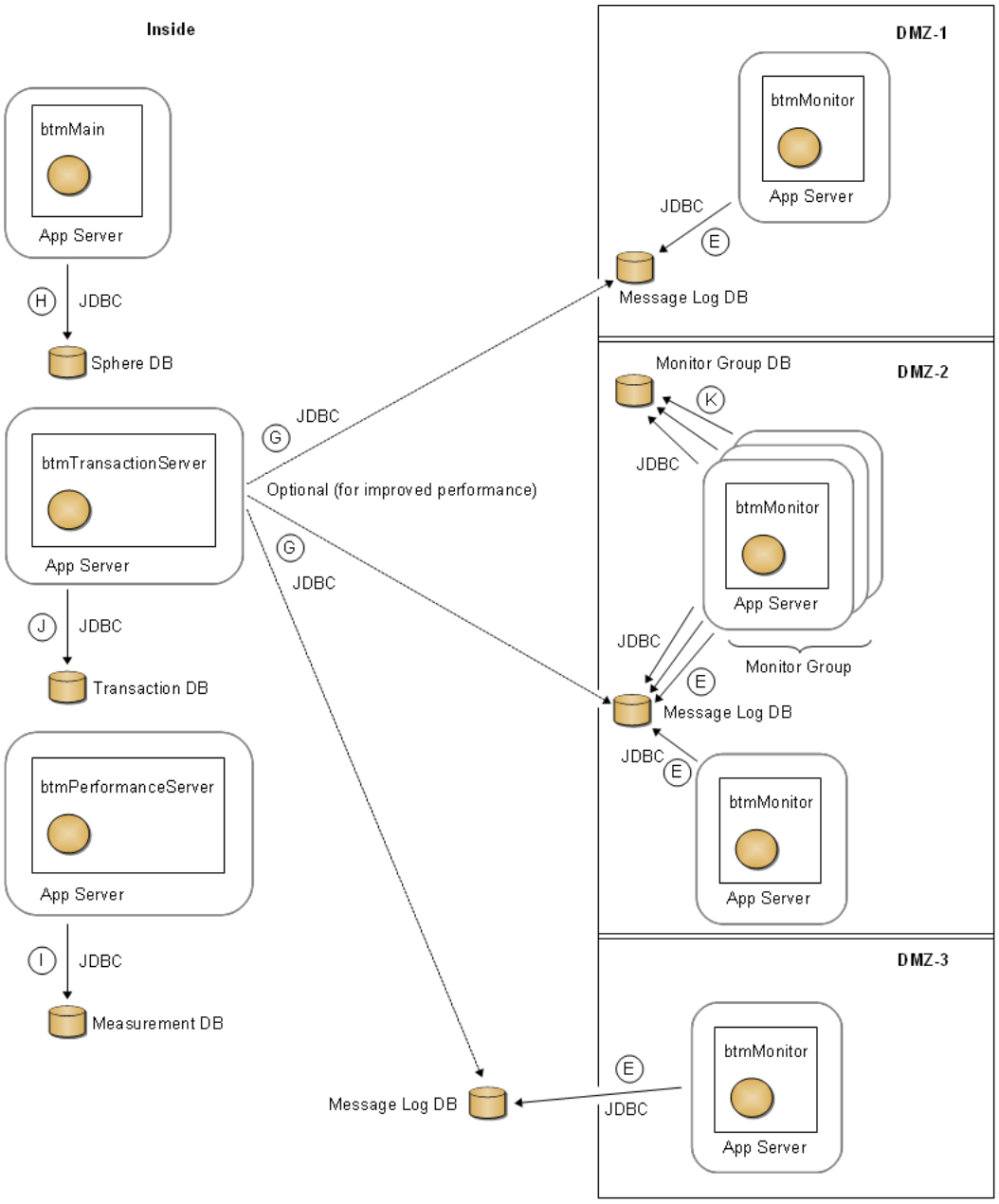


Figure 4–4 Communication Connections Between the Monitors and Observers

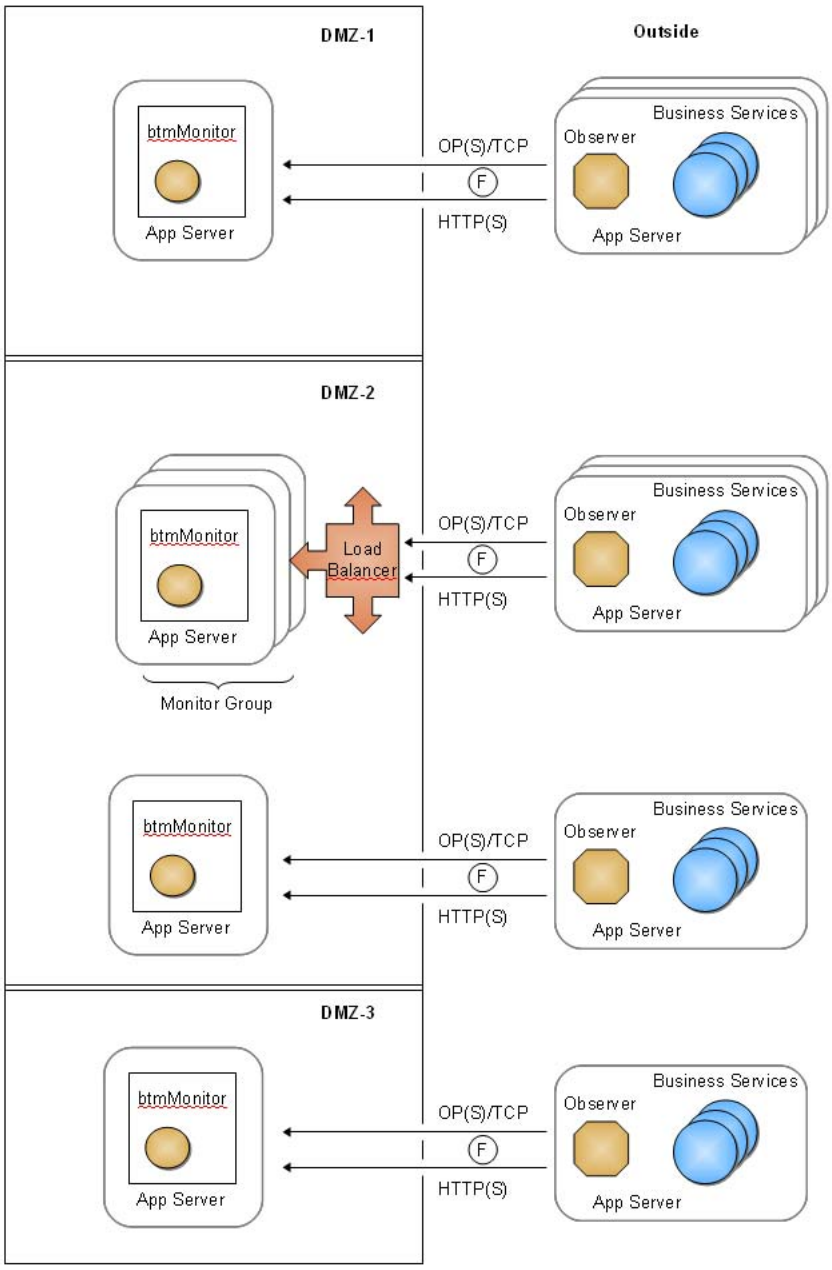
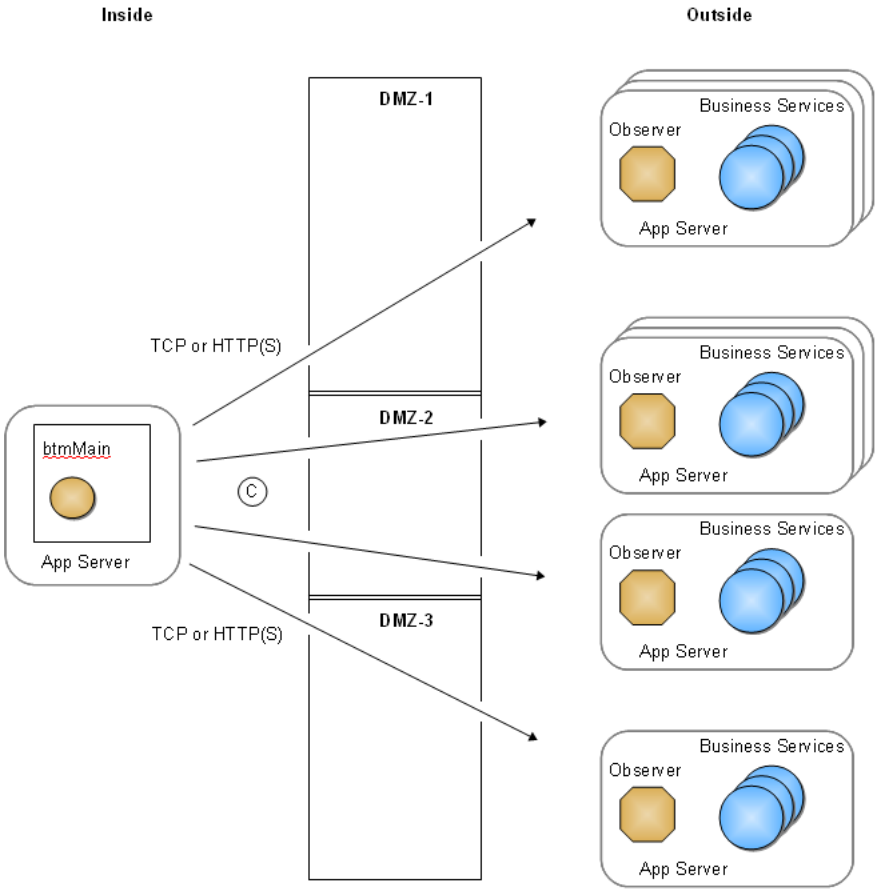


Figure 4–5 Communication Connections Between the Main Server and Business Services



4.2 Setting up Network-Level Security

If you want to enable network-level security for Business Transaction Management components, you can do so using SSL/TLS provided by the application servers in which the components are deployed. SSL provides message integrity and confidentiality for communication among distributed management components, encrypting management traffic as it flows from one server to another. When configured with client certificates, SSL also provides mutual authentication between management components. Business Transaction Management provides no specialized support for SSL and does not interfere with standard SSL configurations. Therefore, any configuration supported by your application server can be used to secure Business Transaction Management traffic. It is strongly recommended that you first ensure that your application servers are properly configured for SSL before you install any Business Transaction Management components.

Once you have configured and tested SSL on the application servers where Business Transaction Management will be deployed, you will install Business Transaction Management and then configure it using the browser-based Initial Configuration wizard. During initial configuration, you will provide the SSL address and port number for your installation. The Business Transaction Management components will automatically communicate over the secured transports.

Note: Observers and monitors communicate with each other by way of two separate channels—a socket connection for transmitting observation messages, and an HTTP/HTTPS connection for administration-related messages. Setting up SSL for the socket connection is covered in Section 4.4, "Setting up a Secure Socket (SSL) for Observation Messages." Setting up an HTTPS connection is a separate and unrelated task that you perform using the security features of your application servers.

4.2.1 Configuring HTTPS

This section pertains to all Business Transaction Management components, including the command line interface (CLI).

If you are installing the central servers into an HTTPS-only environment (that is, an environment in which there is no HTTP traffic between components), you must do one of the following:

- Ensure that the application servers hosting the central servers are listening only on HTTPS ports.

  Or

- Modify each web.xml file packaged in each of the central server WAR files so as to specify that only HTTPS access is allowed. You can do this by simply uncommenting the following <security-constraint> element that is provided in those files:

```
<security-constraint>
  <display-name>Require SSL communication</display-name>
  <web-resource-collection>
    <web-resource-name>All AmberPoint system services</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

4.2.2 Setting JSEE Properties on WebLogic Servers

As a part of configuring WebLogic servers to communicate over HTTPS, you must set a number of Java system properties in your servers. These properties specify the information needed by your WebLogic servers to access their security stores. The information you provide in the properties is as follows:

Note: In the following list, *two-way SSL* refers to an SSL connection that requires mutual authentication. *One-way SSL* refers to an SSL connection that requires only server authentication.

- location of the client's trust store *(required for one- or two-way SSL)*
- type of the trust store *(required for one- or two-way SSL)*
- password for accessing the trust store *(required for one- or two-way SSL)*
- location of the server's key store *(required for two-way SSL)*

- type of the key store *(required for two-way SSL)*
- password for accessing the key store *(required for two-way SSL)*

WebLogic servers allow you to specify this information either by way of javax.net.ssl.* (JSEE) properties or by way of weblogic.security.* properties. WebLogic servers can read this information from either type of property. However, Business Transaction Management also requires access to this same information but can only read the information from JSEE properties. For this reason you can provide the required information by setting JSEE properties only, or by setting both weblogic.security.* and JSEE properties. If you do not set the JSEE properties, your Business Transaction Management components will fail to start up.

> **Note:** You must set the JSEE properties even if you have disabled the **Use JSSE SSL** checkbox on the **Environment > Servers >** *servername* **> Configuration > SSL > Advanced** tab in the WebLogic Administration Console.

You set the JSEE properties as Java options either by way of the WebLogic Administration Console or in the WebLogic server startup script. These are the options to set for two-way SSL:

```
-Djavax.net.ssl.trustStore=path_to_my_trustStore

-Djavax.net.ssl.trustStoreType=type_of_my_trustStore

-Djavax.net.ssl.trustStorePassword=my_trustStore_password

-Djavax.net.ssl.keyStore=path_to_my_keyStore

-Djavax.net.ssl.keyStoreType=type_of_my_keyStore

-Djavax.net.ssl.keyStorePassword=my_keyStore_password
```

The following example shows how to set the JSEE properties on Windows for two-way SSL in the WebLogic server startup script using the WebLogic Demo security stores:

```
set JAVA_OPTIONS=-Djavax.net.ssl.trustStore=C:\Oracle\Middleware\wlserver_
10.3\server\lib\DemoTrust.jks -Djavax.net.ssl.trustStoreType=JKS
-Djavax.net.ssl.trustStorePassword=MyTrustStorePassword
-Djavax.net.ssl.keyStore=C:\Oracle\Middleware\wlserver_
10.3\server\lib\DemoIdentity.jks -Djavax.net.ssl.keyStoreType=JKS
-Djavax.net.ssl.keyStorePassword=MyKeyStorePassword %JAVA_OPTIONS%
```

If you set these properties in your WebLogic server startup script, be sure to insert the line before the following line or the JAVA_OPTIONS value might be overwritten (the first example is for Windows and the second is for UNIX-like operating systems):

```
set SAVE_JAVA_OPTIONS=%JAVA_OPTIONS%

SAVE_JAVA_OPTIONS="${JAVA_OPTIONS}"
```

> **Note:** Use of the WebLogic Scripting Tool (WLST) in an HTTPS environment requires the weblogic.security.* properties.

4.2.3 Setting JSEE Properties for the Command Line Interface

If you are using the Business Transaction Management Command Line Interface (CLI) in an HTTPS environment, then you must set the same JSSE properties that you set on your WebLogic servers in Section 4.2.2. The following example shows how to set the JSSE properties by way of the CLI's built-in variable, AP_OPTS. This example uses the the WebLogic Demo security stores:

```
set AP_OPTS=-Djavax.net.ssl.trustStore=C:\Oracle\Middleware\wlserver_
10.3\server\lib\DemoTrust.jks -Djavax.net.ssl.trustStoreType=JKS
-Djavax.net.ssl.trustStorePassword=MyTrustStorePassword
-Djavax.net.ssl.keyStore=C:\Oracle\Middleware\wlserver_
10.3\server\lib\DemoIdentity.jks -Djavax.net.ssl.keyStoreType=JKS
-Djavax.net.ssl.keyStorePassword=MyKeyStorePassword
```

4.2.4 Configuring Firewalls

This section pertains to the central servers, monitors, and observers.

If you are using a firewall, you must configure it to allow access to each port on which a Business Transaction Management component receives communications. These are:

- the ports on which the hosting application servers listen.
- the sockets on which the monitors receive observation messages from the observers.
- the ports used for JDBC connections to Business Transaction Management databases. These databases are discussed in Section 5.4, "Setting up Business Transaction Management Databases."

4.3 Configuring the Assertion Secret and Encryption Key

Note: If you choose to perform the configuration described in this section, then you must do so on each application server that hosts a Business Transaction Management central server, monitor, or observer. You must also perform this configuration for the execution environment in which you use the Business Transaction Management command line interface (CLI).

Communications between Business Transaction Management components are secured by way of trusted assertions. This means that for your Business Transaction Management components to communicate with each other, and for your Business Transaction Management installation to function properly, every Business Transaction Management component must be configured with an assertion secret of the same value.

Business Transaction Management also encrypts sensitive data contained in the communications between its components. It encrypts this data for both on-the-wire communications and storage in the Business Transaction Management databases.

These security mechanisms are enabled by default, and all Business Transaction Management components are preconfigured with a default value for both the assertion secret and the encryption key. This default security configuration fully enables the security mechanisms and, at the same time, simplifies the installation of Business Transaction Management.

However, because every Business Transaction Management installation has the same default values, using the default values is a potential security threat. For demonstration purposes, and perhaps for development environments, using the default values might be adequate. But, in production environments, you should tighten security by providing your own unique values. You should also use your own values in your test environment before deploying Business Transaction Management into your production environment. If you intend to provide your own values for the assertion secret and encryption key, you should perform that configuration on each application server that hosts a Business Transaction Management component before you deploy the component.

For components deployed to WebLogic servers, you have a choice as to the method you use for configuring and storing these security settings—you can use either Oracle Wallet (see Section 4.3.1) or Business Transaction Management extension properties (see Section 4.3.2). (Oracle Wallet is an implementation of Oracle Credential Store Framework, or OCSF, and is a component of Java Platform Security, or JPS.) For all other Java application servers, and for Oracle Enterprise Gateway, you use Business Transaction Management extension properties. If necessary, you can mix these security configuration methods. For example, you could use Oracle Wallet on some application servers but use Business Transaction Management extension properties on other application servers.

4.3.1 Configuring Security Using Oracle Wallet

This section pertains to central servers, monitors, and observers that will be deployed to WebLogic application servers. It explains how to configure the assertion secret and encryption key for your Business Transaction Management components using Oracle Wallet. You must repeat this procedure for each central server, monitor, and observer for which you want to configure the assertion secret and encryption key using Oracle Wallet.

1. Ensure that the WebLogic domain in which the Business Transaction Management component will be installed includes the Java Required Files (JRF) template.

   If the domain does not yet exist, be sure to add the JRF template when you create the domain. If the domain already exists but doesn't include the JRF template, extend the domain and add the template.

   The JRF template includes the JPS JAR files that are referred to in later steps. (JPS is a component of Oracle Platform Security Services.)

2. Decide on names for the credentials that will hold your assertion secret and encryption key.

   You are free to choose any name you want, but each of the two credentials must have a different name, and you must use the same two credential names for all of your Business Transaction Management components.

3. Decide on values for your issuer name and issuer assertion secret.

   You are free to choose any values you want for these strings, but you must use the same values for all of your Business Transaction Management components.

4. Locate the distribution archive for the Business Transaction Management central servers and expand it into a directory on the machine that hosts the central server, monitor, or observer for which you want to configure security (this directory is henceforth referred to as *BTM_Central_Expanded*).

   The distribution archive is named BTM_Servers*.zip, where * represents the version number.

5. Configure the setBtmOverriderEnv_via_CredStore script file by completing the following substeps.

   Configuring the setBtmOverriderEnv_via_CredStore script file accomplishes these tasks: adds the JPS JAR files to your application server's classpath; overrides the default credential store; and specifies the names of the credentials that hold the shared secret and encryption key (note that you will create the credentials in a later step).

   a. Locate the setBtmOverriderEnv_via_CredStore script file in *BTM_Central_Expanded*/security_add_ons.

      For Windows systems, use setBtmOverriderEnv_via_CredStore.cmd; for UNIX-like systems, use setBtmOverriderEnv_via_CredStore.sh.

   b. Copy the script file to your WebLogic server:

      If you want to configure security for all domains, copy the script file to your WebLogic server's home directory. The home directory is the weblogic92 or wlserver_10.3 directory located in your WebLogic installation directory, for example, C:\bea\wlserver_10.3.

      If you want to configure security for a particular domain, copy the script file to the top level of that domain's directory.

   c. Open the script file in a text editor.

   d. Set the **BTM_TIS_CRED_NAME** variable to the name of the credential that holds the assertion secret by replacing the string **>>> YOUR_ISSUER_SECRET_CREDENTIAL_NAME_HERE <<<** with the credential name you chose for your assertion secret.

   e. Set the **BTM_ENCKEY_CRED_NAME** variable to the name of the credential that holds the encryption key by replacing the string **>>> YOUR_ENCRYPT_KEY_CREDENTIAL_NAME_HERE <<<** with the credential name you chose for your encryption key.

   f. Save and close the script file.

6. Configure your WebLogic domain startup scripts to call the setBtmOverriderEnv_via_CredStore script file:

   Note: This step assumes that you haven't modified your startWebLogic scripts. If you have modified your scripts, you might also have to modify the installation procedure accordingly.

   a. Navigate to the bin directory of one of the WebLogic domains for which you want to configure security and open the startup script in a text editor (open bin\startWebLogic.cmd for Windows systems or bin/startWebLogic.sh for UNIX-like systems; do not edit the startup script located directly within the domain directory).

   b. Locate the following line (the first line is for Windows and the second for UNIX-like systems):

      ```
      call "%DOMAIN_HOME%\bin\setDomainEnv.cmd"

      . ${DOMAIN_HOME}/bin/setDomainEnv.sh
      ```

   c. Directly after that line, add a line that calls the setBtmOverriderEnv_via_CredStore script file:

If you are configuring security for all domains on a Windows system, add this line:

```
call "%WL_HOME%\setBtmOverriderEnv_via_CredStore.cmd"
```

If you are configuring security for all domains on a UNIX-like system, add this line (note the initial period and space):

```
. ${WL_HOME}/setBtmOverriderEnv_via_CredStore.sh
```

If you are configuring security for a particular domain on a Windows system, add this line:

```
call "%DOMAIN_HOME%\setBtmOverriderEnv_via_CredStore.cmd"
```

If you are configuring security for a particular domain on a UNIX-like system, add this line (note the initial period and space):

```
. ${DOMAIN_HOME}/setBtmOverriderEnv_via_CredStore.sh
```

d. Repeat this step for each domain for which you want to configure security.

7. Add permission grants for Business Transaction Management components to the policy store:

a. Navigate to the **config\fmwconfig** directory of one of the WebLogic domains for which you are configuring security.

This directory was created when you extended the domain to include the JRF template and is the default location where the oracle.security.jps.config property looks.

b. Open the file **system-jazn-data.xml** in a text editor.

c. For each deployment in the domain for which you want to configure security, add a **<grant>** element as a child of the **<jazn-data>/<jazn-policy>** element, replacing the string **Your Deployment Name Here** with the name of the deployment, as shown in the following example:

```
<jazn-data>
 <jazn-policy>
 (... other <grant> elements, etc. ...)

  <!-- Begin of Business Transaction Management grants -->
  <grant>
   <grantee>
    <codesource>
     <url>file:${domain.home}/servers/${weblogic.Name}/tmp/_WL_user/Your
Deployment Name Here/-</url>
    </codesource>
   </grantee>
   <permissions>
    <permission>
     <class>
      oracle.security.jps.service.credstore.CredentialAccessPermission
     </class>
     <name>context=SYSTEM,mapName=BTM,keyName=*</name>
     <actions>read,write</actions>
    </permission>
   </permissions>
  </grant>
  <!-- End of Business Transaction Management grants -->
```

```
 (... other <grant> elements, etc. ...)
 </jazn-policy>
</jazn-data>
```

Table 1–1 lists the deployment names for the central servers and monitors. For observers, use the name of the deployment that contains the business application you want to monitor.

d. Save and close the file.

e. Repeat this step for each domain for which you want to configure security.

8. Use the Business Transaction Management command line interface (CLI) to create a "Trusted Issuer and Secret" credential that contains your assertion secret and add it to the Oracle Wallet credential store associated with the policy store that you just configured:

In the following substeps, you will first set up the environment for the CLI, then point the CLI to the Oracle Wallet credential store, and finally use the CLI to create the credential for your assertion secret and add it to the store.

a. Open a command shell or window and ensure that its JAVA_HOME and JPS_11_HOME environment variables are set properly.

These variables can be set at the system or the shell/window level. The JAVA_HOME variable must point to a JDK that is version 6 or higher. The JPS_11_HOME variable must point to your oracle.jps file, for example, to:

```
oracle\Middleware\oracle_common\modules\oracle.jps_11.1.1
```

b. Open the file *BTM_Central_Expanded*/**security_add_ons/jps_config_dir/jps-config.xml** in a text editor and locate the following line:

```
<serviceInstance name="credstore" provider="credstoressp" location="./">
```

c. Change the value of the **location** attribute from **./** to the **/config/fmwconfig/** directory under the WebLogic domain directory, for example:

```
<serviceInstance name="credstore" provider="credstoressp"
location="C:/Oracle/Middleware/user_projects/domains/my_
domain/config/fmwconfig/">
```

The value of the **location** attribute must match the location of the %DOMAIN_HOME%\config\fmwconfig directory where you configured the policy store for your domain, in step 7.

d. In your command shell/window, navigate to the *BTM_Central_Expanded*/tools directory and issue the following command (replacing *my_credential_name* with the name you chose for this credential in step 5d):

```
btmcli credStoreTool -createCred my_credential_name -credType is
```

The CLI then prompts you for the Trusted Assertion Issuer and the Trusted Assertion Secret. The latter is masked with asterisks as you type.

e. Input the values for the Trusted Assertion Issuer and the Trusted Assertion Secret (and keep the command shell/window open for the next step).

Use the values you chose in step 3.

> **Note:** If you prefer not to use the interactive mode, you can provide the issuer name and assertion secret by appending **-credValue** *my_issuer*:*my_secret* to the command line. However, using interactive mode is more secure because of the masking.

The credential for your assertion secret has now been created and added to the Oracle Wallet credential store.

9. In the same command shell/window, use the CLI to create a credential for your encryption key by issuing the following command (replacing *my_credential_name* with the name you chose for this credential in step 5e):

   ```
   btmcli credStoreTool -createCred my_credential_name -credType bin -genKey
   AES:128
   ```

   This command creates a credential with the given name, generates an AES, 128-bit, random encryption key, and adds it to the Oracle Wallet credential store.

10. Retrieve the encryption key for use in configuring the other machines in your system by issuing the following command (replacing *my_credential_name* with the name of your credential):

    ```
    btmcli credStoreTool -getCred my_credential_name -credType bin -showSecret
    ```

    This command returns the string value of your encryption key. You will need to copy this string to the other machines in your system as you configure them.

11. Repeat the pertinent steps of this procedure for each central server, monitor, and observer for which you want to configure the assertion secret and encryption key using Oracle Wallet, but on repetition make the following changes:

    - Skip steps 2 and 3 (you will use the same credential names and issuer name and issuer assertion secret values that you used on the first machine).
    - Issue the following CLI command instead of the commands shown in steps 9 and 10 (replacing *my_credential_name* with the name of your credential, and *my_encryption_key_string* with the string you retrieved in step 10):

      ```
      btmcli credStoreTool -createCred my_credential_name -credType bin
      -credValue my_encryption_key_string
      ```

      This command creates a credential with the given name and encryption key, and adds it to the Oracle Wallet credential store.

4.3.2 Configuring Security Using Extension Properties

This section pertains to central servers, monitors, and observers. It explains how to configure the assertion secret and encryption key using Business Transaction Management extension properties. You must repeat this procedure for each central server, monitor, and observer for which you want to configure the assertion secret and encryption key using extension properties.

There are two basic pieces to this security mechanism:

- The first piece of the mechanism is a set of files in which you declare your extension properties and set their values.

  The assertion secret and encryption key each require their own extension property file. If you are setting both, then you will require two extension property files. You place both extension property files in the same directory and protect the files by

way of your operating system's file security mechanism. The application server in which the Business Transaction Management component is running must have permission to read the extension property files. For details, see Section 4.3.2.1, "Setting up the Extension Property Files."

- The second piece of the mechanism is a Java system property (or for observers in .NET environments, a Windows environment variable) that points to the directory containing the extension property files. For details, see Section 4.3.2.2, "Setting up the Pointer."

4.3.2.1 Setting up the Extension Property Files

1. Create a directory to hold the extension property files.

   You can use any name for the directory. (The remainder of this procedure assumes the directory name is ext_props.)

2. If you want to configure the assertion secret, complete the following substeps:

   a. Inside the ext_props directory, create a file named com.amberpoint.SimpleIdentityAssertion.properties.

   b. Add this line to the file:

   ```
   TrustedIssuerOverriderClassName=com.amberpoint.wsclient.TrustedIssuerOverriderByExtProp
   ```

   c. Then add this line to the file:

   ```
   TrustedIssuerSecretOverride=MySecret
   ```

   where *MySecret* is your own secret string.

   d. By default, the name of the issuer of the security assertion is AmberPoint. If you want to override this default issuer's name, add this third line to the file:

   ```
   TrustedIssuerNameOverride=MyIssuerName
   ```

   where *MyIssuerName* is the name of the issuer of the security assertion.

   e. Close the file.

3. If you want to configure the encryption key, complete the following substeps:

   a. Inside the ext_props directory, create a file named com.amberpoint.security.encryption.properties.

   b. Add these two lines to the file:

   ```
   SystemDefaultAESKeyOverriderClassName=com.amberpoint.security.util.SystemDefaultAESKeyOverriderByExtProp

   aes.defaultKeySize=128
   ```

   c. Then add this line to the file:

   ```
   aes.defaultKey=MyEncryptionKey
   ```

   where *MyEncryptionKey* is a base 64-encoded, AES, 128-bit key.

   After generating your base 64-encoded encryption key, you can copy and paste it in order to set the value of this property. If your key includes special characters, you should enclose it in double quotes, for example:

   ```
   aes.defaultKey="oylJKoTGXTHasOYwtjwA7g=="
   ```

d. Close the file.

4. Using your operating system's file security mechanism, secure your extension property files (com.amberpoint.SimpleIdentityAssertion.properties and com.amberpoint.security.encryption.properties).

   Ensure that the application server in which the Business Transaction Management component is running has permission to read the extension property files.

4.3.2.2 Setting up the Pointer

The procedure you use to set up the pointer to the extension property files depends on the type of server in which your Business Transaction Management component is deployed. This section describes how to set up the pointer for a:

- Java application server (see Section 4.3.2.2.1)
- Oracle Enterprise Gateway server (see Section 4.3.2.2.2)
- .NET Environment (see Section 4.3.2.2.3)

4.3.2.2.1 Setting up the Pointer for a Java Application Server

1. Open your server's startup script in a text editor.
2. Create a Java system property named com.amberpoint.util.Extension.dir and set its value to the location of the extension property directory, for example:

   For Windows systems:

   ```
   set JAVA_OPTIONS=-Dcom.amberpoint.util.Extension.dir=C:\btm\ext_props %JAVA_
   OPTIONS%
   ```

   For UNIX-like systems:

   ```
   JAVA_OPTIONS=-Dcom.amberpoint.util.Extension.dir="/home/btm/ext_props" "${JAVA_
   OPTIONS}"
   ```

   Note: For WebLogic servers, add the line to the startWebLogic script after the call to setDomainEnv.

4.3.2.2.2 Setting up the Pointer for an Oracle Enterprise Gateway Server

1. Open *OEG_HOME*/system/conf/jvm.xml file in a text editor (*OEG_HOME* is the top-level directory in your Enterprise Gateway server installation).
2. Create a Java system property named com.amberpoint.util.Extension.dir and set its value to the location of the extension property directory.

   Set this property by adding a <SystemProperty> element as a child of the <JVMSettings> element, for example:

   For Windows systems:

   ```
   <SystemProperty name="com.amberpoint.util.Extension.dir" value="C:\btm\ext_
   props"/>
   ```

   For UNIX-like systems:

   ```
   <SystemProperty name="com.amberpoint.util.Extension.dir" value="/home/btm/ext_
   props"/>
   ```

4.3.2.2.3 Setting up the Pointer for a .NET Environment

1. Create a Windows environment variable named com.amberpoint.util.Extension.dir and set its value to the location of the extension property directory, for example:

```
set = C:\btm\ext_props
```

4.4 Setting up a Secure Socket (SSL) for Observation Messages

Observers send observation messages to their associated monitor using the Oracle proprietary Observation Protocol (OP) over a socket connection. You can secure your observation messages by setting up a secure socket (SSL) for this communication channel. OP over SSL is known as Observation Protocol Secure (OPS).

This security mechanism provides your choice of server-only or mutual authentication, message integrity, and encryption of observation messages during transmission. Server authentication (authentication of the monitor to the observer) is provided by SSL. Client authentication (authentication of the observer to the monitor) is provided by OPS.

Server authentication requires a properly configured key store that is accessible to the monitor and a properly configured trust store that is accessible to the observers. Business Transaction Management provides built-in, preconfigured key and trust stores so as to minimize the setup required to enable SSL (a preconfigured server certificate is also provided for .NET-based observers). For demonstration purposes, and perhaps for development environments, using these built-in security stores might be adequate. But, in production environments, you should tighten security by providing your own security stores. Note that these security stores are separate from the network-level security stores used for HTTPS.

> **Note:** The client authentication mechanism is entirely built into OPS. You do not need, under any circumstances, to provide security stores for client authentication.

The default Observer Communication policy configures the observation message channel to use OPS/SSL and requires mutual authentication (with the built-in security stores being used for the server authentication). If you intend to use the built-in security stores rather than your own, and you intend to use only Java-based observers, then you can simply accept these default settings. In this case, you do not have to perform any further configuration.

If you want to use your own security stores for server authentication, use the following procedure as a guide. This procedure describes how to create and deploy your own key store and trust store containing a self-signed certificate.

> **Note:** If you intend to use the built-in security stores, but intend to use them with .NET-based observers, then skip to step 4 and deploy the preconfigured certificate to the machines that will host the .NET observers. You can find the preconfigured certificate at nanoagent\config\ssl\server.cer in the ZIP files containing the .NET-based observers.

1. Prepare a key store and trust store containing an appropriate certificate and private key:

a. Locate the keytool application in your JDK.

b. Generate a key store containing a certificate-private key pair.

   For example, the following keytool command generates a key store containing a certificate-private key pair in a file named “mykeystore.ks”. The alias for the certificate-private key pair is “myks”, the common name is “MyMonitor”, the algorithm is RSA, the password for accessing the certificate and private key is “mycertandprivkeypass”, and the password for accessing the key store is “mykeystorepass” (this command creates the file if it does not yet exist):

   ```
   keytool -genkey -alias myks -dname "CN=MyMonitor" -keyalg RSA -keypass
   mycertandprivkeypass -storepass mykeystorepass -keystore mykeystore.ks
   ```

c. Export the certificate from your key store to an external file.

   For example, the following keytool command exports the certificate created in the previous example to a file named “mycertificate.cer”:

   ```
   keytool -export -alias myks -storepass mykeystorepass -file
   mycertificate.cer -keystore mykeystore.ks
   ```

d. Import the certificate (without the associated private key) into the trust store.

   For example, the following keytool command imports the certificate created in the previous example into the trust store contained in the file “mytruststore.ks” (this command creates the trust store file if it does not yet exist).

   ```
   keytool -import -v -trustcacerts -alias myks -file mycertificate.cer
   -keystore mytruststore.ks -keypass mycertandprivkeypass -storepass
   mykeystorepass
   ```

2. Deploy the key store to the machines that will host the monitors.

   Copy the key store (for example, mykeystore.ks) either to the machines that will host the monitors or to a location accessible to the monitors by way of HTTP GET.

3. Deploy the trust store to the machines that will host your Java-based observers (ignore this step if you don’t have Java-based observers).

   Perform this task in one of these two ways:

   - Copy the trust store (for example, mytruststore.ks) either to the machines that will host the observers or to a location accessible to the observers by way of HTTP GET.
   - Copy the trust store (for example, mytruststore.ks) either to the machines that will host the monitors or to a location accessible to the monitors by way of HTTP GET. By using this second method, you can configure the monitor to automatically dispatch the trust store to the observers by enabling the **Auto Dispatch Trust Store to Java Observers** field when you configure the Observer Communication policy (see Section 7.5).

4. Deploy the certificate to the machines that will host your .NET-based observers (ignore this step if you don’t have .NET-based observers).

   Using the Windows Certificate Import Wizard, import the certificate (for example, mycertificate.cer) to the Trusted Root Certification Authorities folder of the certificate store on the machines that will host the .NET-based observers.

Note: The SSL connection for observation messages will not be operational until you have configured the Observer Communication policy with information about your security stores that corresponds with the choices you have made in this section. You will need this information handy when you configure the Observer Communication policy. See step 4 in section 7.5, "Applying an Observer Communication Policy" for details about the information you will need on hand at that point.

5

Prerequisite Requirements and Preliminary Setup

This chapter describes prerequisite requirements and preliminary setup that you must satisfy before you begin installing Business Transaction Management. Business Transaction Management is composed of several types of components. Some of the following requirements pertain to all Business Transaction Management components while other requirements pertain to a subset of components.

5.1 Web Browser Requirements

The requirements in this section pertain to the web browser that you use to perform the initial configuration of Business Transaction Management and to access the Business Transaction Management Console.

- The web browser requires the Adobe Flash plugin version 10.1 or higher.
- If you are using Internet Explorer as your web browser, you must configure it to allow the Flash player's Active X control. Consult your Internet Explorer documentation for instructions on enabling this setting.

5.2 Access to WSDL and Schema Resources

In order to discover and monitor your business services, the Business Transaction Management observers require access to the WSDL and schema resources that describe the services. You must ensure that these resources are not protected in such a way (for example, by authentication) that the observers cannot access them.

For example, in an Oracle Service Bus environment, you must not undeploy **sbresouce.war** or apply security roles to it that will prevent the observer from accessing the WSDL information.

5.3 Setting up your WebLogic Environment

> **Note:** If you are using WebLogic Node Manager, you will use the WebLogic Administrative Console to adjust settings rather than edit scripts.

1. Ensure that the appropriate database driver for your Oracle RDBMS is in the WebLogic server's classpath for each central server and monitor.

Drivers are supplied in the jdbc directory of BTM_Servers_*.zip. Use ojdbc5.jar with JDK 1.5 and ojdbc6.jar with JDK 1.6.

2. Ensure that all WebLogic domains in which the central servers and monitors are installed include the Java Required Files (JRF) template.

   If any of these domains doesn't include the JRF template, extend the domain and add the template. You will get the following exception when you attempt to start the server if the JRF template is not included in the domain:

   ```
   java.lang.ClassNotFoundException:
   oracle.security.jps.wls.listeners.JpsApplicationLifecycleListener
   ```

   Note: The JRF template is part of the Oracle Application Development Framework (ADF) runtime, which means that you must install the ADF runtime into your WebLogic installation before you can extend any domain with the JRF template. When installing the ADF runtime, take care to install the release version that matches your version of WebLogic. You can download the ADF runtime at:

   ```
   http://www.oracle.com/technetwork/developer-tools/adf/downloads/index.html
   ```

3. Ensure that each WebLogic server in which you install a Business Transaction Management central server or monitor is uniquely identified so that the central servers and monitors can reliably connect to each other. You can perform this task in either of the following ways:

   - Ensure that any and all IP addresses assigned to the host machine uniquely identify that machine.
   - Ensure that the WebLogic server's **Listen Address** property is set to a hostname or IP address that uniquely identifies the server.

     To set this property, navigate in the WebLogic Administration Console to **Environment > Servers**, then click your server and display the **Configuration/General** tab.

     Note: If the machine has an IP address that is shared with another machine on the network, or the machine has multiple IP addresses that are treated as separate virtual machines, you must set the domain's **Listen Address** property as described above.

4. For central servers and monitors, ensure that memory allocation for your WebLogic server is set appropriately. Two methods of setting the Java options that control memory allocation are described below. Use the method that is appropriate for how you start and stop your managed servers:

   If you use the Node Manager to remotely start and stop your managed servers, set the Java memory options for your server using this method:

   a. Open the WebLogic Administration Console.
   b. Select your managed server.
   c. Select the **Configuration > Server Start** tab.

d. Enter the following Java options in the **Arguments** field, making sure to separate all entries in the field with a space:

```
-Xms256m -Xmx768m -XX:MaxPermSize=256m
```

These are the minimum recommended settings. Depending on your environment, you might have to set them higher.

If you start and stop your managed servers by executing local script files, set the Java memory options for your server using this method:

a. Open the setDomainEnv script file for your domain in a text editor.

On Windows systems, open setDomainEnv.cmd; on UNIX-like systems, open setDomainEnv.sh. These script files are located in the user_projects\domains*domain_name*\bin directory of your WebLogic installation.

b. Locate the following settings and ensure that they are set to at least the values indicated (depending on your environment, you might have to set them higher.):

```
MEM_ARGS=-Xms256m -Xmx768m

-XX:MaxPermSize=256m
```

There are several of these entries; set them all.

Depending on your version of WebLogic, you might also see separate 32-bit and 64-bit settings like these:

```
set WLS_MEM_ARGS_64BIT=-Xms256m -Xmx512m
set WLS_MEM_ARGS_32BIT=-Xms256m -Xmx512m
set MEM_MAX_PERM_SIZE_64BIT=-XX:MaxPermSize=256m
set MEM_MAX_PERM_SIZE_32BIT=-XX:MaxPermSize=128m
```

In this case, set them all to at least the minimum recommended settings.

5. Set up an administrative user on the WebLogic server in which you will install the Main server (btmMain.ear).

Business Transaction Management maps roles defined in WebLogic to its own application roles. See section 6.4.2, "Mapping WebLogic Users to Business Transaction Management Roles" for more information.

5.4 Setting up Business Transaction Management Databases

Several Business Transaction Management system services use a database to store persistent information and log messages. You must use an Oracle 10g or 11g RDBMS for these databases, and it must be configured to support SQL authentication mode and TCP/IP connections.

Before you configure Business Transaction Management, create the following database users (these are suggested names):

- sphereDB
- measurementDB
- transactionDB
- messageLogDB

You can create the database users in the same Oracle instance or in separate instances. But you must ensure that each Business Transaction Management component can

access its required databases. Refer to Table 4–1, " Business Transaction Management Communication Protocols" for a complete breakdown of the database access requirements for each Business Transaction Management component.

You must create the database users before starting configuration of Business Transaction Management. When you configure Business Transaction Management (see section 6.5, "Initial Configuration of Business Transaction Management"), the system will automatically create the appropriate database tables.

If you prefer to create the schemas manually for the first three of these databases (sphereDB, measurementDB, and transactionDB), your DBA can create them beforehand (see the following note). If you intend to let the system automatically create these tables and indexes, the database users must have create table, create index, create view, and analyze privileges. You cannot create the fourth schema (messageLogDB) beforehand because the system must be able to create and drop tables dynamically in response to changes in your monitored applications. For this database, the user must have create table, drop table, create index, create view, and analyze privileges. Note that it is not sufficient to assign the privileges to the roles associated with the user. You must explicitly assign the privileges to the user.

Note: Your DBA can manually create the tables and indexes for the sphereDB, measurementDB, and transactionDB databases using the `datastoreUtil` utility. This utility generates the appropriate schema definitions. Documentation on using this utility to generate the schema definitions is provided in Chapter 15, "The datastoreUtil Utility."

5.4.1 Setting up a Monitor Group Database

If you plan to deploy a monitor group, then you must also provide a database that the monitor group members can use for sharing information among themselves. Each monitor in a particular monitor group must be able to write to the same monitor group database using JDBC. Although the replicated instances of any one monitor group can be widely dispersed across your network, they must all have access to this same database.

For your monitor group database, you can either use the message log database schema (messageLogDB), or create a separate schema. If you create a separate schema, we suggest that you use monitorGroupDB as the database user name. Using separate schemas gives you the ability to relocate them to alternate drives or physical locations should a performance bottleneck develop.

If you like, you can manually create the monitor group database schema using the `datastoreUtil` utility (see Chapter 15, "The datastoreUtil Utility"). If you don't create the schema manually, then the system will automatically create the schema after you set up the monitor group.

For more information about monitors, including how to set them up, see Chapter 7, "Installing Monitors."

5.4.2 Estimating Database Resource Requirements

For help with sizing Business Transaction Management databases, refer to Doc ID 1487044.1 (Estimating Database Space Requirements for BTM Release 12c) at My Oracle Support (support.oracle.com). This document includes a spreadsheet to help administrators estimate the resources required by their Business Transaction Management databases. You should always use the most current version of this

document from My Oracle Support because it will be updated as we strive to improve its accuracy.

6

Installing and Configuring the Central Servers

This chapter describes how to install and perform the initial configuration of the Business Transaction Management central servers on Oracle WebLogic 10.3.2 through 10.3.5.

6.1 Overview of Installing and Configuring the Central Servers

1. *Optional* – Configure security for the central servers. You can also perform the security configuration for the monitors and observers at this time, if you wish (see Chapter 4, "Configuring Security").
2. Ensure that all prerequisite requirements and setup described in Chapter 5 are satisfied, including:
   - section 5.3, "Setting up your WebLogic Environment"
   - section 5.4, "Setting up Business Transaction Management Databases"
3. *Optional* – Configure the persistent storage directories (see Section 6.2).
4. Deploy the central servers (see Section 6.3).
5. Review the default mapping of users to Business Transaction Management application roles and make adjustments if necessary (see Section 6.4).
6. Perform the initial configuration of Business Transaction Management (see Section 6.5).
7. Configure the location of your Enterprise Manager server (see Section 6.6).

6.2 Configuring Persistent Storage Directories

At initial startup, Business Transaction Management creates a set of persistent storage directories to collect system output log entries and store user preferences for the system deployments. By default, the persistent storage directories are created within the application server's installation directory at *WL_install_dir*/user_projects/domains/*domain_name*/servers/*server_name*/btmstorage/*.

Your company's in-house procedures and rules for persistent storage might require you to place the persistent storage directories in a different location. In such a case, you can reconfigure the location of the persistent storage directories.

An installed Business Transaction Management system is composed of a set of deployments (EAR files), which are themselves composed of subdeployments (WAR files). Each subdeployment has an associated persistent storage directory of the same

name, minus the ".war". The following table lists the names of the deployments, subdeployments, and persistent storage directories.

Table 6–1 Business Transaction Management deployments, subdeployments, and persistent storage directories

Deployments (EARs)	Subdeployments (WARs)	Persistent storage directories
btmMain	btmui	btmui
	btmcentral	btmcentral
	btmcontainer	btmcontainer
btmPerformanceServer	btmcontainer	btmcontainer
	btmperformance	btmperformance
btmTransactionServer	btmcontainer	btmcontainer
	btmtransaction	btmtransaction
btmMonitor	btmmonitor	btmmonitor

6.2.1 Reconfiguring the Location of Persistent Storage Directories

1. Create the persistent storage directories in your file system, in the location that you want them.

   You must name the directories using the default names, as listed in Table 6–1. Leave the directories empty.

2. Locate the distribution archive for the Business Transaction Management central servers and unzip it into a directory.

   The distribution archive is named BTM_Servers*.zip, where * represents the version number. This archive also contains the Business Transaction Management monitor.

3. Modify the persistent storage directory location as specified in each subdeployment's web.xml file:

   a. Locate and expand the WAR file for the deployment whose storage directory location you want to change.

   b. Within the expanded WAR file, open the WEB-INF/web.xml file in a text or XML editor.

   c. Set the new location for the storage directory by editing the value of the storageDirectory parameter:

      As shown in the following example, the default value of the parameter is **AmberPointDefault**:

      ```
      <!-- PERSISTENT STORAGE DIRECTORY
      To set the persistent storage area to some value, change the value of
      param-value to some EXISTING directory where you want things stored.
      -->

      <context-param>
      <param-name>com.amberpoint.storageDirectory</param-name>
      <param-value>AmberPointDefault</param-value>
      </context-param>
      ```

      Delete the value **AmberPointDefault** and replace it with the path to the storage directory you created, as shown in the following examples:

On Windows systems – If you want the persistent storage directory for btmcentral to be C:\btm_data\btmcentral, change the default entry within your btmcentral web.xml file to the following:

```
<context-param>
  <param-name>com.amberpoint.storageDirectory</param-name>
  <param-value>C:\btm_data\btmcentral</param-value>
</context-param>
```

On Unix-like systems – If you want the persistent storage directory for btmcentral to be opt/webserviceapplogs/btm_data/btmcentral, change the default entry within your btmcentral web.xml file to the following:

```
<context-param>
  <param-name>com.amberpoint.storageDirectory</param-name>
  <param-value>/opt/webserviceapplogs/btm_data/btmcentral</param-value>
</context-param>
```

d. Repeat this step for each persistent storage directory whose location you want to reconfigure.

Note: If you are going to reconfigure the location of persistent storage directories for your monitors (as well as for your central servers), this might be a convenient time to do that.

---

4. Repackage your WAR and EAR files.

You should document the location of your persistent storage directories because when you upgrade or reinstall the central servers, you will need to once again define the location of your persistent storage directories for these deployments.

6.3 Deploying the Central Servers

Note: These instructions assume that you are installing into managed instances of WebLogic Server.

---

1. Locate the distribution archive for the Business Transaction Management central servers and unzip it into a directory (henceforth referred to as *Install_Dir*).

The distribution archive is named BTM_Servers*.zip, where * represents the version number. This archive also contains the Business Transaction Management monitor.

Note: If you configured the persistent storage directories, as described in Section 6.2, you will have already completed this step.

---

2. *Optional security step for UNIX-like operating systems* – If you want to set file permissions on the files that make up the distribution to the most restrictive level that still maintains functionality, complete this step:

a. Locate setPermissions.sh at the top level of *Install_Dir*.

This script contains commands for setting file permissions of all regular files to Owner – read/delete; all directories to Owner – read/execute/delete; and all scripts to Owner – read/execute/delete.

Note: These permission levels are extremely restrictive. For example, only the owner can read the files.

---

b. On a command line, at the top level of *Install_Dir*, run this command:

```
source setPermissions.sh
```

This command runs the commands in the script file and sets permissions for all files and directories in the expanded archive.

3. Locate the central server application EAR files inside of *Install_Dir*\archives:
   - btmMain.ear
   - btmPerformanceServer.ear
   - btmTransactionServer.ear
4. Using the WebLogic Server Administration Console, deploy each EAR file once.

   For performance considerations you should deploy each to a separate WebLogic Server instance. You must not deploy any of the central servers to a WebLogic Server instance that hosts services or components you intend to monitor. For more information about the central server applications, see Chapter 1.1, "Architecture."
5. Start all managed servers where you deployed Business Transaction Management components.
6. Start the deployments.
7. If you reconfigured the locations of your persistent storage directories (as described in Section 6.2), confirm that system output log entries have been written in the new locations.
8. If this is a reinstallation of the central servers, notify all Business Transaction Management users to flush their web browser caches.

   The Management Console contains a number of Adobe Flash widgets. Web browsers normally cache these widgets and will continue to use the older cached widgets until you either flush the cache or restart your web browser.

6.4 Mapping Users to Roles

This section describes Business Transaction Management application roles and the default mappings of Business Transaction Management users to these roles. If necessary, you can reconfigure these mappings using your system administration facilities.

Business Transaction Management applications (the central servers and monitors) rely on the application server in which they are deployed for the authentication of users and the association of application roles with users.

By default, authentication is enabled for the Management Console. If you want to disable authentication, use whatever tool or procedure is appropriate for the application server you are using. If you disable authentication, users of the Management Console must still log in. However, they can log in using any user name and are not required to provide a password. Note that all UI personalizations, such as

edits to the Navigator, filters, and column sets are stored as preferences and associated with the user name.

6.4.1 Business Transaction Management Application Roles

The Business Transaction Management Console uses roles to authorize access to various features of the user interface. In order to log into the Management Console, you must use credentials that are mapped to at least one of theses BTM application user roles: btmAdmin, btmUser, or btmObserver. In order to perform the initial configuration of Business Transaction Management, explained in Section 6.5, you must log in as a user with the btmAdmin role.

6.4.1.1 Primary Roles

Each Business Transaction Management user must be assigned at least one primary role. The primary roles are:

btmAdmin – Users with this role are granted all privileges. These users can use all tools and facilities provided by the Management Console, including the ability to view and create sensitive properties and to view all message content.

btmUser – Users with this role have most privileges needed to configure basic monitoring. For example, they can configure monitors; create, edit, and delete policies (does not include system policies); register services; set registry attributes on services and endpoints; and create and edit transactions and conditions. They also have all the privileges of btmObserver. This role does not grant the privilege to modify the Business Transaction Management environment, access message content, or view or edit sensitive properties.

btmObserver – Users with this role have privileges to use most of the basic monitoring facilities. They can view summary, dependency, and administrative information about the monitoring system, but are not allowed to configure any of the policies or settings related to it. They can also view transactions and conditions, but are not allowed to create or edit them. This role does not allow users to modify the Business Transaction Management environment, access message content, or view or edit sensitive properties.

Note: All navigation and views in the Management Console are available to all primary roles. However, some roles cannot access certain menus and menu items and the tools associated with them.

6.4.1.2 Auxiliary Role

In addition to the primary roles, Business Transaction Management defines an auxiliary role. The auxiliary role provides additional privileges that you might want to assign certain users. For example, you might want to let a user access message content but not want to give that user full administrative privileges. You could do this by assigning the user a primary role of btmUser and an auxiliary role of btmInspector. The auxiliary role is:

btmInspector – Users with this role can view message content and view and create properties, including sensitive properties.

Note: The btmAdmin role has all of the privileges of btmInspector.

6.4.2 Mapping WebLogic Users to Business Transaction Management Roles

In a WebLogic server, the role of Business Transaction Management administrator (btmAdmin) is automatically mapped to the Administrators group defined in your WebLogic server. The role of Business Transaction Management user (btmUser) is mapped to the groups Operators and Monitors. The role of Business Transaction Management observer (btmObserver) is mapped to the group Everyone, granting all authenticated users observer privileges. The following table lists the Business Transaction Management application roles and their default mappings to application server groups:

Table 6–2 Role Mapping in WebLogic

Business Transaction Management Application Role	Application Server Group
btmAdmin	Administrators
btmUser	Operators, Monitors
btmObserver	Everyone
btmInspector (Note that this is singular.)	btmInspectors (Note that this is plural.)

> **Note:** By default, the role btmInspector is mapped to a group named btmInspectors. Because this group does not exist by default, the application server administrator must create the group and assign it to the appropriate users.

You can modify these default mappings in the WebLogic deployment descriptor file (weblogic.xml). If you change the mappings, you should change them for all Business Transaction Management applications. The names of the applications (EAR files) begin with the string “btm”. Refer to your WebLogic documentation for further information about editing the default mappings.

6.5 Initial Configuration of Business Transaction Management

Before you can access the facilities of Business Transaction Management, you must perform an initial configuration of the central servers. If this is the first time you have installed Business Transaction Management, you should use the browser-based Configuration wizard, as described below. However, for subsequent installations, you might want to use the Command Line Interface (CLI) to perform a scripted configuration of Business Transaction Management. For more information, see "Scripted Configuration of Oracle Business Transaction Management" on page 14-1.

1. Ensure that the central servers are running.
2. Open the Management Console by pointing a web browser at the server that hosts the btmMain deployment.

   Use a URL in the form of:

   `http://host_name:port_number/btmui`

   The Management Console login page opens.
3. Log in as a user that is in the btmAdmin role (see section 6.4, "Mapping Users to Roles" for information about this role).
4. The introductory page of the Configuration wizard opens.

Click **Next**, and the **Database Type** page opens.

5. Choose **External Database** (the embedded database is not supported for production systems).

   When finished, click **Next**, and the **External Database Configuration** page opens.

6. Provide the connection string to your Oracle instance and the user names and passwords for the database users you created in section 5.4, "Setting up Business Transaction Management Databases".

   If you created the users on separate Oracle instances, first select **Custom** so that you can provide multiple connection strings. If you are using a single connection string for all databases, each database requires a distinct user name.

   When finished, click **Next**, and the **Sphere URL** page opens.

7. Ensure that the URL for the sphere is correct—in most cases the default should be correct (unless you are running in an HTTPS environment).

   > **Note:** If you click the Test Sphere URL link and the sphere URL is correct, the system displays a Sphere Status page that indicates that the sphere is not initialized. This is expected and the sphere service will be initialized at the completion of the configuration. If the URL is incorrect, the browser displays a "page not found" error.

   When finished, click **Next**, and the **Local Container Setup** page opens.

8. This page lets you optionally specify DNS aliases for the network node on which you installed btmMain.

   The use of aliases helps avoid the creation of duplicate endpoints when users register services manually or when the system observes message traffic at an alias address. Use a comma to separate multiple addresses.

   When finished, click **Next**, and the **Performance Monitoring Components** page opens.

9. Enter the URL at which you deployed btmPerformanceServer in the form of http://*HostName:Port*/btmcontainer/container/.

   This URL must end with "/btmcontainer/container/". Make sure that you deploy apPerformanceServer before you exit this screen.

   When finished, click **Next**, and the **Local Container Setup** page opens again (unless you deployed btmPerformanceServer to the same machine as btmMain).

10. This page lets you optionally specify DNS aliases for the network node on which you installed btmPerformanceServer.

    When finished, click **Next**, and the **Transaction Monitoring Components** page opens.

11. Enter the URL at which you deployed btmTransactionServer in the form http://*HostName:Port*/btmcontainer/container/.

    This URL must end with "/btmcontainer/container/". Make sure that you deploy btmTransactionServer before you exit this screen.

    When finished, click **Next**, and the **Local Container Setup** page opens again (unless you deployed btmTransactionServer to the same machine as btmPerformanceServer or btmMain).

12. This page lets you specify DNS aliases for the network node on which you installed btmTransactionServer.

    When finished, click **Next**, and the **Summary of Configuration** page opens.

13. Ensure that the configuration information is correct and click **Finish**.

    Business Transaction Management validates the information you supplied during configuration, and if all the information is valid, configuration completes successfully and the Business Transaction Management Console appears.

    If any configuration information cannot be validated, Business Transaction Management displays an error message as well as a check box with the following text: "Ignore these errors and proceed with a potentially incompletely configured system." In general you should attempt to correct any configuration errors. However, you have the option of proceeding with Business Transaction Management partially configured by enabling the checkbox and clicking **Finish**. If you choose to access the partially configured system, you will need to correct the configuration errors before you can successfully use the product. For example, if your database connection information was not accurate, you will need to re-configure it from within the Management Console. For information about configuring databases after initial configuration, refer to the Business Transaction Management online help.

6.6 Configuring the Connection to Enterprise Manager

If you have Oracle Enterprise Manager installed, perform the configuration procedure described in this section so that Business Transaction Management can connect to your Enterprise Manager installation. During this procedure you will:

- Provide the location of your Enterprise Manager server.

  This information enables Business Transaction Management to locate your Java Virtual Machine Diagnostics server, which in turn enables you to drill down from various places in the Business Transaction Management UI into the Java Virtual Machine Diagnostics UI.

  If you don't provide this information or provide the wrong information, you will receive an error message when you attempt to drill down into the Java Virtual Machine Diagnostics UI.

  Even though you might not have Java Virtual Machine Diagnostics currently installed, you should perform this step of the procedure in case you decide to install it later on.

- Provide the connection string and user credentials for accessing the Enterprise Manager repository.

  This information enables Business Transaction Management to send transaction-related SLA events to Enterprise Manager, where they are displayed as events or incidents. Note that you must also associate the transaction with a business application target in Enterprise Manager for the events to be displayed. For information about associating transactions with business application targets, refer to the Business Transaction Management online help after you finish installing the product.

1. Open the Business Transaction Management console.
2. In the Navigator, select **Administration > System Services**.
3. In the main area, select **AP_Sphere_Service**.

4. From the menu bar, choose **Admin > Edit Setup Data for AP_Sphere_Service**.
5. Click the **Edit XML** button.

   The sphere services setup XML file is displayed.
6. Scroll to the bottom of the file and locate the **<pfx6:emgcURL/>** element.
7. Add the base URL of your Enterprise Manager server as the value of this element, for example:

   ```
   <pfx6:emgcURL>https://myEMhost:5416/</pfx6:emgcURL>
   ```

   Note: The namespace prefix might be other than pfx6; use whatever value appears in the XML text.
8. Directly following the **<pfx6:emgcURL/>** element, locate the **<pfx6:SphereSetupDataVersion/>** element.
9. Replace the **<pfx6:SphereSetupDataVersion/>** element with the following code:

   ```
   <pfx6:SphereSetupDataVersion>
     <pfx6:emgcRepos>
     <pfx1:User>myUserName</pfx1:User>
     <pfx1:Password>myPassword</pfx1:Password>
     <pfx1:Driver>oracle.jdbc.OracleDriver</pfx1:Driver>
     <pfx1:Connection>myDatabaseConnectionString</pfx1:Connection>
     <pfx1:DatabaseConnectionVersion/>
     </pfx6:emgcRepos>
     <pfx6:SphereSetupDataVersion/>
   </pfx6:SphereSetupDataVersion>
   ```

   In this code, replace:

   - *myUserName* and *myPassword* with the username and password of an account that has privilege to access your Enterprise Manager repository.
   - *myDatabaseConnectionString* with the connection string to your Enterprise Manager Repository, for example:

     ```
     jdbc:oracle:thin:@myhost.mydomain.com:15044:mySID
     ```
10. Click the **Apply** button.

7

Installing Monitors

This chapter describes how to install Business Transaction Management monitors. Instructions are provided for installing both singleton monitors and monitor groups.

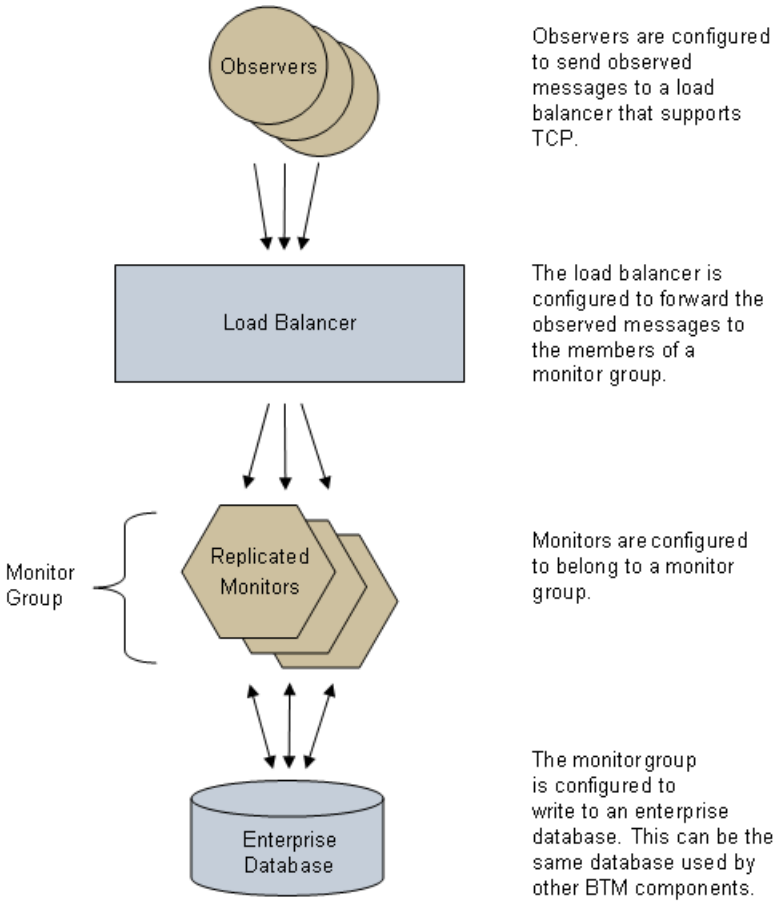
A *singleton monitor* is a single instance of a monitor that receives observations directly from an observer. A *monitor group* is a group of replicated monitor instances situated behind a load balancer. In the case of a monitor group, the observer sends observations to the load balancer, which then distributes the observations to the monitors in the group.

A monitor can receive observations from an unlimited number of observers. Theoretically, you could deploy a single monitor to process observations from all the observers in your system. However, if the rate and/or size of observations produced by your system is large, a single monitor might be overwhelmed. Such a situation could result in reduced performance of your monitoring system and possible loss of monitoring data.

You can eliminate such a potential bottleneck by deploying multiple monitors. You can deploy as many monitors as required for your monitoring needs. You can deploy the monitors as singleton monitors, as monitor groups, or as a mix of the two. You could, for example, deploy a dedicated monitor for each observer. Note that an observer can send observations to only one singleton monitor or one monitor group. If you have an observer that produces observations faster than a singleton monitor can process them, you will need to deploy a monitor group for that observer.

Using monitor groups not only gives you the ability to scale your monitoring system, but makes it more fault tolerant. Monitor groups require a third-party load balancer that supports TCP for routing messages from your observers to your monitors. Monitor groups also require access to an Oracle 10g or 11g database, which is used to share data between the replicated instances of the monitor. This database can be the same database used by other Business Transaction Management components. Although the replicated instances of any one monitor group can be widely dispersed across your network, they must all have access to this same database (for more information about the monitor group database, see Section 5.4.1, "Setting up a Monitor Group Database."). A generalized deployment of a monitor group is shown below.

Figure 7–1 Generalized deployment of a monitor group



7.1 Overview of Installing Monitors

This section provides an overview of the tasks you must perform to install monitors. Subsequent sections provide details on how to perform these tasks. To ensure a properly configured system, you must perform the steps in the order shown.

1. Configure security if needed (see Chapter 4, "Configuring Security").

> **Note:** If you configured the assertion secret and encryption key on your central servers, then you must also configure your monitors to use the same assertion secret and encryption key. If, on the other hand, you are using the default security configuration on your central servers, then you must use the default security configuration on your monitors (see Section 4.3, "Configuring the Assertion Secret and Encryption Key").

2. Ensure that all prerequisite requirements and setup described in Chapter 5 are satisfied, including:
   - Section 5.3, "Setting up your WebLogic Environment"
   - Section 5.4, "Setting up Business Transaction Management Databases"

3. *Optional* – Configure the persistent storage directories for the monitors. Use the same procedure as described for the central servers in Section 6.2.
4. Deploy and register the monitors (see Section 7.2).
   a. Ensure that the central servers are installed and running.
   b. Deploy monitors to the target application servers using your application server's deployment tools.
   c. Register your monitors using the registerMonitor CLI command.
5. *For replicated monitors only* – Set up a monitor group (see Section 7.3).
   a. Create a monitor group.
   b. Assign monitors to the monitor group.
6. *For replicated monitors only* – Configure your load balancer (see Section 7.4).
7. Apply an Observer Communication Policy (see Section 7.5).

7.2 Deploying and Registering Monitors

This section pertains to both singleton and replicated monitors. You must perform this task for all installation scenarios.

1. Ensure that the Business Transaction Management central services are installed and running.
2. Using your application server's deployment tools, install the btmMonitor.ear deployment package on each application server that will host a monitor.

> **Notes:** Do not deploy more than one monitor per application server. Also, do not deploy the monitor into an application server that is hosting any of the Business Transaction Management central services.
>
> (An application server is sometimes referred to as a container in the Business Transaction Management Console and elsewhere in the documentation.)

3. Start the monitor deployments.
4. Use the registerMonitor CLI command to register each of the monitors that you deployed, for example:

```
btmcli registerMonitor -e http://my_monitor_host:8080/btmmonitor/agent/agent
                       -fn My_BTM_Monitor
                       -s http://localhost:8080/btmcentral/sphere
                       -l my_admin_username:my_admin_password
```

Table 7–1 Flags for the registerMonitor CLI command

Flag	Description
-e | -endpointUrl	Required. Specify the URL of the monitor to register, for example: `http://my_monitor_host:8080/btmmonitor/agent/agent` Replace the host name and port number with appropriate values. This URL always ends with **btmmonitor/agent/agent**.

Table 7–1 (Cont.) Flags for the registerMonitor CLI command

Flag	Description
-fn | -friendlyName	Optional. Specify a friendly name for the endpoint of the monitor.
-s | -sphereUrl	Required unless the AP_SPHERE_URL environment variable is set. Specify the URL of the sphere, for example: `http://mySphereHost:8080/btmcentral/sphere/` Replace the host name and port number with appropriate values. This URL always ends with **btmcentral/sphere/**.
-l | -userLogin	Required unless the AP_USER_LOGIN environment variable is set. Specify the credentials of a user belonging to the **btmadmin** role in the format: *username:password*. You can encrypt passwords using the encryptPassword CLI command, for example: `btmcli encryptPassword -password "myPassword"`

See Chapter 14.2, "Invoking the CLI" for information about how to invoke a CLI command. Refer to the Business Transaction Management online help for other information about the CLI.

5. Verify that all monitors and system services are running properly.

7.3 Setting Up a Monitor Group

This section pertains to replicated monitors, only. You must perform this task if you plan to use replicated monitors.

1. Create a monitor group:
   a. In the Business Transaction Management Console, choose **Admin > Create System Policy > Monitor Agent Group**.
   b. Enter a name for your monitor group (you can optionally provide descriptive information in the **Version** and **Notes** fields).
   c. Specify the connection string for your database.

      If you use the default string, replace the text within the curly braces (and the curly braces themselves) with values appropriate for your database. Each member of the monitor group must have access to this database. For more information about the monitor group database, see Section 5.4.1, "Setting up a Monitor Group Database."
   d. Specify a user name and password for accessing your database.
   e. Click **Apply**.
2. Assign monitors to the monitor group you just created:

Note: If your monitor will not be using the default Observer Communication policy, you must complete Section 7.5, "Applying an Observer Communication Policy" before continuing with this section. One scenario in which this would be necessary is if multiple monitors are running on the same machine, in which case they would have to listen on different socket ports and would therefore require separate Observer Communication policies.

a. In the Business Transaction Management Console's Navigator, choose **Administration > Monitors**.

b. In the main area, select the monitor that you want to assign to a group.

c. Choose **Modify > Edit Profile for Monitor Agent**.

d. Type the name of the monitor group in the **Monitor Group** field.

   This name must match the value of the **Name** field in the monitor group policy you used to register the monitor group.

e. Click **Apply**.

7.4 Configuring Your Load Balancer

This section pertains to replicated monitors, only. You must perform this task if you plan to use replicated monitors.

1. Configure your load balancer to communicate with the observers by defining both an HTTP virtual server and a socket virtual server.

   Your load balancer requires two observer communication channels—one referred to as the *HTTP virtual server*, and the other referred to as the *socket virtual server*:

   - The HTTP virtual server is used for administrative purposes. For example, observers retrieve their configurations from any one of the replicated monitors by way of the HTTP virtual server.
   - The observers transmit observational data to the socket virtual server, and the load balancer distributes this data across the replicated monitors. We refer to the messages containing such data as observation messages.

   Note: The replicated monitors communicate with the Business Transaction Management central servers and database directly—not by way of the load balancer.

2. Take note of the IP address and port of your socket virtual server.

   You will need this information when you configure the Observer Communication policy (see Section 7.5).

3. Configure your load balancer to distribute observation messages across the replicated monitors as follows:

   a. Create a pool for the socket virtual server.

   b. Assign each monitor to the pool by assigning the port on which the monitor will receive observations.

4. Take note of the monitor port number(s) that you assign to the socket virtual server's pool.

   You will need this information when you configure the Observer Communication policy (see Section 7.5).

   Note: Configuring the Observer Communication policy will be easier if all the monitors listen on the same port number.

5. Configure your load balancer to distribute administrative messages across the replicated monitors as follows:

   a. Create a pool for the HTTP virtual server.

   b. Assign each monitor to the pool by assigning the port on which the monitor's application server listens.

6. Ensure that the application servers that host the observers have their AP_NANO_CONFIG_URL Java system property or AmberPoint:NanoConfigUrl Windows key (depending on the platform) set to the URL of the monitor by way of the load balancer's HTTP virtual server.

   For example, if the HTTP virtual server's IP address is 10.147.46.152, and its port number is 5072, then the URL of the monitor by way of the HTTP virtual server would be:

   ```
   http://10.147.46.152:5072/btmmonitor/agent/agent/
   ```

   For more information on this topic, see Chapter 8, "Installing Observers Overview".

7.5 Applying an Observer Communication Policy

This section pertains to both singleton and replicated monitors. You must ensure that a correctly configured Observer Communication policy is applied to all of your monitors. The following subsections explain how to configure and apply this policy:

- Section 7.5.1, "About the Observer Communication Policy"
- Section 7.5.2, "Procedure for Applying an Observer Communication Policy"
- Section 7.5.3, "How Many Observer Communication Policies Do I Need to Apply?"
- Section 7.5.4, "Preconfigured Observer Communication Policies"
- Section 7.5.5, "Targeting Observers Reference"

7.5.1 About the Observer Communication Policy

The Observer Communication policy sets up communication between observers and a monitor or monitor group. By default, this policy configures both monitors and observers by:

- setting up the observer-to-monitor communication channel (including security settings)
- configuring the runtime settings for the observer (including selection of the types of components discovered and monitored)

Prior to release 12.1.0.4, you could apply only a single Observer Communication policy to any one monitor, which meant that all observers associated with the same

monitor would receive the same configuration. To provide greater flexibility in configuring observers, release 12.1.0.4 introduces the ability for you to apply multiple Observer Communication policies to a single monitor and to then *target* each of these policies at a different observer or set of observers.

In such a scenario, you designate one policy as the source for generating the monitor configuration (we will refer to this as the "monitor" policy). You then designate each remaining policy as the source for generating an observer configuration (we will refer to these as "observer-only" policies).

Note that the monitor policy generates an observer configuration in addition to the monitor configuration. You can either target this configuration at a set of observers, just as you would with an observer-only policy, or leave it untargeted, in which case it serves as the default configuration for observers that are not targeted by a configuration.

As in previous releases, you can use a single policy for configuring monitors and all of their associated observers, if you desire (in other words, you don't have to target specific observers). In such a scenario, you would use a "monitor policy" that is not targeted at specific observers. This policy will generate a default observer policy, and, since no specific observers are targeted, all observers associated with the monitors will receive this default configuration.

7.5.2 Procedure for Applying an Observer Communication Policy

The following procedure describes at a high level how to apply an Observer Communication policy. This procedure is followed by subsections that detail how to configure the monitor-related and observer-related fields of the policy.

1. Open the Observer Communication policy you will use for configuring your monitor (this is your "monitor" policy).

   You can either edit one of the preconfigured policies or create a new policy (see Section 7.5.4, "Preconfigured Observer Communication Policies" for descriptions of the preconfigured policies):

   - To edit one of the preconfigured Observer Communication policies:

     Select **Administration > System Policies** in the Navigator.

     Select the preconfigured policy you want to edit in the main area.

     Choose **Modify > Edit Definition for *My_Policy***, where *My_Policy* is the name of the policy.

   - To create a new Observer Communication policy:

     Choose **Admin > Create System Policy > Observer Communication**.

     Note: By default, the preconfigured policy named **Observer Communication Policy - Default** both generates a monitor configuration and applies it to every monitor in the system. You should use this default policy for configuring your monitors if possible. If you use a different policy for configuring your monitors, then you must first edit the default policy and ensure that it either does not generate a monitor configuration or is not applied to the monitors that you want to configure (or both). You are allowed to target a monitor with only one monitor configuration.

2. Designate this policy as the source for generating the monitor configuration by ensuring that the **Generate Configuration for Observers Only** checkbox is disabled.
3. Configure the policy fields that pertain to the monitor.

   Refer to Section 7.5.2.1, "Configuring Monitor-Related Fields in the Observer Communication Policy" for detailed instructions.
4. Decide whether you want to use the observer configuration generated by your monitor policy as a default observer configuration or whether you want to target it at specific observers.

   Note: By default, the policy named **Observer Communication Policy - Default** generates an untargeted observer configuration and is applied it to all monitors in the system. This means that by default this observer configuration serves as the default configuration for all observers in the system. If you want a default configuration for your observers, then you could simply leave this policy untargeted in regards to observers. If, on the other hand, you don't want a default observer configuration, then you must edit this policy by targeting it at some number of observers.
5. Configure the policy fields that pertain to the observer.

   Refer to Section 7.5.2.2, "Configuring Observer-Related Fields in the Observer Communication Policy" for detailed instructions.
6. *Optional* - Target additional observers:
   a. Open one of the preconfigured policies or create a new policy for use as an "observer-only" policy.
   b. Designate this policy as the source for generating only an observer configuration by ensuring that the **Generate Configuration for Observers Only** checkbox is enabled.
   c. Ensure that the values in the monitor-related fields match the corresponding fields in your monitor policy.
   d. Configure the policy fields that pertain to the observer.

   Refer to Section 7.5.2.2, "Configuring Observer-Related Fields in the Observer Communication Policy" for detailed instructions.

7.5.2.1 Configuring Monitor-Related Fields in the Observer Communication Policy

The following procedure describes how to configure the monitor-related fields of the Observer Communication policy. This procedure assumes that you have already opened the policy you want to edit as described in Section 7.5.2, "Procedure for Applying an Observer Communication Policy."

1. The way you perform this step depends on whether your monitors are singleton or replicated.

   For singleton monitors:
   a. Set the **Communication path** field to: **Direct to monitor** (this is the default setting).

b. Specify the port number on which your monitors listen in the **Monitor port number** field.

This is the port to which the observers send observations and at which the monitors receive observations. This setting configures both the monitors and observers.

Note: You do not have to specify the host name for the monitor. The host name is obtained from the AP_NANO_CONFIG_URL Java system property or AmberPoint:NanoConfigUrl Windows key. You will set this system property/key on the application servers on which you install observers (see Section 8.2).

For replicated monitors:

a. Set the **Communication path** field to: **Through router to monitor group**

b. Specify the IP address and port number of your load balancer's socket virtual server in the **Router IP address** and **Router port number** fields.

These settings configure your observers to send their observations to your load balancer's incoming address (that is, to the load balancer's socket virtual server that you configured in Section 7.4).

c. Specify the port number on which your monitors will receive observations in the **Monitor port number** field.

This setting configures the monitors to listen on the specified port. This port number should coincide with the monitor port number that you assigned in the pool of your load balancer's socket virtual server in Section 7.4.

Note: If you later add or delete a monitor to or from a monitor group, you must reconfigure your load balancer accordingly.

---

2. If you want to use a secure connection for transporting observation messages from observers to monitors, leave the **Enable SSL** checkbox enabled and continue to the next step (SSL is enabled by default).

If, on the other hand, you prefer to use a nonsecured connection, disable the checkbox and then skip to step 6.

Additional fields are displayed if the **Enable SSL** checkbox is enabled.

3. If SSL is enabled, specify the protocol for the connection in the **Protocol** field.

Choose **TLSv1** to use TLS 1.0.

Choose **SSLv3** to use SSL 3.0 (**SSLv3** is not supported by the .NET-based observers.).

Choose **Any** to let the components decide on the best protocol at runtime. This is the default setting.

4. By default, built-in, preconfigured security stores are used when SSL is enabled. If you want to use these built-in security stores, proceed to substep d.

If you would rather use your own security stores, disable the **Use Default Stores** checkbox and complete all of the following substeps (in this case, additional fields are displayed):

a. Specify the information for the Monitor's key store as follows:

   Key Store Location – The location of the key store. You can specify this location as either an absolute path, if the key store file is local to your monitor, or as an HTTP(S) URL, if the file is accessible by HTTP GET.

   Key Store Password – The password for accessing the key store.

   Key Store Type – The type of the key store, for example, JKS, JCEKS, or PKCS12.

   Key Name – The certificate and private key. You can specify a key alias or a certificate attribute of the form CN=*value*, UID=*value*, etc.

   Key Password – The password for accessing the certificate and private key. If unspecified, the password for the key store is used.

b. If you want the monitor to automatically dispatch the trust store to your Java-based observers, enable the **Auto Dispatch Trust Store to Java Observers** checkbox.

c. Specify the information for the trust store as follows:

   Trust Store Location – The location of the trust store. You can specify this location as either an absolute path, if the key store file is local to your observer (or local to your monitor, if you are using auto dispatch), or as an HTTP(S) URL, if the file is accessible by HTTP GET.

   Trust Store Password – The password for accessing the trust store.

   Trust Store Type – The type of the trust store, for example, JKS, JCEKS, or PKCS12.

d. If you are using .NET-based observers, ensure that you have deployed the appropriate certificate to the machines hosting those observers as described in Section 4.4, "Setting up a Secure Socket (SSL) for Observation Messages."

5. Enabling SSL requires the monitor to authenticate itself to the observer. Whether the observer authenticates itself to the monitor, however, depends on the setting of the **Observer Authentication** field. Adjust the setting of this field to one of the following values:

   Use Connection Authentication – This is the default value and requires the observer to authenticate itself to the monitor each time it establishes a connection.

   None – This value turns observer authentication off.

   Use Message Authentication – This value requires the observer to authenticate itself each time it sends a message to the monitor. Note that the use of message authentication can significantly degrade performance. You should use this setting only when necessary. If your observer sends its messages to a monitor group whose load balancer is configured for per-message balancing (rather than per-connection), then you cannot use connection authentication. In this case, you must set this field to either **None** or **Use Message Authentication**.

6. In the **Criteria** section, ensure that the policy is applied to the correct monitors.

   You can apply the policy to all monitors in a particular monitor group by adding a **Monitor Group** clause, for example:

Note: You can apply only one Observer Communication policy that generates a monitor configuration to any one monitor. You must ensure that the **Criteria** section of all other Observer Communication policies that generate a monitor configuration either do not include, or explicitly exclude the monitors to which you apply this policy.

7. Click **Apply**.

For information about advanced settings, refer to the Business Transaction Management online help.

7.5.2.2 Configuring Observer-Related Fields in the Observer Communication Policy

The following procedure describes how to configure the observer-related fields of the Observer Communication policy. This procedure assumes that you have already opened the policy you want to edit as described in Section 7.5.2, "Procedure for Applying an Observer Communication Policy."

1. In the **Active Probes** section, choose the types of business components you want to discover and monitor by activating or deactivating the appropriate probes.

   Notes: A probe is the component within an observer that is responsible for discovering and monitoring a particular type of business component. Most types of observers contain multiple probes. For more information about probes, see Section 1.1, "Architecture".

   The JAVA probe monitors local Java calls, which in most cases is not needed and can be distracting because of the typically large number of local Java calls that occur. In order to use the JAVA probe, you must first deploy and configure it. For information about deploying and configuring the JAVA probe, enter a service request at My Oracle Support (support.oracle.com).

   In most situations, you should deactivate the RMI probe if it is not already deactivated. Most applications utilize RMI by way of higher level APIs, such as JAX-RPC, JAX-WS, EJB, and JMS. In such cases, it is better to activate only the probes for these higher-level components. However, if your application makes RMI calls directly you might want to activate the RMI probe.

   There is no need to explicitly deactivate probes that are not installed—neither for the sake of performance nor for any other reason. Uninstalled probes are inherently not activated (a probe is installed if an observer that contains that probe is installed). The only reason to deactivate a probe is if: (1) the probe is installed, AND (2) you do not want to monitor the type of business component the probe monitors. Furthermore, if you deactivate (or activate) any of the SOA probes, you must deactivate (or activate) all of the SOA probes as a group.

   Enable/disable the **Enable Discovery** checkbox to activate/deactivate the discovery mechanism for the associated component type. When discovery is activated for a component type, the components are discovered and displayed in the Management Console the next time they receive a message or call.

   Enable the **Monitor Upon Discovery** checkbox for a component type if you want to immediately begin monitoring those components as they are discovered. Disable the checkbox if you don't want to monitor components of that type.

> **Note:** If you enable discovery but not monitoring and then later edit the policy and enable monitoring, the system will not begin monitoring previously discovered components. The system will begin monitoring only the components discovered after you enable monitoring. For information on enabling monitoring for previously discovered components, see the topic "Start and Stop Monitoring of Endpoints" in the online help.

2. If you are using this policy to configure observers only, complete the following substeps:
   a. Set all the fields in the **Communication Channel** section (except for the **Generate Observer Configuration Only** checkbox, but including the fields in the **SSL Configuration** section) so that they match the policy you use to configure your monitor.
   b. Enable the **Generate Observer Configuration Only** checkbox.
3. If you want only specific observers to receive the configuration generated by this policy, target those observers by completing the following substeps:
   a. Use the **Configuration Label** and/or **Observer Base Address** fields to specify which observers you want to target (for more information about these and related fields, refer to Section 7.5.5.1, "Observer Configuration Labels" and Section 7.5.5.4, "Field Reference for Targeting Observers").
   b. Ensure that labels and addresses specified in one "observer policy" are not specified in any other policy that is applied to the same monitor (for additional information, see Section 7.5.5.2, "Rejection of Observer Communication Policies").
   c. *Optional* – Enable the **Validate Addresses** checkbox to ensure that all targeted observers are known to Business Transaction Management.

      If you target an unknown observer and enable this field, the policy will be rejected. If you want to target an observer that is not currently known but will be later, you should disable this field.
   d. *Optional* – (Do not enable this checkbox if you apply the policy to multiple monitors.) Enable the **Enforce to Monitor** field to validate that all the targeted observers are associated with the monitor to which you apply the policy.
4. In the **Criteria** section, ensure that the policy is applied to the appropriate monitors.

   If you are using this policy to configure observers only, these settings must match the settings in the policy you use to configure your monitor.
5. Click **Apply**.

For information about advanced settings, refer to the Business Transaction Management online help.

7.5.3 How Many Observer Communication Policies Do I Need to Apply?

The preconfigured Observer Communication policy named **Observer Communication Policy - Default** generates both a monitor and an observer configuration and by default is applied to all monitors in the Business Transaction Management sphere. This means that by default all monitors and observers are configured by way of this single policy. For simple deployment topographies and monitoring needs, you can simply

edit this default policy as needed. For more complicated deployment topographies and monitoring needs, you might need to configure some monitors and observers differently than others by applying additional policies.

If all of the following points are true for your environment, then you require only a single Observer Communication policy and you can simply edit the default policy:

- All monitors are either singleton or are replicated behind a single load balancer. You cannot use a single policy if you have a mix of singleton and replicated monitors or if you use multiple load balancers.
- All monitors receive observations on the same port number.
- If monitors receive observations over SSL, they all use the same private key/certificate and the security stores are in the same relative locations for all monitors and observers.
- All observers will be configured to have the same component monitoring capabilities (in other words, they will all have the same probes enabled).
- Advanced settings do not need to be edited for a subset of monitors or observers. In most cases you can leave the advanced settings at their default values.

Each of the following scenarios require you to apply additional Observer Communication policies:

- Your monitors receive observations on different port numbers. In this case, you must apply a separate policy for each port number.
- Your monitors are replicated behind multiple load balancers. In this case, you must apply a separate policy for each load balancer. In addition, if you want to configure the individual monitors to receive observations (from the load balancer) on different port numbers, you must apply a separate policy for each port number, per load balancer.
- You have a mix of singleton and replicated monitors. In this case, you must apply one policy per port number for the singleton monitors and one policy per port number per load balancer for the replicated monitors.
- Monitors will receive observations over SSL but not all monitors will use the same private key/certificate pairs. In this case, you must apply a separate policy for each private key/certificate pair.
- Monitors will receive observations over SSL but the location of the key store is different for different monitors, or the location of the trust store is different for different observers. In this case, you must apply a separate policy for each location.
- Different observers require different component monitoring capabilities. For example, some observers need their OSB probe activated but their SOA probes deactivated, while others need their SOA probes activated but their OSB probe deactivated. In this case, you must apply a separate policy for each combination of activated probes that your observers require.
- You need to edit the advanced settings for a subset of monitors or observers. In this case, every combination of advanced settings requires a separate policy.

7.5.4 Preconfigured Observer Communication Policies

Business Transaction Management provides a number of Observer Communication policies that are preconfigured for monitoring particular types of applications. You can

edit any of these policies and tailor them to your monitoring needs. The name of each policy is displayed in bold, followed by a description:

- **Observer Communication Policy - Default**

  This policy generates both a monitor configuration and an untargeted (default) observer configuration. By default, this policy is applied to all monitors in the system. The observer configuration is distributed to all associated observers that are not specifically targeted by a different policy.

- **Observer Communication Policy - Fusion Applications**

  This policy generates only an observer configuration. By default, this policy is applied to all monitors in the system. The observer configuration is targeted at observers tagged with the label CONFIG_LABEL_FAPPS. This configuration activates probes and adjusts observer runtime settings for monitoring Oracle Fusion Application components.

- **Observer Communication Policy - JavaEE**

  This policy generates only an observer configuration. By default, this policy is applied to all monitors in the system. The observer configuration is targeted at observers tagged with the label CONFIG_LABEL_JAVAEE. This configuration activates probes and adjusts observer runtime settings for monitoring JavaEE components.

- **Observer Communication Policy - OSB**

  This policy generates only an observer configuration. By default, this policy is applied to all monitors in the system. The observer configuration is targeted at observers tagged with the label CONFIG_LABEL_OSB. This configuration activates probes and adjusts observer runtime settings for monitoring Oracle Service Bus components.

- **Observer Communication Policy - SOA**

  This policy generates only an observer configuration. By default, this policy is applied to all monitors in the system. The observer configuration is targeted at observers tagged with the label CONFIG_LABEL_SOA. This configuration activates probes and adjusts observer runtime settings for monitoring Oracle SOA components.

7.5.5 Targeting Observers Reference

The following subsections provide background and reference information related to the targeting of observers:

- Section 7.5.5.1, "Observer Configuration Labels"
- Section 7.5.5.2, "Rejection of Observer Communication Policies"
- Section 7.5.5.3, "Order of Precedence"
- Section 7.5.5.4, "Field Reference for Targeting Observers"

7.5.5.1 Observer Configuration Labels

An observer configuration label is a simple text string that conceptually identifies a set of observers (for example, CONFIG_LABEL_MY_OBSERVERS). You target an observer by specifying either a label or the address of the application server in which the observer is deployed (for example, http://my_host.com:7011). You can specify any number of labels and/or addresses, and target any number of observers with a single policy.

Note: Labels are supported only on Java platforms. If you want to target an observer deployed to .NET, you must use an address.

Labels enable you to group observers logically rather than physically for configuration purposes. Applying a label is a two-step procedure that you can perform in either order:

- In the application server that hosts the observer, create a system property named ap.nano.config.label and set its value to your label string (this task is described in the instructions for installing observer libraries in your application server).
- Set the **Configuration Label** field of the Observer Communication policy that you will use to configure your observer to the same value as ap.nano.config.label (this task is described in Section 7.5.2.2, "Configuring Observer-Related Fields in the Observer Communication Policy").

7.5.5.2 Rejection of Observer Communication Policies

There are a numbers of ways, related to the targeting of observers, in which you might inadvertently cause an Observer Communication policy to be rejected. All of the following scenarios will cause a policy to be rejected:

- Attempting to apply more than one untargeted policy (default observer configuration) to the same monitor
- Attempting to specify the same observer configuration label in two different policies that are applied to the same monitor
- Attempting to specify the same observer base address in two different policies that are applied to the same monitor

Labels are scoped to the monitor to which a policy is applied. This means that you can reuse a particular label name across policies if the policies are applied to different monitors, but you cannot reuse a label name across policies applied to the same monitor. This scoping principle also pertains to untargeted policies.

If a policy is rejected, select the policy in the work area of the console and display the **Targets** tab. This tab provides information about the cause of a rejected policy.

7.5.5.3 Order of Precedence

The order of precedence that determines which configuration an observer receives is as follows:

1. Observer base address

   If a policy specifies the observer's base address, then the observer will receive the configuration generated by that policy.

2. Observer configuration label

   If a policy specifies the observer's configuration label and no policy specifies the base address, then the observer will receive the configuration generated by the policy that specifies the configuration label.

3. Untargeted policy

   If an untargeted policy exists and no policy specifies the observer's base address or configuration label, then the observer will receive the configuration generated by the untargeted policy.

7.5.5.4 Field Reference for Targeting Observers

Field Name (boldface denotes a section name)	Description
Generate Configuration for Observers Only	Leave this checkbox disabled if you want this policy to generate both a monitor and observer configuration. You can apply only one such policy to any monitor. If you apply additional policies to a monitor, they must all have this checkbox enabled. With this checkbox enabled, the policy generates only an observer configuration. Even if this checkbox is enabled, you must still provide values for all other fields in the **Communication Channel** section of the policy, and their values must match all other policies applied to the same monitor.
Targeted Observers	---------- This is a section label ---------- Use this section to specify which observers receive the observer configuration generated by this policy.
Target Specific Observers	Enable this checkbox if you want to configure only specific observers with the observer configuration generated by this policy. Enabling this checkbox displays additional fields that let you specify which observers should receive the configuration. These additional fields let you target observers by way of addresses and/or labels. If you leave this checkbox disabled, the policy is untargeted and will generate a default configuration for all observers associated with the monitors to which the policy is applied. If an associated observer is not targeted by a policy, it will receive this default configuration. You can apply only one untargeted policy to any monitor. If you attempt to apply a second untargeted policy to a monitor, the policy will be rejected. **Notes:** You can target observers by way of two different mechanisms—addresses and labels. But, only one policy can target any particular observer by way of the same mechanism. For example, if policy A targets an observer by way of a label, then policy B cannot target that same observer by way of a label. In this case, policy B would be rejected. Policy B could, however, target the observer by way of an address. In this case, the observer would receive its configuration from policy B because addresses take precedence over labels.
Configuration Label	*Optional* – Specify a comma-delimited list of observer configuration labels. Observers tagged with any of the specified labels will receive configurations generated by this policy (unless a different policy targets the observer by way of an address). This field accepts text input and is case insensitive. **Note:** You tag observers by way of a system property named ap.nano.config.label in the application server hosting the observer. Labels are supported only on Java platforms. If you want to target an observer deployed to .NET, you must use an address.
Known Address	---------- This is a section label ---------- *Optional* – Use this section to choose observer addresses from a drop-down list of addresses known by Business Management Transaction. The specified observers will receive configurations generated by this policy.
Observer Base Address	Use this drop-down list to select the address of the container where the observer is deployed.
[add observer address]	Click this link to add an **Observer Base Address** drop-down list.

Field Name (boldface denotes a section name)	Description
Any Address	---------- This is a section label ---------- *Optional* – Use this section to manually enter observer addresses into a text field. The specified observers will receive configurations generated by this policy.
Observer Base Address	Use this field to manually enter the address of the application server where the observer is deployed, for example, http://my_host.com:7011.
[add observer address]	Click this link to add an **Observer Base Address** text field.
Validate Addresses	Enable this checkbox to ensure that all observers targeted by the policy are known to Business Transaction Management. If you target an unknown observer and enable this field, the policy will be rejected. If you want to target an observer that is not currently known but will be later, you should disable this field.
Enforce to Monitor	Enable this checkbox to validate that all the targeted observers are associated with the monitor to which you apply the policy. Do not enable this checkbox if you apply the policy to multiple monitors.

7.6 Adding and Removing Monitor Group Members

Replicated monitors in a monitor group are known as members. After setting up a monitor group, you might find that you need to add or remove members. You are free to do this, but you must perform the steps of the procedure as described in this section.

7.6.1 Adding Members to a Monitor Group

1. Deploy, and then register the monitor you want to add as described in Section 7.2, "Deploying and Registering Monitors".
2. Assign the monitor to membership in the monitor group as described in Section 7.3, "Setting Up a Monitor Group".
3. Assign the monitor to your load balancer's socket virtual server pool as described in Section 7.4, "Configuring Your Load Balancer".

7.6.2 Removing Members from a Monitor Group

1. Remove the member from your load balancer's socket virtual server pool.
2. Remove the member from the monitor group as follows:
   a. In the Business Transaction Management Console's Navigator, choose **Administration > Monitors**.
   b. In the main area, select the member that you want to remove from the monitor group.
   c. Choose **Modify > Edit Profile for** *Your_Monitor*, where *Your_Monitor* is the name of your member you want to remove.

      The **Edit Profile** tool opens.
   d. Delete the name of the monitor group from the **Monitor Group** field.

e. Click **Apply**.

f. If you want to remove the monitor from membership in the monitor group but leave it as part of your Business Transaction Management system, stop now. If you want to remove the monitor completely from your system, perform the remaining steps.

3. Use your application server's management tools to undeploy the monitor.

4. Unregister the monitor as follows:

   a. In the Business Transaction Management Console's Navigator, choose **Administration > Monitors**.

   b. In the main area, select the monitor.

   c. Choose **Modify > Delete** Your_Monitor **Registration**, where Your_Monitor is the name of your monitor.

      The **Delete Monitor Registration** tool opens.

   d. Click **Delete**.

8

Installing Observers Overview

This chapter describes the procedure for installing Business Transaction Management observers. Installation of observer libraries are covered in separate chapters and are cross referenced from this chapter.

8.1 Prerequisite and Preliminary Setup Checklist

- Ensure that the Business Transaction Management central servers are installed and running (see Chapter 6, "Installing and Configuring the Central Servers").
- Ensure that at least one monitor is installed, registered, and running (see Chapter 7, "Installing Monitors").
- Verify that all system services are running properly.

  You can verify that the system services are running properly by choosing **Administration > System Services** in the Navigator. The summary area then lists all system services. A round, green icon should be displayed in the **System Service Status** column for each system service to indicate that it is running properly.
- If there is a firewall between the observer and its associated monitor (the monitor from which the observer retrieves its configuration and sends observations), ensure that the firewall is configured to allow the observer access to the monitor:
  - In step 4 of Section 8.2, you will associate the observer with a monitor by configuring the observer to retrieve its configuration at a particular URL. You must configure the firewall so that the observer can access this URL.
  - When you set up the Observer Communication policy, you specified a host and port to which the observer will send observations. You must configure the firewall so that the observer can access this host/port. For more information about this host/port, see step 1 of Section 7.5, "Applying an Observer Communication Policy".

8.2 General Steps for Installing Observers

1. Ensure that the assertion secret and encryption key for your observer are set to the same values that are used by the other components of your Business Transaction Management system.

   The components of your Business Transaction Management system use an assertion secret and an encryption key in order to secure communications between themselves. If you have configured your Business Transaction Management server components to use nondefault values for these settings, you must, likewise,

configure each JVM or .NET execution environment that hosts observers to use identical values for these settings. For an in-depth explanation of these security settings and how to configure them, see Section 4.3, "Configuring the Assertion Secret and Encryption Key".

2. *Optional* – Configure SSL security on execution platforms that will host observers for the observer-to-monitor transport of observation messages. If you have configured SSL security for any of your monitors, then you must also configure it for the associated observers.

   See Section 4.4, "Setting up a Secure Socket (SSL) for Observation Messages."

3. Ensure that an Observer Communication policy is set up.

   See Section 7.5, "Applying an Observer Communication Policy".

4. Install the observer libraries and configure them into your application server.

   The procedure for performing this task is specific to the application server and observer type. See the following chapters for detailed instructions:

   - Chapter 9, "Installing Observer Libraries on WebLogic".
   - Chapter 11, "Installing Observer Libraries on Oracle Enterprise Gateway".
   - Chapter 10, "Installing Observer Libraries for WCF".

   Notes: This installation guide provides instructions for installing observers that are included as part of release 12.1.0.4. If you want to install an observer from an earlier release, consult the installation guide from that earlier release. In particular, if you want to install the observer for WebLogic 9.2, WebSphere, JBoss, or ASP.NET, consult the installation guide for release 12.1.0.2.2.

   For detailed information about a specific observer's compatibility and functionality, refer to the README.txt file located in the observer's nanoagent directory after you expand the observer ZIP file.

   ---

5. Set the AP_NANO_CONFIG_URL Java system property or AmberPoint:NanoConfigUrl Windows key on the application server.

   This property/key associates the observer with the monitor whose URL you specify in the property/key. At startup, the observer retrieves its configuration from the specified monitor and begins sending observations to the monitor. Following is an example of the URL. Edit only the host name and port number:

   ```
   http://my_host:8080/btmmonitor/agent/agent/
   ```

   Setting this property/key is also explained in the application server-specific instructions for installing the observer libraries.

Note: If you are using replicated monitors, you must set the host and port portion of the URL to the host and port of your load balancer's HTTP virtual server. For example, if the HTTP virtual server's IP address is 10.147.46.152, and its port number is 5072, then you would set the URL to:

```
http://10.147.46.152:5072/btmmonitor/agent/agent/
```

For more information about the load balancer's HTTP virtual server, see Section 7.4.

6. *Optional* – If you want to target this observer by way of an observer configuration label:
   - *On Java application servers* – Create a Java system property named ap.nano.config.label and set its value to your label string.
   - *For WCF* – Create an environment variable named AP_NANO_CONFIG_LABEL and set its value to your label string.

   For more information about targeting observers refer to Section 7.5, "Applying an Observer Communication Policy."
7. Restart your application server.

8.3 Overriding the Default Location of Observer Libraries

This section applies only to Enterprise Gateway and Java application servers (WCF and ASP .NET observer libraries are installed in the GAC).

By default, the observer library is located at AP_NANO_HOME/lib. If you want to place the observer library in a different location, create a Java system property in your server named AP_NANO_CLASSLOADER_BASEDIR and set its value to the location of the library.

The observer uses the following order of precedence to locate its library (from highest to lowest):

1. the value of the system property AP_NANO_CLASSLOADER_BASEDIR
2. the value of the deprecated system property apclassloader.basedir
3. AP_NANO_HOME/lib

9

Installing Observer Libraries on WebLogic

This chapter provides instructions for installing and uninstalling observer libraries on WebLogic 10.3 and 12 servers.

Business Transaction Management observers are distributed by way of ZIP files. Each ZIP file contains one type of observer that is suitable for installation into a particular application server. The ZIP files suitable for installing an observer into a WebLogic application server are as follows:

- **BTMObserver_Wls_10.3_Universal_*.zip** – Use this ZIP file to install the observer for JavaEE, Oracle SOA Suite, Oracle Service Bus 11gR1, and Oracle Fusion Applications (ADF-UI, ADF-BC and SOA deployments) into a WebLogic 10.3 server.
- **BTMObserver_Wls_12_JavaEE_*.zip** – Use this ZIP file to install the observer for JavaEE into a WebLogic 12 server.
- **BTMObserver_Wls_10.3_Osb11gR1_*.zip** – Use this ZIP file to install the observer for Oracle Service Bus 10gR3 into a WebLogic 10.3 server. Note that although the ZIP file name contains the string "Osb11gR1", this observer supports Oracle Service Bus 10gR3. This observer is not recommended for Oracle Service Bus 11gR1 (you should instead use BTMObserver_Wls_10.3_Universal_*.zip for that purpose).

> **Notes:** In the complete ZIP file name, the asterisk (*) is replaced with the observer version number.
>
> For a list of the exact platform and application server versions supported by these observers, refer to the Business Transaction Management (BTM) Certification Matrix. You can find this document online at `http://support.oracle.com`.
>
> For detailed information about a specific observer's compatibility and functionality, refer to the README.txt file located in the observer's nanoagent directory after you expand the observer ZIP file.

Separate installation instructions are provided for:

- Installation on Node Manager-Configured Servers
- Installation on Script-Configured Servers

9.1 Installation on Node Manager-Configured Servers

1. Locate the appropriate observer distribution ZIP file for your environment and monitoring needs.
2. Unpack the observer ZIP file into your WebLogic managed server's home directory (*WL_HOME*).

   The home directory is the wlserver_10.3 or wlserver_12.1 directory located in your WebLogic installation directory, for example, C:\WL_10-3-2-0\wlserver_10.3. For the remainder of this procedure, replace the string *WL_HOME* with the actual path to the WebLogic home directory.

   Unpacking the ZIP file creates directories named nanoagent and security_add_ons. The nanoagent directory contains three subdirectories named bin, config, and lib.

   Note: By default, the observer looks in the lib directory for its libraries. For information on overwriting this default location, see Section 8.3, "Overriding the Default Location of Observer Libraries."

3. Ensure that the user account running WebLogic has at least the following privileges:
   - read permission on the nanoagent/config and nanoagent/lib directories (on UNIX-like systems traverse permission is also required)
   - read permission on all JAR files in the lib directory
4. Open the WebLogic Administration Console (the default URL is http://*machine_name*:7001/console).
5. Prepare to edit your WebLogic startup arguments:
   a. Using the Domain Structure pane (on the left), navigate to **Environment > Servers**.
   b. In the **Servers** table, click your managed server.
   c. Display the **Configuration / Server Start** tab.
   d. Click **Lock & Edit**.

   Note: These instructions assume you are operating in a production environment and that your WebLogic server's **Automatically Acquire Lock and Activate Changes** setting is therefore disabled. However, if this setting is enabled as it might be in a development environment, you do not have to click **Lock & Edit** in order to make changes and you do not have to activate changes after saving them.

6. With the **Configuration / Server Start** tab displayed, edit your WebLogic startup arguments as follows:
   a. Create an AP_NANO_HOME system property and set it to the location of the observer's nanoagent directory by adding the following string to the **Arguments** field:

      Add this string for Windows systems, using spaces to separate entries:

```
-DAP_NANO_HOME=WL_HOME\nanoagent
```

Add this string for UNIX-like systems, using spaces to separate entries:

```
-DAP_NANO_HOME=WL_HOME/nanoagent
```

Note: Do not use new lines to separate argument entries.

---

b. Create an AP_NANO_CONFIG_URL system property by adding the following string to the **Arguments** field:

```
-DAP_NANO_CONFIG_URL=http://Host:Port/btmmonitor/agent/agent/
```

Replace *Host:Port* with the host name and port number of the monitor to which the observer will forward messages.

This property associates the observer with the monitor whose URL you specify. At startup, the observer retrieves its configuration from the specified monitor and begins sending observations to the monitor.

Note: If you are using replicated monitors, you must set the host and port portion of the URL to the host and port of your load balancer's HTTP virtual server. For example, if the HTTP virtual server's IP address is 10.147.46.152, and its port number is 5072, then you would set the URL to:

```
http://10.147.46.152:5072/btmmonitor/agent/agent/
```

For more information on this topic, see Section 7.4.

---

c. *Optional* – If you want to target this observer by way of an observer configuration label, create a system property named **ap.nano.config.label** and set its value to your label string by adding the following string to the **Arguments** field:

```
-Dap.nano.config.label=My_Label_String
```

Replace *My_Label_String* with the string you want to use as a label for this observer. For more information about targeting observers refer to Section 7.5, "Applying an Observer Communication Policy."

d. Configure the observer bootstrap module into your server by adding one of the following JVM arguments to the **Arguments** field.

Add this argument for Windows systems:

```
-javaagent:WL_HOME\nanoagent\lib\bootstrap\ap-nano-bootstrap.jar
```

Add this argument for UNIX-like systems:

```
-javaagent:WL_HOME/nanoagent/lib/bootstrap/ap-nano-bootstrap.jar
```

7. Ensure that the user under which your WebLogic server is running has permission to write to the observer's error log directory.

By default, the observer's error log directory is the WebLogic domain directory. For information about configuring error logging, see Chapter 13, "Logging Observer Errors and Debugging Information".

8. Click **Save** and then click **Activate Changes**.
9. Restart your managed server.
10. Ensure that the monitor to which your observer forwards messages has an Observer Communication policy applied to it.

    For information on applying an Observer Communication policy, see Section 7.5, "Applying an Observer Communication Policy."

9.2 Installation on Script-Configured Servers

> **Note:** Depending on how you perform this procedure, you can install the observer into either all servers defined in the WebLogic installation or into servers of a specific domain. In this procedure, the term *global install* refers to installing the observer into all servers, and the term *domain install* refers to installing the observer into a specific domain.

1. Locate the appropriate observer distribution ZIP file for your environment and monitoring needs.
2. Unpack the observer ZIP file into either your WebLogic server's home directory (to perform a global install) or into one of its domain directories (to perform a domain install).

   The home directory is the wlserver_10.3 or wlserver_12.1 directory located in your WebLogic installation directory, for example, C:\WL_10-3-2-0\wlserver_10.3. For the remainder of this procedure, replace *WL_HOME* with the actual path to the WebLogic home directory.

   Unpacking the ZIP file creates directories named nanoagent and security_add_ons. The nanoagent directory contains three subdirectories named bin, config, and lib.

   > **Note:** By default, the observer looks in the lib directory for its libraries. For information on overwriting this default location, see Section 8.3, "Overriding the Default Location of Observer Libraries."

3. Ensure that the user account running WebLogic has at least the following privileges:
   - read permission on the nanoagent/config and nanoagent/lib directories (on UNIX-like systems traverse permission is also required)
   - read permission on all JAR files in the lib directory
4. Configure your WebLogic domain startup scripts to call the observer script file:

   > **Note:** This step assumes that you haven't modified your startWebLogic scripts. If you have modified your scripts, you might also have to modify the installation procedure accordingly.

   a. Navigate to the bin directory of one of the WebLogic domains whose services you want to monitor and open the startup script in a text editor (open

bin\startWebLogic.cmd for Windows systems or bin/startWebLogic.sh for UNIX-like systems; do not edit the startup script located directly within the domain directory).

b. Locate the following line (the first line is for Windows and the second for UNIX-like systems):

```
call "%DOMAIN_HOME%\bin\setDomainEnv.cmd"

. ${DOMAIN_HOME}/bin/setDomainEnv.sh
```

c. Directly after that line, add a line that calls the observer script file:

If you are performing a global install on a Windows system, add this line:

```
call "%WL_HOME%\nanoagent\bin\nanoEnvWeblogic.cmd"
```

If you are performing a global install on a UNIX-like system, add this line (note the initial period and space):

```
. ${WL_HOME}/nanoagent/bin/nanoEnvWeblogic.sh
```

If you are performing a domain install on a Windows system, add this line:

```
call "%DOMAIN_HOME%\nanoagent\bin\nanoEnvWeblogic.cmd"
```

If you are performing a domain install on a UNIX-like system, add this line (note the initial period and space):

```
. ${DOMAIN_HOME}/nanoagent/bin/nanoEnvWeblogic.sh
```

Note: If you performed the security-related configuration described in Section 4.3.1, "Configuring Security Using Oracle Wallet," you will have already added a call to the setBtmOverriderEnv_via_CredStore script in this location. The relative order in which these scripts are called does not matter.

---

d. Perform this step for each domain whose services you want to monitor.

5. Open the observer script file in a text editor.

The observer script file is the nanoEnvWeblogic.cmd or nanoEnvWeblogic.sh file that your startWebLogic script file calls.

6. If you are performing a domain install, make the following change (otherwise, skip to the next step):

For Windows systems, locate the line:

```
set NANOAGENT_HOME=%WL_HOME%\nanoagent
```

and change it to:

```
set NANOAGENT_HOME=%DOMAIN_HOME%\nanoagent
```

For UNIX-like systems, locate the line:

```
NANOAGENT_HOME=$WL_HOME/nanoagent
```

and change it to:

```
NANOAGENT_HOME=$DOMAIN_HOME/nanoagent
```

Keep the file open.

7. Associate your observer with a monitor.

   Note: The observer script file uses the NANOAGENT_CONFIGURATION_URL variable to set the value of the system property AP_NANO_CONFIG_URL. This system property associates the observer with the monitor whose URL you specify. At startup, the observer retrieves its configuration from the specified monitor and begins sending observations to the monitor.

   a. In the observer script file, locate the following variable definition (the first line is for Windows and the second for UNIX-like systems):

   ```
   set NANOAGENT_CONFIGURATION_URL=http://HOST:PORT/btmmonitor/agent/agent/

   NANOAGENT_CONFIGURATION_URL=http://HOST:PORT/btmmonitor/agent/agent/
   ```

   b. Replace HOST:PORT with the host name and port number of the monitor to which you want the observer to send observations, for example:

   ```
   set NANOAGENT_CONFIGURATION_URL=http://myhost:7002/btmmonitor/agent/agent

   NANOAGENT_CONFIGURATION_URL=http://myhost:7002/btmmonitor/agent/agent
   ```

   Note: If you are using replicated monitors, you must set the host and port portion of the URL to the host and port of your load balancer's HTTP virtual server. For example, if the HTTP virtual server's IP address is 10.147.46.152, and its port number is 5072, then you would set the URL to:

   ```
   http://10.147.46.152:5072/btmmonitor/agent/agent/
   ```

   For more information about the load balancer's HTTP virtual server, see Section 7.4.

   c. If you are performing a domain install, perform steps 6 and 7 for each domain.

8. *Optional* – If you want to target this observer by way of an observer configuration label, perform the following substeps:

   Note: The observer script file uses the NANOAGENT_CONFIGURATION_LABEL variable to set the value of the system property ap.nano.config.label. For information about targeting observers refer to Section 7.5, "Applying an Observer Communication Policy."

   a. In the observer script file, locate the following variable definition (the first line is for Windows and the second for UNIX-like systems):

   ```
   set NANOAGENT_CONFIGURATION_LABEL=

   NANOAGENT_CONFIGURATION_LABEL=
   ```

b. Append your label string to the variable definition, for example:

```
set NANOAGENT_CONFIGURATION_LABEL=My_Label

NANOAGENT_CONFIGURATION_LABEL=My_Label
```

9. Ensure that the user under which your WebLogic server is running has permission to write to the observer's error log directory.

   By default, the observer's error log directory is the WebLogic domain directory. For information about configuring error logging, see Chapter 13, "Logging Observer Errors and Debugging Information".

10. Restart your server.

11. Ensure that the monitor to which your observer forwards messages has an Observer Communication policy applied to it.

    For information on applying an Observer Communication policy, see Section 7.5, "Applying an Observer Communication Policy."

9.3 Uninstalling Observer Libraries for WebLogic

This section describes how to uninstall observer libraries from a WebLogic 10.3 server. The procedure differs according to whether you configure your server using the Node Manager or local scripts.

- If you configure your server using the Node Manager, see Section 9.3.1, "Uninstallation from a Managed Server Configured by the Node Manager."
- If you configure your server using local scripts, see Section 9.3.2, "Uninstallation from a Server Configured by a Local Script."

9.3.1 Uninstallation from a Managed Server Configured by the Node Manager

1. Stop your WebLogic managed server.
2. Delete the nanoagent folder located in your WebLogic home directory (*WL_HOME*).

   The home directory is the wlserver_10.3 or wlserver_12.1 directory located in your WebLogic installation directory, for example, C:\WL_10-3-2-0\wlserver_10.3. For the remainder of this procedure, replace *WL_HOME* with the actual path to the WebLogic home directory.

3. Open the WebLogic Administration Console (the default URL is http://*Machine_Name*:7001/console).
4. Remove all observer-related startup arguments:
   a. Using the navigation pane (on the left), navigate to **Environment > Servers**.
   b. In the **Servers** table, click your managed server.
   c. Display the **Configuration / Server Start** tab.
   d. Click **Lock & Edit**.
   e. Remove any and all of the following strings from the **Arguments** field:

```
-DAP_NANO_HOME=WL_HOME/nanoagent
-DAP_NANO_HOME=WL_HOME\nanoagent
-DAP_NANO_CONFIG_URL=http://HOST:PORT/btmmonitor/agent/agent/
```

```
-Dap.nano.config.label=My_Label
-javaagent:WL_HOME\nanoagent\lib\bootstrap\ap-nano-bootstrap.jar
-javaagent:WL_HOME/nanoagent/lib/bootstrap/ap-nano-bootstrap.jar
```

5. Click **Save** and then click **Activate Changes**.
6. Restart your WebLogic server.

9.3.2 Uninstallation from a Server Configured by a Local Script

1. Stop your WebLogic server.
2. Delete the nanoagent folder located in your WebLogic home directory (*WL_HOME*).

   The home directory is the wlserver_10.3 or wlserver_12.1 directory located in your WebLogic installation directory, for example, C:\WL_10-3-2-0\wlserver_10.3. For the remainder of this procedure, replace *WL_HOME* with the actual path to the WebLogic home directory.
3. Navigate to the bin directory of the WebLogic domain whose services you no longer want to monitor and edit the startWebLogic script (startWebLogic.cmd for Windows or startWebLogic.sh for UNIX-like systems). Delete the following line from the script (the first line is for Windows and the second for UNIX-like systems):

   ```
   call "%WL_HOME%\nanoagent\bin\nanoEnvWeblogic.cmd"
   ```

   ```
   . ${WL_HOME}/nanoagent/bin/nanoEnvWeblogic.sh
   ```
4. Restart your WebLogic server.

10

Installing Observer Libraries for WCF

This topic provides instructions for installing and uninstalling observer libraries for monitoring WCF 3.5 and 4.0 services. The observer requires that you have the .NET 3.5 Framework installed on your machine for monitoring WCF 3.5 services or the .NET 4.0 Framework for monitoring both WCF 3.5 and 4.0 services. This observer monitors only IIS 7.5-hosted, WCF Services.

Notes: For a list of the exact application server versions supported by this observer, refer to the Business Transaction Management (BTM) Certification Matrix. You can find this document online at http://support.oracle.com.

For detailed information about a specific observer's compatibility and functionality, refer to the README.txt file located in the observer's nanoagent directory after you expand the observer ZIP file.

10.1 The Observer Distribution File

The Business Transaction Management observers are distributed by way of ZIP files. Each ZIP file contains one type of observer that is suitable for installation into a particular application server. The ZIP file containing the observer for monitoring WCF 3.5 services is BTMObserver_Iis_7.5_DotNet4_*.zip.

Notes: In the complete ZIP file name, the asterisk (*) is replaced with the observer version number.

10.2 Installing the Observer Libraries for WCF 3.5 and 4.0

1. Locate the observer distribution ZIP file for WCF (BTMObserver_Iis_7.5_DotNet4_*.zip).
2. Unpack the ZIP file into a temporary directory (referred to henceforth as observer_temp).

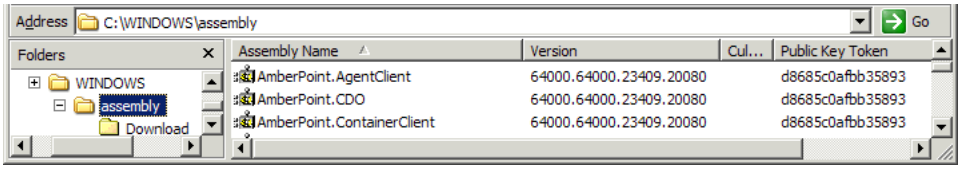
   Unpacking the ZIP file creates a nanoagent directory containing two subdirectories—config and lib.
3. Use gacutil.exe to copy all of the DLL files from observer_temp\nanoagent\lib to the global application cache (GAC; normally located at C:\WINDOWS\assembly).
4. Configure the observer into your applications by editing the application configuration file as follows:

Caution: Make a backup copy before editing your application configuration file.

To monitor all WCF applications running on the machine, edit the machine.config file (see Section 10.3).

To monitor one or more specific web applications, edit the web.config file appropriate for each web application (see Section 10.4).

5. Ensure that the version numbers of the AmberPoint assemblies in the GAC match the version number in your application configuration file, for example:




```
<soapExtensionTypes>
  <add type="AmberPoint.NanoAgent.DotNet.AspNet.Handlers.SoapExtensionHandler,
   AmberPoint.NanoAgentToolkit, Version=64000.64000.23409.20080,
   Culture=neutral,
   PublicKeyToken=d8685c0afbb35893" priority="1" group="0" />
</soapExtensionTypes>
```

The version numbers must match for the assemblies to be found.

6. Make the following edits to your application configuration file:

   a. Set the value of the AmberPoint:NanoConfigUrl key to the URL of the monitor to which you want the observer to forward messages.

   Use the following form, replacing *Host:Port* with the host name and port number of the monitor:

   ```
   <add key="AmberPoint:NanoConfigUrl"
   value="http://Host:Port/btmmonitor/agent/agent"/>
   ```

   This key associates the observer with the monitor whose URL you specify in the key. At startup, the observer retrieves its configuration from the specified monitor and begins sending observations to the monitor.

   As an alternative to setting the monitor URL in the AmberPoint:NanoConfigUrl key, you can create an environment variable named AP_NANO_CONFIG_URL and use it to set the monitor URL.

Notes: If you are using replicated monitors, you must set the host and port portion of the URL to the host and port of your load balancer's HTTP virtual server. For example, if the HTTP virtual server's IP address is 10.147.46.152, and its port number is 5072, then you would set the URL to:

```
http://10.147.46.152:5072/btmmonitor/agent/agent/
```

For more information about the load balancer's HTTP virtual server, see Section 7.4.

b. Set the value of the AmberPoint:NanoHome key to the location in which you want the observer to cache its configuration file, for example:

```
<add key="AmberPoint:NanoHome" value="C:/nanohome"/>
```

c. Set the value of the AmberPoint:NanoLogBaseDir key to the location in which you want error logs created. Specify the location as an absolute path. If you want the directory created in case it doesn't exist, set the value of the AmberPoint:NanoCreateLogBaseDir key to true. For example:

```
<add key="AmberPoint:NanoLogBaseDir" value="C:/nanohome/log"/>
<add key="AmberPoint:NanoCreateLogBaseDir" value="true"/>
```

The AmberPoint:NanoLogBaseDir key does not have a default value. If it is set to null, log files will not be generated.

In order for the observer to generate the log files, ensure that the user under which the observer is running has permission to write to the log directory. The observer runs as the user named NETWORK SERVICE.

For information about configuring error logging, see Chapter 13, "Logging Observer Errors and Debugging Information."

7. *Optional* – If you want to target this observer by way of an observer configuration label, create an environment variable named AP_NANO_CONFIG_LABEL and set its value to your label string.

For more information about targeting observers refer to Section 7.5, "Applying an Observer Communication Policy."

8. Ensure that the monitor to which your observer forwards messages has an Observer Communication policy applied to it.

For information on applying an Observer Communication policy, see Section 7.5, "Applying an Observer Communication Policy."

10.3 Editing the machine.config File

Caution: Make a backup copy before editing your machine.config file.

To monitor all WCF applications running on the machine, edit the machine.config file. This file is typically located at C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CONFIG.

An annotated example of a machine.config file is provided at observer_temp\nanoagent\config\Machine.config.part. Use this file as a guide in making your edits.

10.4 Editing the web.config File

Caution: Make a backup copy before editing your web.config file.

You can monitor all web applications or only specific web applications depending on which web.config file you edit. For example, if you want to monitor all the applications installed in the web site, edit the web.config file located in the web site's

home directory. If you want to monitor only specific web applications, edit the web.config file located in the directories of those specific applications.

Annotated examples of web.config files are provided at observer_temp\nanoagent\config\Web.config.part and Web.config.wcf4.part. If you are installing the observer into a WCF 3.5 environment, use Web.config.part as a guide in making your edits. If you are installing the observer into a WCF 4.0 environment, you can use either file as a guide, although the procedure described in Web.config.wcf4.part is more streamlined.

10.5 Uninstalling the Observer Libraries for WCF 3.5 and 4.0

1. Delete the apobserver.configuration file from the location specified in your application configuration file as the value of the AmberPoint:NanoConfig key.

   Your application configuration file is either machine.config or web.config.

2. Revert your application configuration file to the state it was in before you installed the observer.

   The observer distribution file (BTMObserver_Iis_7.5_DotNet4_*.zip) contains annotated examples of application configuration files that explain how to edit an application configuration file to insert the observer. You can also use these as a guide in reverting your application configuration file.

   If you need to revert your machine.config file, use the config\Machine.config.part file as a guide. If you need to revert your web.config file, use either config\Web.config.part or Web.config.wcf4.part as a guide, depending on which you used to configure your web.config file.

3. Remove the observer DLLs from the GAC (unless they are being used by another observer on the machine).

   The name of each observer DLL begins with the string "AmberPoint". To remove a DLL, right-click it and choose **Uninstall**.

   Note: The observer for ASP.NET uses many of the same DLLs as the observer for WCF. If you have the observer for ASP.NET installed on the machine, you must not remove the DLLs that are being used by it.

11

Installing Observer Libraries on Oracle Enterprise Gateway

This chapter provides instructions for installing and uninstalling observer libraries into Oracle Enterprise Gateway 11.1.1.6.

11.1 The Observer Distribution File

The Business Transaction Management observers are distributed by way of ZIP files. Each ZIP file contains one type of observer that is suitable for installation into a particular application server. The ZIP file suitable for installing an observer into an Enterprise Gateway server is BTMObserver_OEG_11.1.1.6_OEG_*.zip.

Notes: In the actual ZIP file, the asterisk (*) is replaced with the observer version number.

For a list of the exact platform and application server versions supported by this observer, refer to the Business Transaction Management (BTM) Certification Matrix. You can find this document online at `http://support.oracle.com`.

For detailed information about a specific observer's compatibility and functionality, refer to the README.txt file located in the observer's nanoagent directory after you expand the observer ZIP file.

11.2 Installing Observer Libraries on Enterprise Gateway 11.1.1.6

1. Shut down your Enterprise Gateway server.
2. Unpack the observer ZIP file (BTMObserver_OEG_11.1.1.6_OEG_*.zip) into a temporary directory, referred to henceforth as *observer_temp*.

   Unpacking the ZIP file creates three directories named config, lib, and scripts.
3. Copy all of the JAR files located in the lib directory to *OEG_HOME*/ext/lib, where *OEG_HOME* is your Enterprise Gateway server's home directory (the top-level installation directory).

Note: By default, the observer looks in the lib directory for its libraries. For information on overriding this default location, see Section 8.3, "Overriding the Default Location of Observer Libraries."

---

4. Ensure that the user account running Enterprise Gateway has at least the following privileges:
   - read permission on the nanoagent/config and nanoagent/lib directories (on UNIX-like systems traverse permission is also required)
   - read permission on all JAR files in the lib directory
5. Open *OEG_HOME*/system/conf/jvm.xml in a text editor and make the following changes inside the <JVMSettings> element:
   a. Associate your observer with a monitor by adding the following line as the first child of the <JVMSettings> element:

      ```
      <SystemProperty name="AP_NANO_CONFIG_URL" value="http://Host_Name:Port_
      Number/btmmonitor/agent/agent/"/>
      ```

      Replace *Host_Name:Port_Number* with the host name and port number of the monitor you want to associate with your observer.

   b. Then add the following lines as the 2nd, 3rd, and 4th children of the <JVMSettings> element:

      ```
      <SystemProperty name="AP_NANO_HOME" value="$VDISTDIR/NanoAgent"/>
      <SystemProperty name="AP_NANO_LOG_BASEDIR" value="$VDISTDIR/NanoAgent"/>
      <SystemProperty name="AP_NANO_CLASSLOADER_BASEDIR"
      value="$VDISTDIR/ext/lib"/>
      ```

   c. *Optional* – If you want to target this observer by way of an observer configuration label, add the following line as the 5th child of the <JVMSettings> element:

      ```
      <SystemProperty name="ap.nano.config.label" value="My_Label_String"/>
      ```

      Replace *My_Label_String* with the string you want to use as a label for this observer. For more information about targeting observers refer to Section 7.5, "Applying an Observer Communication Policy."

   d. Locate this line:

      ```
      <ClassPath name="$VDISTDIR/system/lib/system/iaik_jce.jar"/>
      ```

      and then add this line just before it:

      ```
      <ClassPath name="$VDISTDIR/ext/lib/orawsdl_1.0.0.jar"/>
      ```

6. Restart your Enterprise Gateway server.
7. Publish the interceptor module to finish the installation of the observer as follows:
   a. Open the *observer_temp*/scripts/publishLoadableModule.py script file in a text editor.
   b. Set the defUsername and defPassword variables using the credentials of an Enterprise Gateway administrator's account (the user name of the default administrator account is admin), for example:

      ```
      defUserName = "admin"
      defPassword = "mypassword"
      ```

   c. Set the defServer variable to the base URL of the Enterprise Gateway server, for example:

      ```
      defServer = "http://myOegHost:8090"
      ```

Do not add a trailing slash (/) to this URL.

d. Save and close the script file.

e. Run the script, for example:

On Windows systems, run:

```
%OEG_HOME%/Win32/bin/jython.bat publishLoadableModule.py
```

On UNIX-like systems, run:

```
$OEG_HOME/posix/bin/jython publishLoadableModule.py
```

8. Ensure that the user under which your Enterprise Gateway server is running has permission to write to the observer's error log directory.

By default, the observer's error log directory is *OEG_HOME*. For information about configuring error logging, see Chapter 13, "Logging Observer Errors and Debugging Information."

9. Ensure that the monitor to which your observer forwards messages has an Observer Communication policy applied to it.

For information on applying an Observer Communication policy, see Section 7.5, "Applying an Observer Communication Policy."

11.3 Uninstalling Observer Libraries from Enterprise Gateway 11.1.1.6

This section describes how to uninstall observer libraries from an Enterprise Gateway 11.1.1.6 server.

1. Remove the observer interceptor module from your Enterprise Gateway server as follows:

a. Open the *observer_temp*/scripts/removeLoadableModule.py script file in a text editor.

Note: If *observer_temp* no longer exists, refer to Section 11.2, "Installing Observer Libraries on Enterprise Gateway 11.1.1.6" for information about recreating it.

---

b. Set the defUsername and defPassword variables using the credentials of an Enterprise Gateway administrator's account (the user name of the default administrator account is admin), for example:

```
defUserName = "admin"
defPassword = "mypassword"
```

c. Set the defServer variable to the base URL of the Enterprise Gateway server, for example:

```
defServer = "http://myOegHost:8090"
```

Do not add a trailing slash (/) to this URL.

d. Save and close the script file.

e. Run the script, for example:

On Windows systems, run:

```
%OEG_HOME%/Win32/bin/jython.bat removeLoadableModule.py
```

On UNIX-like systems, run:

```
$OEG_HOME/posix/bin/jython removeLoadableModule.py
```

2. Shut down your Enterprise Gateway server.
3. Remove all of the observer-related JAR files located in *OEG_HOME*/ext/lib.

   The observer-related JAR files are those prefixed with "ap-" plus the following files:

   ```
   orahttp_client_1.0.0.jar
   orawsdl_1.0.0.jar
   xstream-1.2.2.jar
   ```

   Note: Take care not to remove JAR files that are being shared with another module.

4. Open *OEG_HOME*/system/conf/jvm.xml in a text editor and remove any and all observer-related elements from inside the <JVMSettings> element, including the following:

   ```
   <SystemProperty name="AP_NANO_CONFIG_URL"
   value="http://host-name:port-number/btmmonitor/agent/agent/"/>

   <ClassPath name="$VDISTDIR/ext/lib/orawsdl_1.0.0.jar"/>
   ```

   Also remove all <VMArg> elements whose name attribute begins with the following string:

   ```
   -Dcom.amberpoint
   ```

5. Save and close the jvm.xml file.
6. Restart your Enterprise Gateway server.

12

Starting and Shutting Down Business Transaction Management

This chapter explains how to:

- start and shutdown Business Transaction Management components
- shutdown and restart members of a monitor group without losing observations
- log in to and out of the Management Console
- access the online help

12.1 Starting Business Transaction Management Components

Business Transaction Management components start automatically when the application server in which they are installed starts. Because the central servers and monitor are deployed applications, you can also start them manually using your application server's management facilities. Observers, however, are installed into your business applications and start along with those applications.

If all of the components of your system are shut down, the best order in which to start them is from the center out, that is:

1. Central servers
2. Monitors
3. Observers

However, you can start them in any order and achieve a fully functioning system after all the components have recognized each other and configured themselves.

Note: If the observers, and therefore your business applications, are running, but the monitors are not, the observers' outgoing queue will eventually fill up. At that point, either observation messages will be dropped from the queue or your business applications will stall. The default behavior is for observation messages to be dropped from the queue, which ensures that the performance of your business applications is not degraded. This behavior is configurable by way of the Observer Communication policy. However, this is an advanced option and you should leave it at its default setting unless you are instructed by the Oracle support team to edit it. If you require the ability to shut down monitors without the risk of dropping observations, you should replicate your monitors as described in Chapter 7, "Installing Monitors."

12.2 Shutting Down Business Transaction Management Components

Business Transaction Management components shut down when the application server in which they are installed shuts down. Because the central servers and monitor are deployed applications, you can also shut them down them using your application server's management facilities. Observers, however, are installed into your business applications and shut down along with those applications.

If you want to shut down all of the components of your system, the best order to shut them down is from the outside to the center, that is:

1. Observers
2. Monitors
3. Central servers

However, you can shut down and then restart individual components without harming the system. After all the components have recognized each other and reconfigured themselves, you should once again have a fully functioning system.

The note in Section 12.1 that describes a possible loss of observations, however, also applies in this case.

12.3 Shutting Down and Restarting Monitor Group Members

Replicated monitors in a monitor group are known as members. This section explains how to shut down individual members of a monitor group without losing in-flight data. It also explains how to properly restart monitor group members.

12.3.1 Shutting Down Monitor Group Members

1. Disable the monitor's associated pool member in the load balancer's socket virtual server.
2. Wait for all in-flight data to clear from the monitor.

   You can check the status of in-flight data as follows:

   a. Display the monitor's status page in either of the following ways:

      In the Business Transaction Management Console's Navigator, choose **Administration > Monitors**, select the monitor in the main area, and click the **Status** tab.

      Or, access the following URL in a web browser, where *host:port* are the host name and port number where the monitor is running:

      ```
      http://host:port/btmmonitor/agent/agent?status
      ```

   b. Scroll to the **Nano Observation Queue** section.

   c. All in-flight data has cleared from the monitor if both of the following equations are true:

      ```
      perf.SimpleMeasurementsProcessed = perf.SimpleMeasurementsQueued

      Perf.RequestsQueued = Perf.ResponsesQueued + Perf.QueueSize
      ```

3. Use your application server's administration tools to shut down the monitor deployment.

12.3.2 Restarting Monitor Group Members

1. Use your application server's administration tools to start the monitor deployment.
2. Enable the monitor's associated pool member in the load balancer's socket virtual server.

12.4 Logging in to the Management Console

You administer and access the facilities of Business Transaction Management using a web-based interface called the Business Transaction Management Console (sometimes referred to in this document simply as the Management Console).

In order to access the Management Console, the Main server (btmMain.ear) must be running. To log in to the Management Console, open a URL of the following form in a web browser:

`http://hostname:port/btmui`

Note: The web browser must meet the requirements listed in Section 5.1, "Web Browser Requirements."

Replace *hostname:port* with the name of the machine on which you installed the Main server, and the port number on which it is accessible.

The web browser will display a log-in page. Log in using the appropriate credentials, which are described in section 6.4, "Mapping Users to Roles".

12.5 Logging out of the Management Console

To log out of the Management Console, click the **Log out** link in the upper-right corner of the page.

12.6 Online Help

You can access Business Transaction Management online help by clicking the **Help** menu in the Management Console and choosing **Help**, or by clicking the **Help** button from within a tool dialog box. Once the online help opens, use the navigation facilities on the left to locate the appropriate help topic.

13

Logging Observer Errors and Debugging Information

The observer writes error and debugging information to the following log files:

- **NanoAgentErrorTrace.log** – contains single occurrences of all errors and warnings logged to the other log files. Each error and warning entry is referenced by a unique identifier within a <Ref> element, for example:

  ```
  <Ref: Dq/QGNWqOmbdXPigC+vsO40eXgs=>
  ```

  You can use this identifier to search for all occurrences of the error or warning in the other log files, typically within NanoAgent.log. This is generally the first log file you should check when a problem occurs.

  The default size of this log file is 10M and it is recreated on each restart of the server. However, because its default rotation is set to 2, the previous log file is retained after a server restart.

- **NanoAgent.log** – contains runtime error and debugging information (you can adjust this logger's settings using the Enable trace logging option in the Observer Communication policy.)

- **NanoAgentPreprocessTrace.log** – contains information about bytecode instrumentation errors and debugging, class-loading, and preprocessing. This file is regenerated on each restart of the server. The maximum size of this log file is 10 MB.

  This file was renamed for release 12.1.0.2.2. For observers of previous releases, the file was named AWTrace.log.

> **Note:** You can also configure observers to log observed messages. For information on this topic, refer to the online help for information about the **Log Observed Messages to File** field in the Observer Communication policy.

The default location of the log files is as follows:

- **WebLogic** – the *domain_root_directory*/nanoagent/logs/*server_name* directory (if that directory cannot be determined, then it defaults to the domain root directory)
- **OC4J** – the j2ee\home directory inside your SOA Suite installation directory
- **Enterprise Gateway** – the home directory (top-level installation directory) of the Enterprise Gateway server
- **WebSphere** – the profile directory

- **JBoss** – the JBOSS_HOME/bin directory
- **WCF and ASP.NET** – the C:/temp/NanoAgentBaseDir directory

> **Note:** The default log location for WCF and ASP.NET is not a true default. It is simply the default setting of the AmberPoint:NanoLogBaseDir key. If you set this key to null, log files will not be created.

If you want the log files generated in a different directory, set the AP_NANO_LOG_BASEDIR Java property or AmberPoint:NanoLogBaseDir Windows key. For Java application servers, you can set the property to either an absolute path or a path that is relative to the default log directory. For Enterprise Gateway, WCF, and ASP.NET, you must set the property or key to an absolute path. The following examples illustrate how to set this property or key:

- On WebLogic, if you configure your server by editing local scripts, edit the nanoEnvWeblogic script located in WL_HOME/nanoagent/bin directory. In the options section of the file, add **-DAP_NANO_LOG_BASEDIR="my_log_dir"** to the end of the NANOAGENT_JAVA_OPTIONS. This relative path would generate the log files in the directory my_log_dir under your domain directory.

  If you configure you WebLogic server using the Node Manager, open the WebLogic Administration Console, select your server, and display the **Configuration / Server Start** tab. Then add **-DAP_NANO_LOG_BASEDIR=my_log_dir** to the **Arguments** field. This relative path would generate the log files in the directory my_log_dir under your domain directory.
- On OC4J, add **-DAP_NANO_LOG_BASEDIR=my_log_dir** to the Java startup options. This relative path would generate the log files in the directory my_log_dir under the j2ee\home directory inside your SOA Suite installation directory.
- On Enterprise Gateway, open OEG_HOME/system/conf/jvm.xml in a text editor and add **<SystemProperty name="AP_NANO_LOG_BASEDIR" value="C:\OEG\my_log_dir"/>** as a child of the <JVMSettings> element. This absolute path would generate the log files in the directory C:\OEG\my_log_dir.
- On WebSphere, in the WebSphere Administrative Console, navigate to Servers > Application servers > server1 > Server Infrastructure > Java and Process Management > Process Definition > Java Virtual Machine > Custom Properties (you might have to substitute a different server name for server1). Create a custom property named **AP_NANO_LOG_BASEDIR** and set it's value to **my_log_dir**. This relative path would generate the log files in the directory my_log_dir under your profile directory.
- On JBoss, edit your server startup script JBOSS_HOME/bin/run. In the options section of the file, add **set JAVA_OPTS=-DAP_NANO_LOG_BASEDIR="my_log_dir".** This relative path would generate the log files in the directory JBOSS_HOME/bin/my_log_dir.
- For WCF or ASP.NET, edit the application configuration file (for example, Web.config) and set the value for the AmberPoint:NanoLogBaseDir key to **C:/Inetpub/wwwroot/my_log_dir**. This absolute path would generate the log files in the directory my_log_dir under your default web site directory, for example:

```
<configuration>
 <configSections>
  ...
 </configSections>
```

```
 <AmberPoint>
  <NanoAgentDataSection>
   <add key="AmberPoint:NanoConfig"
value="c:/temp/NanoAgentLogBaseDir/nanoagentDiscovery.CONFIGURATION"/>
   <add key="AmberPoint:NanoLogBaseDir" value="c:/Inetpub/wwwroot/my_log_dir"/>
   <add key="AmberPoint:NanoCreateLogBaseDir" value="false"/>
  </NanoAgentDataSection>
 </AmberPoint>
 <system.web>
  ...
 </system.web>
</configuration>
```

In order for the observer to generate the log files, ensure that the user under which the observer is running has permission to write to the log directory. For Java observers, the user is the user that is running the application server. For IIS observers (WCF and ASP.NET), the user is as follows:

- IIS 5.x – the observer user is ASPNET
- IIS 6.x and 7.x – the observer user is NETWORK SERVICE

By default, the directory specified by the AP_NANO_LOG_BASEDIR property is automatically created if it does not exist. If you do not want this directory to be automatically created, set the property AP_NANO_CREATE_LOG_BASEDIR to **false**. In this case, you must create the directory yourself. Set this property in the same way you set AP_NANO_LOG_BASEDIR.

Notes: *For Java application servers* – If the log directory does not exist and AP_NANO_CREATE_LOG_BASEDIR is set to false, runtime errors might occur and the observer might not initialize.

For IIS – If the NanoLogBaseDir Windows key is set to null, log files are not created.

14

Scripted Configuration of Oracle Business Transaction Management

When you configure Business Transaction Management for the first time, we recommend that you use the browser-based Configuration Wizard. For users who want to later execute various configuration tasks from the command line, a command line interface (CLI) is provided.

The CLI provides command-line equivalents to many configuration tasks. The CLI command relevant to this document is the configure command. This command provides an alternative to the wizard approach to configuration.

You can find complete documentation for the configure command, as well as the entire CLI, in the Business Transaction Management online help.

14.1 The configure Command

The configure command takes an XML configuration file as an argument. This configuration file specifies all the setup information for Business Transaction Management, including database type and connection information, deployment credentials, and so on.

You can develop this configuration input file by using the generated configuration file that is output when you perform initial configuration of Business Transaction Management using the browser-based Configuration Wizard. The wizard produces a configuration file that contains all the configuration information (sphere URL, database connection information, performance server location, authentication credentials, and so on). You can edit this configuration file and use it as input to the configure command. The generated configuration file is named essentialConfiguration.xml and is located inside the WebLogic installation directory at:

```
user_projects/domains/MyDomain/servers/MyServer/btmstorage/btmMain
/globalPreferences
```

14.2 Invoking the CLI

The CLI executable is located in the *Install_Dir*/tools directory—btmcli.bat for Windows and btmcli.sh for Unix-like systems.

Command syntax for use with the configure command:

```
btmcli configure -i inputFile -s sphereUrl -l username:password
```

You can avoid placing the username and password on the command line by setting the AP_USER_LOGIN environment variable before executing the CLI, for example:

```
set AP_USER_LOGIN=MyUsername:MyPassword
```

15

The datastoreUtil Utility

Section 5.4, "Setting up Business Transaction Management Databases," instructs you to create database users for the sphere, performance, and transaction databases using the following suggested names:

- **sphereDB** (for the sphere database)
- **measurementDB** (for the performance database)
- **transactionDB** (for the transaction database)

This chapter assumes you are using the suggested names. If not, substitute your names as appropriate.

When you configure Business Transaction Management (see section 6.5, "Initial Configuration of Business Transaction Management"), the system automatically creates the appropriate database tables for these users, unless you choose to create them beforehand with the datastoreUtil Utility.

The datastoreUtil utility enables you to generate the DDL that you can use as input (with sqlplus) to create the necessary tables and views for Business Transaction Management. You can alternatively use datastoreUtil to connect to a database instance and create the tables and views directly.

15.1 Usage

To invoke the datastoreUtil utility, use a command window or shell to navigate to the tools directory of your Business Transaction Management installation and execute the datastoreUtil script that is appropriate for your operating system (either datastoreUtil.bat or datastoreUtil.sh).

After starting the utility, you can run any of the commands listed in Section 15.2, "Commands." For commands that have multiple arguments, you must call the arguments in the order described.

15.2 Commands

The datastoreUtil utility provides the following commands:

- generateSchema (or generate)
- connect
- createSchema (or create)
- close
- exit

- help

15.2.1 generateSchema (or generate)

Generates a DDL of the specified schema definition. You can use this command without being connected to a database.

```
generateSchema schemaType databaseType [[directory] targetSchema]
-partition|-nopartition
```

- *schemaType* – Specify one of the following schema types:
  - **sphere** – Creates a schema for the sphere database (the sphereDB user).
  - **exm** – Creates a schema for the transaction database (the transactionDB user).
  - performance – Creates a schema for the performance database (the measurementDB user).
  - **monitorgroup** – Creates a schema for a monitor group.
  - **msglog** – Creates a schema for the system message log.
- *databaseType* – Specify **oracle**. This is the only supported value.
- *directory* – Specify a location to generate the DDL file (defaults to the local directory).
- *targetSchema* – Optional. This argument scopes the generated schema to a specific user, for example, **sphereDB**, **transactionDB**, or **measurementDB**.
- **-partition** or **-nopartition** – This flag is required if your specified *schemaType* is **performance** or **monitorgroup**. If your specified *schemaType* is any other value, this flag is not required and is ignored if you use it.

  If you are using Oracle Enterprise Edition, you can create a performance or monitorgroup schema that takes advantage of Oracle's partitioning feature by specifying the **-partition** flag. If you do not want to take advantage of this feature, or if your Oracle edition does not provide the partitioning feature, you must specify the **-nopartition** flag (if you are creating a performance or monitorgroup schema).

15.2.2 connect

Connect to a database using the user-specified connection information.

```
connect databaseType|filename
```

- *databaseType* – Specify **oracle**. This is the only supported value.
- *filename* – Specify a file output by the **saveConnection** command.

Use the **connect** command to enter database connection information and connect to the database. You must have the following information for the database to which you want to connect:

- driver name
- username
- password
- URL connection string

Once connected, you might issue the **saveConnection** command to save the connection information within a file. The next time you want to connect to the same database, you can provide the file name with the connect command. If you provide the database type, the utility automatically selects the corresponding default driver.

15.2.3 createSchema (or create)

Create the specified schema within the connected database.

```
createSchema schemaType -partition|-nopartition
```

- *schemaType* – Specify one of the following schema types:
  - **sphere** – Creates a schema for the sphere database (the sphereDB user).
  - **exm** – Creates a schema for the transaction database (the transactionDB user).
  - performance – Creates a schema for the performance database (the measurementDB user).
  - **monitorgroup** – Creates a schema for a monitor group.
  - **msglog** – Creates a schema for the system message log.
- **-partition** or **-nopartition** – This flag is required if your specified *schemaType* is **performance** or **monitorgroup**. If your specified *schemaType* is any other value, this flag is not required and is ignored if you use it.

  If you are using Oracle Enterprise Edition, you can create a performance or monitorgroup schema that takes advantage of Oracle's partitioning feature by specifying the **-partition** flag. If you do not want to take advantage of this feature, or if your Oracle edition does not provide the partitioning feature, you must specify the **-nopartition** flag (if you are creating a performance or monitorgroup schema).

15.2.4 close

Close a connection previously opened through the connect command.

```
close
```

15.2.5 exit

Exit the utility.

```
exit
```

15.2.6 help

Use the help command to view help for all commands, or enter a command name after help to receive help for a single command.

```
help | help command
```

Index

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W