Oracle
Primavera
P6 EPPM WebLogic Configuration Guide for On-Premises

Version 22
December 2022

ORACLE®

Contents

About This Guide

Scope

Oracle WebLogic server is an application server you can use to deploy your P6 EPPM applications. This guide describes how to manage and configure WebLogic settings to optimize the performance of your P6 EPPM applications:

- ▶ P6
- ▶ P6 Team Member
- ▶ P6 Professional Cloud Connect
- ▶ P6 EPPM Web Services, and P6 Integration API)

It also describes how to configure eventing for P6.

Audience

System and network administrators should use this guide.

Using This Guide

This guide assumes that you have already installed a supported version of WebLogic and a JDK. For more information, see the *Tested Configurations* document.

Prerequisites

Deleting Applications and Help from WebLogic

To delete your P6 EPPM and help from WebLogic:

1) Launch the WebLogic **Administration Console** using the following URL:

```
http://serverIP:listenport/console
```

2) In the **Domain Structure** pane, click **Deployments**.
3) In the **Control** tab, in the **Summary of Deployments** pane, complete the following:
   a. Select the following applications:
      - ▪ p6
      - ▪ p6procloudconnect
      - ▪ p6tm
      - ▪ p6ws
      - ▪ pr
      - ▪ PrimaveraAPI
      - ▪ p6help.war

- p6tmhelp.war
- prhelp.war

b. Click **Stop** and then click **Force Stop Now**.

c. In the **Force Stop Application Assistant** pane, click **Yes**.

d. For each application and help, ensure that the **State** column shows **Prepared**.

e. Select the applications and help that you stopped.

f. Click **Delete**.

g. In the **Delete Application Assistant** pane, click **Yes**.

> **Tip:** If you plan to use the same home directory for the new help or tutorial deployments, delete the existing WAR or EAR file from that directory to avoid conflict with the new files.

Uninstalling P6 EPPM Applications

If you are upgrading your P6 EPPM applications to the latest version, you must uninstall all earlier versions of your P6 EPPM applications. You can uninstall your P6 EPPM applications using Oracle Universal Installer.

To uninstall your P6 EPPM applications:

1) Delete your Applications from WebLogic. For more information, see *Deleting Applications and Help from WebLogic* (on page 7).

2) From the P6 EPPM installation folder, do one of the following:

   ‣ With Windows, navigate to the **win\Disk1\install** folder, and double-click **setup.exe**.

   ‣ With UNIX or Linux, enter the following commands:

```
cd <Operating System>/Disk1/install
chmod 755 runInstaller
chmod 755 unzip
./runInstaller
```

   where `<Operating System>` is linux, or solaris_64.

3) On the **Welcome** screen, click **Deinstall Products**.

4) In the **Inventory** dialog box, select the following:

   ‣ **P6 <Release_Level>**

   ‣ **P6 Professional Cloud Connect <Release_Level>**

   ‣ **P6 Integration API <Release_Level>**

   ‣ **P6 Web Services <Release_Level>**

   ‣ **P6 Team Member <Release_Level>**

   ‣ **P6 Progress Reporter <Release_Level>**

5) Click **Remove**.

6) On the **Confirmation** dialog box, click **Yes**.

7) Click **Close** then **Cancel** to exit the dialog box and Oracle Universal Installer.

Uninstalling Previous Versions of P6

You must uninstall any previous versions of P6 before upgrading to the latest version of P6.

Cautions:

- Before upgrading P6, you should upgrade the P6 EPPM database to the current version. See *P6 EPPM Upgrade and Configuration Guide* or *P6 EPPM Manual Upgrade Guide* for your database for details on how to upgrade your database and for information on potential impact areas to your environment. For the full list of tested configurations for P6, go to the \Documentation\<language>\Tested_Configurations folder of the P6 EPPM physical media or download.

▶ If you are a current Apache JackRabbit user and want to upgrade to the latest version of P6 EPPM, JackRabbit documents data will not migrate automatically. Refer to My Oracle Support's Knowledge Articles for information on manually migrating JackRabbit documents to Oracle Universal Content Management. Oracle recommends that you migrate the data before upgrading.

▶ If you are a current jBPM user and want to upgrade P6 EPPM, workflows and reviews data will not be available. You should close all workflows and reviews that are in progress before upgrading. You cannot migrate any of this data to the latest version of P6 EPPM.

▶ Starting with P6 EPPM R8, all recurring job service functions are hosted by P6. Due to this change, after upgrading to P6 EPPM R8 or later, you must configure Scheduled Services settings in the Primavera P6 Administrator to use this functionality. Also, if you're upgrading from version 7.0 or earlier, you must RESUMMARIZE ALL PROJECTS to accurately reflect your summary data. See the *P6 EPPM System Administration Guide* for information on Scheduled Services and configuring separate servers for job services.

Tips

▶ For security reasons, Oracle strongly recommends that on-premises users replace the default Admin Superuser (admin) immediately after a manual database installation or an upgrade from P6 version 7.0 and earlier. See information about the Admin Superuser in the *P6 EPPM Application Administration Guide*.

▶ P6 EPPM does not support Summary-Only projects. During the P6 EPPM database upgrade, existing Summary-Only projects convert to standard projects, but lose all summary data. You can import the summary project from Microsoft Project into the converted blank project, then summarize the data. See the *P6 Professional Help* or the *P6 Help*.

▶ During the upgrade to the latest version of P6 EPPM, some P6 Activity Views settings will reset. After the upgrade, use the Customize Detail Windows feature to modify the settings that should appear for each view. See the *P6 Help* for information on how to edit Activity Views.

▶ Starting with P6 EPPM R8, P6 saves filter definitions globally. Filters still work for Activity Views, but all standard filter assignments reset during the upgrade. Due to this change, views that had Standard Filters applied will show all activities after the upgrade. Reapply filters after the upgrade finishes. See the *P6 Help* for information on how to edit Activity Views.

Installing P6 EPPM Applications

For information about how to install the P6 EPPM applications, see the *P6 EPPM Installation and Configuration Guide*.

Help Directory Installation

You have two options for accessing Help:

Use the Default P6 Help URL (Recommended)

The default URL will take you to the version hosted by Oracle, and is the most up-to-date documentation.

The default help URLs are in the Primavera P6 Administrator.

> **Note**: You can choose not to use the default help URL, but you will have to ensure you deploy it in your application server and enter the server URL into the Primavera P6 Administrator. If any updates are made to the help, these will not be visible if you are using the p6help.war file.

Use the P6 Help WAR File

You can download the online help WAR files from the following URLs:

P6 Help: https://docs.oracle.com/cd/F25599_01/p6help/war/P6_Help.zip

P6 Team Member Help:
https://docs.oracle.com/cd/F25599_01/team_member/web/help/war/TM_Help.zip

> **Note**: For the Helps to launch, you must enter the server URLs in the Primavera P6 Administrator. See the *P6 EPPM System Administration Guide* for more information.

Tips

If you receive a "server not found" or a similar error when accessing the help in the application, it means one of the following:

▸ The application server cannot access the Internet.
▸ The help site is not available.

If the application server cannot access the Internet, check your Primavera P6 Administrator settings and ensure they are configured to the correct URL.

If the URL is configured correctly, but you still cannot access the application help, you will need to deploy the help using the WAR file.

Configuring WebLogic for P6 EPPM

This section details the basic configuration steps for P6 EPPM applications in a WebLogic environment when opting for an Admin Server and Managed Server deployment.

Oracle recommends you create a Managed Server deployment. Each managed server requires exclusive access to its own rotating logfiles. If your managed and clustered servers are on the same machine, you must create a different P6 home folder for each managed server and then copy the BREBootstrap file into each folder. You must edit the BREBootstrap file to ensure the logDir setting is set to the folder unique to each managed server. If your Admin, Managed, and clustered servers are all on the same machine, you do not need to copy the BREBootstrap file. This section assumes that you set up separate domains for your applications; however, you can create one domain and configure all P6 EPPM applications to run in this domain.

Although not required for the P6 EPPM server setup, WebLogic has additional settings that can be used to enhance your environment. For example, when using clustering, enabling the session replication setting will seamlessly transfer users to another server if a server shuts down unexpectedly.

See the official WebLogic documentation for more information on setting up WebLogic Windows Services and all available WebLogic configuration and deployment options.

Creating a Domain with Servers

A domain can include multiple WebLogic Server clusters and non-clustered WebLogic Server instances. A minimal domain can contain only one WebLogic Server instance, which functions as both an Administration Server and as a Managed server. This domain can be useful while developing applications, but it is not recommended for use in a production environment. Although the scope and purpose of a domain can vary significantly, most WebLogic Server domains contain the components described in this section.

By default, the WebLogic domain created by the P6 EPPM Configuration Wizard creates a domain with an Administration Server (AdminServer) and a Managed Server for each application that you deploy. If you are installing P6 EPPM for the first time, Oracle recommends that you create a domain for each application that you plan to deploy. For example:

- P6
- P6 Team Member
- P6 Professional Cloud Connect
- P6 EPPM Web Services
- P6 Integration API

> **Note:** If you plan to enable Oracle Web Services Manager Authentication, skip the following procedure and refer to *Creating a Domain for Oracle Web Services Manager Authentication* (on page 14) for instructions on creating a domain.

To create a domain:

To create a WebLogic Domain:

1) Run the WebLogic **Configuration Wizard**.
2) In the **Create Domain** window:
   a. Select **Create a new WebLogic domain**.
   b. In the **Domain Location** field, enter the location that you want to create the domain.
   c. Click **Next**.
3) In the **Templates** window, select the templates that you want to apply to the domain and click **Next**.
4) In the **Administrator Account** window:
   a. Enter the user name and password information.
   b. Click **Next**.
5) In the **Domain Mode** window:
   a. Select **Production Mode**.
   b. Select a JDK.
   c. Click **Next**.
6) In the Advanced **Configuration** window:
   a. Select the **Administration Server**, **Managed Servers**, and **Managed Servers, Clusters and Coherence**.
   b. Click **Next**.
7) In the **Administration Server** window:
   a. (Optional) Select **Enable SSL** and set the SSL listen port if you are enabling Secure Sockets Layer communication.
   b. Click **Next**.
      See https://docs.oracle.com/middleware/1221/core/ASADM/sslconfig.htm for more details on setting SSL for WebLogic.
8) In the **Node Manager** window:
   a. Select a **Node Manager Type**.
   b. In the **Username** and **Password** fields, enter the credentials for the Node Manager.
   c. Click **Next**.
9) In the **Managed Servers** window:
   a. Click **Add**.
   b. Enter the **Name** and select the **Listen address** information.
   c. (Optional) Select the **SSL enabled** option and set the SSL listen port.
   d. (Optional) Add or delete other managed servers.
   e. Click **Next**.
10) In the **Clusters** window:

> **Note**: Do not add clusters if you are not using multiple WebLogic server instances for scalability.

   a. (Optional) Click **Add**.

    b. (Optional) Enter the name of the cluster.

    c. (Optional) Enter the following information: **Cluster messaging mode**, **Multicast address**, **Multicast port**, and **Cluster address**.

    d. (Optional) Add or delete other configured clusters.

    e. Click **Next**.

> **Note:** For information on setting up clusters, use Oracle's WebLogic Server documentation: https://docs.oracle.com/middleware/1221/wls/CLUST/setup.htm.

11) (Optional) In the **Assign Servers to Clusters window**, assign a server to a cluster and then click **Next**.

> **Note:** Select the Cluster in the right pane, then select the Server in the left pane. Assign the server to the cluster by clicking the right arrow button.

12) In the **Machines** window, select **Machine** (with Windows) and then complete the following:

    a. Click **Add**.

    b. Enter a machine name.

    c. (Optional) Select the **Node manager listen address** from the list.

> **Note:** If you specify an address for a machine that hosts the Administration Server and you need to access the WebLogic Server Node Manager, you must disable the host name verification.

    d. (Optional) Enter the **Node manager listen port**.

    e. (Optional) Add or delete configured machines.

    f. (Optional) Click **Next**.

13) In the **Machines** window, select **Unix Machine** (with UNIX) and the complete the following:

    a. Click **Add**.

    b. Enter a machine name.

    c. (Optional) Select the **Post bind GID enabled** option to enable a server running on this machine to bind to a UNIX group ID (GID) after it finishes all privileged startup actions.

    d. (Optional) Enter the **Post bind GID** where a server on this machine will run after it finishes all privileged startup actions. If you do not enter a GID, the server will continue to run under the group where it was started. For this setting to work, you must select the **Post bind GID enabled** option.

    e. (Optional) Select the **Post bind UID enabled** option to enable a server running on this machine to bind to a UNIX user ID (UID) after it finishes all privileged startup actions.

    f. (Optional) Enter **Post bind UID** where a server on this machine will run after it finishes all privileged startup actions. If you do not enter a UID, the server will continue to run under the account where it was started. For this setting to work, you must select the **Post bind UID enabled** option.

    g. (Optional) Add or delete configured machines.

h. Click **Next**.

> **Notes:**
>
> - You might want to create machine definitions for the following situations: (1) The Administration Server uses the machine definition (along with the Node Manager application) to start remote servers. (2) WebLogic Server uses configured machine names when determining the server in a cluster that can handle certain tasks, such as HTTP session replication. The WebLogic Server then delegates those tasks to the identified server.
> - You must configure machines for each product installation that runs a Node Manager process. The machine configuration must include values for the listen address and port number parameters.

14) In the **Assign Servers to Machines** window:

> **Note:** A machine is a physical server that will host a WebLogic managed server. Depending on your resource needs and data load, the machines may be in the same physical server where the WebLogic Admin Server is installed or in separate physical servers.

   a. In the **Machine** list, select the machine where you want to assign a WebLogic Server instance.
   b. Assign the managed and administrative servers that you created to the selected machine.
   c. Review the machine assignments.
   d. Click **Next.**

15) In the **Configuration Summary** window, click **Create**.

   If given the option, you can click **Done** now. Otherwise, continue to the next step.

16) If you are using Windows, in the **Creating Domain** window:
   a. Select **Start Admin Server**.
   b. Click **Done**.

17) When prompted, enter the administrator user name and password that you entered above.

18) Repeat this procedure for each new domain.

Creating a Domain for Oracle Web Services Manager Authentication

To create a domain and administration server for P6 EPPM Web Services:

1) Open the **Fusion Middleware Configuration Wizard** by completing the following:
   a. Navigate to `<Oracle_Home>\oracle_common\common\bin`.
   b. Run **config.cmd** (if Windows) or **config.sh** (if UNIX).
2) On the **Configuration Type** screen, complete the following:
   a. Select **Create a new domain**.
   b. Browse to the location at which the new domain will be created.
3) On the **Template** screen, complete the following:

   a. Select **Create Domain Using Product Templates:**

   b. In the **Template Categories** menu, select **All Templates**.

   c. Select the following from the Available Templates window:

- **Basic WebLogic Server Domain**
- **Oracle Enterprise Manager**
- **Oracle WSM Policy Manager**
- **Oracle JRF**
- **WebLogic Coherence Cluster Extension**

4) On the **Administrator Account** screen, complete the following:

   a. In the **Name** field, enter a name for the domain administrator.

   b. In the **Password** field, enter a password for the domain administrator.

   c. In the **Confirm Password** field, enter the password that you entered in the **Password** field.

5) On the **Domain Mode and JDK** screen, complete the following:

   a. Under **Domain Mode**, select one of the following:

- **Development**
- **Production**

   b. Under **JDK**, select a supported JDK. For supported JDKs, see *Tested Configurations*.

6) On the **Database Configuration Type** screen, complete the following:

   a. Under **Specify AutoConfiguration Options Using:**, select **RCU Data**.

   b. Fill in each field, using the connection information that you specified for the **Service Table** component in the Repository Creation Utility (RCU).

   c. Click **Get RCU Configuration** to retrieve the schema information.

   d. In the Schema Password field, enter the password for the schema owner.

7) On the **JDBC Components Schema** screen, complete the following:

   a. Fill in each field, using the connection information that you specified for the **Service Table** component in the Repository Creation Utility (RCU).

   b. In the **Schema Password** fields of the table, enter the password for each component schema.

   c. Select the schema in the table.

   d. Click **Test Selected Connections** to ensure that the credentials for each schema are valid.

8) On the **Advanced Configuration** screen, select **Administration Server**.

9) On the **Administration Server** screen, complete the following:

   a. In the **Server Name** field, enter a name for the administration server.

   b. In the **Listen Address** field menu, select the appropriate listen address.

   c. In the **Listen Port** field, enter the listen port number.

   d. In the **Server Groups** menu, select the following:

- **WSM-CACHE-SVR**
- **WSMPM-MAN-SVR**

10) On the **Configuration Summary** screen, click **Create**.

11) After the domain has been created, complete the tasks in the ***Authentication Using Oracle Web Services Manager*** (on page 30) section.

Adding Applications and Help to Managed Servers

After creating your WebLogic domains, you must add your P6 EPPM applications to your WebLogic domains.

To add P6 EPPM applications to WebLogic:

1) In the **Change Center** pane, click **Lock & Edit**.

2) In the **Domain Structure** pane, click **Deployments**.

3) In the **Configuration** tab on the **Summary of Deployments** page, click **Install**.

4) In the **Install Application Assistant** window, complete the following:

   a. Select the EAR or WAR file for the application or Help that you want to deploy. For each application and Help, you can find the EAR and WAR files at the following locations:

   For P6, `<P6_EPPM_Home>\p6\p6.ear`

   For P6 Team Member, `<P6_EPPM_Home>\tmws\p6tm.ear`

   For P6 Professional Cloud Connect, `<P6_EPPM_Home>\p6procloudconnect\p6procloudconnect.war`

   For P6 Integration API, `<P6_EPPM_Home>\api\applications\PrimaveraAPI.war`

   For P6 EPPM Web Services, `<P6_EPPM_Home>\ws\server\p6ws.ear`

   For P6 and P6 Team Member helps:

   1. https://docs.oracle.com/en/industries/construction-engineering/primavera-p6-project/index.html.

   2. Click the **22 Documentation Library** link.

   3. Navigate to the **Using** page.

   4. Click the **Downloadable P6_Help** link and the **Downloadable TM_Help** link.

   > **Note:** For the Help to launch, you must enter the server URL in the Primavera P6 Administrator. See the *P6 EPPM System Administration Guide* for more information. Oracle recommends that you do not deploy the P6 or P6 Team Member Help because the default URL will take you to the version hosted by Oracle. The hosted version contains the most up-to-date documentation.

   b. Click **Next**.

5) On the **Install Application Assistant** page:

   a. Select **Install this deployment as an application**.

   b. Click **Next**.

6) On the **Install Application Assistant** page:

   a. Select the managed server.

   b. Click **Next**.

7) In the **Install Application Assistant** pane, click **Next** to accept the default options.

8) Review the configuration settings you have chosen and then click **Finish** to complete the installation.

9) Repeat this procedure for each application on each domain.

10) In the **Change Center** pane, click **Activate Changes**.

Setting Arguments

Use these instructions for each P6 EPPM application that you deploy in a managed or clustered server. To continue configuring WebLogic for P6, you must add the following arguments to your managed or clustered server in the WebLogic Administration console.

To set WebLogic Server arguments:

1) In the **Change Center** pane of the Administration Console, click **Lock & Edit**.

2) On the **Domain Structure** pane, expand **Environment** and then click **Servers**.

3) In the **Summary of Servers** page, click the managed server on which you deployed your application or Help.

4) In **Configuration** tab, on the **Settings for <managed server name>** page, navigate to the **Server Start** tab.

5) In the **Arguments** field, enter the following arguments as necessary in the order that they appear:

   ▸ `-server`

   ▸ `-Dprimavera.bootstrap.home=<P6_EPPM_BootStrap_Home>`

   where `<P6_EPPM_BootStrap_Home>` is the home directory of `BREBootStrap.xml`. If you do not have a `BREBootStrap.xml` file for your P6 EPPM implementation, you can create one by creating a P6 EPPM configuration using the **Database Configuration** wizard. For more information about using the **Database Configuration** wizard, see the *P6 EPPM Database Administration Guide*.

   ▸ `-Ddatabase.instance=`*db_id*

   Include this argument to establish a default database connection for P6 Team Member, P6 EPPM Web Services, and P6 Integration API if you use more than one database instance. You do not need to set this argument for P6.

   ▸ `-Dweblogic.webservice.i18n.charset=utf-8`

   (Optional) Include this argument to allow WebLogic to process UTF-8 characters.

   ▸ `-Djava.awt.headless=true`

   (With UNIX only) Include this argument if you are deploying P6 EPPM on a non-graphical UNIX environment. This argument sets the server's JVM to bypass server-side rendering optimizations. If you do not add the argument, you may not be able to see graphics on pages and the server logs will report NoClassDefFoundError errors.

   ▸ `-Djavax.xml.stream.XMLInputFactory=com.ctc.wstx.stax.WstxInputFactory`

   (Optional) Include this argument for improved performance when starting the WebLogic domain.

   ▸ `-XX:MaxPermSize=`

(Optional) Include this argument to set the new generation heap size. The MaxPermSize setting should be set to at least 256m. You can modify memory settings to maximize performance.

▶ `-XX:NewSize=`

(Optional) Include this argument to set the maximum size of the new generation heap size. As a general rule, set -XX:NewSize to be one-fourth the size of the heap size. Increase the value of this option for larger numbers of short-lived objects. Be sure to increase the new generation as you increase the number of processors. Memory allocation can be parallel, but garbage collection is not parallel.

▶ `-XX:MaxNewSize=`

(Optional) Include this argument to set the maximum size of the new generation heap size.

▶ `-XX:SurvivorRatio=`

(Optional) Include this argument to set new heap size ratios.

The New generation area is divided into three sub-areas: Eden, and two survivor spaces that are equal in size. Configure the ratio of the Eden/survivor space size. Try setting this value to 8, and then monitor your garbage collection.

▶ `-Xms -Xmx`

(Optional) Include this argument to set the initial and maximum heap size.

As a general rule, set the initial heap size (`-Xms`) equal to the maximum heap size (`-Xmx`) to minimize garbage collection.

▶ `-XX:+UseParallelGC -XX:+UseParallelOldGC -XX:GCTimeRatio=19`

(Optional) Include this argument if you are running Publication Services on a larger database.

▶ `-Xloggc:<Log_Path>`

Include this argument to collect garbage logs if you have memory issues on JVM.

For example:

(With Windows)

```
-server -Dprimavera.bootstrap.home=<P6_EPPM_Home>
-Djavax.xml.stream.XMLInputFactory=com.ctc.wstx.stax.WstxInputFactory
-XX:MaxPermSize=256m -XX:NewSize=256m -XX:MaxNewSize=256m -XX:SurvivorRatio=8
-Xms2048m -Xmx2048m -XX:+UseParallelGC -XX:+UseParallelOldGC -XX:GCTimeRatio=19
-Xloggc:<Log_Path>
```

(With UNIX)

```
-server -Dprimavera.bootstrap.home=<P6_EPPM_Home> -Djava.awt.headless=true
-Djavax.xml.stream.XMLInputFactory=com.ctc.wstx.stax.WstxInputFactory
-XX:MaxPermSize=256m -XX:NewSize=256m -XX:MaxNewSize=256m -XX:SurvivorRatio=8
-Xms2048m -Xmx2048m -XX:+UseParallelGC -XX:+UseParallelOldGC -XX:GCTimeRatio=19
-Xloggc:<path to the log file>
```

Note: If you use additional arguments that are not listed here, be sure to list them in the correct sequence for your environment.

6) Click **Save**.

7) Add arguments to the managed servers on which applications or Help are deployed.

8) In the **Change Center** pane, click **Activate Changes**.

9) Restart your managed servers.

Modifying setDomainEnv for Applications Deployed on Administration Servers

Use these instructions for each P6 EPPM application that you deploy on an administration server.

To edit the setDomainEnv file:

1) Navigate to `<WebLogic_Home>/user_projects/<Domain_Home>/bin` and make a backup copy of the `setDomainEnv.cmd` (on Windows) `setDomainEnv.sh` (for UNIX).

2) Edit the `setDomainEnv` file.

3) After `%WL_HOME%/common/bin/commEnv.cmd"` (on Windows) `${WL_HOME}/common/bin/commEnv.sh` (on UNIX), enter the following arguments as necessary in the order that they appear:

   ▸ `-server`

   ▸ `-Dprimavera.bootstrap.home=<P6_EPPM_BootStrap_Home>`

   where `<P6_EPPM_BootStrap_Home>` is the home directory of `BREBootStrap.xml`. If you do not have a `BREBootStrap.xml` file for your P6 EPPM implementation, you can create one by creating a P6 EPPM configuration using the **Database Configuration** wizard. For more information about using the **Database Configuration** wizard, see the *P6 EPPM Database Administration Guide*.

   ▸ `-Ddatabase.instance=`*db_id*

   Include this argument to establish a default database connection for P6 Team Member, P6 EPPM Web Services, and P6 Integration API if you use more than one database instance. You do not need to set this argument for P6.

   ▸ `-Dweblogic.webservice.i18n.charset=utf-8`

   (Optional) Include this argument to allow WebLogic to process UTF-8 characters.

   ▸ `-Djava.awt.headless=true`

   (With UNIX only) Include this argument if you are deploying P6 EPPM on a non-graphical UNIX environment. This argument sets the server's JVM to bypass server-side rendering optimizations. If you do not add the argument, you may not be able to see graphics on pages and the server logs will report NoClassDefFoundError errors.

   ▸ `-Djavax.xml.stream.XMLInputFactory=com.ctc.wstx.stax.WstxInputFactory`

   (Optional) Include this argument for improved performance when starting the WebLogic domain.

   ▸ `-XX:MaxPermSize=`

   (Optional) Include this argument to set the new generation heap size. The MaxPermSize setting should be set to at least 256m. You can modify memory settings to maximize performance.

- ▸ `-XX:NewSize=`

  (Optional) Include this argument to set the maximum size of the new generation heap size. As a general rule, set -XX:NewSize to be one-fourth the size of the heap size. Increase the value of this option for larger numbers of short-lived objects. Be sure to increase the new generation as you increase the number of processors. Memory allocation can be parallel, but garbage collection is not parallel.

- ▸ `-XX:MaxNewSize=`

  (Optional) Include this argument to set the maximum size of the new generation heap size.

- ▸ `-XX:SurvivorRatio=`

  (Optional) Include this argument to set new heap size ratios.

  The New generation area is divided into three sub-areas: Eden, and two survivor spaces that are equal in size. Configure the ratio of the Eden/survivor space size. Try setting this value to 8, and then monitor your garbage collection.

- ▸ `-Xms -Xmx`

  (Optional) Include this argument to set the initial and maximum heap size.

  As a general rule, set the initial heap size (`-Xms`) equal to the maximum heap size (`-Xmx`) to minimize garbage collection.

- ▸ `-XX:+UseParallelGC -XX:+UseParallelOldGC -XX:GCTimeRatio=19`

  (Optional) Include this argument if you are running Publication Services on a larger database.

- ▸ `-Xloggc:<Log_Path>`

  Include this argument to collect garbage logs if you have memory issues on JVM.

  For example:

  (With Windows)

```
-server -Dprimavera.bootstrap.home=<P6_EPPM_Home>
-Djavax.xml.stream.XMLInputFactory=com.ctc.wstx.stax.WstxInputFactory
-XX:MaxPermSize=256m -XX:NewSize=256m -XX:MaxNewSize=256m -XX:SurvivorRatio=8
-Xms2048m -Xmx2048m -XX:+UseParallelGC -XX:+UseParallelOldGC -XX:GCTimeRatio=19
-Xloggc:<Log_Path>
```

  (With UNIX)

```
-server -Dprimavera.bootstrap.home=<P6_EPPM_Home> -Djava.awt.headless=true
-Djavax.xml.stream.XMLInputFactory=com.ctc.wstx.stax.WstxInputFactory
-XX:MaxPermSize=256m -XX:NewSize=256m -XX:MaxNewSize=256m -XX:SurvivorRatio=8
-Xms2048m -Xmx2048m -XX:+UseParallelGC -XX:+UseParallelOldGC -XX:GCTimeRatio=19
-Xloggc:<path to the log file>
```

  Note: If you use additional arguments that are not listed here, be sure to list them in the correct sequence for your environment.

4) **Save** and **Close** the file.

5) Restart the admin server.

Product-Specific Configurations

Finalizing P6 mobile

Use the steps in the related topics to begin using P6 mobile.

Downloading P6 mobile Apps

To download the P6 mobile apps, do one of the following:

▶ If you are using an iOS device, go to the Apple App Store to download the mobile application.
▶ If you are using an Android device, go to the Google Play Store to download the mobile application.

> **Note**: If you are in a region without access to the Google Play Store, or your organization is using a Content Security Service or Mobile Device Management solution and requires that users do not download P6 mobile from the Apple App Store or Google Play Store, submit a Service Request in My Oracle Support to request versions of the P6 mobile apps for those scenarios.

Configuring Login and Authentication Settings to Use P6 for iOS

Follow these steps to start the app for the first time. When you return to the app after working in other apps, the last page you were on will appear. Once you configure these settings, you won't need to perform these steps again unless your SSO cookies expire. If your cookies expire, you will need to enter your user name and password again.

> **Note:** You may need to activate your device's VPN feature to access your company's deployment of P6. Contact your administrator for more information.

To start the app:

1) On your device's **Home** screen, tap 🔲 **P6**.
2) On the **Welcome to P6 Team Member** page, slide the **Single Sign On** (SSO) switch to either **ON** or **OFF**.

> **Note**: Team Member Web Services supports LDAP, Native, or SSO mode. Your administrator will select the authentication mode when they configure P6.

3) If you turn SSO on:
   a. Tap the **URL** field and enter the URL to your server (for example, http://*server.port*/p6tmws).

   > **Note**: You will need to specify the server name and port number in the

URL.

b. Tap **Authenticate**.

c. Enter your SSO username and password.

4) If you turn SSO off:

a. Tap the **URL** field and enter the URL to your server (for example, https://*server*.*port*/p6tmws).

b. Enter your P6 username.

c. Enter your P6 password.

d. Tap **Sign In**.

> **Notes**:
>
> - You will need to configure Email Statusing Service separately. See the *P6 EPPM System Administration Guide*.
> - P6 for iOS supports SSL (https) or HTTP only when it has a certificate signed by a trusted authority.
> - P6 for Android requires SSL (https) when you are not using SSO authentication. If you are using SSO authentication, you can use HTTP or HTTPs protocols. HTTPS requires a valid certificate from an Android trusted certifying authority.

Tips

▸ To access server information in the app—which includes the SSO setting, the URL to access the server, and your user name—navigate to the **app menu**, and then tap **Settings**.

▸ For more information on the different types of authentication modes (Single Sign-On, Native, or LDAP), see "Authentication Modes in P6 EPPM" in the *P6 EPPM System Administration Guide*.

▸ You can modify the settings for the app from the **Settings** page on your iPhone. See the *P6 Team Member User's Guide* for more information.

Configuring Login and Authentication Settings to Use P6 for Android

Follow these steps to start the app for the first time. When you return to the app after working in other apps, the last page you were on will appear. Once you configure these settings, you won't need to perform these steps again unless your SSO cookies expire. If your cookies expire, you will need to enter your user name and password again.

> **Note:** You may need to activate your device's VPN feature to access your company's deployment of P6. Contact your administrator for more information.
>
> P6 for Android only accepts SSL certificates from a certifying authority and must use an HTTPS connection.

To start the app:

1) On your device's **Home** screen, tap **P6**.

2) On the **Welcome to P6 Team Member** page, slide the **Single Sign On** (SSO) switch to either **ON** or **OFF**.

> **Note**: Team Member Web Services supports LDAP, Native, or SSO mode. Your administrator will select the authentication mode when they configure P6.

3) If you turn SSO on:
   a. Tap the **URL** field and enter the URL to your server (for example, http://*server*:*port*/p6tmws).

   > **Note**: You will need to specify the server name and port number in the URL.

   b. Tap **Authenticate**.
   c. Enter your SSO username and password.

4) If you turn SSO off:
   a. Tap the **URL** field and enter the URL to your server (for example, https://*server*:*port*/p6tmws).
   b. Enter your P6 username.
   c. Enter your P6 password.
   d. Tap **Sign In**.

   > **Notes**:
   >
   > - You will need to configure Email Statusing Service separately. See the *P6 EPPM System Administration Guide*.
   > - P6 for iOS supports SSL (https) or HTTP only when it has a certificate signed by a trusted authority.
   > - P6 for Android requires SSL (https) when you are not using SSO authentication. If you are using SSO authentication, you can use HTTP or HTTPs protocols. HTTPS requires a valid certificate from an Android trusted certifying authority.

Tips

▶ To access server information in the app—which includes the SSO setting, the URL to access the server, and your user name—navigate to the **app menu**, and then tap **Settings**.

▶ For more information on the different types of authentication modes (Single Sign-On, Native, or LDAP), see "Authentication Modes in P6 EPPM" in the *P6 EPPM System Administration Guide*.

▶ You can modify the settings for the app from the **Settings** button in the app. See the *P6 Team Member User's Guide* for more information.

Finalizing P6 Team Member Web

Use the steps in the related topics to begin using P6 Team Member Web.

Configuring P6 Team Member Web Settings

To configure the settings for P6 Team Member Web, you must have a P6 Team Member Admin Configuration. See *Installing and Configuring P6 EPPM* for more information on how to create this configuration during a new installation of P6 Team Member Web or P6 EPPM.

To configure the settings for P6 Team Member Web:

1) Launch **Primavera P6 Administrator**.
2) Navigate to your P6 Team Member configuration, and then expand it.
3) Expand **Application**:
   a. Set the Help server URL to one of the following:
      ‣ `http://localhost:`*listenport*`/`*ContextRoot*

        where *localhost* is your server's host name, *listenport* is your P6 Team Member port number, and *ContextRoot* is the root for your Help.
      ‣ https://docs.oracle.com/cd/E90746_01/team_member/web/help/

        Note: Oracle recommends that you use the second URL option because it ensures that you always access the latest version of the P6 Team Member Help, including critical corrections and enhancements.

4) Expand **Team Member**:
   a. Expand **Connection Pool** if you want to alter the following default values:
      ▪ **Initial connection pool size**: 10
      ▪ **Maximum active database connections**: 150
      ▪ **Maximum idle database connections**: -0
      ▪ **Minimum idle database connections**: 10
   b. Expand **Logging** if you want to alter the following default values:
      ▪ **Enable**: enabled
      ▪ **Severity Level**: Error
      ▪ **Log File Name**: p6tmweb.log

        Note: The **primavera.bootstrap.home** property determines the location of the log file.

Configuring Email Statusing Service for P6

To configure Email Statusing Service for P6:

1) From the P6 Team Member home directory (for example, C:\P6EPPM_1\tmws), extract the files from the **p6tm-email-service.zip** to any location on the application server. (That location will be referred to as EMAIL_HOME).
2) Make a backup copy of the **config.properties** file before you edit it.
3) Edit the **config.properties** file to configure the settings for the email application.

> **Note:** When you are finished editing the config.properties file, move it to a secure location.

4) In the **P6 Configuration** section, locate and specify the following:

   a. In the **p6.admin.user =** line, specify a P6 user who is assigned to at least one module access and has resource access.

   b. In the **p6.notify.user =** line, specify the email addresses, separated by commas, of users who can send Notify Task lists to other users. As you see the *P6 Help* for more information on sending team members their tasks.

   c. In the **p6.max.tmusers =** line, specify the maximum number of email recipients that can receive notification emails at one time. You should keep the number below 500 to avoid performance problems.

   d. In the **p6.bootstrap =** line, specify the location of the BREBootstrap.xml file.

      For example, the location could be one of the following: C:/P6EPPM_1/p6, C:/EMAIL_HOME, /usr/P6EPPM_1/p6, or /usr/EMAIL_HOME or you can use the following: C:\\P6EPPM_1\p6, C:\\EMAIL_HOME.

   e. In the **mail.read.schedule =** line, specify how often your email should process new messages. Use cron expressions for your formatting. For example:

      `0 */5 * * * ?`

      will process new messages every five minutes.

      > **Note:** For more information on cron expressions, see http://www.quartz-scheduler.org/documentation/quartz-2.2.x/tutorials/tutorial-lesson-06.html.

5) In the **Outgoing mail (SMTP Settings)** section, locate and specify the following:

   a. In the **send.host =** line, specify the SMTP server hostname or IP address that you will use to send emails.

   b. In the **send.port =** line, specify the port number that connects to the SMTP server.

   c. In the **send.ssl =** line, specify **true** to use SSL or **false** if you will not use SSL.

      > **Note:** Oracle recommends that you always use SSL in a production environment for secure communications.

   d. In the **send.starttls =** line, specify **true** to use TLS or **false** if you will not use TLS.

   e. In the **send.email =** line, specify the email address or User Principal Name (UPN) where team members will send their status update requests.

   f. In the **send.replyto =** line, specify an email address for team members to respond to when they receive emails. If this field is left blank, the reply address will be the **send.email** address you specified above.

   g. In the **send.password =** line, specify the password for the email address or UPN you just specified.

      > **Notes:**
      > - If you do not specify a password now, the email application prompts

you to specify one when it runs.

- You can also specify a password by running the following parameter when you run the 'run.bat' or 'run.sh' file:
  `-Dsend.password=YOUR_PASSWORD`
  where *Your_Password* is the password for your status email address.
  For example:
  In Windows: `./run.bat -Dsend.password=`*YOUR_PASSWORD*
  In UNIX: `./run.sh -Dsend.password=`*YOUR_PASSWORD*

- Oracle recommends using the option in the **config.properties** file and then moving that file to a secure location when you are finished modifying it.

- If you want to set up anonymous authentication for an SMTP server, comment this field out.

6) In the **Incoming Mail (IMAP Settings)** section, locate and specify the following:

a. In the **imap.host =** line, specify the IMAP protocol host to read emails.

b. In the **imap.port =** line, specify the IMAP port number.

c. In the **imap.ssl=** line, specify **true** to use SSL or **false** if you will not use SSL.

> **Note**: Oracle recommends that you always use SSL in a production environment for secure communications.

d. In the **imap.username =** line, specify the IMAP user name (which will be your email address) or User Principal Name (UPN).

e. In the **imap.password =** line, specify the password for the user name you just specified.

> **Notes**:
>
> - If you do not specify a password now, the email application will ask you to specify one when it runs.
>
> - You can also specify a password by running the following parameter when you run the 'run.bat' or 'run.sh' file:
>   -Dimap.password=*YOUR_PASSWORD*
>   where *Your_Password* is the password for your status email address.
>   For example:
>   In Windows: ./run.bat -Dimap.password=*YOUR_PASSWORD*
>   In UNIX: ./run.sh -Dimap.password=*YOUR_PASSWORD*
>
> - Oracle recommends using the option in the **config.properties** file and then moving that file to a secure location when you are finished modifying it.
>
> - If you want to setup anonymous authentication for an SMTP server, comment this field out.

f. In the **imap.search.folder =** line, specify the folder to use for processing incoming emails. For example:

```
imap.search.folder = inbox
```

> **Note**: The email application will use this folder to search for incoming mail. Oracle recommends you use the inbox for this folder. If you select another folder, you must ensure that all emails get filtered there, or the email application will not be able to detect the messages.

7) In the **IMAP Folder configuration** section, locate and specify the following:

a. In the **imap.success.folder =** line, specify the folder where messages that process successfully will go. For example:

```
imap.success.folder = P6Success
```

> **Note:** The email application will automatically create this folder if you do not create it now.

b. In the **imap.failure.folder =** line, specify the folder where messages that failed will go. For example:

```
imap.failure.folder = P6Fail
```

> **Note:** The email application will automatically create this folder if you do not create it now.

c. In the **imap.temp.folder =** line, specify the folder where messages that are being processed will go. For example:

```
imap.temp.folder = P6Temp
```

> **Note:** The email application will automatically create this folder if you do not create it now.

8) In the **Task update options** section, locate and specify the following:

a. In the **reply.label.cleanup** field, enter true if you want to remove HTML tags attached to the task field labels. Enter false if you don't want to remove the tags.

b. In the **reply.label.tag.search** field, enter the regular expression to remove unexpected HTML tags during reply message parsing.

> **Note**: See information on "regular expression" for more details on this field. For example, see https://developer.mozilla.org/en-US/docs/JavaScript/Guide/Regular_Expressions#Writing_a_Regular_Expression_Pattern.

c. In the **update.debug** field, enter true if you want to output the contents of the email message as it is parsed for updates. Enter false if you don't want to output the contents of the email message as it is parsed for updates. You should only enter true when you are troubleshooting issues with updating tasks.

9) In the **Message Format options** section, locate and specify the following:

a. In the **update.task.search.term =** line, specify the text that the email application should search for in the subject line of incoming messages. For example:

```
update.task.search.term = Your Requested P6 Tasks
```

b. In the **notify.update.task.search.term =** line, specify the text that the email application should search for in the subject line of incoming messages that originated from a Notify message. For example:

```
update.task.search.term = Your P6 Tasks
```

10) If the EMAIL_HOME is on a different server from your P6 home, copy the BREBootstrap.xml from your P6 home to your EMAIL_HOME.

If the EMAIL_HOME is on the same server as your P6 home, you will use the BREBootstrap.xml that is in your P6 home folder.

11) Run one of the following:

‣ In Windows, launch the **run.bat** file.

‣ In UNIX, launch the **run.sh** file.

When you launch the run file, you may have to complete the following steps depending on how you configured your Email Statusing Service:

a. If the config.properties file and the run file are in different folders, you will need to specify the config.properties location with the -Dconfig.file parameter.

▪ In Windows (add a space between run.bat and -Dconfig):

```
run.bat
-Dconfig.file=C:\home\auser\somelocation\config.properties
```

▪ In UNIX (add a space between run.sh and -Dconfig):

```
./run.sh
-Dconfig.file=/home/auser/somelocation/config.properties
```

b. If you didn't specify the **send.password=** or **imap.password=**, you can specify them now. If you do not specify them now, the email application will prompt you to do so when it runs.

▪ In Windows:

```
run.bat -Dsend.password=YOUR_PASSWORD
```

▪ In UNIX:

```
./run.sh -Dsend.password=YOUR_PASSWORD
```

▪ In Windows:

```
run.bat -Dimap.password=YOUR_PASSWORD
```

▪ In UNIX:

```
./run.sh -Dimap.password=YOUR_PASSWORD
```

where *Your_Password* is the password for your status email address

c. If your bootstrap file contains more than one database, you need to clarify which database you are using when you launch the run file.

▪ In Windows:

```
run.bat -Ddatabase.instance=db_id
```

where *db_id* is your database instance (for example, -Ddatabase.instance=2).

▪ In UNIX:

```
./run.sh -Ddatabase.instance=db_id
```

where *db_id* is your database instance (for example, -Ddatabase.instance=2).

Once the run file finishes, you can verify that the email status application works. See **Verifying Email Statusing Service** (on page 29).

> **Note**: Anonymous SMTP authentication is supported.

Verifying Email Statusing Service

Once you have installed the email status application, you can send an email to verify that the server is setup properly.

To verify your Email Statusing Service setup:

1) Enter an email address that P6 recognizes in the 'From' line.
2) Enter the email address you specified in **Configuring Email Statusing Service for P6** (on page 24) in the 'To' line.
3) Enter **Ping** in the subject line.
4) Send the message.

If the message is successful, you will receive a message containing the status of the email service. The message will tell you how many users have a blank email address in the database.

If the message failed, you will not receive a message. The failed message will appear in the failure folder that you specified in the **config.properties** file. See **Configuring Email Statusing Service for P6** (on page 24).

Deploying P6 EPPM Web Services in WebLogic

Follow the instructions to deploy P6 Integration API into the WebLogic domain.

> **Notes**:
>
> - Consult WebLogic's documentation for additional methods of deploying a Web application, such as using a Managed Server or Clustering.
> - P6 EPPM Web Services supports the HTTPS communication protocol. See the application server documentation for instructions on configuring HTTPS on your application server.

Creating an Endorsed Folder

If you do not have a JDK Home folder specified, you must create an endorsed folder to deploy P6 EPPM Web Services in WebLogic.

To create an endorsed folder:

1) Locate the folder that contains the JDK you used when you created the WebLogic domain. This folder is known as <JDK_HOME>.

   If it does not already exist, create an endorsed folder: **<JDK_HOME>/jre/lib/endorsed**.
2) In the **<WS_INSTALL_HOME>/endorsed** folder, locate the **saaj-impl.jar** file.

3) Copy the saaj-impl.jar file to the endorsed folder you created
   (**<JDK_HOME>/jre/lib/endorsed**).

4) Restart the WebLogic domain.

Adding a JVM Option to the startWeblogic Script

Before starting P6 EPPM Web Services, add the following JVM option to the startWeblogic
startup script:

1) Make a backup copy of the **startWebLogic** file in case you need to undo any changes.

   ▸ In Windows, the file is named "startWebLogic.cmd" and is located in:
     weblogic_home\user_projects\domains*your_domain*\bin\

   ▸ In Unix, the file is named "startWebLogic.sh" and is located in:
     weblogic_home/user_projects/domains/*your_domain*/bin/

2) Right-click the startWebLogic file and select Edit.

3) Locate the line that begins with "set JAVA_OPTIONS=" and add these variables (as all one
   line with no space between "-" and "Dprimavera" or "-" and "Djavax").

   ▸ On Windows:

   ```
   set JAVA_OPTIONS=%SAVE_JAVA_OPTIONS%
   -Djavax.xml.soap.MessageFactory=com.sun.xml.messaging.saaj.soap.ve
   r1_1.SOAPMessageFactory1_1Impl
   -Djavax.xml.soap.SOAPConnectionFactory=weblogic.wsee.saaj.SOAPConn
   ectionFactoryImpl
   -Dprimavera.bootstrap.home=<WS_INSTALL_HOME>
   ```

   where *<WS_INSTALL_HOME>* is the P6 EPPM Web Services home directory the you
   set during installation

   ▸ On Linux:

   ```
   JAVA_OPTIONS="${SAVE_JAVA_OPTIONS}
   -Djavax.xml.soap.MessageFactory=com.sun.xml.messaging.saaj.soap.ve
   r1_1.SOAPMessageFactory1_1Impl
   -Djavax.xml.soap.SOAPConnectionFactory=weblogic.wsee.saaj.SOAPConn
   ectionFactoryImpl
   -Dprimavera.bootstrap.home=<WS_INSTALL_HOME>"
   ```

   where *<WS_INSTALL_HOME>* is the P6 EPPM Web Services home directory the you
   set during installation

Authentication Using Oracle Web Services Manager

Oracle Web Services Manager (OWSM) provides the business agility to respond to security
threats and security breaches by allowing policy changes to be enforced in real time without
interrupting running business processes.

The benefits of using OWSM with P6 EPPM includes:

▸ Centrally define and store security policies applied to the Web services.

▸ Monitor run time security events such as failed authentication or authorization.

▸ Avoids the need for developers to understand security specifications and security
  implementation details.

▸ Provides visibility and control of the policies through a centralized administration interface offered by Oracle Enterprise Manager.

Prerequisites for Oracle Web Services Manager Authentication

Ensure that you have installed Oracle Fusion Middleware SOA Suite and have created the Oracle MDS and OPSS schema. Oracle MDS schema is a prerequisite for the Oracle Fusion Middleware SOA Suite. Oracle OPPS schema is needed to extend the P6 EPPM Web Services domain to work with the Fusion MIddleware SOA Suite.

For information on how to create the Oracle MDS and OPSS schema, see the *Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility* guide on Oracle Technical Network.

Enabling Oracle Web Services Manager Authentication Using WebLogic Enterprise Manager

To enable OWSM authentication using WebLogic Enterprise Manager:

1) Create a user in security realms of the Weblogic Administration Console.

   a. Log in to the WebLogic Administration Console as an administrative user using the following URL:

   http://<Host_Name>:<Port>/console

   b. In the Change Center pane, select Lock & Edit.

   c. In the Domain Structure pane, select Security Realms.

   d. Select myrealm in the security realm list.

   e. In the Settings for myream page, select Users and Groups.

   > **Note:** The name of the tab reflects the name of the security realm that you selected.

   f. In the **Users** toolbar, click **New** and then complete the following:
   - In the Name field, enter a name for the user.
   - In the Password field, enter a password for the user.
   - In the Confirm Password field, re-enter the password from the previous field.
   - Click OK.

   > **Note:** The user added in security realms should also exist in the P6 EPPM database.

2) Log in to WebLogic Enterprise Manager using the following URL:

   http://<Host_Name>:<Port>/em

3) Navigate to the P6 EPPM domain using the **Target Navigation** directory.

4) Deploy P6 EPPM Web Services, using p6ws.ear, in WebLogic Enterprise Manager.

   For information on how to deploy P6 EPPM Web Services in WebLogic Enterprise Manager, refer to the *Deploying ADF Applications Using Fusion Middleware Control section* of the *Oracle Fusion Middleware Administering Oracle Fusion Middleware* guide.

5) Create a WSM Policy Set using the following settings:

a. On the **Create Policy Set: Enter Resource Scope** page, in the **Type of Resources** menu, select **SOAP Web Service**.

b. On the **Create Policy Set: Enter Resource Scope** page, complete the following:

▸ In the **Domain Name** field, enter the name of the domain on which <0P_ProdName_P6_WS is deployed.

▸ In the **Application Module Name or Connection Name** field, enter **p6ws.war**.

a. On the **Edit Policy Set: Add Policy References** page, attach an OWSM policy to the Policy Set.

> **Note:** For detailed instructions on how to create a WSM Policy Set, refer to the instructions in *Using Fusion Middleware Control* section under *Creating a Policy Set* in the *Fusion Middleware Security and Administrator's Guide for Web Services* guide.

6) Select OWSM as the authentication mode in Primavera P6 Administrator.

a. Log in to Primavera P6 Administrator using the following URL:

http://<Host_Name>:<Port>/p6/action/adminconfig

b. Expand your P6 EPPM configuration.

c. Expand Web Services/Security/Authentication.

d. In the **Mode** field, select **Owsm**.

e. Click **Save**.

7) Restart the P6 EPPM Web Services server.

Enabling Oracle Web Services Manager Authentication for P6 Team Member Using WebLogic Enterprise Manager

To enable OWSM authentication for P6 Team Member using WebLogic Enterprise Manager:

1) Log in to WebLogic Enterprise Manager using the following URL:

http://<Host_Name>:<Port>/em

2) Navigate to the P6 Team Member domain using the **Target Navigation** directory.

3) Right-click the P6 Team Member domain and then select Web Services and WSM Policy Sets.

4) Click **Create**.

5) Create a WSM Policy Set using the following settings:

a. On the **Create Policy Set: Enter Resource Scope** page, in the **Type of Resources** menu, select **SOAP Web Service**.

b. On the **Create Policy Set: Enter Resource Scope** page, complete the following:

▸ In the **Domain Name** field, enter the name of the domain on which P6 EPPM Web Services is deployed.

▸ In the **Application Module Name or Connection Name** field, enter **p6tm.war**.

a. On the **Edit Policy Set: Add Policy References** page, select **oracle/multi_token_rest_service_policy**.

> **Note:** For detailed instructions on how to create a WSM Policy Set, refer

to the instructions in *Using Fusion Middleware Control* section under *Creating a Policy Set* in the *Fusion Middleware Security and Administrator's Guide for Web Services* guide.

6) Create a user in security realms of the Weblogic Administration Console.

   a. Log in to the WebLogic Administration Console as an administrative user using the following URL:

   ```
   http://<Host_Name>:<Port>/console
   ```

   b. In the **Change Center** pane, select **Lock & Edit**.

   c. In the **Domain Structure** pane, select **Security Realms**.

   d. Select **myrealm** in the security realm list.

   e. In the Settings for myream page, select Users and Groups.

   > **Note:** The name of the tab reflects the name of the security realm that you selected.

7) Restart WebLogic and ensure that you can see the LDAP users in Users and Groups.

   > **Note:** The user added in security realms should also exist in the P6 EPPM database.

8) Navigate to the **Provider** tab.

9) Click **New**.

10) In the **Create a New Authentication Provider** window, complete the following:

   a. In the **Name** field, enter a name for the authentication provider.

   b. In the **Type** list, select **LDAPAuthenticator**.

   c. Click **OK**.

11) In the **Provider** tab, select the provider that you just created.

12) In the **Control Flag** list, select **SUFFICIENT** and then click **Save**.

13) Navigate to the **Provider Specific** tab.

14) Enter the details for your LDAP server in the **Connection** and **Users** sections.

15) Restart the WebLogic server.

Installing P6 Integration API

This chapter describes how to install the P6 Integration API.

In This Section

Creating a Shared Folder for P6 Integration API and P6

When P6 or `p6service.jar` runs on a server other than a P6 Integration API Remote Mode server and the "Job Service Shared Folder Location" configuration in the Primavera P6 Administrator is set to null, attempting an XML import from a P6 Professional deployment that uses Cloud Connect will cause the XML import process to fail. The import process fails because the temporary XML file for the project import is created under the P6 Integration API home directory but not the P6 or `p6services.jar` server, which runs the import code.

Setting the Job Service Shared Folder Location configuration to a shared directory that can be utilized by all P6 modules allows the temporary XML file to be stored in a common location that can be accessed by P6 Integration API Remote Mode server as well as the modules which have potential to run services, such as XML import.

For more information about when you must create a shared folder for P6 Integration API and P6, see the following article on My Oracle Support:

Primavera P6 XML Import Error "Cannot Schedule job. Please contact administrator to correct the problem" or "The job request you submitted did not complete successfully" And The Log File Displays "$path\importer_RMI_####.xml (No such file or directory)" (Doc ID 1987004.1)

To ensure that you can successfully implement an XML import when P6 Integration API Remote Mode and P6 or `p6service.jar` resides on different servers:

1) Add a shared mount (if Linux) or commonly mapped drive (if Windows) to the following locations:
    ‣ The P6 application servers
    ‣ The P6 Integration API application servers
    ‣ The P6 EPPM Web Services application servers
    ‣ The server that runs `p6services.jar`

    > **Note:** If you are using Windows, the drive letter that you use for the mapping must be the same across all servers.

2) Create a directory on the mount or drive on which temporary files can be created (for example, `/P6EPPM/Temp)`.
3) Log in to Primavera P6 Administrator.
4) Expand **Services** and then click **Job Service Shared File Location**.
5) In the **Job Service Shared File Location** field, enter the location of the shared directory.
6) Click **Save Changes**.

7) Restart the application servers that are associated with the changes.

Starting and Stopping Servers

This section describes how to start and stop server instances in WebLogic Server. The method you choose depends on whether you prefer using the WebLogic Server Administration Console or a command line, and on whether you are using Node Manager to manage the server's life cycle.

In This Section

Starting and Stopping the Domain

A domain can include multiple WebLogic Server clusters and non-clustered WebLogic Server instances. A minimal domain can contain only one WebLogic Server instance, which functions both as an Administration Server, and as a Managed server—such a domain can be useful while developing applications, but is not recommended for use in a production environment. Although the scope and purpose of a domain can vary significantly, most WebLogic Server domains contain the components described in this section.

By default, the WebLogic domain created by the P6 EPPM Configuration Wizard creates a domain with an administration server (AdminServer) and a managed server for each application that you deploy.

Starting the Domain

To start WebLogic:

1) Navigate to the *weblogic_home*/user_projects/domains/*your_domain* directory.
2) Run **startWeblogic.cmd** (with Windows) **startWeblogic.sh** (with UNIX).
3) If prompted for a user name and password in the WebLogic console window, enter the administrative user name and password you specified when creating the domain.

> **Note**: If you enabled the WebLogic precompile option, the WebLogic console displays "Server started in RUNNING mode" when precompiling finishes. For detailed information about enabling precompilation, see your WebLogic Server documentation.

Stopping the Domain

To stop WebLogic:

1) **Navigate to** `<WebLogic_Home>/user_projects/domain/<P6_EPPM_Domain>/bin`.

Where:

- <WebLogic_Home> is the WebLogic home directory. The default is Oracle/Middleware/Oracle_Home.
- <P6_EPPM_Domain> is the WebLogic domain for P6 EPPM that was set during configuration.

2) Run **stopWeblogic.sh**.
3) If prompted for a user name and password in the WebLogic console window, enter   the administrative user name and password you specified when creating the domain.

Starting and Stopping Managed Servers

You have several options for starting and stopping managed servers. Oracle recommends that you use the startNodeManager file and start the server in the WebLogic Administration Console (see *Starting a Managed Server* (on page 38)). However, you can view other ways to stop and start managed servers at "Managing Server Startup and Shutdown for Oracle WebLogic Server" at https://docs.oracle.com/middleware/1221/wls/START/overview.htm.

> **Note**: You can use different ways to start the managed servers, but you must ensure that the managed servers recognize the arguments required for your application (for example, the argument for where the bootstrap is located) and how the application environment will start.

Starting a Managed Server

> **Note**: When starting/stopping an environment using the node manager, the **StartScriptEnabled** setting in the **nodemanager.properties** file must equal true. Example: StartScriptEnabled=true

To start a managed server in the WebLogic Administration Console:

1) Run one of the following:
    - (with Windows)

```
<weblogic_home>\server\bin\startNodeManager.cmd
```

    - (with UNIX or Linux)

```
<weblogic_home>\server\bin\startNodeManager.sh
```

2) Launch the WebLogic Administration Console.

> **Note**: You can open the Administration Console via a web browser at `http://<serverIP>:<listenport>/console`. The default

`<listenport>` is 7001.

3) In the **Welcome** window, log in using the user name and password that you created when you created your WebLogic domain.

4) On the **Change Center** pane of the Administration Console, click **Lock & Edit**.

5) In the **Domain Structure** pane:

   a. Expand **Environment**.

   b. Click **Servers**.

6) In the **Summary of Servers** pane:

   a. Select the **Control** tab.

   b. Select the option for your managed server.

   c. Click **Start**.

7) In the **Server Life Cycle Assistant** pane, click **Yes**.

8) In the **Summary of Servers** pane, click the 'Start Refresh' icon in the middle of the pane to see when the **State** column says 'RUNNING.'

Stopping a Managed Server

To stop a managed server:

1) In the **Change Center** pane, click **Lock & Edit**.

2) In the **Domain Structure** pane, click **Deployments**.

3) In the **Summary of Deployments** pane, select the managed server that you want to stop.

4) In the **Summary of Deployments** pane, navigate to the **Control** tab and then complete the following:

   a. Click the down arrow to the right of the **Stop** button.

   b. Click one of the options to stop the managed server.

   c. In the **Start Application Assistant** pane, click **Yes**.

5) In the **Change Center** pane, click **Release Configuration**.

Configuring the Node Manager to Start and Stop Servers

Node Manager is a Java utility that runs as a separate process from WebLogic Server and allows you to perform common operations for a managed server regardless of its location to the Administration Server. The WebLogic Scripting Tool (WLST) is a command-line scripting interface that system administrators use to monitor and manage WebLogic Server instances and domains.

Using the node manager and WLST is optional, but it provides benefits if your WebLogic Server environment hosts applications with high availability requirements. If you run the node manager on a machine that hosts managed servers, you can start and stop the managed servers remotely using the WebLogic Console or from the command line using WLST as a node manager client. Using the node manager and WLST to control a WebLogic domain will ensure you do not need to manually modify the domain file to pass customized application java parameters, which eliminates the risk of distorting the files.

Oracle recommends controlling a WebLogic Domain through the node manager for the P6 EPPM WebLogic domain, instead of using individual Windows services per administration and managed server.

> **Note**: A node manager process is not associated with a specific WebLogic domain but with a machine. You can use the same node manager process to control server instances in any WebLogic Server domain as long as the server instances reside on the same machine as the node manager process. The node manager must run on each computer that hosts WebLogic server instances -- whether administration server or managed server -- that you want to control with the node manager.

Follow the steps in the following sections to control a WebLogic domain with the node manager. If you intend to control the node manager with Windows, refer to *Using the WebLogic Scripting Tool on Windows* (on page 41). If you intend to control the node manager with Linux or UNIX, refer to *Using the WebLogic Scripting Tool on Linux or UNIX* (on page 50).

For more information on the node manager and WLST, refer to the Oracle WebLogic server documentation.

In This Section

Using the WebLogic Scripting Tool on Windows

Refer to the following sections to configure the node manager with Windows.

Starting the Node Manager with WebLogic

In previous releases of WebLogic, a node manager process was not associated with a specific WebLogic domain but with a host machine. You used the same node manager process to control server instances in any WebLogic domain as long as the server instances resided on the same machine, a machine-scoped, per host node manager. In this release of WebLogic Server, the node manager is now configured individually, per domain, and each domain will contain a unique node manager process.

To install the node manager as a Windows service:

1) Log in to the WebLogic server with administrative privileges.

2) Modify the `installNodeMgrSvc.cmd` file so that it does not hard code the node manager hostname and port in the registry.

   a. Browse to `<WebLogic_Server_Home>\server\bin`.

   b. Open `installNodeMgrSvc.cmd` in a text editor.

   c. Locate the following lines in the file:

      `set NODEMGR_HOST=localhost`

      `set NODEMGR_PORT=5556`

      > **Note:** If lines `set NODEMGR_HOST=localhost` and `set NODEMGR_PORT=5556` do not exist, close the file and skip steps 2d and 2e.

   d. Comment out the lines by adding @rem to the beginning of the lines. For example:

      `@rem set NODEMGR_HOST=localhost`

      `@rem set NODEMGR_PORT=5556`

   e. Save the changes, and then close the file.

   f. Browse to `<WebLogic_domain_home>\bin`.

   g. Open `installNodeMgrSvc.cmd` in a text editor.

   h. Locate the following lines in the file:

      `set NODEMGR_HOST=localhost`

      `set NODEMGR_PORT=5556`

   i. Comment out the lines by adding @rem to the beginning of the lines. For example:

      `@rem set NODEMGR_HOST=localhost`

      `@rem set NODEMGR_PORT=5556`

   j. Save the changes, and then close the file.

3) Open the command window, and then complete the following steps:

   a. Change the directory to `<WebLogic_domain_home>\bin`.

   b. Enter the following command:

      `installNodeMgrSvc.cmd`

4) Close the command window.

5) Browse to Windows Services, and then start the **Oracle WebLogic <domain_name> NodeManager** service.

Starting an Administration and Managed Servers Using WLST With Node Manager

To automate the startup of the Admin and managed server using WLST:

> **Note**: If you are using P6 EPPM R8.2 or higher and you used the P6 EPPM Configuration Wizard to create the WebLogic domain, steps 1-6 below have already been completed.

1) Create a **userConfigFile** and **userKeyFile**:
   a. Start the WebLogic Domain's AdminServer by launching **weblogic_home\user_projects\domains\your_domain\bin\startWebLogic.cmd**.
   b. Browse to the P6 home directory and create a new directory called **scripts**.
   c. Open the command prompt.
   d. Change the directory to **weblogic_home\user_projects\domains\your_domain\bin**.
   e. Run the following commands:

   ```
   call
   setDomainEnv.cmd
   java_home\bin\java.exe weblogic.WLST
   ```

   where java_home is the path to the JDK installed and being used by WebLogic.

   > **Note**: If the "java_home" directory contains a space in the path, it should include quotations around the full path and executable being called. For example: "C:\Program Files\Java\jdk1.6.0_27\bin\java.exe" weblogic.WLST

   f. At the **wls:/offline>** prompt, enter the following command:

   ```
   connect('weblogic','password','AdminServerHostName:AdminServerPort')
   ```

   where:
   - *weblogic* is the weblogic admin user for the domain
   - *password* is the password for the weblogic admin user for the domain
   - *AdminServerHostName* is the host name where administration server was installed
   - *AdminServerPort* is the port number of administration server

   g. At the **wls:/mydomain/serverConfig>** prompt, enter the following command:

   ```
   storeUserConfig('p6home/scripts/primaveraConfig.secure',
   'p6home/scripts/primaveraKeyfile.secure')
   ```

   where *p6home* is the P6 home directory the you set during installation (for example, C:/p6home).

   h. Enter **y** at the following prompt:

   Creating the key file can reduce the security of your system if it is not kept in a secured location after it is created. Do you want to create the key file? y or n

   > **Note**: Confirm that the following two files were created under the **p6home/scripts** directory: **primaveraConfig.secure** and

primaveraKeyfile.secure. If these files were not created, repeat the previous steps.

i. Stop the WebLogic Domain's administration server from running by launching **weblogic_home\user_projects\domains\your_domain\bin\stopWebLogic.cmd**.

2) Ensure that AdminServer is assigned to the machine created for the WebLogic Domain:

a. Browse to the **weblogic_home\user_projects\domains\your_domain\config** directory.

b. Make a copy of the **config.xml** file to back up the existing file.

c. Open the **config.xml** file with a text editor.

d. Search for the following:

`<name>AdminServer</name>`

e. Locate the </ssl> element and add the following line under the element:

`<machine><!--NameOfMachine--></machine>`

where `<!--NameOfMachine-->` is the name of the logical machine created for the WebLogic domain.

> **Notes**:
>
> - If an </ssl> element does not exist, locate and add the line `<machine><!--NameOfMachine--></machine>` above the `<listen-address>` element.
> - If a <machine> element already exists for AdminServer, this step can be skipped.

f. Save and close the **config.xml** file.

3) Ensure the administration server and managed servers in the weblogic domain are configured with a boot.properties file for node manager startup:

a. Browse to the **weblogic_home\user_projects\domains\your_domain\servers\AdminServer\data** directory.

b. Check if you have a directory called **nodemanager**. If you do not, create a directory called **nodemanager**.

c. Create a text file in the nodemanager directory called **boot.properties**.

d. Open **boot.properties** using a text editor and add:

- usename=<werblogic>

  where <weblogic> is the Administrator user created for the WebLogic domain.

- password=<password>

  where <password> is the password for the Administrator user created for the WebLogic domain

e. Save and close the file.

f. Repeat steps a – e for every managed server in the domain.

Example path:

weblogic_home\user_projects\domains\your_domain\servers\<NameOfManagedServer>\data

> **Note**: The username and password variables will become encrypted in the boot.properties file the first time the servers start using the node manager.

4) Create a text file in the **p6home/scripts** directory called **start_primavera.py** (this is also known as a python file).

5) Open **start_primavera.py** using a text editor and add the following 6 lines (each line should be entered as a single line):

```
nmConnect(userConfigFile='p6home/scripts/primaveraConfig.secure',use
rKeyFile='p6home/scripts/primaveraKeyfile.secure',host='NodeMangerHos
t',port='5556',domainName='Mydomain',domainDir='DomainLocation',nmType=
'ssl')
```

```
arg = "Arguments=\" -server –Xms128m –Xmx512m –XX:MaxPermSize=256m \""
```

```
prps = makePropertiesObject (arg)
```

```
nmStart('AdminServer',props=prps)
```

```
connect(userConfigFile='p6home/scripts/primaveraConfig.secure',userK
eyFile='p6home/scripts/primaveraKeyfile.secure',url='t3://Adminserverh
ost:Adminserverport')
```

```
start('myManagedServer')
```

where:

▸ *p6home* is the home directory for one of the P6 EPPM applications. For example: C:/p6home

▸ *NodeManagerHost* is the hostname where the node manager is installed

▸ *Adminserverhost* is the hostname where the administration server is running

▸ *Adminserverport* is the port number of the administration server (default is 7001)

▸ *Mydomain* is the name of the WebLogic domain

▸ *DomainLocation* is the path to the WebLogic domain (C:/Oracle/Middleware/user_projects/domains/your_domain)

▸ *AdminServer* is the Logical Name of the administration server

▸ *myManagedServer* is the Logical Name of the managed server

> **Note**: If multiple managed servers exist, repeat the line 'start('myManagedServer')' with the name of each managed server.

6) Save and close the file.

7) Create a command file that will call the Python script you created above:

a. Open a text editor and enter the following lines:

```
SETLOCAL
SET CLASSPATH=
call
weblogic_home\user_projects\domains\your_domain\bin\setDomainEnv.cmd
call java_home\bin\java.exe weblogic.WLST
p6home\scripts\start_primavera.py
```

where:

- *weblogic_home* is the WebLogic Server home directory. The default is C:\Oracle\Middleware
- *your_domain* is the domain name directory in your environment
- *java_home* is the path to the JDK installed and being used by WebLogic
- *p6home* is the home directory for the P6 EPPM folder containing the scripts directory. For example: C:\p6home

Note: If the "java_home", "p6home" or "weblogic_home" directories contain a space in the path, it should include quotations around the full path and executable/command file being called. For example: call "C:\Program Files\Java\jdk1.6.0_27\bin\java.exe" weblogic.WLST C:\P6WebAccess\scripts\start_primavera.py

b. Save the file in the p6home\scripts directory with the following name: **start_primavera.bat**.

Adding the start_primavera.bat to the Windows Task Scheduler

To add the **start_primavera.bat** file to the Windows Task Scheduler:

1) Browse to **Start**, **All Programs**, **Accessories**, **System Tools**, **Task Scheduler**.
2) Under **Actions**, select **Create Task**.
3) Under the **Create Task**, **General** tab:
   a. Provide a name for the Scheduled Task.
   b. Provide a description of the Scheduled Task
   c. Select **Change User or Group**.
   d. Under **Security Options** in the **Enter the object name to select** field, enter **System** and click **OK**.
   e. Select **Run with the highest privileges**.
4) Under the **Create Task**, **Trigger** tab:
   a. Select **New**.
   b. In the **Begin the task** list, select **At Startup**.
   c. Under **Advanced Settings**, select **Enabled**.
   d. Select **OK**.
5) Under the **Create Task**, **Actions** tab:
   a. In the **Action** list, select **Start a program**.
   b. In the **program/script** field, browse to the **start_primavera.bat** file.
   c. Select **OK**.
6) View the scheduled task by selecting **Task Scheduler Library**.

Executing the start_primavera.bat file

Oracle recommends executing the start_primavera.bat file using the command prompt before running it through the scheduled task window so you can identify and resolve issues that occur after you create the batch file and python script. See *Troubleshooting* (on page 49) for common issues that can occur after you create the batch file and python script.

To run `start_primavera.bat`:

1) Browse to **Start**, **Run**.
2) In the **Run** window, enter **cmd**.
3) Select **OK** to open the command prompt.
4) Enter the following:

   p6home`\scripts\start_primavera.bat`

   where *p6home* is the home directory for one of the P6 EPPM applications. For example: C:\p6home

   > **Notes**:
   >
   > - Each Java process that loads should run under the SYSTEM account.
   > - Later, if the start_primavera.bat file is not run automatically through the scheduled task, you can run it directly or run it manually using the Windows Task Scheduler.

Using WebLogic Scripting to Stop the WebLogic Domain for P6 EPPM

To stop the WebLogic domain with WebLogic Scripting:

1) Create a text file in the **p6home/scripts** directory called **stop_primavera.py**.
2) Open **stop_primavera.py** using a text editor and enter the following lines:

   > **Note**: You must indent the commands under "try" and "except".

```
print('PRM-Stopping Primavera administration server, managed servers,
and deployments')
print('PRM-This will NOT attempt to stop the node manager as it may be')
print('PRM-needed for other servers on this machine.')
connect(userConfigFile= '
p6home/scripts/primaveraConfig.secure',userKeyFile='p6home/scripts/p
rimaveraKeyfile.secure',url='t3://Adminserverhost:Adminserverport')
try:
   print('PRM-Shutting down managed server myManagedServer')
   shutdown('myManagedServer','Server','true',100,'true')
except:
   print('PRM-Failed to stop managed server myManagedServer')
try:
```

```
print('PRM-Shutting down AdminServer')
shutdown('AdminServer','Server','true',100,'true')
except:
print('PRM-Failed to stop AdminServer')
print('PRM-Shutdown Complete')
exit()
```

where:

▸ *p6home* is the home directory for one of the P6 EPPM applications. For example: C:\p6home

▸ *Adminserverhost* is the hostname where the administration server is running

▸ *Adminserverport* is the port number of the administration server (default is 7001)

▸ *myManagedServer* is the Logical Name of the managed server

▸ *AdminServer* is the Logical Name of the administration server

3) If multiple managed servers exist, repeat the following lines from above with the name of each managed server:

> **Note**: You must indent the commands under "try" and "except".

```
try:
print('PRM-Shutting down managed server myManagedServer')
shutdown('myManagedServer','Server','true',100,'true')
except:
print('PRM-Failed to stop managed server myManagedServer')
```

4) Save and close the file.

5) Browse to `<P6_EPPM_Home>\scripts` and then create a text file called **stop_primavera.bat**.

6) Edit **stop_primavera.bat** and then add the following lines:

```
SETLOCAL
SET CLASSPATH=
call
weblogic_home\user_projects\domains\your_domain\bin\setDomainEnv.c
md
call java_home\bin\java.exe weblogic.WLST
p6home\scripts\stop_primavera.py
```

where:

▪ *weblogic_home* is the WebLogic Server home directory. The default is C:\Oracle\Middleware

▪ *your_domain* is the domain name directory in your environment

▪ *java_home* is the path to the JDK installed and being used by WebLogic

▪ *p6home* is the home directory for the P6 EPPM folder containing the scripts directory. For example: C:\p6home

> **Note**: If the "java_home", "p6home" or "weblogic_home" directories contain a space in the path, it should include quotations around the full path and executable/command file being called. For example: call "C:\Program Files\Java\jdk1.6.0_27\bin\java.exe" weblogic.WLST C:\P6WebAccess\scripts\start_primavera.py

    a. Save and close the file.

Troubleshooting

To troubleshoot the following start_primavera.bat error:

WLSTException: Error occured while performing nmConnect : Cannot connect to Node Manager. : Access to domain '<domainName>' for user '<username>' denied.

1) Ensure that the **userConfigFile** and **userKeyFile** variables entered in the **start_primavera.py** file point to a valid path and file name for the primaveraConfig.secure and primaveraKeyfile.secure files.

2) Ensure that the node manager username and password are correct:
   a. Start the AdminServer for the WebLogic Domain.
   b. Launch the WebLogic Administrative Console.
   c. In the **Change Center** pane of the Administration Console, click **Lock & Edit**.
   d. Under the **Domain Structure** pane, select the link for your domain name.
   e. Select the **Security** tab and then select the **Advanced Options** link.
   f. Enter the valid credentials in the console for NodeManager username and NodeManager password.
   g. Activate the changes.
   h. Logout of the WebLogic Administrative Console.
   i. Stop the AdminServer for the WebLogic Domain.
   j. Retry executing the **start_primavera.bat** file.

For the following error, the managed server did not start using the Node Manager and you need to create a **startup.properties** file before you can start the managed server using WLST Scripting.

weblogic.management.ManagementException: Booting as administration server, but servername, <ManagedServerName>, does not match the administration server name, AdminServer.

To create a `startup.properties` file:

1) Start the AdminServer for the WebLogic Domain.
2) Launch the WebLogic Administrative Console.
3) Under the **Domain Structure** pane, select 'servers' link.
4) On the **Servers** page, select the **Control** tab.
5) Select **Startup**.
6) Once the server is in a running state, on the **Servers** page, select the **Control** tab.
7) Select **Shutdown**.

8) Verify the **startup.properties** file is in the
`weblogic_home\user_projects\domains\<domain>\servers\<managedServer>`
`\data\nodemanager` directory. You can now startup through the WebLogic scripting files.

Using the WebLogic Scripting Tool on Linux or UNIX

Refer to the following sections to configure the node manager with Linux or UNIX.

Creating a Configuration and Key File

In order to automate the startup of the administration server and managed servers of your P6 EPPM deployment, you must create a configuration (`userConfigFile`) and key (userKeyFile) file. For more information about configuration and key files, refer to the *Oracle Fusion Middleware Command Reference for Oracle WebLogic Server* guide.

To create a configuration and key file:

1) Start the WebLogic Domain's Administration Server by launching **startWebLogic.sh** from
`<WebLogic_Home>/user_projects/domains/<P6_EPPM_Domain>/bin/`

> **Where:**
> - <WebLogic_Home> is the WebLogic home directory. The default is Oracle/Middleware/Oracle_Home.
> - <P6_EPPM_Domain> is the WebLogic domain for P6 EPPM that was set during configuration.

2) Browse to the P6 home directory and create a new directory called **scripts**.
3) Open the terminal console and run the following commands:
`cd <WebLogic_Home>/user_projects/domains/<P6_EPPM_Domain>/bin`

`source setDomainEnv.sh`

`<Java_Home>/bin/java.exe weblogic.WLST`

> **Note**: If the <Java_Home> directory contains a space in the path, it should include quotations around the full path and executable being called.
>
> **Where:**
> - <Java_Home> is the path to the JDK that is used by WebLogic.
> - <WebLogic_Home> is the WebLogic home directory. The default is Oracle/Middleware/Oracle_Home.
> - <P6_EPPM_Domain> is the WebLogic domain for P6 EPPM that was set during configuration.

4) At the **wls:/offline>** prompt, enter the following command:
`connect('<WebLogic_Username>','<WebLogic_User_Password>','<Admin_Ser`
`ver_Hostname>:<Admin_Server_Port>')`

Where:

- <WebLogic_Username> is the name of the WebLogic administrator user for the domain.
- <WebLogic_Password> is the password for the weblogic administrator.
- <Admin_Server_Hostname> is the hostname where the administration server was installed.
- <Admin_Server_Port> is the port number of administration server.

5) At the **wls:/mydomain/serverConfig>** prompt, enter the following command:

```
storeUserConfig('<P6_EPPM_Home>/scripts/primaveraConfig.secure',
'<P6_EPPM_Home>/scripts/primaveraKeyfile.secure')
```

Where: <P6_EPPM_Home> is the P6 EPPM home directory that was set during installation.

6) Enter **y** at the following prompt:

Creating the key file can reduce the security of your system if it is not kept in a secured location after it is created. Do you want to create the key file? y or n

7) Confirm that the following two files were created under the `<P6_EPPM_Home>/scripts` directory:

primaveraConfig.secure and **primaveraKeyfile.secure**

If these files were not created, repeat the previous steps.

Associating the Administration Server with Machines on a P6 EPPM Domain

If a <machine> element already exists for `AdminServer`, this procedure can be skipped.

To associate the administration server with machines on the P6 EPPM domain:

1) Browse to the `<Weblogic_Home>\user_projects\domains\<P6_EPPM_Domain>\config`.
2) Make a copy of **config.xml** as a back up of the existing file.
3) Open **config.xml** with a text editor.
4) Search for **<name>AdminServer</name>**.
5) Locate the </ssl> element and then add the following line beneath it:

```
<machine><!--Machine_Name--></machine>
```

> **Note**: If an </ssl> element does not exist, locate and add the line <machine><Machine_Name></machine> above the <listen-address> element.
>
> **Where:** `<!--Machine_Name-->` is the logical name of the machine created for the WebLogic domain.

6) Save and close the file.

Configure a boot.properties File for the Administration and Managed Servers

Ensure that the administration server and managed servers in the WebLogic domain are configured with a `boot.properties` file.

To create a boot.properties file for the administration and managed servers:

1) Browse the `data` directory of either the administration server or a managed server.

   For example,
   `<Weblogic_Home>\user_projects\domains\<P6_EPPM_Domain>\servers\Admin Server\data.`

2) Check if you have a directory called `nodemanager`. If you do not, create a directory called **nodemanager** beneath the `data` directory.

3) Create a text file in the `nodemanager` directory called **boot.properties**.

4) Edit **boot.properties** and then add the following lines:
   ‣ username=<WebLogic_Username>
   ‣ password=<WebLogic_Password>

   > **Notes:**
   >
   > ▪ <WebLogic_Username> is the name of the WebLogic administrator for the domain.
   > ▪ <WebLogic_Password> is the password for the weblogic administrator.

5) Save and close the file.
6) Repeat this procedure for the administration server and every managed server at `<WebLogic_Home>\user_projects\domains\<P6_EPPM_Domain>\servers\.`

   > **Note**: The username and password variables are encrypted in the `boot.properties` file the first time the servers start using the Node Manager.
   >
   > **Where:**
   >
   > ▪ <WebLogic_Home> is the WebLogic home directory. The default is `Oracle/Middleware/Oracle_Home`.
   > ▪ <P6_EPPM_Domain> is the WebLogic domain for P6 EPPM.

Using WebLogic Scripting to Start the P6 EPPM WebLogic Domain

To automate the startup of the Admin and managed server using WLST:

1) Browse to `<P6_EPPM_Home>\scripts` and then create a text file called **start_primavera.py**.

> **Note:** <P6_EPPM_Home> is the P6 EPPM home directory that was set during installation.

2) Edit **start_primavera.py** and then add the following lines:

```
nmConnect(userConfigFile='<P6_EPPM_Home>/scripts/primaveraConfig.sec
ure',userKeyFile='<P6_EPPM_Home>/scripts/primaveraKeyfile.secure',ho
st='<Node_Manager_Hostname>',port='5556',domainName='<P6_EPPM_Domain
>',domainDir='<Domain_Path>',nmType='ssl')

arg = "Arguments=\" -server -Xms128m -Xmx512m -XX:MaxPermSize=256m \""

prps = makePropertiesObject (arg)

nmStart('AdminServer',props=prps)

connect(userConfigFile='<P6_EPPM_Home>/scripts/primaveraConfig.secur
e',userKeyFile='<P6_EPPM_Home>/scripts/primaveraKeyfile.secure',url=
't3://<Admin_Server_Hostname>:<Admin_Server_Port>')

start('<Managed_Server_Name>')

#If multiple managed servers exist, repeat the previous line for each
managed server.
```

> **Where:**
> - <P6_EPPM_Home> is the P6 EPPM home directory that was set during installation.
> - <Node_Manager_Hostname> is the name of the host where the Node Manager is running.
> - <Admin_Server_Hostname> is the name of the host where the administration server was installed.
> - <Admin_Server_Port> is the port number of administration server. The default is 7001.
> - <P6_EPPM_Domain> is the WebLogic domain for P6 EPPM that was set during configuration.
> - <Domain_Path> is the full path to the P6 EPPM domain. For example,
> `<WebLogic_Home>\user_projects\domains\<P6_EPPM_Domain>`
> - <Managed_Server_Name> is the logical name of the managed server. The default managed server names for P6 EPPM are P6, P6IntegrationAPI, p6procloudconnect, P6WebServices, and TeamMember.

3) Save and close the file.

4) Browse to `<P6_EPPM_Home>\scripts` and then create a text file called **start_primavera.sh**.

5) Edit **start_primavera.sh** and then add the following lines:

```
export CLASSPATH=
<weblogic_home>/wlserver/server/bin/setWLSEnv.sh
<Java_Home>/bin/java weblogic.WLST <P6_EPPM_Home>/scripts/start_primavera.py
```

> **Where:**
> - <WebLogic_Home> is the WebLogic home directory. The default is Oracle/Middleware/Oracle_Home.
> - <Java_Home> is the path to the JDK that is used by WebLogic.
> - <P6_EPPM_Home> is the P6 EPPM home directory that was set during installation.

6) Save and close the file.

Using WebLogic Scripting to Stop the WebLogic Domain for P6 EPPM

To stop the WebLogic domain with WebLogic Scripting:

1) Browse to `<P6_EPPM_Home>\scripts` and then create a text file called **stop_primavera.py**.

> **Note:** <P6_EPPM_Home> is the P6 home directory that was set during installation.

2) Edit **stop_primavera.py** and then add the following lines:

```
print('Stopping Primavera servers. This will not stop the Node Manager because it
may be needed for other servers.')

scripts_dir='<P6_EPPM_Home>/scripts'
adminserver_host='<Admin_Server_Hostname>'
adminserver_port='<Admin_Server_Port>'
<Managed_Server_Variable>='<Managed_Server_Name>'
# Create a variable, similar to the one above, for each managed server in the P6
EPPM domain.

connect(userConfigFile=scripts_dir+'/primaveraConfig.secure',userKeyFile=scrip
ts_dir+'/primaveraKeyfile.secure',url='t3://'+adminserver_host+':'+adminserver
_port)

# The following section of the script tries to shutdown a managed server. Copy and
modify this section of the script for each managed server in the P6 EPPM domain.
try:
  print('PRM-Shutting down the managed server for <Managed_Server_Variable>')
  shutdown(<Managed_Server_Variable>,'Server','true',100,'true')
except:
  print('PRM-Failed to stop managed server: ' + <Managed_Server_Variable>)
```

```
# The following section of the script tries to shutdown the administration server.
try:
  print('PRM-Shutting down <Admin_Server_Name>.')
  shutdown('<Admin_Server_Name>','Server','true',100,'true')
except:
  print('PRM-Failed to stop <Admin_Server_Name>.')
print('PRM-Shutdown Complete')
exit()
```

Notes:

- <P6_EPPM_Home> is the P6 EPPM home directory that was set during installation.
- <Admin_Server_Hostname> is t*he* name of the host where the administration server was installed.
- <Admin_Server_Port> is the port number of administration server. The default is 7001.
- <Managed_Server_Variable> is the variable that you created for a P6 EPPM application.
- <Managed_Server_Name> is the logical name of the managed server. The default managed server names for P6 EPPM are P6, P6IntegrationAPI, p6procloudconnect, P6WebServices, and TeamMember.
- <Admin_Server_Name> is the logical name of the administration server. The default is AdminServer.

3) Save and close the file.

4) Browse to `<P6_EPPM_Home>\scripts` and then create a text file called **stop_primavera.sh**.

5) Edit **stop_primavera.sh** and then add the following lines:

```
export CLASSPATH=
<WebLogic_Home>/wlserver/server/bin/setWLSEnv.sh <Java_Home>/bin/java
weblogic.WLST <P6_EPPM_Home>/scripts/stop_primavera.py
```

Notes:

- <WebLogic_Home> is the WebLogic home directory. The default is Oracle/Middleware/Oracle_Home.
- <Java_Home> is the path to the JDK that is used by WebLogic.
- <P6_EPPM_Home> is the P6 EPPM home directory that was set during installation..

6) Save and close the file.

Daemonizing the Node Manager for Primavera Applications

WebLogic does not provide scripts for uninstalling and re-installing the Node Manager daemon process. If your WebLogic server implementation includes a Node Manager daemon process, uninstall it. For more information about removing a daemon process, refer to your operating system documentation.

To create a daemon process:

1) Open a terminal and log in to the system as a super user.
2) Change the directory to the following location:

   `/etc/init.d`
3) Create a text file called **wlsnodemgr**.
4) Edit **wlsnodemgr** and add the following lines:

```
#!/bin/bash
# wlsnodemgr startup script for Node Manager
# chkconfig: - 74 15
# description: weblogic Node Manager
# processname: wlsnodemgr

# Source function library
. /etc/rc.d/init.d/functions

# Modify below values appropriately
export WLS_DOMAIN_HOME=<WebLogic_Home>/user_projects/domains/<P6_EPPM_Domain>
export P6HOME=[<P6_EPPM_Home>]
export LOG_FILE_LOC=/home/oracle/wlsnodemgr.log
export DAEMON_USER=oracle

start(){
    nohup su - $DAEMON_USER -c $ <P6_EPPM_Domain>/bin/startNodeManager.sh >
$LOG_FILE_LOC &
    sleep 2m
    nohup su - $DAEMON_USER -c $ <P6_EPPM_Home>/scripts/start_primavera.sh >>
$LOG_FILE_LOC &
}

stop(){
    nohup su - $DAEMON_USER -c $<P6_EPPM_Home>/scripts/stop_primavera.sh >>
$LOG_FILE_LOC &
    nohup su - $DAEMON_USER -c $<P6_EPPM_Domain>/bin/stopNodeManager.sh >>
$LOG_FILE_LOC &
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    *)
```

```
        echo $"Usage: $PROG {start|stop}"
        exit 1
esac
```

Where:

- <WebLogic_Home> is the WebLogic home directory. The default is Oracle/Middleware/Oracle_Home.
- <P6_EPPM_Domain> is the WebLogic domain for P6 EPPM that was set during configuration.
- <P6_EPPM_Home> is the P6 EPPM home directory that was set during installation.

---

5) Save and close the file.
6) Create a background service by running the following command:

```
chkconfig -- add wlsnodemgr
```

7) Activate the background service by running the following command:

```
chkconfig wlsnodemgr on
```

Note: The background service can be stopped by running the following command:

```
chkconfig wlsnodemgr off
```

Starting the Node Manager with WebLogic

To start the Node Manager with WebLogic:

1) Open a terminal console.
2) Run **startNodeManager.sh** from the bin folder of the P6 EPPM domain by using the following scripts:

```
cd <WebLogic_Home>/user_projects/domains/<P6_EPPM_Domain>/bin

./startNodeManager
```

Where:

- <WebLogic_Home> is the WebLogic home directory. The default is Oracle/Middleware/Oracle_Home.
- <P6_EPPM_Domain> is the WebLogic domain for P6 EPPM that was set during configuration.

Setting the Node Manager as a Windows Service

You can set the WebLogic node manager as a windows service in your production environments.

To set up node manager as a windows service:

1) Stop any running instances of the WebLogic node manager.
2) Navigate to the WebLogic Server installation home directory, then navigate to **server\bin**.
3) In the **server\bin** folder modify the **installNodeMgrSvc.cmd** file using a text editor of your choice, and comment out the following lines:

> - **Note:** Add **@REM** at the beginning of a line to comment out the line.

```
set NODEMGR_HOST=localhost
set NODEMGR_PORT=5556
```

4) Run the **installNodeMgrSvc.cmd** file to create the node manager windows service.

> **Note:** To modify the port number on which node manager runs, navigate to the WebLogic Server installation home directory, then navigate to **common\nodemanager**. Open the **nodemanager.properties** file in a text editor of your choice and set the value of **ListenPort** to the port number you'd like to use in your environment.

Configuring Your Environment to Support Event Notification

Configuring your Environment

If you would like to receive notification when events occur, you must configure the JMS message queue and P6 to send the events in which you are interested.

To configure your environment to support notification:

1) Configure the WebLogic message queue:
   a. (Optional) Target the JMS server to a migratable target. See *Targeting the JMS Server to a Migratable Target* (on page 59)
   b. Determine if the WebLogic message queue will be on the same domain as P6.
      - See *Configuring the WebLogic Message Queue* (on page 59) if the queue and the application are on the same domain.
      - See *Sending Events to a Remote WebLogic JMS Server* (on page 66) if the queue and the application are on different domains.
   c. (Optional) Configure the message queue security policy. See *Configuring the Security Policy for the WebLogic Message Queue* (on page 70).
2) Configure the application settings in P6 to send event notification. See *Configuring Eventing in P6* (on page 63).

   An event may be one of two types, a business object event or a special operation event.

Targeting the JMS Server to a Migratable Target

JMS-related services are hosted on individual server instances within a cluster and for these services, the WebLogic Server migration framework supports failure recovery with service migration, as opposed to failover. In a clustered server environment, a recommended best practice is to target the JMS server to a migratable target, so that a member server will not be a single point of failure. Therefore, the following prerequisite instruction is required if configuring The WebLogic Message Queue In A Cluster.

> **Note**: The instruction below assumes the WebLogic domain being configured has managed servers already created and assigned to a cluster parameter.

1) Configure the cluster migration Basis to Consensus:
    a. Log in into the WebLogic Administration console.
    b. In the Change Center of the Administration Console, click **Lock & Edit**.
    c. Navigate to the following location under the Domain Structure:
        Environment -> **Clusters** -> **ClusterName** -> **Migration**
    d. Locate the **Migration Basis** drop-down list and change its value to **Consensus**.
    e. Select **Save**.
    f. Select **Active Changes**.
2) Configure a migratable target for one of the managed servers in the cluster:
    a. In the Change Center of the Administration Console, click **Lock & Edit**.
    b. In the **Domain Structure** tree, expand **Environment**, then select **Migratable Targets**.
    c. On the Summary of Migratable Targets page, select the link for one of the managed servers in the list.
    d. Select the migration tab:
    e. In Service Migration Policy, select the **Auto-Migrate Exactly-Once Services** migration policy.
    f. In **Constrained Candidate Servers**, move the user-preferred server and the additional managed servers in the cluster you want to support failure recovery from the **Available** column to the **Chosen** column.
    g. Select **Save**.
    h. In the Change Center of the Administration Console, click **Active Changes**.
    i. Restart the Admin server and the managed servers in the cluster for changes to take effect.

Configuring the WebLogic Message Queue

When an event is triggered, the P6 Event Notification system sends the event message to a message queue. To receive these notifications, you must first configure the message queue.

The following procedure indicates how to set up a WebLogic Java Messaging Service (JMS) message queue when the queue and P6 are on the same domain. See ***Sending Events to a Remote WebLogic JMS Server*** (on page 66) if the application and the queue are on different domains. For information about setting up other JMS-based message queues, see the vendor documentation.

To set up the WebLogic JMS message queue:

1) In either a new or existing WebLogic domain, launch the WebLogic **Administration Console** if it is not already open.
2) In the WebLogic **Administration Console**:
   a. Create a new JMS server and persistence store. See ***Creating a JMS Server and Persistence Store*** (on page 60).
   b. Create a JMS module. See ***Creating a JMS Module*** (on page 61).
   c. Create a new connection factory. See ***Creating a JMS Connection Factory*** (on page 61).
   d. Create a new queue or topic. See ***Creating a JMS Message Queue and Subdeployment*** (on page 62) to see how to create a new queue.

   > **Note**: Create a queue to deliver a message to a specific group of users. Create a topic to distribute a message amongst several users.

Creating a JMS Server and Persistence Store

Create a JMS server to hold queues and topics.

To create a JMS server and persistence store:

1) If it is not already open, launch the WebLogic **Administration Console** on the remote or local server.
2) In the WebLogic **Administration Console**, expand **Services**/**Messaging** and click **JMS Servers** in the **Domain Structure** pane.
3) On the **Summary of JMS Servers** page, click **New**.
4) On the **Create a New JMS Server** page:
   a. Under **JMS Server Properties**:
      1. Enter a name in the **Name** field.
      2. Click **Create a New Store**.
   b. Under **Select a store type**, select **File Store** from the **Type** list and click **Next**.
   c. Under **File Store Properties**:
      1. Enter a name in the **Name** field.
      2. Select a server instance from the **Target** list.

      > **Note**:   If you are configuring the WebLogic Message Queue in a cluster, select the migratable target you configured earlier.

      3. Specify a location for the file store in the **Directory** field.

> ▪ This location must already exist on your machine and you must have read/write rights to this folder. No error messages appear on the WebLogic console if the filestore is not configured correctly.

> 4. Click **OK**.

d. Under **JMS Server Properties**, select the new store from the **Persistent Store** list and click **Next**.

e. Under **Select Targets**:

> ▪ If you are deploying to a standalone server, select the administration server (for example, AdminServer) as the target from the **Target** list and click **Finish**.

> ▪ If you are deploying to a cluster, select the migratable target configured earlier.

Creating a JMS Module

Create a JMS module to manage and configure resources.

To create a JMS module:

1) If it is not already open, launch the WebLogic **Administration Console** on the remote or local server.

2) In the WebLogic **Administration Console**, expand **Services**/**Messaging** and click **JMS Modules** in the **Domain Structure** pane.

3) On the **JMS Modules** page, click **New**.

4) On the **Create JMS System Module** page:

a. Under **The following properties will be used to identify your new module**, enter a name in the **Name** field and click **Next**.

> **Note**: Make a note of this name, you will need to know which module to expand in the *Configuring the Security Policy for the WebLogic Message Queue* (on page 70) topic.

b. Under **The following properties will be used to target your new JMS system module**, select a target server in the **Servers** box and click **Next**.

> **Note**: If you are deploying to a cluster, select the cluster from the targets list.

c. Under **Add resources to this JMS system module**, select the **Would you like to add resources to this JMS system module** option and click **Finish**.

Creating a JMS Connection Factory

Create a connection factory to enable connections between your JMS elements.

To create a JMS connection factory:

1) Log in to the WebLogic **Administration Console**.

2) In the WebLogic **Administration Console**, expand **Services**/**Messaging** and click **JMS Modules** in the **Domain Structure** pane.

3) On the **JMS Modules** page, select the module to which you want to add the connection factory.

4) On the **Settings** page, click the **Configuration** tab and under **Summary of Resources** click **New**.

5) On the **Create a New JMS System Module Resource** page:

   a. Under **Choose the type of resource you want to create**, select the **Connection Factory** option and click **Next**.

   b. Under **Connection Factory Properties**:

      1. Enter a name for the connection factory in the **Name** field.

      2. Enter a name in the **JNDI** field.

         > **Note**: Make note of the JNDI name. You will need to enter this name in the **JMS Connection Factory** field of P6 Application Settings Eventing page. If you are using a remote server, you will need to enter this name in the **Remote JNDI Name** field on the **Connection Factories** tab in the *Creating a Foreign JMS Server* (on page 69) topic.

      3. Click **Next**.

   c. Under **The following properties will be used to target your new JMS system module resource**, ensure the correct server is targeted and click **Finish**.

      > **Note**: If you are deploying to a cluster, the cluster should be selected as the target.

Creating a JMS Message Queue and Subdeployment

Create a message queue to act as a receptacle for event messages sent from P6.

To create a JMS message queue and subdeployment:

1) Log in to the WebLogic **Administration Console**.

2) In the WebLogic **Administration Console**, expand **Services**/**Messaging** and click **JMS Modules** in the **Domain Structure** pane.

3) On the **JMS Modules** page, select the module to which you want to add the queue.

4) On the **Settings** page, click the **Configuration** tab.

5) On the **Configuration** tab under **Summary of Resources**, click **New**.

6) On the **Create a New JMS System Module Resource** page:

   a. Under **Choose the type of resource you want to create**, select the **Queue** option and click **Next**.

   b. Under **JMS Destination Properties**:

      1. Enter a name for the queue in the **Name** field.

      2. Enter a name in the **JNDI** field.

         > **Note**:   Make note of the JNDI name. You will need to enter this name in the **JMS Destination Name** field of the P6 Application Settings Eventing

page. If you are using a remote server, you will need to enter this name in the **Remote JNDI Name** field on the **Destinations** tab in the Creating a Foreign JMS Server topic.

   3. Click **Next**.

c. Under **The following properties will be used to target your new JMS system module resource**, click **Create a New Subdeployment**.

7) On the **Create a New Subdeployment** page, enter a name in the **Subdeployment Name** field and click **OK**.

8) On the **Create a New JMS System Module Resource** page:

a. Select the new subdeployment from the **Subdeployments** list.

b. Select the JMS server you created when creating the Persistence Store as the target from the **JMS Servers** box.

c. Click **Finish**.

Configuring Eventing in the Primavera P6 Administrator

Depending on administrative settings, events can be triggered when P6, P6 EPPM Web Services, or the P6 Integration API is used to update or create objects in the P6 database or when one of the special operations completes. When a business object change or special operation triggers an event, the P6 Event Notification system sends an event message to a user-configured message queue. If you are planning to use P6 Event Notification with P6 EPPM products, follow the steps below to configure the notification to work with your Java Messaging Service (JMS), the application server, and P6. Refer to the message queue vendor documentation.

Configuring Eventing in P6

If you want events to occur when business objects are created or updated, or when an operation is performed, you must configure Eventing page in the Application Settings of P6.

To configure the Eventing page:

1) Launch P6.
2) Click **Administration**.
3) On the Administration navigation bar, click **Application Settings**.
4) On the Application Settings page, click **Eventing**.
5) Click the **General** tab.
6) In the Eventing section:

a. Select **Eventing**.

b. Set additional settings as appropriate for your implementation. See *Eventing Settings* (on page 64) for details on available settings.

7) In the Directory Services section:

a. In the **Provider URL** field, enter the URL of the JNDI provider for the Connection Factory. For example:

- If you are using a standalone server: t3://<Host_Name>:<Port>
- If you are using a cluster:
  t3://<Host_Name_1>:<Port_1>,<Host_Name_2>:<Port_2>

  b. In the **Initial Context Factory** field, enter the class name of the initial context factory for the JNDI connection. For example, weblogic.jndi.WLInitialContextFactory.

  c. In the **Lookup Name** field, enter the lookup used to test the directory connection. This can be the JNDI name of the JMS connection factory created earlier or the JMS destination.

  d. In the **Security Principal** field, enter the principal to connect to the JNDI provider. If you are using WebLogic, this is the name of a WebLogic user.

  e. In the **Security Credentials** field, enter the credentials to connect to the JNDI provider. If you are using WebLogic, this is the password for the WebLogic user you entered in the **Security Principal** field.

  f. In the **Security Level** field, enter the security level to use when authenticating to the directory service.

8) Click the **Configuration** tab.

9) In the Business Objects section:

  a. Expand a business object type and configure the options to determine the type of notifications you will receive.

  - Select or clear the **Create** option for an object to determine whether you will receive a notification when that object is created.
  - Select or clear the **Update** option for an object to determine whether you will receive a notification when that object is updated.

10) In the Special Operations section:

  a. Expand an operation type and select or clear the **Enabled** option for each operation to determine if it is enabled or disabled.

11) Click **Save and Close**.

Eventing Settings

Setting Name and Description	Default	Valid Ranges/Values
Eventing Select to enable the sending of events for P6, P6 EPPM Web Services, and P6 Integration API.	selected	—

Setting Name and Description	Default	Valid Ranges/Values
Interval The length of time that the Event Notification System uses to determine how often it sends events to the message queue. Specifying a smaller time increases the frequency with which the Event Notification System reports event occurrences to the message queue.	5m	1s-10m
Max Queue Size The amount of memory allocated to the queue for events. Once exceeded, events will be published immediately.	1000	10-5000
Show Costs Select to enable the display of cost fields in event notifications.	selected	—
JMS Connection Factory Specify the JNDI name of the JMS Connection Factory.	—	—
JMS Connection Name Specify the JNDI name of the queue or topic where events are published.	—	—
JMS Destination Security Select to use the user name and password specified when sending messages to JMS queue.	not selected	—
JMS Connection Username Specify the user name to use when sending events to the specified JMS destination specified.	—	—
JMS Connection Password Specify the password to use when sending events to the JMS Destination specified.	—	—

Testing Event Notification

Test event notification to ensure event messages are sent when an event occurs.

To test event notification:

1) If it is not already installed, install P6. See the *P6 EPPM Installation and Configuration Guide*.

> **Note**: If you are using more than one server, install P6 on the local server.

2) Configure WebLogic for eventing:
   - See *Configuring the WebLogic Message Queue* (on page 59) if the queue and the application are on the same domain.
   - See *Sending Events to a Remote WebLogic JMS Server* (on page 66) if the queue and the application are on different servers.

3) Open P6 and create a project.

4) In P6:
   a. Add one or more activities to the project.
   b. Summarize the project to test if an event is generated.

5) Launch the WebLogic **Administration Console** to verify that the event is generated and sent to the queue.

6) In the WebLogic **Administration Console**, expand **Services/Messaging** and click **JMS Modules** in the **Domain Structure** pane.

7) On the **JMS Modules** page, click the module you created for the remote server.

8) On the **Settings** page for the module, click the queue you created for the remote server.

9) On the **Settings** page for the queue, click the **Monitoring** tab.

10) On the **Monitoring** tab, select the option for the remote server destination you created and click **Show Messages**. The event message should be visible in the **JMS Messages** list.

Sending Events to a Remote WebLogic JMS Server

When an event is triggered, the P6 Event Notification system sends the event message to a message queue. If you are using a remote JMS server, then you must configure the local and remote servers to receive these notifications.

The following procedure should be used when the queue is on a different domain than P6. For information about setting up other JMS-based servers, see the vendor documentation.

> **Note**: If you are configuring the WebLogic messaging queue in a cluster, ensure that you have targetted the JMS server to a migratable target. See: *Targeting the JMS Server to a Migratable Target* (on page 59)

To send events to a remote WebLogic JMS server:

1) Start the WebLogic **Configuration Wizard**.

2) In the wizard, create a WebLogic domain on the remote server to which you will be sending the events. See *Creating a WebLogic Domain on a Remote or Local Server* (on page 68).

> ▪ **Note**: Rename the WebLogic administration server to a name that is different from the name you used to deploy the P6 application. For example, RemoteAdminServer.

3) Start the new server and launch the WebLogic **Administration Console**. The new server will act as the remote server.

4) In the WebLogic **Administration Console**:
   a. Create a WebLogic message queue.
      1. Create a new JMS server. See *Creating a JMS Server and Persistence Store* (on page 60).
      2. Create a JMS module. See *Creating a JMS Module* (on page 61).
      3. Create a connection factory. See *Creating a JMS Connection Factory* (on page 61).
      4. Create a new queue or topic. See *Creating a JMS Message Queue and Subdeployment* (on page 62) to see how to create a queue.

      > **Note**: Create a queue to deliver a message to a specific group of users. Create a topic to distribute a message amongst several users.

   b. Configure the trust relationship on the remote server. See *Configuring a Trust Relationship* (on page 68).

5) Create a WebLogic domain on the local server from which the events will be sent. See *Creating a WebLogic Domain on a Remote or Local Server* (on page 68).

6) Start the local server and launch the WebLogic **Administration Console**.

7) In the WebLogic **Administration Console**:
   a. Target the JMS Server to a migratable target. See *Targeting the JMS Server to a Migratable Target* (on page 59)
   b. Create a new JMS server. See *Creating a JMS Server and Persistence Store* (on page 60).
   c. Create a JMS module. See *Creating a JMS Module* (on page 61).
   d. Create a foreign server. See *Creating a Foreign JMS Server* (on page 69).
   e. Configure the trust relationship on the local server. You must use the same credentials that were used on the remote server. See *Configuring a Trust Relationship* (on page 68).

8) Restart both Weblogic **Administration Console** servers (restart the domains, not the machines).

9) If it is not already installed, install P6 EPPM Web Services or P6 on the local server.

10) Launch P6 to configure message queue settings and event notification options. See *Configuring Eventing in the Primavera P6 Administrator* (on page 63). The values you enter in the **Directory Service** and **Eventing** sections of the P6 Application Settings Eventing page should be the the Local JNDI names of the Destination and the Connection Factory in the Foreign JMS Server.

Creating a WebLogic Domain on a Remote or Local Server

Create a WebLogic domain on a remote or local server to define how the server and domain interact.

To create a WebLogic domain:

1) Start the Weblogic **Configuration Wizard** on the local or remote server.
2) In the **Welcome** window, select **Create a new WebLogic domain** and click **Next**.
3) In the **Select Domain Source** window, click **Next** to accept the default selections.
4) In the **Specify Domain Name and Location** window:
    a. Enter the domain name. If you are creating a domain on both a local and remote server, give the domains a different name.
    b. Select the domain location.
    c. Click **Next**.
5) In the **Configure Administrator User name and Password** window, enter the user name and password information and click **Next**.

> **Note**: Make a note of the name and password, you will need this information for the **JNDI Properties Credential** fields and the **JNDI Properties** box in the *Creating a Foreign JMS Server* (on page 69) topic.

6) In the **Configure Server Start Mode and JDK** window:
    a. Select **Production Mode** in the left pane.
    b. Select an appropriate JDK in the right pane.
    c. Click **Next**.
7) In the **Select Optional Configuration** window, click **Next**.
8) In the **Configuration Summary** window, click **Create**.
9) In the **Creating Domain** window, select **Start Admin Server** and click **Done**.

Configuring a Trust Relationship

If you are sending events between different servers, you must establish a trust relationship between the local server, on which P6 is installed, and the remote server, on which you have setup the JMS message queue.

To configure a trust relationship:

1) If it is not already open, launch the WebLogic **Administration Console** on the server where you need to configure the trust relationship.
2) In the WebLogic **Administration Console**, click the name of your domain which is the top element in the **Domain Structure** pane.
3) On the **Settings** page, click the **Security** tab and then the **General** tab.
4) On the **General** tab, expand the **Advanced** section.
5) In the **Advanced** section:

    a. Enter and confirm credentials in the **Credential** and **Confirm Credential** fields.

> **Note**: Make a note of the credentials you enter for the remote server; you must enter the same credentials for both servers.

    b. Click **Save**.

Creating a Foreign JMS Server

Create a foreign JMS server to establish a link between the WebLogic domains.

To create a foreign JMS server:

1) If it is not already open, launch the WebLogic **Administration Console**.
2) In the WebLogic **Administration Console**, expand **Services/Messaging** and click **JMS Modules** in the **Domain Structure** pane.
3) On the **JMS Modules** page, select the module you created for the remote server.
4) On the **Settings** page, click the **Configuration** tab and under **Summary of Resources** click **New**.
5) On the **Create a New JMS System Module Resource** page:
   a. Under **Choose the type of resource you want to create**, select the **Foreign Server** option and click **Next**.
   b. Under **Foreign Server Properties**, enter a name in the **Name** field and click **Next**.
   c. Under **The following properties will be used to target your new JMS system module resource**, ensure the correct server is targeted and click **Finish**.

   > **Note**: If you are deploying to a cluster, the cluster should be selected as the target.

6) On the **Configuration** tab of the **Settings** page, click the name of the new foreign server.
7) On the **Settings** page, click the **Configuration** tab and then click the **General** tab.
8) On the **General** tab:
   a. Enter the URL of the remote server created from step 4 of section **Sending Events to a Remote WebLogic JMS Server** in the **JNDI Connection URL** field.
   For example:
   - If you are using a standalone server: t3://<hostname>:7001.
   - If you are using a cluster: t3://<hostname>:7003,<hostname>:7004.
   b. Enter the password that you used to log on to the remote WebLogic server in the **JNDI Properties Credential** field.
   c. Reenter this password in the **Confirm JNDI Properties Credential** field.
   d. Enter *java.naming.security.principal*=<name> in the **JNDI Properties** box, where *name* is the user name you used to log on to the remote WebLogic server.
   e. Click **Save**.
9) On the **Configurations** tab, click the **Destinations** tab and click **New** under **Foreign Destinations**.

10) On the **Create a New Foreign JMS Destination** page:

    a. Enter a name in the **Name** field.

    b. Enter a JNDI name in the Local JNDI Name field.   Ensure that the Local JNDI name is different from the JNDI name that you had previously assigned to the message queue for the remote server.

    c. Enter the JNDI name that you assigned to the message queue for the remote server in the **Remote JNDI Name** field.

    d. Click **OK**.

11) On the **Configurations** tab, click the **Connection Factories** tab and click **New** under **Foreign Connection Factories**.

12) On the **Create a New Foreign JMS Connection Factory** page:

    a. Enter a name in the **Name** field.

    b. Enter a JNDI name in the **Local JNDI Name** field.   Ensure that the Local JNDI name is different from the JNDI name that you had previously assigned to the connection factory for the remote server.

    c. Enter the JNDI name that you assigned to the connection factory for the remote server in the **Remote JNDI Name** field.

    d. Click **OK**.

Configuring the Security Policy for the WebLogic Message Queue

You can configure the WebLogic security policy to allow only specific users, roles, or groups to access the queue. The following is an example a security policy configured for one user. If you need more information, refer to the WebLogic documentation.

To configure the WebLogic message queue security policy for one user:

1) Launch the WebLogic **Administration console**.

2) In the WebLogic **Administration console**, click **Security Realms** in the **Domain Structure** pane.

3) On the **Summary of Security Realms** page, click the security realm you are using in the **Name** column under **Realms**.

4) On the **Settings** page for the realm, click the **Users and Groups** tab and then click the **Users** tab.

5) On the **Users** tab, under **Users** click **New**.

6) On the **Create a New User** page, enter a name and password in the appropriate fields and click **OK**.

> **Note**: Make a note of the name and password as they will be needed in the *Configuring and Testing the WebLogic Message Queue Security* (on page 71) topic.

7) On the **Settings** page for the user, click the **Roles and Policies** tab and then click the **Realm Policies** tab.

8) On the **Realm Policies** tab, under **Policies** expand **JMS** then expand the module you created for the remote server and click the queue you created for the remote server.

9) On the **Settings** page for the queue, click the **Security** tab and then click the **Policies** tab.

10) On the **Policies** tab:

    a. In the **Policy Conditions** section, click **Add Conditions**.

    b. Under **Choose a Predicate**, select **User** from the **Predicate** list and click **Next**.

    c. Under **Edit Arguments**:

        1. Enter the user name for the user you just created in the **User Argument Name** field and click **Add**. You can now use this user in P6.

        2. Click **Finish**.

11) On the **Settings** page for the queue, click **Save**.

12) Log in to P6 as an administrator.

13) From the **Administer** menu, select **Applications Settings**.

14) On the **Application Settings** page, click **Eventing**.

15) Enter information in the required fields. For more information about information that is required for each field, refer to *Configuring Eventing in P6* (on page 63).

16) Change and test the **Directory Services** user name and password. See *Configuring and Testing the WebLogic Message Queue Security* (on page 71).

Configuring and Testing the WebLogic Message Queue Security

After you create a new user on the Users and Groups tab in Security Realms, you must change the Security Principal and the Security credentials listed under Directory Services in P6.

To configure and test the WebLogic message queue security:

1) Log in to P6 as an administrator.

2) From the **Administer** menu, select **Applications Settings**.

3) On the **Application Settings** page, click **Eventing**.

4) Under **Directory Services**, complete the following:

    a. Enter the new username in the **Security Principal** field.

    b. Enter the new password in the **Security Credentials** field.

    c. Enter the provider URL in the **Provider URL** field.

    d. Enter the initial context factory name in the **Initial Context Factory** field.

    e. Enter the lookup name in the **Lookup Name** field.

> **Note**: The user name and password were set in the *Configuring the Security Policy for the WebLogic Message Queue* (on page 70) topic.

    f. Click Test Connection.

    g. Repeat the steps above with the original username and password to ensure that they no longer work.

5) Under Eventing,enter your eventing settings and test the connection. See *Configuring Eventing in P6* (on page 63).

Using Events with an SSL Connection

Configuring a WebLogic Deployment of P6 to Use an SSL connection

You must configure your WebLogic deployment of P6 if you would like to use a Secure Sockets Layer (SSL) connection.

> **Note**: Please ensure that you have completed the steps in the *Sending Events to a Remote WebLogic JMS Server* (on page 66) section before attempting to configure your WebLogic deployment of P6 to use SSL.

To configure a WebLogic deployment of P6 to use SSL:

▶ See *Configuring an SSL Connection on One Server* (on page 72) if P6 and the WebLogic message queue are on the same server.

▶ See *Configuring an SSL Connection on Different Servers* (on page 72) if P6 and the WebLogic message queue are on different servers.

Configuring an SSL Connection on One Server

You can configure your WebLogic deployment of P6 to use a Secure Sockets Layer (SSL) connection when P6 and the WebLogic message queue are on the same server.

> **Note**: When P6 and the WebLogic message queue are on the same server, they will use the same certificate and trust store.

To configure an SSL connection on one server:

1) Launch the WebLogic **Administration Console**.
2) In the WebLogic **Administration Console**, enable the SSL port. See *Enabling the SSL Port* (on page 73).
3) Log in to P6 as an administrator.
4) From the **Administer** menu, select **Applications Settings**.
5) On the **Application Settings** page, click **Eventing**.
6) In the **Provider URL** field, under **Directory Services**, enter the protocol over SSL (t3s:// or iiops://) and the SSL port.For example, t3s://<hostname>:7002

Configuring an SSL Connection on Different Servers

You can configure your WebLogic deployment of P6 to use a Secure Sockets Layer (SSL) connection when P6 and the WebLogic message queue are on different servers.

> **Note**: This topic assumes that you have already set up a foreign JMS server. See *Creating a Foreign JMS Server* (on page 69).

To configure an SSL connection on different servers:

1) Launch the WebLogic **Administration Console**.

2) In the WebLogic **Administration Console**, enable the SSL port on both servers. See *Enabling the SSL Port* (on page 73).

3) Log in to P6 as an administrator.

4) From the **Administer** menu, select **Applications Settings**.

5) On the **Application Settings** page, click **Eventing**.

6) In the **Provider URL** field, under **Directory Services**, enter the protocol over SSL (t3s:// or iiops://) and the SSL port. For example, t3s://<hostname>:7002.

7) Import the certificate of the server hosting the WebLogic message queue into the trust store of the server hosting P6. Please refer to the WebLogic documentation for detailed instructions.

   ‣ http://download.oracle.com/docs/cd/E21764_01/web.1111/e13707/identity_trust.htm
   ‣ http://download.oracle.com/docs/cd/E21764_01/apirefs.1111/e13952/taskhelp/security/ConfigureIdentityAndTrust.html

8) In the WebLogic **Administration Console**, expand **Services** and click **JMS Modules** in the **Domain Structure** pane.

9) On the **JMS Modules** page, click the module that contains the foreign JMS server you are using.

10) On the **Settings** page for the module, click the foreign JMS server on the **Configuration** tab.

11) On the **Settings** page for the foreign JMS server, use the protocol over SSL (t3s:// or iiops://) and the SSL port in the **JNDI Connection URL** on the **General** tab. For example, t3s://<hostname>:7002.

Enabling the SSL Port

To enable the SSL port:

1) Launch the WebLogic **Administration Console**.

2) In the WebLogic **Administration Console**, expand **Environment** and click **Servers** in the **Domain Structure** pane.

3) On the **Summary of Servers** page, click your server in the **Name** column under **Servers**.

4) On the **Settings** page for the server, click the **Configuration** tab and then click the **General** tab.

5) On the **General** tab, select the **SSL Listen Port Enabled** option and click **Save**.

Configuring your Java client to use an SSL connection

You may configure the Java client to use a Secure Sockets Layer (SSL) connection. See the documentation for WebLogic for more information.

To configure Java to use an SSL connection:

1) In order to use an SSL connection, include the following jar files in you classpath:

   ‣ **wlfullclient.jar**
   ‣ **wlcipher.jar** (This can be found in the <WL_HOME>/server/lib (e.g. C:\Oracle\Middleware\wlserver_10.3\server\lib).)

a. To obtain the jar file *wlfullclient.jar*, run *java –jar wljarbuilder.jar* in <WL_HOME>/server/lib.

b. To obtain the provider URL, you can use either *t3s://* or *iiops://* protocol (t3 or iiop over SSL).

> **Note**: Use the SSL port (7002 by default).

2) To obtain the keystore file that contains the server's certificate:

a. Export the server's certificate from the browser to a certificate file.

b. Import the certificate file into your keystore using java's keytool:

- **keytool -import -trustcacerts -alias demotrust -file server_cert.crt -keystore mykeystore.jks**

3) When running the client, specify the following parameters:

▸ **Dweblogic.security.TrustKeyStore=CustomTrust**

▸ **Dweblogic.security.CustomTrustKeyStoreFileName=mykeystore.jks**

▸ **Dweblogic.security.CustomTrustKeyStorePassPhrase=<your keystore password>**

▸ (Optional) **Dweblogic.security.SSL.ignoreHostnameVerification=true** (Specify this parameter to ignore hostname verification.)

WebLogic Clustering for High Availability

Introduction to High Availability

High availability allows application processing to continue when a server instance fails and provides a reliable environment with minimal or no loss of service. High Availability is often achieved through the use of clusters, units of servers running concurrently to provide application failover and load balancing. Enterprise application deployments can benefit from the additional reliance and flexibility high availability systems provide.

Reasons for Implementing High Availability

Mission critical computer systems need to be available 24 hours a day, 7 days a week, and 365 days a year. However, part or all of the system may be inoperable during planned or unplanned downtime. A system's availability is measured by the amount of time that it provides service over the total time elapsed since the system's initial deployment.

System downtime may be categorized as planned or unplanned. Planned downtime refers to scheduled operations that render the system unavailable. The effect of planned downtime on end users is typically minimized by scheduling downtime when system traffic is slow. Unplanned downtime is any sort of unexpected failure. Unplanned downtime may have a greater effect than planned downtime because it can happen at peak hours, disrupt business operations, or lead to lost productivity or revenue.

When designing your P6 EPPM implementation, it is important to recognize the cost of downtime to understand how your services can benefit from availability improvements.

High Availability Options for P6 EPPM

WebLogic Clustering enables you to provide high availability for P6 EPPM applications, ensuring your services remain scalable and resilient against unexpected failures.

About WebLogic Clustering

WebLogic Server clusters provide scalability and reliability for your applications by distributing the work load among multiple instances of WebLogic Server. Incoming requests can be routed to a WebLogic Server instance in the cluster based on the volume of work being processed. In case of hardware or other failures, session state is available to other cluster nodes that can resume the work of the failed node.

A WebLogic server cluster consists of multiple WebLogic server instances running simultaneously to provide increased scalability and reliability. A cluster appears to clients to be a single WebLogic server instance. The server instances that constitute a cluster can run on the same machine, or be located on different machines. You can increase a cluster's capacity by adding additional server instances to the cluster on an existing machine, or you can add machines to the cluster to host the incremental server instances.

> **Note:** Each server instance in a cluster must run the same version of WebLogic.

Relationship between Clusters and Domains

A domain is an interrelated set of WebLogic server resources that are managed as a unit. A domain includes one or more WebLogic server instances, which can be clustered, non-clustered, or a combination of clustered and non-clustered instances. A domain can include multiple clusters. A domain also contains the application components deployed in the domain, and the resources and services required by those application components and the server instances in the domain. Examples of the resources and services used by applications and server instances include machine definitions, optional network channels, connectors, and startup classes.

In each domain, one WebLogic server instance acts as the Administration Server—the server instance that configures, manages, and monitors all other server instances and resources in the domain. Each domain contains one Administration Server only. If a domain contains multiple clusters, each cluster in the domain has the same Administration Server. All server instances in a cluster must reside in the same domain; you cannot split a cluster over multiple domains. Similarly, you cannot share a configured resource or subsystem between domains.

Clustered WebLogic server instances behave similarly to non-clustered instances, except that they provide failover and load balancing. The process and tools used to configure clustered WebLogic server instances are the same as those used to configure non-clustered instances. However, to achieve the load balancing and failover benefits that clustering enables, you must adhere to certain guidelines for cluster configuration.

High Availability with WebLogic Clustering

P6 EPPM can utilize WebLogic clustering to continue processing data when a server instance fails. You cluster P6 EPPM by deploying it on multiple server instances in the cluster. If a server instance on which P6 EPPM is running fails, then another running server instance on which P6 EPPM is deployed can continue to process data.

For a more comprehensive product overview, documentation, and community forums for WebLogic and its clustering functionality, see http://www.oracle.com/technetwork/middleware/weblogic/overview/index.html

Prerequisites

Prerequisites for WebLogic Clustering

This guide assumes that you have already installed and configured P6 EPPM.

In addition to the prerequisites to enable WebLogic clustering on your initial P6 EPPM deployment, which are described in either Installation Prerequisites of the Installation and Configuration Guide or Prerequisites in Manual Deployment Guide, you will need to install a supported version of Oracle HTTP Server. For a supported version of Oracle HTTP Server, refer to Tested Configurations.

Deploying P6 EPPM with WebLogic clustering requires you to install WebLogic on each machine that will use the P6 EPPM domain, which was created by the P6 EPPM configuration wizard.

> **Note:** You must install the same version of WebLogic on each machine.

Setting Up WebLogic Clustering for P6 EPPM Applications

After you have installed and configured P6 EPPM, you can add new machines, servers, and clusters using the WebLogic Administration Console. You must then add any new servers to different machines and create a cluster to utilize high availability for your application deployments. After creating a cluster, you must add servers to the cluster. To add new servers to your clusters and establish high availability for your P6 EPPM applications, complete the following topics in the order that they are listed:

Adding New Machines in WebLogic

Adding New Servers in WebLogic (on page 78)

Associating Servers with Machines (on page 80)

Adding New Clusters in WebLogic (on page 79)

Assigning Servers to Clusters (on page 79)

After the servers have been added to your cluster, deploy the P6 EPPM WebLogic domain to new machines by creating a zip file of the domain, extracting its contents to the domains folder in Oracle Middleware Home, and then updating nodemanager.properties with the correct ListenAddress hostname. For the procedures to deploy the P6 EPPM domain to new machines, refer to Copying the P6 EPPM Domain to Additional Machines.

When deploying P6 EPPM to managed servers in a cluster running across different physical machines to the domain's administration server, you must modify the Web Service Manager (WSM) Policy Manager on the remote machines so that the additional managed servers can use the WSM Policy Framework. To modify the WSM Policy Manager, refer to Modifying the WSM Policy Manager.

Lastly, configure the WebLogic proxy plugin driver (mod_wl_ohs) of an Oracle HTTP Server (OHS) instance to connect to the P6 EPPM WebLogic cluster using the instructions in Configuring the WebLogic Proxy Plugin Driver of an Oracle HTTP Server and then start your node managers and managed servers using the instructions in Starting the Node Managers and Managed Servers on Additional Machines.

Adding New Machines in WebLogic

To add new machines in WebLogic:

1) Log in to the WebLogic Administration Console with the following URL:

    http://<host_name>:<port>/console

2) In the **Change Center** pane, click **Lock & Edit**.

3) In the **Domain Structure** pane, click **Environment** and then click **Machines**.

4) Click **New**.

5) On the **Create a New Machine** page, complete the following:

    a. In the **Name** field, enter a logical name for the machine (for example, Machine 1).

    b. In the **Machine OS** list, select **Unix** if the machine uses a Unix operating system or select **Other** if the machine uses a non-Unix operating system, such as Windows.

    c. Click **Next**.

6) On the **Create a New Machine** page, complete the following:

    a. In the **Type** list, select the protocol with which the node manager on the machine communicates with its servers. The protocol options are **SSL** (default option), **Plain**, **RSH**, and **SSH**.

    b. In the **Listen Address** field, enter the hostname or IP Address of the remote server.

    c. In the **Port** field, enter the port number for the remote server on which the node manager will run.

    d. Click **Finish**.

7) Click **Activate Changes**.

Adding New Servers in WebLogic

To add new servers in WebLogic:

1) Log in to the WebLogic Administration Console with the following URL:

    http://<host_name>:<port>/console

2) In the **Change Center** pane, click **Lock & Edit**.

3) In the **Domain Structure** pane, click **Environment** and then click **Servers**.

4) Click **New**.

5) On the **Create a New Server** page, complete the following:

    a. In the Server Name field, enter a name for the managed server.

    b. In the Server Listen Address field, enter the hostname or IP Address of the remote server.

    > **Note:** If you plan to add a new managed server on the same physical server as the Administration server, you can leave this field blank. This allows all local listening addresses on the server to be utilized.

    c. In the **Server Listen Port** field, enter the port number from which you want to access the server instance.

> **Tip:** You can more easily manage and maintain the servers in your cluster by using the same port number across all of your managed servers.

    d. Select **Yes, make this server a member of an existing cluster**.

    e. Select a cluster.

    f. Click **Finish**.

6) Click **Activate Changes**.

Adding New Clusters in WebLogic

To add new clusters in WebLogic:

1) Log in to the WebLogic Administration Console with the following URL:

    http://<host_name>:<port>/console

2) In the **Change Center** pane, click **Lock & Edit**.

3) In the **Domain Structure Pane**, click **Environment**, then click **Clusters**.

4) Click **New**, then select **Cluster**.

5) On the **Create a New Cluster** page, complete the following:

    a. Enter a name for the cluster in the **Name** field.

    b. Select a messaging mode for the cluster, **Unicast** or **Multicast**.

      ▸ If you selected **Unicast**, enter a **Unicast Broadcast Channel**.

      ▸ If you selected **Multicast,** enter a **Multicast Address** and a **Multicast Port** number.

6) Click **Ok**.

7) Click **Activate Changes**.

To learn more about WebLogic clusters, cluster configuration, and application deployment, refer to the official WebLogic documentation.

Assigning Servers to Clusters

To assign servers to clusters in WebLogic:

1) Log in to the WebLogic Administration Console with the following URL:

    http://<host_name>:<port>/console

2) In the **Change Center** pane, click **Lock & Edit**.

3) In the **Domain Structure** Pane, click **Environment**, then click **Clusters**.

4) Click the name of the cluster you want to which you want to assign servers.

5) On the **Settings for <Cluster_Name>** page, click the **Servers** tab.

6) On the **Servers** tab, in the **Servers** table, click **Add**.

7) Select **Select an existing server, and add it as a member of this cluster**, to add a preexisting server to the cluster.

8) Select a server in the **Select a server** list.

9) Click **Next**.

10) Click **Activate Changes**.

Associating Servers with Machines

To associate a server with a machine:

1) Log in to the WebLogic Administration Console with the following URL:

   http://<host_name>:<port>/console

2) In the **Change Center** pane, click **Lock & Edit**.

3) In the **Domain Structure** pane, click **Environment** and then click **Machines**.

4) Click the name of the machine to which you want to add a server.

5) Under the Configuration tab, click **Servers**.

6) Click **Add**.

7) On the **Add Server to Machine** page, complete the following:

   a. Select **Select an existing server, and associate it with this machine**.

   b. In the **Select a Server** list, select the name of the server that you want to add to the machine.

   c. Click **Finish**.

8) Click **Activate Changes**.

Modifying the Web Services Manager Policy Manager

P6 EPPM uses Fusion Middleware Infrastructure, which includes Oracle Web Services Manager (WSM) Policy Manager to manage Oracle WSM policies and run the Oracle WSM policy framework. The Oracle WSM Policy Manager reads from a configuration file called wsm-ccw-config.xml that is stored in the <Primavera_Domain>/config/fmwconfig directory. In wsm-ccw-config.xml, the administration server URL is referred as http://localhost:<port>. The wsm-ccw-config.xml file will need to be updated to change localhost references to the actual hostname of the administration server before you copy the domain for every additional managed server that has been added to the P6 EPPM cluster.

To modify the WSM Policy Manager:

1) Go to <Oracle_Middleware_Home>/user_projects/domains/<Primavera_Domain>/config/fmwconfig /.

2) Edit **wsm-ccw-config.xml**.

3) Replace localhost references with the hostname of the machine on which the administration servers runs and then save your changes.

Copying the P6 EPPM Domain to Additional Machines

After you have added new managed servers to your P6 EPPM cluster and modified the WSM Policy Manager, you need to copy the domain to your new machines. WebLogic offers two command-line utilities for this process. The `pack` command creates a template that contains a snapshot of your domain. The `unpack` command uses this template to re-create a domain that contains the packaged applications. For more information on using `pack` and `unpack`, see the WebLogic documentation.

To copy the P6 EPPM domain:

1) Navigate to the `<WebLogic_Home>/wlserver/common/bin` of the machine on which you created the initial P6 EPPM domain.

2) Depending on your operating system, complete the following:

   a. For Windows, run:

```
pack.cmd -managed=true -domain=<Domain_Path>
-template=<Template_Location>\Clustered_Domain.jar
-template_name="Clustered_Domain"
```

   b. For Unix, run:

```
pack.sh -managed=true -domain=<Domain_Path>
-template=<Template_Location>/Clustered_Domain.jar
-template_name="Clustered_Domain"
```

> **Where**:
>
> - <Domain_Path> is the full path to the P6 EPPM domain. For example,
>   <WebLogic_Home>\user_projects\domains\<P6_EPPM_Domain>.
> - <Template_Location> is the full path where P6 EPPM Web Services creates the template. For example, C:\oracle\user_templates.

3) Copy the `Clustered_Domain.jar` file from <Template_Location> to the machines on which you have created additional servers.

4) Navigate to `<WebLogic_Home>/wlsever/common/bin`.

5) Depending on your operating system, compete the following:

   a. For Windows, run:

```
Unpack.cmd -template=<Copy_Location>/Clustered_Domain.jar
-domain=<WebLogic_Home>/user_projects/domains/Clustered_Domain
```

   b. For Unix, run:

```
Unpack.sh -template=<Copy_Location>/Clustered_Domain.jar
-domain=<WebLogic_Home>/user_projects/domains/Clustered_Domain
```

Where:

- <Copy_Location> is the path to where you copied the `Clustered_Domain.jar` file. For example, C:\temp.
- <WebLogic_Home> is the WebLogic home directory. The default is Oracle/Middleware/Oracle_Home.

6) Repeat these steps for each clustered server.

Running Node Manager as a Service

To run the node manager as a background process upon system startup, you can run the node manager as a service on Windows or you can configure the node manager as a daemon on Unix.

For Unix

To start the node manager upon system startup and to run it in the background:

1) Open a terminal and run the following script:

```
nohup
<Oracle_Middleware_Home>/user_projects/domains/<Primavera_Domain>/bi
n/startNodeManager.sh > logfilename.log 2>&1 &
```

2) Create a shell script using the following sample script:

```
#! /bin/bash
#
# wlsnmd Oracle Weblogic NodeManager service
#
# chkconfig:   345 85 15
# description: Oracle Weblogic NodeManager service
case "$1" in
   start)
       export WL_HOME=<Oracle_Middleware_Home>/wlserver/

$WL_HOME/../user_projects/domains/<Primavera_Domain>/bin/startNodeManager.sh
       ;;
   *)
       echo "Usage: $0 {start}"
       exit 1
esac
exit_status
```

3) Save the script to /etc/init.d/nodemgr.
4) Run the following command with root privileges:

```
# chmod +x /etc/init.d/nodemgr
# chkconfig --add nodemgr
```

For Windows

To create and run a node manager as a Windows service:

1) Log in to a machine with administrative privileges.
2) Go to <Oracle_Middleware_Home>/wlserver/server/bin.
   a. Edit **installNodeMgrSvc.cmd**.
   b. Comment out the following lines:

```
set NODEMGR_HOST=<host>
set NODEMGR_PORT=<port>
```

   c. Save your changes and then close the file.
3) Go to <Oracle_Middleware_Home>/user_projects/domains/<Primavera_Domain>/bin.
   a. Edit **installNodeMgrSvc.cmd**.
   b. Comment out the following lines:

```
set NODEMGR_HOST=<host>
set NODEMGR_PORT=<port>
```

   c. Save your changes and then close the file.
4) Open a command prompt and then complete the following:
   a. Change your directory to
      <Oracle_Middleware_Home>/user_projects/domains/<Primavera_Domain>/bin.
   b. Run **installNodeMgrSvc.cmd**.

   - **Note:** If the node manager service has already been installed, the following message will display: CreateService failed - The specified service already exists.

   c. Close the command prompt.
5) Go to Control Panel\All Control Panel Items\Administrative Tools and then click **Services**.
6) Right-click **Oracle WebLogic <primavera_domain> NodeManager** and then click **Start**.

Configuring the WebLogic Proxy Plugin Driver of an Oracle HTTP Server

Typically, you use Oracle HTTP Server to front end a cluster of WebLogic servers. Oracle HTTP Server acts as a front end for the cluster of WebLogic servers. When a request is made to a managed server, the mod_wl_ohs directive specifies a comma-separated list of cluster members to balance the server load. The process works as follows:

1) When mod_wl_ohs receives a request that requires a managed server, it sends that request to one of the WebLogic cluster members that is listed in the directive. At least one server must be available to service the request.
2) The Managed server processes the request and sends a complete list of cluster members back to mod_wl_ohs.

3) When mod_wl_ohs receives the updated list, it dynamically adds any previously unknown servers, that may have been down during the initial request, to the list of known servers, which enables all future requests to be load balanced across the full cluster member list defined in the mod_wl_ohs configuration.

This process also enables new managed servers to be added to the cluster without having to manually update mod_wl_ohs as long as the new managed server host and port is determined ahead of time.

In an example scenario, assume that the WebLogic Cluster consists of WLS1, WLS2, and WLS3. Also assume that all three managed servers have been added to the mod_wl_ohs clustering directive, but only WLS1 and WLS2 are known to mod_wl_ohs during an initial request because WLS3 is offline.

When mod_wl_ohs first receives a request, it attempts to send that request to either WLS1 or WLS2. If during the request WLS3 becomes active, the list of available servers is updated to include WLS3 for future requests. If, however, WLS1 and WLS2 are unavailable, but WLS3 becomes available during the request, the request is rejected because mod_wl_ohs has no way of knowing WLS3 exists.

To configure mod_wl_ohs:

1) Go to <OHS_Middleware_Home>/user_projects/domains/<Primavera_Domain>/config/fmwconfig/components/OHS/instances/<instance_name>.

2) Edit **mod_wl_ohs.conf**.

3) Add the following information to mod_wl_ohs:

```
<Location /p6>
WLSRequest On
WebLogicCluster <WebLogic_server1>:<port1>, <WebLogic_server2>:<port2>,
<WebLogic_server3>:<port3>
</Location>
<Location /p6tmweb>
WLSRequest On
WebLogicCluster <WebLogic_server1>:<port1>, <WebLogic_server2>:<port2>,
<WebLogic_server3>:<port3>
</Location>
<Location /p6ws>
WLSRequest On
WebLogicCluster <WebLogic_server1>:<port1>, <WebLogic_server2>:<port2>,
<WebLogic_server3>:<port3>
</Location>
```

4) Restart Oracle HTTP Server.
   a. Go to <OHS_Middleware_Home>/user_projects/domains/base_domains/bin.
   b. Depending on your operating system, complete the following:
      ▪ For Unix, run the following in a terminal:

```
./stopComponent.sh <component_name>
./startComponent.sh <component_name>
```

▸ For Windows, run the following in a command prompt:

```
stopComponent.bat <component_name>
startComponent.bat <component_name>
```

Starting Node Managers and Managed Servers on Additional Machines

After the P6 EPPM domain has been modified on machines on which they reside, you must start the node manager on each machine to control the managed servers from the WebLogic Administration Console.

To start a managed server:

1) Log in to the Domain's Administration Console.
2) In the left pane of the Administration Console, expand **Environment** and select **Servers**.
3) In the right pane, select the **Control** tab.
4) In the **Server Status** table, select the check box next to the name of the server you want to start and click **Start**.
5) Click **Yes** to confirm.

To start the node manager:

1) Go to
   `<Oracle_Middleware_Home>/user_projects/domains/<Primavera_Domain>/bin`.
2) Depending on your operating system, complete the following:
   ▸ For Windows, run:

```
startManagedWeblogic.bat <server_name> t3://<administration_server_name>:<port>
```

   ▸ For Unix, run

```
./startNodeManager.sh
```

Where to Go from Here

Now that you have deployed P6 in WebLogic, you can begin using P6. You may find the following guides useful for help getting started:

▶ *P6 EPPM System Administration Guide*
▶ *P6 EPPM Application Administration Guide*
▶ *P6 EPPM Database Administration Guide*
▶ *P6 Help*
▶ *P6 Team Member User's Guide*

Depending on what other applications you are using, you may also need the following guides:

P6 Professional

▶ *P6 Professional Installation and Configuration Guide* for your database

Primavera Data Warehouse

▶ *Primavera Data Warehouse Installation and Configuration Guide*

Analytics

▶ *Analytics Installation and Configuration Guide*

Additional Applications

▶ *P6 EPPM Oracle Analytics Server Configuration Guide*
▶ *P6 EPPM BPM Configuration Guide*
▶ *P6 EPPM Content Repository Configuration Guide*
▶ *P6 EPPM Oracle Identity Manager Configuration Guide*
▶ *Primavera Oracle Access Manager Configuration Guide*
▶ *Primavera Oracle Enterprise Manager Configuration Guide*

Copyright

Oracle Primavera P6 EPPM WebLogic Configuration Guide for On-Premises