# HP Project and Portfolio Management Center

Software Version: 7.1

# System Administration Guide and Reference

Document Release Date: March 2007

Software Release Date: March 2007

## Legal Notices

## Documentation Updates

This manual’s title page contains the following identifying information:

- Software version number, which indicates the software version
- Document release date, which changes each time the document is updated
- Software release date, which indicates the release date of this version of the software

To check for recent updates, or to verify that you are using the most recent edition of a document, go to: http://ovweb.external.hp.com/lpe/doc_serv/.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

# Support

## Mercury Product Support

You can obtain support information for products formerly produced by Mercury as follows:

- If you work with an HP Software Services Integrator (SVI) partner (www.hp.com/managementsoftware/svi_partner_list), contact your SVI agent.
- If you have an active HP Software support contract, visit the HP Software Support site and use the Self-Solve Knowledge Search to find answers to technical questions.
- For the latest information about support processes and tools available for products formerly produced by Mercury, we encourage you to visit the HP-Mercury Software Support web site at: support.mercury.com.
- Contact your HP Sales Representative if you have additional questions.

## HP Software Support

You can visit the HP Software Support web site at www.hp.com/managementsoftware/services.

HP Software online support provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To find more information about access levels, go to: www.hp.com/managementsoftware/access_level.

To register for an HP Passport ID, go to: www.managementsoftware.hp.com/passport-registration.html.

# Table of Contents

# List of Figures

# List of Tables

# 1 Introduction

In This Chapter:

# Administering the HP Project and Portfolio Management Center System

This document provides information about how to install, configure, and maintain the HP Project and Portfolio Management Center (PPM Center) system, including:

- The PPM Server or server cluster
- The Oracle® database and database schema used with PPM Center
- Other system components

If you are not installing PPM Center for the first time, but need instructions on how to upgrade from an earlier version, see the *Upgrade Guide.*

The chapters in this document provide the following information about PPM Center and how to administer the system:

- Overview of PPM Center system architecture and of single-server and server cluster system configuration (Chapter 2, *System Overview,* on page 23)
- Information about product licensing and optional programs that you can install (Chapter 3, *Installing PPM Center,* on page 37)
- Instructions on how to create the required database schemas, verify installation, and install service packs and HP Deployment Management Extensions and Accelerators (Chapter 3, *Installing PPM Center,* on page 37)
- Details on how to configure all components of the PPM Center system and to start and stop the PPM Server. (Chapter 4, *Configuring the System,* on page 65)
- Information that PPM Center users need to know in order to use the PPM Workbench (Chapter 4, *Configuring the System,* on page 65)
- Advanced configuration information, including details on how to configure an external Web server and PPM Server clusters (Chapter 5, *Advanced System Configuration,* on page 95)
- Information on how to integrate PPM Center with an LDAP server (Chapter 5, *Advanced System Configuration,* on page 95)
- Details on how to maintain the PPM Center and the database after installation and configuration (Chapter 6, *Maintaining the System,* on page 133)

- Information about the kinds of performance issues that can arise, and how to identify and resolve them (Chapter 7, *Improving System Performance,* on page 157)

- Information on how to migrate entire instances of PPM Center, and on how to migrate just the database schemas (Chapter 9, *Migrating Instances,* on page 211)

- Details on how to use the HP entity migrators to migrate specific kinds of PPM Center entities and associated objects between instances of PPM Center (Chapter 8, *Migrating Entities,* on page 175)

- PPM Server configuration parameters (Appendix A, *Server Configuration Parameters,* on page 225)

- Details about PPM Center directories and the scripts and tools they contain (Appendix B, *Server Directory Structure and Server Tools,* on page 277)

- Checklists of the tasks to perform on the application server (or servers), database server, client machines, and the network before you install and configure PPM Center for your organization (Appendix C, *Preinstallation Checklists,* on page 291)

This document is written for:

- Application developers and configurators

- System and instance administrators

- Database administrators (DBAs)

The information in this document is directed toward users who are moderately knowledgeable about enterprise application development and skilled in enterprise system and database administration.

# Related Documents

The following documents provide installation information for system administrators and DBAs:

- *System Requirements and Compatibility Matrix*

  Before you install PPM Center, check this document to make sure that your operating environment meets *all* of the minimum system requirements.

- *Release Notes*

  This document provides product information that is not included in the regular documentation set.

- *Customizing the Standard Interface*

  Refer to this document for information on how to tailor the look and feel of the PPM Center standard interface for your organization.

- *Creating Portlets and Modules*

  Refer to this document for information on how to configure the PPM Center standard interface to suit your organization's business needs.

- *Upgrade Guide*

  If you plan to upgrade from an earlier version of PPM Center, see this guide.

Additional documents that you might find useful as you configure or maintain PPM Center include:

- *Commands, Tokens, and Validations Guide and Reference*
- *Open Interface Guide and Reference*
- *Service-Oriented Architecture: Web Services Guide*
- *Reports Guide and Reference*
- *Security Model Guide and Reference*
- *HP-Supplied Entities Guide* (includes descriptions of all portlets, request types, and workflows in PPM Center)

# Accessing PPM Center Documentation

You can access PPM Center documentation from the product itself.

To access the PPM Documentation Library from the standard interface:

1. Log on to PPM Center.
2. From the menu bar, select **Product Information > Library.**

To access the documentation library from the PPM Workbench:

1. Log on to PPM Center.
2. From the menu bar, select **Administration > Open Workbench.**

   The PPM Workbench opens.
3. From the shortcut bar, select **Product Information > Library.**

The PPM Download Center gives you access to the same PDF files that are available through the product itself, and to documents that are only available at that location. To access the login page for the PPM Download Center, go to itg.merc-int.com/support/download/login.jsp.

# 2 System Overview

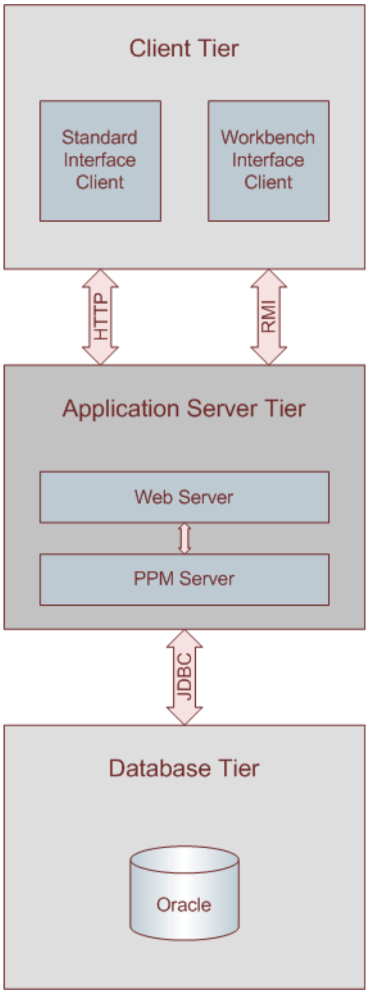
In This Chapter:

# Overview of PPM Center Architecture

PPM Center is based on a three-tier architecture that consists of:

- An unlimited number of client browsers (client tier)
- One or more middle-tier J2EE servers (application server tier)
- A single Oracle relational database (database tier)

Figure 2-1. PPM Center Architecture



Browser clients use HTTP or HTTPS (HTTPS requires an external Web server) to communicate with the PPM Center Web and application servers. PPM Workbench clients (Java™ applet) use Remote Method Invocation (RMI). The following sections provide information about each tier.

## Client Tier

The client tier of the system consists of:

- The PPM Center standard interface. The standard interface is rendered using Java Server Pages (JSP) and is accessed using a Web browser.
- The PPM Workbench interface is displayed using a Java applet installed on the client machine, and is started using the Sun Java plug-in to a Web browser.

The client and application server tiers communicate as follows:

- For the standard interface, the client and application server communicate using HTTP or HTTPS, with no code required on client machines. The client accesses information from the database through the J2EE application server using a shared database session pool.

To use HTTPS, you must also use an external Web server.

- For the PPM Workbench interface, the client and application server communicate using Remote Method Invocation (RMI) or Secure Remote Method Invocation (SRMI), which is optimized for use in PPM Center.

  The architecture and communication protocols are created to minimize the number of round trips between the applet and server, and the volume of data transferred.

For more information about the PPM Center standard and PPM Workbench interfaces, see the *Getting Started* guide.

## Application Server Tier

The application server:

- Runs on the Microsoft® Windows®, Sun Solaris, HP-UX, IBM AIX, and Red Hat Linux, and SUSE Linux platforms
- Uses the JBoss Application Server, which has full J2EE 1.3 (Java 2 Platform, Enterprise Edition) support
- Houses workflow, scheduling, notification, and execution engines that drive automated tasks such as code deployment to remote systems, dynamic routing, and email notifications
- Can run on multiple machines as a cluster to improve performance and scale hardware as usage increases

- Can run with external Web servers such as Sun Java System Web Server, Microsoft IIS, Apache HTTP Server, Apache-based Web Server (from HP), and IBM HTTP Server

➤ For detailed information on which Web server versions PPM Center supports, and on which platforms, see the *System Requirements and Compatibility Matrix.*

- Maintains a database connection pool that caches connections to the database, which eliminates the need to restart the application server if the database shuts down for scheduled maintenance or because of system failure

The application server and the PPM Web server communicate using Apache JServ Protocol version 1.3, or AJP13. The AJP13 protocol is similar to HTTP that has been optimized for performance. The application server and database tiers communicate using Java Database Connectivity (JDBC).

For more information about configuring an external Web server, see *Configuring an External Web Server* on page 101.

## Database Tier

The database tier consists of an Oracle database that contains the tables, procedures, PL/SQL packages, and other components that the PPM Center products use. All transaction, setup, and auditing data is stored in the database. PPM Center can run on a single database instance, or can leverage Oracle RAC (Real Application Cluster) configuration for load balancing, redundancy, and failover.

➤ The database consists of the following two database schemas:

The central schema (typically named `mitg`) contains the core PPM Center data model and PL/SQL package code. The core data model contains all PPM Center configuration and transaction data.

The Reporting Meta Layer (RML) schema contains a set of database views to facilitate reporting on PPM Center data.

PPM Center supports the following Oracle database features:

- A relational data model
- Use of Oracle stored procedures to implement business logic (for example, workflow processing)
- Use of a database connection pool to eliminate the need to create a separate database session for each user or transaction

- Database caching of frequently used data, programs, and procedures to improve performance

# System Configurations

The three-tier architecture of PPM Center supports a variety of system configurations. You can deploy PPM Servers in a single-server configuration or a server cluster configuration. The following sections provide detailed information about these configurations.

## Single-Server Configurations

PPM Center configurations are typically single-server configurations that consist of one PPM Server and one Oracle database. The single PPM Server handles the entire user load and functions as the Web server. It also houses the file system for the program code, reports, execution logs, and attachments files. The Oracle database stores all other data.

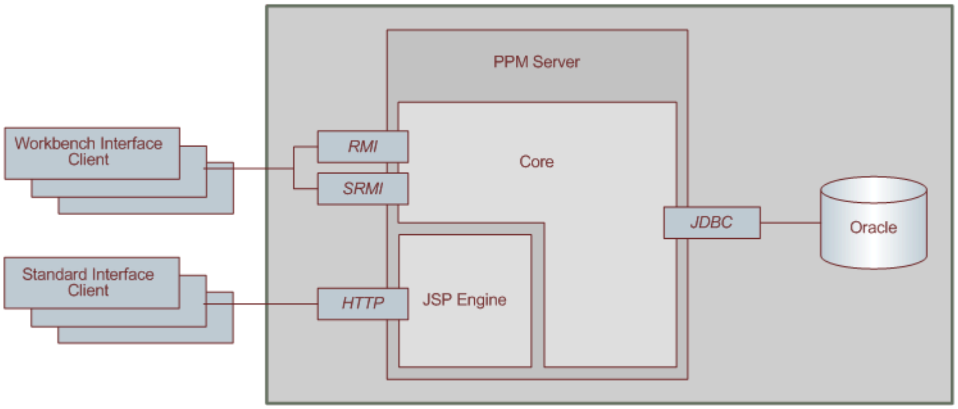
You can set up single-server configurations in the following ways:

- Single-server/single-machine configuration
- Single-server/multiple-machine configuration
- Single-server/external Web server configuration

### Single-Server/Single-Machine Configuration

The single-server/single-machine configuration illustrated in *Figure 2-2* consists of one machine that hosts both the PPM Server and the Oracle database.

Figure 2-2. Single-server/single-machine configuration



Standard interface clients communicate with the PPM Server using HTTP, or, for secure communication, HTTPS (requires that you use an external Web server). PPM Workbench interface clients communicate with the PPM Server using RMI, or, for secure communication, SRMI.

The machine that houses the PPM Server also contains the Oracle database. The PPM Server uses JDBC to communicate with the Oracle database.

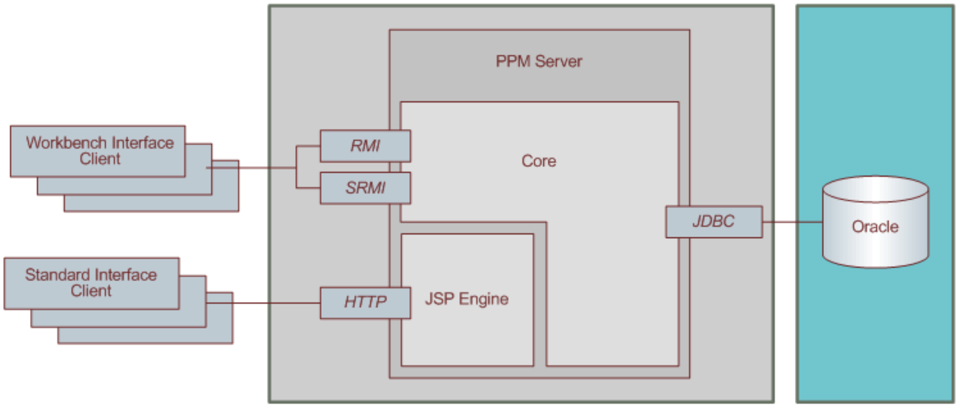
An organization typically uses this configuration if it requires a dedicated machine for all PPM Center services and database operations. User load, transaction capacity, and system performance depend on the available resources on a machine. This configuration does not support load balancing or server failover features.

For information about how to set up a single-server/single-machine configuration, see Chapter 3, *Installing PPM Center,* on page 37.

## Single-Server/Multiple-Machine Configuration

In the single-server/multiple-machine configuration illustrated in *Figure 2-3*, the PPM Server and the Oracle database reside on separate machines. This configuration offers additional performance capacity and modularizes the maintenance of the application server and database tiers. The separate machines can run on different operating systems, thereby allowing greater flexibility.

Figure 2-3. Single-server/multiple-machine configuration



Standard interface clients communicate with the PPM Server using HTTP, or HTTPS for secure communication. (To use HTTPS, you must use an external Web server.) PPM Workbench interface clients communicate with the PPM Server using RMI, or SRMI for secure communication. The PPM Server and Oracle database use JDBC to communicate.

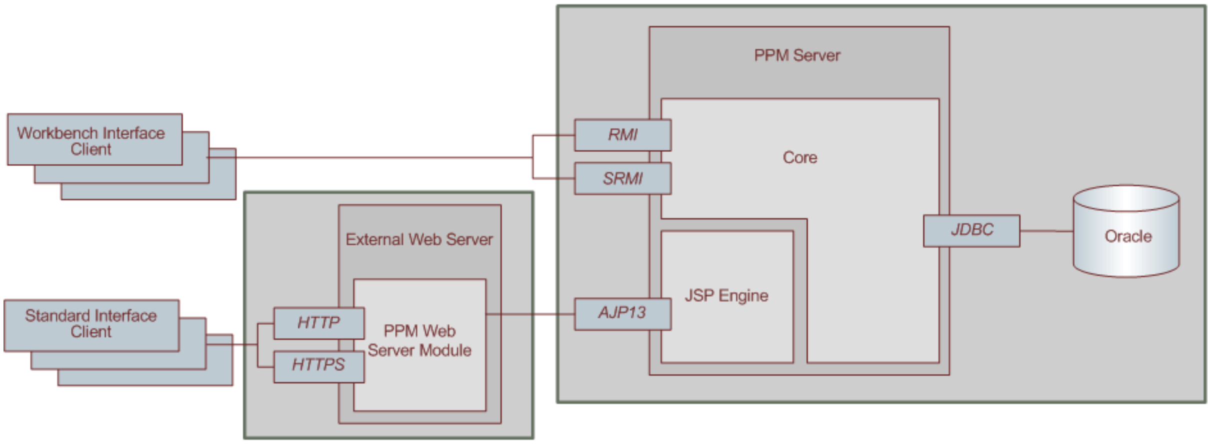
An organization typically uses the single-server/multiple-machine configuration if it requires a separate machine for database operations. User load, transaction capacity, and system performance depend on the resources available on the PPM Server machine. This configuration does not support load balancing or server failover features.

For information about how to set up a single-server/multiple-machine configuration, see Chapter 3, *Installing PPM Center,* on page 37.

## Single-Server/External Web Server Configuration

In the single-server/external Web server configuration illustrated in *Figure 2-4*, Web traffic comes into the Web server and is then passed to PPM Center. The external Web server and the PPM Server communicate using AJP13, a proprietary protocol that is more efficient for this configuration type than HTTP or HTTPS.

Figure 2-4. Single-server/external Web server configuration



- Standard interface clients communicate with an external Web server using HTTP, or, for secure communication, HTTPS. The external Web server and PPM Servers use AJP13 to communicate.
- PPM Workbench interface clients communicate directly with the PPM Server using RMI, or, for secure communication, SRMI.
- The machine that houses the PPM Server also contains the Oracle database. The PPM Server communicates with the Oracle database using JDBC.
- The PPM Server and Oracle database can reside on separate machines.

This configuration is suitable if your organization:

- Already uses a standard Web server within the network infrastructure.
- Must prevent clients from having direct access to the PPM Server.

IT departments often have standards for the Web server used for HTTP traffic. Running the HTTP listener allows for PPM Center integration with enterprise-specific architecture.

System administrators typically prefer HTTP traffic configured on port 80. On UNIX® systems, processes must run as root to listen on a port number below 1024. However, HP recommends that you not run the PPM Server run as root. If you want to configure HTTP traffic on a port number below 1024, consider integrating with an external Web server.

As with other single-server configurations, user load, transaction capacity, and system performance depend on available resources on the PPM Server machine. This configuration does not support load balancing and server failover features.

➤ HP recommends that you use the internal Web server built into the PPM Server unless you have the kind of special Web server requirements described in this section.

For information about how to set up a single-server/external Web server configuration, see Chapter 3, *Installing PPM Center,* on page 37 and Chapter 5, *Advanced System Configuration,* on page 95.

For a list of supported Web servers, see the *System Requirements and Compatibility Matrix*. For information on how to access this and other PPM Center documents, see *Accessing PPM Center Documentation* on page 21.

## Server Cluster Configurations

Server cluster configurations improve performance on systems that handle high transaction volumes or large numbers of concurrent users. In addition to handling higher user loads and providing greater scalability, server cluster configurations support load balancing and server failover features to help ensure that mission-critical systems provide constant and optimal access to users.

To handle large numbers of concurrent users, server cluster configurations use either an external Web server or a hardware-based load balancer to distribute user connections evenly across multiple PPM Servers. If a PPM Server shuts down, the activities running on that server are automatically transferred to an available PPM Server in the cluster. This server failover feature helps ensure that PPM Center system services such as email notifications and scheduled executions remain operational.

Server cluster configurations contain two or more PPM Servers and an Oracle database. The first PPM Server installed and configured is the *primary server*. The other server (assuming a two-server setup) is the *secondary server*. The two servers can act as peers in a load-balancing situation, or one can act as a backup machine for the other.

➤ A server cluster setup can include multiple databases. If a database in a setup such as this goes down, the Oracle JDBC driver manages database connectivity.

You can implement server cluster configurations on a single machine or on multiple machines. To run multiple PPM Servers on a single machine, the machine's memory capacity and CPU usage must meet the same memory and CPU requirements for multiple servers. To run multiple servers on multiple machines, the servers must share a common file system for reports, execution logs, and attachment files. Although each machine can contain its own instance of the PPM Center application code, only a single copy is required for each machine, regardless of the number of servers running on that machine.

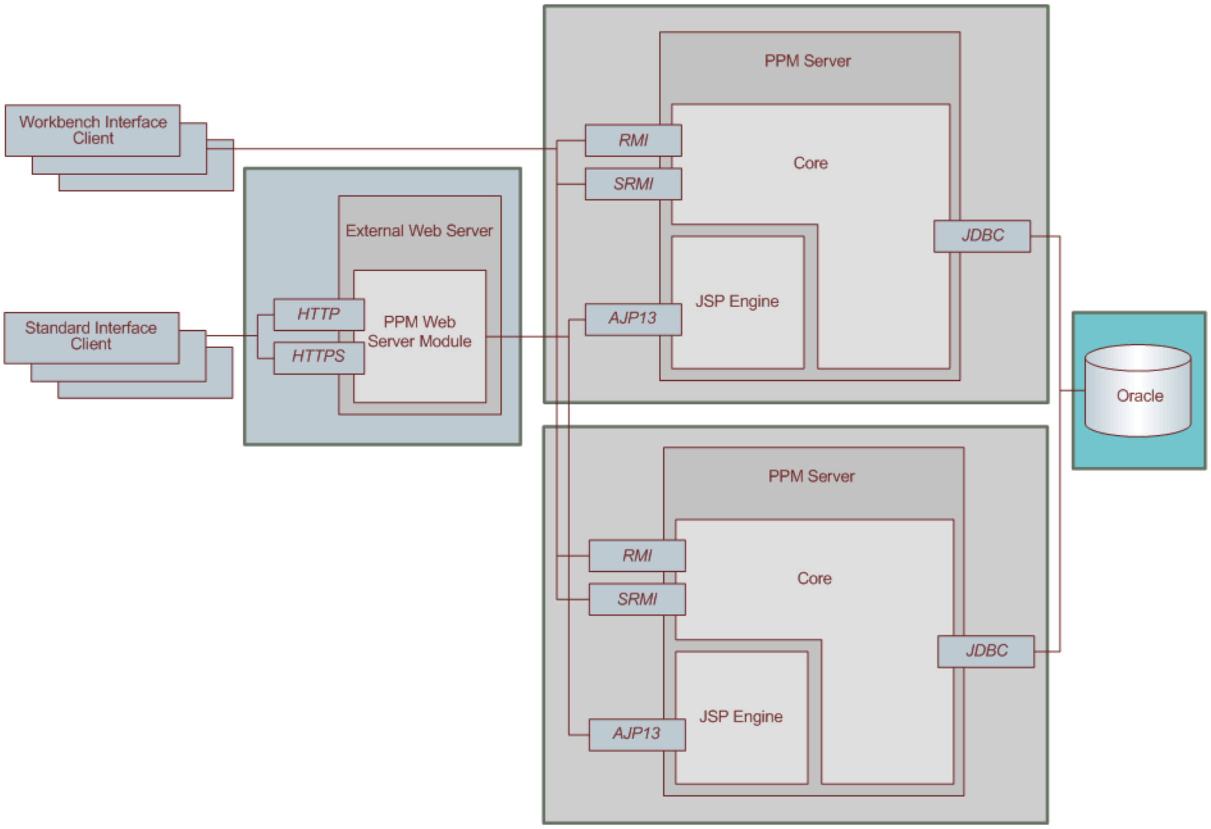
You can set up server clusters with an external Web server, or with a hardware load balancer. The following sections describe these two setups.

### Server Cluster/External Web Server Configuration

The server cluster/external Web server configuration (see *Figure 2-5* on page 32) distributes client connections evenly among any number of PPM Servers, based on Web traffic and server load. This configuration is typically used for organizations that need to load-balance Web traffic across multiple PPM Servers (as an alternative to hardware-based load balancing). It can also be useful to an organization that already uses a standard Web server within its network infrastructure.

You can usually improve user load, transaction capacity, and system performance with this configuration. The extent of improvement depends on the number of PPM Servers in the cluster and their available resources. This configuration supports load balancing and server failover features.

Figure 2-5. Server cluster/external Web server configuration



The external Web server listens for HTTP or HTTPS requests from standard interface clients. PPM Servers run in the background and are transparent to users. Users see only the URL to the external Web server.

The HP PPM Web server module forwards HTTP or HTTPS requests to one of the PPM Servers. The HP PPM Web server module and the PPM Servers communicate using AJP13.

The PPM Servers also accept RMI or SRMI connections from PPM Workbench users who run applets in browsers to directly connect to the PPM Server using this protocol. The PPM Server uses JDBC to communicate with the Oracle database.

➤ You cannot use a single Web server installation on a Windows-based system for multiple PPM Center instances. If you must use an external Web server for multiple PPM Center instances, HP recommends that you either use a UNIX machine to host the Web server, or use a hardware load balancer.

### Software Load Balancing

You can use the PPM Center Web server module as the software load balancer for a PPM Server cluster configuration. In this configuration, the PPM Servers running in the cluster do not accept HTTP requests directly.

The request sequence is as follows:

1. A user submits an HTTP request to the Web server.
2. The Web server forwards the request to the HP PPM Web server module.
3. The HP PPM Web server module sends the request to a PPM Server.

### Integrating with a Single Sign-On Product

With the server cluster/external Web server configuration, you can implement single sign-on using a product such as *e*Trust SiteMinder. For instructions on how to implement single sign-on with PPM Center, see *Implementing Single Sign-On with PPM Center* on page 128.

### Using SSL Accelerators

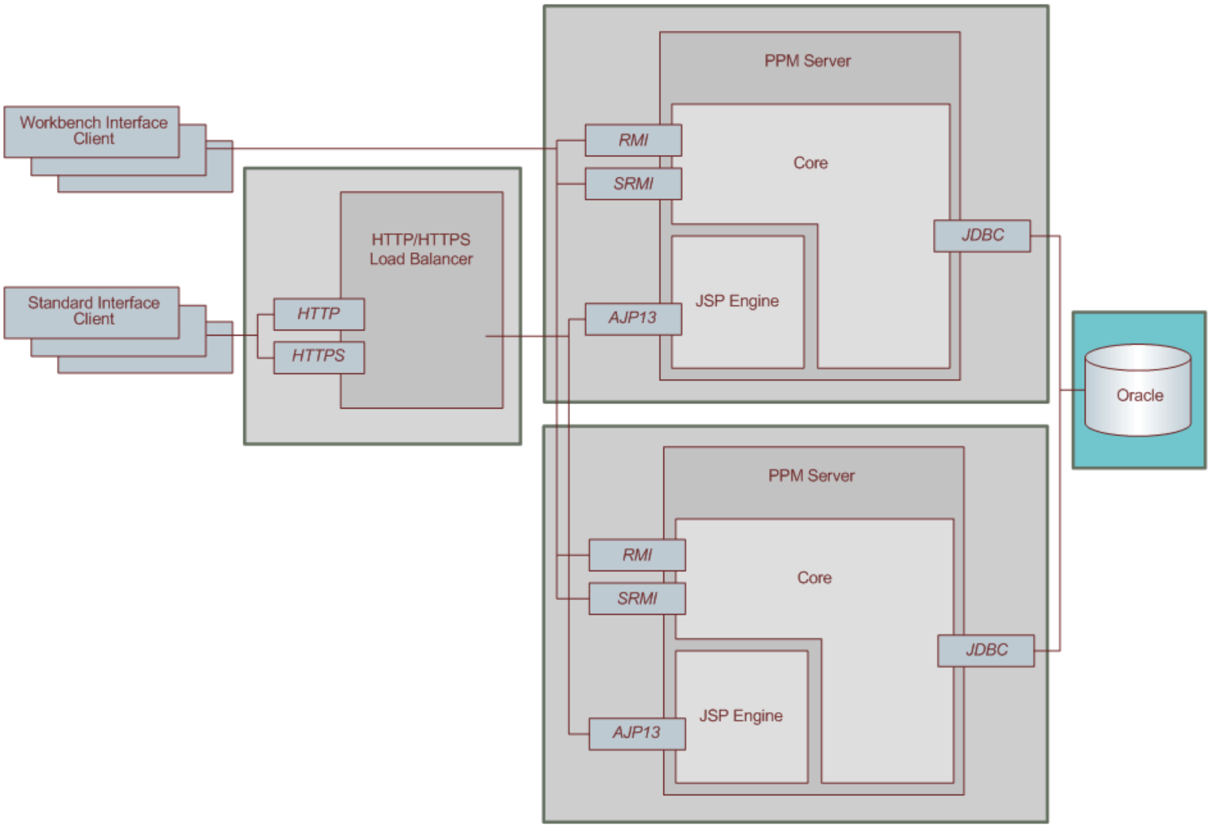
For PPM Server cluster configurations running HTTPS, you must integrate an external Web server that supports the appropriate accelerator to leverage a hardware-based SSL accelerator.

The external Web server and PPM Servers communicate using AJP13, a proprietary protocol that can be more efficient than HTTP for communicating with PPM Servers using an external Web server. For information about how to set up a server cluster with an external Web server, see Chapter 5, *Advanced System Configuration,* on page 95.

## Server Cluster Hardware Load Balancer Configuration

The server cluster/hardware load balancer configuration (illustrated in *Figure 2-6*) is similar to the server cluster/external Web server configuration. However, in place of an external Web server, a hardware load balancer is used to balance client HTTP sessions across PPM Servers. This configuration enables the even distribution of client connections among PPM Servers based on server load and availability.

Figure 2-6. Server cluster/hardware load balancer configuration



In this configuration:

- Standard interface clients communicate with PPM Servers using HTTP (or, for secure communication, HTTPS) through the use of a hardware load balancer. The hardware load balancer behaves like a reverse proxy server and PPM Servers listen for HTTP or HTTPS requests that it distributes.

➤ To use HTTPS, you must also use an external Web server.

  Many hardware load balancers support handling HTTPS and forwarding plain HTTP. In this case, the hardware load balancer handles the encryption and decryption of requests, and the PPM Servers perform other tasks. Setting up your system this way can improve system performance.

- PPM Workbench interface clients communicate directly with the PPM Server using RMI, or SRMI for secure communication.
- The PPM Server and Oracle database reside on separate machines and communicate with each other using JDBC.

➤ Although *Figure 2-6* on page 34 illustrates multiple servers and just a single database, the system can support multiple databases or a single database mirrored for redundancy across multiple machines (equal to the number of PPM Servers.)

Using this configuration improves user load distribution, transaction capacity, and system performance. The degree of improvement depends on the number of PPM Servers in the cluster and the resources available to each. Load balancing and server failover features are supported in this configuration.

For information about how to set up a server cluster/hardware load balancer configuration, see Chapter 5, *Advanced System Configuration,* on page 95.

# 3 Installing PPM Center

In This Chapter:

# Key Considerations

To prepare to install PPM Center, review the key considerations addressed in this section.

➤ For information about how to upgrade to PPM Center 7.1, see the *Upgrade Guide.*

## Installing for the First Time

To prepare to install PPM Center 7.1, perform the following tasks:

1. Read the rest of this chapter.

2. Read the *System Requirements and Compatibility Matrix,* which is described in *Related Documents* on page 20.

3. Read the *Release Notes,* which are described in *Related Documents* on page 20.

4. To ensure that you have performed all required preinstallation tasks, use the preinstallation checklists provided in Appendix C, *Preinstallation Checklists,* on page 291.

5. If you plan to install one of the HP Deployment Management Extensions, or an HP Accelerator, see the documentation for the product.

   ➤ For information on how to access documentation for HP Migrators, Accelerators, and HP Deployment Management Extensions, see *Accessing PPM Center Documentation* on page 21.

6. Make certain that you have the valid licenses required for all of the products you plan to install.

7. For instructions on how to install PPM Center, see *Installing PPM Center* on page 55.

   That section addresses how to:

   - Prepare to install the product
   - Install the product
   - Verify the installation

8. Configure the PPM Server and system environment.

   For information about how to configure PPM Center, see Chapter 4, *Configuring the System,* on page 65.

9. Install and configure optional products you have purchased to work with PPM Center.

   ➤ After you install and configure PPM Center, you can install Extensions, Accelerators or Migrators in any order you choose. For information about how to install and configure optional products, see *Optional Installations* on page 63.

## Installing Document Management

HP provides you with both the HP-configured EMC Documentum code and the EMC Documentum documentation required to install the PPM Center document management functionality. If you plan to set up document management, you must perform a separate installation. For more information, see the *Document Management Guide and Reference.*

## Installing HP Object Migrator or HP GL Migrator

If you are running PPM Center in the Oracle environment, and plan to use HP Object Migrator or HP GL Migrator software, you must consult not only the installation instructions in this document, but also the instructions in the HP Object Migrator or HP GL Migrator documentation.

For information about the HP Object Migrator and the HP GL Migrator, see the *HP Object Migrator Guide* and the *HP GL Migrator Guide,* respectively.

## Installing an HP Deployment Management Extension

If you purchased an HP Deployment Management Extension, be sure to consult not only the installation instructions in this document, but also the instructions in the HP Deployment Management Extensions documentation.

To complete an Extension installation successfully, you must ensure that you have the required system privileges. For information about these privileges, and how to grant them, see *Key Decisions* on page 41.

## Obtaining License Keys

Make sure that you have purchased the HP products you intend to install (you can purchase and install additional products later), and that you have obtained the required license file. You must have a license file for the purchased version. PPM Center license keys are delivered in the `license.conf` file, which you can find in the `<*PPM_Home*>/conf` directory after installation.

After you purchase HP Deployment Management Extensions or HP Migrators, you receive a user name and password that you can use to download product code and documentation from the PPM Download Center. To access the login page for the PPM Download Center, go to itg.merc-int.com/support/download/login.jsp.

## Checking System Requirements

Before you start to install PPM Center, make sure that your system environment meets all the requirements. For information about the system requirements, see the *System Requirements and Compatibility Matrix.* For information on how to access this and other PPM Center documents, see *Accessing PPM Center Documentation* on page 21.

Appendix C, *Preinstallation Checklists,* on page 291 provides checklists that include all of the preliminary tasks you must perform on your network, the PPM Server(s), the database server, and PPM Center clients. HP recommends that you use these checklists to track the tasks that you must perform before you start to install PPM Center.

## Key Decisions

This section addresses several decisions you must make before you begin to install your PPM Center products.

Table 3-1. Decisions to make before you install (page 1 of 2)

| Decision | What to Consider |
|---|---|
| When do I configure the PPM Server? | Before you can start the PPM Server, you must configure it. The installer prompts you for several server parameter values.<br>If you do not configure during installation, the installer saves the values you provide to the server configuration file, and you can complete server configuration after installation, without having to reenter the values.<br>If the server information you provide (for example, valid port numbers) is unavailable during installation, you might have to configure the server after installation. For instructions, see *Configuring or Reconfiguring the Server* on page 69. |
| When do I create the database schemas? | The PPM Server requires two database schemas to store application data. You can create these schemas before you install PPM Center, or you can create them automatically during installation.<br>To create the schemas before installation, follow the instructions provided in *Creating the Database Schemas* on page 52. If you set up the schemas before installation, the installer populates them with the entities and data required to run the PPM Server. |
| When do I set up grants to the database schema? | To improve PPM Center performance, the installer rebuilds statistics for the Oracle optimizer during installation.<br>You cannot successfully complete the installation until you grant privileges and rebuild the statistics. |

Table 3-1. Decisions to make before you install (page 2 of 2)

| Decision | What to Consider |
|---|---|
| What privileges do I grant the database schema user? | To rebuild the statistics, the PPM Center database schema user must be granted the following privileges (as SYS DBA on Oracle):<br>■ `grant select on v_$parameter to <PPM_Schema>`<br>■ `grant select on v_$mystat to <PPM_Schema>`<br>■ `grant select on v_$process to <PPM_Schema>`<br>■ `grant select on v_$session to <PPM_Schema>`<br>■ `grant execute on dbms_stats to <PPM_Schema>`<br>If you have access to SQL*PLUS, you can run the script `sys/GrantSysPrivs.sql` (located in the `mitg710/sys` directory), which grants all required privileges for you. You can run the script before installation (as SYS DBA) or during installation. If you are logged on as SYS DBA, you can run the script after installation. In this case, the installer does not gather statistics or install Best Practices. |
| Do I run the install program in graphic mode or in console mode? | On Windows platforms, you can only install the PPM Server in graphic (or *swing*) mode. On UNIX platforms, you can either install the PPM Server in graphic mode or in console mode (from the command line). |

# About PPM Center Best Practices Installation

In addition to installing the Foundation product, the database, and the application server on your system, the PPM Center installer places Best Practices on your system. Best Practices provides you with experience-derived information and advice on how to configure and use HP Portfolio Management and HP Program Management. Best Practices installation places various workflows and request types on your system.

Best Practices is automatically installed during PPM Center installation if *all* of the following conditions are met:

- You have Oracle SYS DBA privileges.
- You have licenses for both the HP Portfolio Management and HP Program Management.
- You elect to run the access grants script during installation. (During installation, the installer program gives you this option.) This requires that you provide the SYS DBA password.

If these conditions are not met during PPM Center installation, you can install Best Practices later if you have the Portfolio Management and Program Management licenses, and if you provide the SYS DBA password. For detailed instructions on how to install Best Practices separately, see *Installing PPM Center* on page 55.

# Preparing to Install PPM Center

Before you start to install PPM Center, complete the following tasks:

1. Check the *System Requirements and Compatibility Matrix* to make sure that your system meets *all* of the minimum requirements.
2. Collect the information required for installation.
3. Download the installation bundle (`mitg-710-install.zip`) from the PPM Download Center (itg.merc-int.com/support/download/login.jsp).

   ➤ The installation files for PPM Center and HP Deployment Management Extensions and Migrators are distributed from the PPM Download Center. To access the PPM Download Center and the files, you must have the user name and password provided when you purchased the software.

4. Extract the installation files to a temporary directory.

5. Install the SDK.

   For information on which version of the SDK to install, see the *System Requirements and Compatibility Matrix.*

6. Verify that the `JAVA_HOME` parameter is set.
7. Verify that the `ORACLE_HOME` parameter is set.

   To set the `ORACLE_HOME` parameter, you must first install Oracle client on the server machine.

   To install PPM Center and maintain the system after installation, you must create a system user. After you do, always log on to the server machine as this user to perform any PPM Server maintenance—for example, stopping and restarting the PPM Server. This helps to avoid file system permission issues, which can be difficult to track.

8. Create a PPM Center user.

   To create PPM Center users, you must have the Demand Management User Administration License.

9. Set up the Oracle tablespaces required to create the schemas and database objects.

   To create schemas and database objects, you must first create the data, index, and character large object data type (CLOB) tablespaces.

10. Verify that the required ports are open through the firewall and that other applications are not using them.

The following sections provide detailed information about each of these tasks.

The variable *<PPM_Home>*, which is used throughout this document, refers to the root directory where PPM Center is installed. The name and location of this directory are up to you.

Do not unzip the installation files in your *<PPM_Home>* directory—instead, choose a temporary directory in another location.

For checklists that include all of the preliminary tasks you must perform on your network, the PPM Server(s), the database server, and PPM Center clients, see *Preinstallation Checklists* on page 291. HP recommends that you use these checklists to track the tasks that you must perform before you start to install PPM Center.

After you complete the checklists, give them to your HP Professional Services representative. The checklists will help your representative make the necessary preparations and speed up installation.

## Collecting Required Information

The PPM Center installer prompts you to enter several parameters values that are used to create and configure the PPM Server. The installer validates each value you enter before it continues the installation. *Table 3-2* lists the information required for installation.

Table 3-2. Required installation information (page 1 of 3)

| Prompt | Description |
|---|---|
| Install Location | Directory in which the PPM Server is to be installed and configured. If the directory does not exist, the installer creates it.<br>The directory path cannot contain spaces. |
| License Configuration File | This file contains valid PPM Center license keys.<br>The PPM Server is enabled by license keys provided in a `license.conf` file, which you must obtain before installation. If you do not have a valid `license.conf` file, contact HP-Mercury support (support.Mercury.com ). |
| JAVA_HOME | On Windows and UNIX systems, the directory in which Java is installed. On UNIX systems, this parameter is set in the profile file (a `*.profile` or `*.cshrc` file) of the user who is installing PPM Center.<br>Windows example: `C:\j2sdk1.4.2_08` |
| ORACLE_HOME | The PPM Server machine must have Oracle client 9.2.0.7, or later, installed to communicate with the PPM Center database schema.<br>Specify the home directory for the Oracle client tools on the PPM Server machine. The directory path cannot contain spaces. |
| SQL*PLUS | Location of the SQL*Plus utility.<br>SQL*Plus is not required for installation, but is required for the PPM Server.<br>Example: `C:\Oracle\bin\sqlplus.exe`<br>If the `ORACLE_HOME` environment variable is set, then this parameter is detected automatically. |
| System Password | If you create database users during installation, use your system password. |

Table 3-2. Required installation information (page 2 of 3)

| Prompt | Description |
|---|---|
| Database Access Information | In addition to installing the PPM Center file system, the installer can create and populate the database schemas needed to store application data. To access the database, the installer prompts you for a user name and password, and the valid components of a JDBC URL.<br>If you want the installer to create the database schemas, you must provide the system user name and password. If you created the database schemas before installation, you provide the PPM Center database schema user name and password.<br>The PPM Server uses the JDBC URL to connect to the Oracle database.<br>The URL short format is<br>`jdbc:oracle:thin:@<hostname>:<port>:<SID>`<br>where:<br>- `<hostname>` is the DNS name or IP address of the computer running the database<br>- `<port>` is the port number that SQL*Net uses to connect to the database. Its value is typically 1521. To get the actual value, look at the corresponding entry in `tnsnames.ora`.<br>- `<SID>` is the security identifier of the database. This is usually identical to the database connect string. If it is different, an extra parameter is required.<br>For Oracle Real Application Clusters (RAC), the `JDBC_URL` parameter must contain the host and port information for all databases to which the PPM Server is to connect.<br>Following is an example of database access information used to allow the PPM Server to communicate with databases on two servers named Jaguar1 and Jaguar2:<br>`jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP) (HOST=jaguar1) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP) (HOST=jaguar2)(PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=J920)))` |
| PPM Center Schema | If you create the database schema during installation, supply the user name and password for the PPM Center database schema. |
| Reporting Meta Layer Schema | User name and password of the PPM Center Reporting Meta Layer (RML) schema. |
| Tablespaces | Data, index, and character large object data type (CLOB) tablespaces of the Oracle database. These are required to create schemas and database objects. |

Table 3-2. Required installation information (page 3 of 3)

| Prompt | Description |
|---|---|
| Windows Service Name | Name of the Windows service for the PPM Server. The installer prefixes the service name with “HP PPM” to identify it. The installer also uses the service name to create the **Start** menu item. |
| Holiday Schedule | Holiday schedule on which the PPM Center regional calendar is to be based. If you choose **None,** a new calendar with no holidays is set as the system default regional calendar, which you must name in the System Calendar prompt. |
| System Calendar | If you specify a Holiday Schedule value of **None,** the name of the system default regional calendar. |
| Currency Code | Three-letter code for the default currency.<br>The system default is US dollars (USD).<br>**Warning:** Once you choose your currency during installation, you cannot change it. |
| Region Name | Name of the region for the installation, which is defined by a combination of calendar and currency.<br>If your organization operates in only one region, use “Enterprise” or your company name. |
| Configure Server | If you answer **Yes** to this prompt, a wizard prompts you for values for the required (also called “standard”) set of server configuration parameters. You can configure the server now or later.<br>*Table A-1* on page 228 lists the server configuration parameters. Required parameters are marked with an asterisk. |

## Downloading the Installation Files

The installation files for PPM Center and HP Deployment Management Extensions and Migrators are distributed from the PPM Download Center (itg.merc-int.com/support/download/login.jsp). To access the files, you must have the user name and password provided when you purchased the software.

Download the PPM Center installation file (`mitg-710-install.zip`). If you plan to install one or more HP Deployment Management Extensions or Migrators, see the corresponding HP product documentation for specific download and installation instructions.

## Unzipping the Installation Files

Before you run the installation driver script, extract the installation files for the PPM Center software to a temporary directory. You can do this with a graphical application such as WinZip, or use a command-line tool such as Unzip. You can also extract bundles with `jar xvf <>`. The unzip procedure creates a new subdirectory named `mitg710/`. Run the command in a directory other than the `<`*`PPM_Home`*`>` directory.

## Installing the Java Software Developer Kit (SDK)

Because the PPM Server is based on Java, the machine that hosts it must also host a Java Virtual Machine (JVM), which is part of a Software Development Kit (SDK). SDKs native to the operating systems supported by PPM Center are available from either Sun Microsystems or from the operating system vendor.

➤ You must install the complete SDK. The Java Runtime Environment (JRE) alone is not supported.

For a list of required SDKs, see the *System Requirements and Compatibility Matrix.* For information on how to access this and other PPM Center documents, see *Accessing PPM Center Documentation* on page 21.

To install the SDK:

1. Download the SDK for your operating system from the Javasoft Web site or from your operating system vendor's Web site. For example:

   java.sun.com

2. Install the SDK according to the instructions provided by the vendor.

   ➤ Many operating systems require that you apply operating system–specific patches before you install the SDK. Make sure that you follow all instructions that the vendor provides.

Some vendors provide custom installation packages that you can install automatically using a command such as `pkgadd`. Other vendors provide a TAR file that you must extract.

The directory in which you install the SDK is referred to in this document as `SDK_Install_Dir`.

➤ The directory path name cannot contain spaces.

3. Verify that your user name has the Java executable in its path by logging on and running the following the command:

```
java -version
```

   This returns the Java version. If you see an error message, modify the path environment variable, as required.

4. Verify that the `JAVA_HOME` environment variable is set correctly. If the path set for `JAVA_HOME` is not correct, set it to the correct value.

➤ For information about how to check for and set the `JAVA_HOME` environment variable, see *Verifying that the JAVA_HOME Parameter Is Set*.

## Verifying that the JAVA_HOME Parameter Is Set

PPM Center requires that you set `JAVA_HOME` in the system environment of the user account to be used to start the PPM Server. It is important that the `JAVA_HOME` parameter be set for the same shell and user who runs the installation.

### Determining the JAVA_HOME Path in DOS

To determine the `JAVA_HOME` path in DOS:

- At the command line, type **`echo %JAVA_HOME%`**.

### Determining the JAVA_HOME Path in UNIX

To determine the `JAVA_HOME` path in a UNIX shell (SH, BASH, or KSH):

- At the UNIX prompt, type **`echo $JAVA_HOME`**.

### Setting the JAVA_HOME Parameter in Windows

To set the value of `JAVA_HOME` in Windows:

1. Open the Control Panel.
2. Open the System Properties window.
3. Click the **Advanced** tab.
4. Click **Environment Variables.**
5. Under **System Variables,** click **New.**

   The New System Variable dialog box opens.
6. In the **Variable name** field, type **`%JAVA_HOME%`**.
7. In the **Variable Value** field, type the full Java install directory path.
8. Click **OK.**
9. Click **OK.**

### Setting the JAVA_HOME Parameter in DOS

To set the value of `JAVA_HOME` in DOS:

- Run the following:

```
set JAVA_HOME=<JVM_Install_Directory>
```

### Setting the JAVA_HOME Parameter in UNIX

To set the value of `JAVA_HOME` in UNIX using the Bourne shell (SH, BASH, or KSH):

- Run the following:

```
JAVA_HOME=<JVM_Install_Dir>
export JAVA_HOME
```

# Creating a PPM Center User

To install PPM Center and maintain the system after installation, you must create a system user. After you do, always log on to the server machine as this user to perform any PPM Server maintenance—for example, stopping and restarting the PPM Server. This helps to avoid file system permission issues, which can be difficult to track.

## Creating the User in Windows

In Windows, configure the user to be a member of the Administrators and Domain Users groups, at a minimum. Provide the user with full access to the installation directory for PPM Center and all of its subdirectories. Provide the Administrators screen group with at least read access to these directories.

## Creating the User in UNIX

In UNIX, PPM Center does not require root access for installation. Do not install the server as the root user.

To create a user in UNIX:

1. In the `.profile` file, set the `JAVA_HOME` environment variable.
2. In the `.profile` file, set the `ORACLE_HOME` environment variable.
3. Set the term to dump option.

# Installing a UNIX Emulator and Telnet Server (Windows)

To run PPM Center on Microsoft Windows, you must have a UNIX emulator such as Cygwin, and a Telnet server such as Microsoft Telnet. For a list of supported UNIX emulators and Telnet servers, see the *System Requirements and Compatibility Matrix.*

➤ To configure private key authentication with secure shell (see *Configuring Private Key Authentication with Secure Shell* on page 74), you use the ssh-keygen utility, which is part of the Cygwin installation. To get this utility, you must enable the Open SSH components during Cygwin installation.

## Creating the Database Schemas

To create the empty database schemas (with tables to be populated during installation):

1. Set up the required data, index, and CLOB tablespaces for the PPM Center database schema.

   ➤ For information on the minimum size recommended for these tablespaces, see the *System Requirements and Compatibility Matrix.*

   Use locally-managed SYSTEM tablespaces with automatic segment-space management.

   ➤ Locally-managed tablespaces eliminate extent fragmentation and provide better performance than dictionary-managed tablespaces.

2. Create each tablespace as shown in the following example for a data tablespace:

```
CREATE TABLESPACE <PPM_DATA>
datafile <'/u0/oracle/oradata/G1010/ppm_data01.dbf'>
size <1024m>
AUTOEXTEND ON MAXSIZE <4096m>
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
SEGMENT SPACE MANAGEMENT AUTO;
```

3. Generate at least one rollback segment for each tablespace and use an undo tablespace.

   These rollback segments must reside in a separate tablespace reserved for rollback segments. Generate them with the `OPTIMAL` size constraint so that the rollback segments automatically deallocate space as it becomes free.

4. Generate unlimited quota on the data, index, and CLOB tablespaces for PPM Center.

   ➤ Oracle has the default TEMP tablespace, which you can resize to improve performance.

The PPM Server requires two separate database schemas to store application data. A DBA can create these schemas before installation. Creating database schemas requires privileges that a DBA might not want to grant to a PPM Center administrator. Either create the database schemas before installation or make sure that a DBA is available during installation.

To create the database schemas and grant the permissions between them:

1. Unpack the PPM Center installation bundle as outlined in *Installing PPM Center* on page 55.

   The `mitg710` directory is created. The `mitg710/sys` and `mitg710/system` directories contain the scripts required to create the database schemas.

2. Run the script `CreateKintanaUser.sql` (located in `mitg710/system`) against the database into which you plan to install PPM Center.

   The script prompts for a user name and password, and the tablespaces that the PPM Center database schema are to use.

   ```
   sh> sqlplus system/<password>@<SID> \
   @CreateKintanaUser.sql \
   <PPM_username> \
   <password> \
   <data_tablespace> \
   <index_tablespace> \
   <CLOB_tablespace>
   ```

3. Run the `CreateRMLUser.sql` script, which is located in the `/system` directory.

   The script prompts for a user name and password for the Reporting Meta Layer (RML) schema, tablespace information, and the PPM Center database schema user name. The script creates the RML schema and establishes the permissions between the RML and the PPM Center database schema.

   Because the RML schema contains only views (and no physical objects), it does not require a separate tablespace.

   ```
   sh> sqlplus system/<password>@<SID> \
   @CreateRMLUser.sql \
   <RML_username> \
   <RML_password> \
   <data_tablespace> \
   ```

4. As the SYS DBA user, run the `GrantSysPrivs.sql` script, which is located in the `mitg710/system` directory.

   This script grants the privileges that the PPM Server requires.

If you created the schemas before installation, select **Please use existing schemas** when prompted during installation. Supply the same values as those used in this procedure (that is, the values *<PPM_username>* and *<RML_username>*).

## Verifying Port Availability

To successfully install and configure PPM Center, specific ports must be available through the firewall. To expedite installation, make sure that the ports are available before you start to install the product. *Table 3-3* contains summary information about the ports and protocols that PPM Center system components use to communicate.

If you are using an external Web server, you must assign it a port number other than the one assigned to the internal Web server.

Table 3-3. Summary of PPM Center ports and protocols

| Communication Channel | Protocols | Ports |
|---|---|---|
| Web Browser <—> Web Server | HTTP/HTTPS | 80/443 (configurable) |
| | ■ If you do not use the default port, you must specify the port number in the URL. For example, http://Mercury.com versus http://Mercury.com:<PORT>. You may also be required to open the firewall for ports other than the defaults.<br>■ On UNIX systems, only processes started by the root user can be assigned a port number that is less than 1024. | |
| PPM Workbench <—> App Server | RMI / SRMI | 1099 (configurable) |
| External Web Server <—> App Server | AJP13 | 8009 (configurable) |
| App Server <—> Database | JDBC | 1521 (configurable) |
| App Server <—> Mail Server | SMTP | 25 |
| App Server <—> LDAP Server | LDAP | 389 |
| App Server <—> LDAP Server | LDAP over SSL | 636 |
| App Server <—> External System | Telnet | 23 |
| App Server <—> External System | SSH | 22 |

Table 3-3. Summary of PPM Center ports and protocols

| Communication Channel | Protocols | Ports |
|---|---|---|
| App Server <—> External System | FTP (control) | 21 |
| App Server <—> External System | FTP Data | Dynamic |
| App Server <—> External System | SCP (Secure Copy) | 22 |

# Installing PPM Center

This section provides the detailed steps used to install the database objects and data that the PPM Server uses. You can perform these steps on any computer (running UNIX or Windows) that has SQL*Net connected to the database on which the PPM Center database objects are to be installed.

## Installing PPM Center on Windows

The installation utility for a Windows server is an executable file that performs the steps required for a basic server installation. The executable and supporting files are contained in a Zip file. The typical installation automatically installs the following components on the server:

- PPM Center program files
- PPM Center database objects
- Start menu item
- Windows service

‼ You cannot successfully install PPM Center unless you have SYS DBA privileges or a DBA has already created the required schemas. For more information, see *Key Decisions* on page 41.

To install the PPM Server on Windows:

1. Ensure that you have a UNIX emulator (such as Cygwin) and a Telnet server (such as MSFT Telnet) installed.

   For a list of supported UNIX emulators and Telnet servers, see the *System Requirements and Compatibility Matrix.*

   ➤ To configure private key authentication with secure shell (see *Configuring Private Key Authentication with Secure Shell* on page 74), you use the ssh-keygen utility, which is part of the Cygwin installation. To get this utility, you must enable the Open SSH components during Cygwin installation.

2. Extract all files from `mitg-710-install.zip` to the file system.

   The extraction creates the `mitg710` directory, which includes the `install.exe` file, JAR files, a `system` directory, a `sys` directory, various scripts, and more.

3. Locate, and then double-click the `install.exe` file.

   The installer prompts for the directory for the software installation (the <*PPM_Home*> directory. You can specify any install path.

4. Provide all required information as the installer prompts you for it (see *Collecting Required Information* on page 45).

   After you provide all required information, the installer installs the PPM Center files and configures the database. Status bars indicate installation progress. An installation summary page displays any problems encountered during installation.

   After successful installation, PPM Center is installed as a Windows service. You can view the properties for this service through the Services Control Panel item.

5. To complete the service setup:

   a. Open the Control Panel.

   b. Double-click **Administrative Tools.**

   c. Double-click **Services.**

   d. Right-click the PPM Center service name, and then click **Start** on the shortcut menu.

      HP recommends that you set the startup type to **Automatic** so that the PPM Server restarts automatically after the computer is restarted. If you have generated a custom PPM Center user (as recommended), specify this user name for the “Log On As” value.

e. Close the Administrative Tools window.

6. Click **Save.**

An item that corresponds to the Windows service name that you specified during installation is added to the **Start** menu. The menu provides links to PPM Center documentation and an uninstall program.

If you did not configure the PPM Server during installation, see *Configuring or Reconfiguring the Server* on page 69.

➤ Do not map the <*PPM_Home*> directory so that it is accessible from an external Web server. This introduces a potential security risk. HP recommends that you use the HP-supplied Web server, unless you have the special requirements described in *Single-Server/External Web Server Configuration* on page 29.

## Installing PPM Center on UNIX

To install the PPM Center on UNIX:

1. To extract the files into an empty directory from the download bundle, at a Telnet command prompt, type one of the following:

```
unzip mitg-710-install.zip
```

Alternatively,

```
jar xvf mitg-710-install.zip
```

All the files and scripts required for PPM Center installation are extracted. The installer prompts for the software install directory. You can specify any directory for installation.

The `mitg710` directory resulting from the extraction contains:

- The `install.sh` shell script
- Several JAR files
- A `system` directory
- A `sys` directory

2. To start the installation, run the installation script (as the SYSTEM user) and specify the installation mode.

Example: `sh install.sh [-swing|-console]`

Table 3-4. UNIX installation modes

| Mode | Meaning |
|---|---|
| -swing | GUI mode. A wizard guides you through the installation steps. |
| -console | Command-line mode. The installation script runs within the terminal session. |

The installation script performs the following actions:

- Prompts for information required to install the server (see *Collecting Required Information* on page 45).
- Generates all database tables in the specified tablespace.
- Creates all database objects (indexes, packages, views) and application data.
- Generates password security keys.
- Generates the server configuration file.
- Rebuilds statistics for the Oracle optimizer. This is done to optimize system performance. For the installation procedure to perform this step, the following grants to the schema must be in place:

  ```
  grant select on v_$parameter to <PPM_Schema>
  grant select on v_$mystat to <PPM_Schema>
  grant select on v_$process to <PPM_Schema>
  grant select on v_$session to <PPM_Schema>
  grant execute on dbms_stats to <PPM_Schema>
  ```

  The `GrantSysPrivs.sql` script (located in the `mitg710/sys` directory) performs these required grants.

  ➤ To run this script, you must have SYS DBA privileges.

  You cannot run PPM Center until the grants are in place.

  If you did not run the `GrantSysPrivs.sql` script before you started installation, do it now (with SYS DBA access).

➤ HP recommends that, after you install PPM Center, you change the password for the administrator user.

# Configuring the FTP Server on Windows

PPM Center uses FTP to move files between machines. To transfer files between machines on a network, each source and destination machine must be running an FTP server. On UNIX platforms, this is standard functionality, but machines running Windows require additional FTP server configuration to function with PPM Center.

Before you configure the FTP server on a machine, make sure that the Windows user account (which PPM Center uses to open a connection) has access to the directories to which files will be moved. Some FTP servers require that you map these directories to FTP aliases, and a configuration utility is usually provided for this (for example, for Microsoft IIS, the utility is Internet Services Manager).

➤ On Windows, most FTP servers, including Microsoft IIS, do not support drive letters. If you use FTP in PPM Center, the drive letter is removed from the base path. If your base path is `d:\ppm710`, then FTP tries to start from the ftp root directory and FTP fails.

To work around this, you must create an FTP alias. (For example, map `/ppm710` to `D:\ppm710`.) This way, FTP and Telnet point to the same disk location.

Configure the FTP server according to directions that the vendor has provided. For the File and Directory Chooser components to work, you must set the FTP server directory listing style to UNIX, and not to MS-DOS.

To set the directory listing style to UNIX:

1. In Windows, open the Internet Services Manager.
2. In the left pane, under **Console Root,** open the Internet Information Server.
3. Select the machine name.
4. Right-click the Default FTP site displayed in the right pane, and then click **Properties** on the shortcut menu.

   The Default FTP Site window opens.
5. Click the **Home Directory** tab.
6. Under **Directory Listing Style,** click **UNIX.**
7. Click **OK.**

To test the connection, try to open a session manually. If you can open an FTP session and navigate from one directory to another, then PPM Center can do this too.

## Verifying the Installation

To verify the installation, perform the following tasks:

1. Check the logs produced during installation.
2. Log on to PPM Center.
3. Start the PPM Workbench.
4. Run a report.
5. Create a request.
6. Test the graphical view of the request.

If you encounter a problem that you cannot solve, contact HP-Mercury support (support.Mercury.com).

## Contacting Support

If you encounter problems with your installation or have questions, contact HP-Mercury support (support.Mercury.com). Before you contact HP-Mercury support, have the following information ready:

1. Open the `mitg_install.log` file (located in the `<PPM_Home>` directory) in a text editor such as Notepad.

   This file provides information about what part of the installation failed.

2. Search the `mitg_install.log` file for an error message that is specific to installation failure.

3. Place all of the files in the `<PPM_Home>/Install_710/logs` directory in a compressed file.

   The installation utility creates a separate log directory for each installation attempt. In the most recent directory, examine each file to see exactly where the PPM Server has failed. The log file contains information about which failed action it attempted.

# Installing Service Packs

HP occasionally delivers product service packs to licensed PPM Center customers. You can use the `kDeploy.sh` script (a command-line tool) to install service packs.

➤ To install a service pack, you must ensure that you have the required system privileges. For information about these privileges, and when to grant them, see *Key Decisions* on page 41.

PPM Center service packs are distributed as deployments. Deployments are software bundles that contain files and data, and are in the following format:

```
mitg-<ver>-<id>[.#].jar
```

Where:

`mitg` is the product code

`<ver>` is the PPM Center version for which you can install the service pack

`<id>` is the unique identifier for service pack

`[.#]` represents an optional revision number for the deployment, and may not be included in the deployment name.

For example, to install Service Pack 1:

1. Back up your database.
2. Stop the PPM Server.

   ➤ You cannot install the service pack on an active server. For information about how to start and stop the server, see *Starting and Stopping the PPM Server* on page 66

3. Issue the following command:

   ```
   sh kDeploy.sh -i SP1
   ```

4. As the script runs, follow the prompts.
5. Start the PPM Server.

To obtain a list of all service packs applied to your PPM Center instance, run the command `sh kDeploy.sh -l`.

For more information about the `kDeploy.sh` script, see *kDeploy.sh* on page 280.

## Handling Backup Files Related to Service Pack Installation

During a service pack installation, the installer backs up all of the existing files that are to be replaced. After multiple service pack installations, the backup files can take up significant space.

Eventually, the backed up files can consume so much space that service pack installation fails. To prevent this from occurring, do one of the following:

- Install service packs without creating backup files. To do this, run the `kDeploy.sh` script, as follows:

  ```
  sh kDeploy.sh -i SP3 -B
  ```

- Specify that backed up files are deleted after service pack installation. To do this, run the `kDeploy.sh` script, as follows:

  ```
  sh kDeploy.sh -tidy
  ```

## Contacting Support

If you encounter problems with service pack installation, contact HP-Mercury support (support.Mercury.com). Before you contact HP-Mercury support, prepare information about the installation problem, as follows:

- Compress all the files in `<`*`PPM_Home`*`>/logs/deploy/710/directory/<`*`SP#`*`>`.

  where `<`*`SP#`*`>` is the service pack version you are installing. For example, for PPM Center version 7.1, Service Pack 2, you would compress the directory `<`*`PPM_Home`*`>/logs/deploy/710/directory/SP2.`

# Optional Installations

This section provides descriptions of additional products that you can install and set up to work with PPM Center.

➤ This chapter does not provide instructions on how to install PPM Center language packs. You can find that information in the README file associated with the language pack.

## Installing HP Project and Portfolio Management Best Practices

HP PPM Best Practices provides customers with experience-derived information and advice about configuring and using HP Portfolio Management and HP Program Management. Best Practices installation places various workflows and request types on your system to help optimize your use of HP Program Management and HP Portfolio Management.

If you did not, or could not, install Best Practices during PPM Center installation, you can install it separately. For more information, see *About PPM Center Best Practices Installation* on page 43.

Before you can perform a separate installation of Best Practices, ensure that *all* of the following conditions are met:

- You have installed and configured PPM Center.
- You are logged on to your system with administrator privileges.
- You have licenses for both the HP Portfolio Management and HP Program Management.
- You choose to run the access grants script during installation. (During installation, the installer program gives you this option.) This requires that you have SYS DBA privileges.
- You have created a valid PPM Center user name.

To install Best Practices:

1. Start the PPM Server.
2. Run the `kDeploy.sh` script, as follows:

   ```
   sh kDeploy.sh -best-practices
   ```

➤ For more information about the `kDeploy.sh` script, see *kDeploy.sh* on page 280.

### Verifying HP Project and Portfolio Management Best Practices Installation

To verify that Best Practices is successfully installed, run the `kDeploy.sh` script, as follows:

```
kDeploy.sh -l
```

This returns a list of the deployed bundles in an instance.

## Installing HP Accelerators and HP Deployment Management Extensions

If you plan to install any HP Accelerators or HP Deployment Management Extensions, you must do so after you install and configure PPM Center, and before you use PPM Center for processing.

You are not required to stop the PPM Server(s) before you install an Extension. However, HP recommends that you install the Extension when no users are logged on to the system. Consider placing the server in “restricted” mode before you install.

For specific information on how to install an HP Accelerator or an HP Deployment Management Extension, see the documentation for the Extension or Accelerator you purchased.

➤ To complete an Accelerator or Extension installation successfully, you must ensure that you have the required system privileges. For information about these privileges, and how to grant them, see *Key Decisions* on page 41.

# What to Do Next

After you have successfully installed PPM Center, delete all subdirectories of the `install_710` directory, except for the `logs` subdirectory.

Proceed to Chapter 4, *Configuring the System,* on page 65.

# 4 Configuring the System

In This Chapter:

# Starting and Stopping the PPM Server

This section provides information about how to start the PPM Server on a single-server system. For information about configuring and running a clustered configuration, see *Server Cluster Configurations* on page 30 and *Configuring a Server Cluster* on page 115.

➤ Unless otherwise indicated, “the server” refers to the PPM Server, and not the server machine.

## Setting the Server Mode

PPM Center supports the following server modes:

- **Normal.** In normal mode, all enabled users can log on, and all services are available, subject to restrictions set in `server.conf` parameters.
- **Restricted.** In restricted mode, the server allows only users with Administrator access granted to log on. The server cannot run scheduled executions, notifications, or the concurrent request manager while in this mode.

  Before you can install an HP Deployment Management Extension, you must set the server to restricted mode.
- **Disabled.** Disabled mode prevents server startup. A server enters disabled mode only after a PPM Center upgrade exits before the upgrade is completed.

### Setting the Server Mode with setServerMode.sh

The `setServerMode.sh` script, located in the `<PPM_Home>/bin` directory, sets the server mode in situations where you want to obtain exclusive access to a running server.

To set the server mode using the `setServerMode.sh` script:

1. From the desktop, select **Start > Run.**

   The Run dialog box opens.

2. In the **Open** field, type the following:

   ```
   sh setServerMode.sh <MODE NAME>
   ```

For example, to set the server in restricted mode, type the following:

```
sh setServerMode.sh RESTRICTED
```

3. Click **OK.**

### Setting the Server Mode Using kConfig.sh

You can use the `kConfig.sh` script to set the server mode.

To set the server mode using the `kConfig.sh` script:

1. Run `sh kConfig.sh` (located in the `<PPM_Home>/bin` directory).
2. Select **Set Server Mode.**
3. In the list, select **Restricted Mode.**
4. Click **Finish.**

For more information about the `setServerMode.sh` script, see *Setting the Server Mode* on page 66. For more information about the `kConfig.sh` script, see *kConfig.sh* on page 280.

## Starting and Stopping the Server on Windows

To start the server on a Windows system:

1. If you are installing one of the HP Deployment Management Extensions, set the server to restricted mode.

   For information about how to set the server mode, see *Setting the Server Mode* on page 66.

2. Open the Control Panel.
3. Double-click **Administrative Tools.**
4. Double-click **Services.**
5. Right-click the name of the PPM Center service, and then click **Start** on the shortcut menu.

   The service name starts with “HP PPM.”

6. If you have installed an Extension, set the server to Normal mode.

   For information about how to set the server mode, see *Setting the Server Mode* on page 66.

➤ If you prefer to use the Windows shell command line to start servers instead of using Windows Services, you can use the `kStarts.sh` script.

To stop the server on a Windows system:

1. Open the Control Panel.
2. Double-click **Administrative Tools.**
3. In the Administrative Tools window, double-click **Services.**
4. In the Services window, right-click the name of the PPM Center service, and then click **Stop** on the shortcut menu.

   The service name starts with “HP PPM.”

## Starting and Stopping the Server on UNIX

To start the server on UNIX:

1. If you are installing an HP Deployment Management Extension, set the server to restricted mode.

   For information about how to set the server mode, see *Setting the Server Mode* on page 66.

2. Change to the `<PPM_Home>/bin` directory.
3. Run the `kStart.sh` script, as follows:

   ```
   sh ./kStart.sh
   ```

4. If you have installed an Extension, set the server to normal mode.

For more information about `kStart.sh`, see *kStart.sh* on page 283. For information about how to start servers in a cluster, see *Starting and Stopping Servers in a Cluster* on page 125.

To stop the server on UNIX:

1. Navigate to the `<PPM_Home>/bin` directory.
2. Run the `kStop.sh` script as follows:

   ```
   sh ./kStop.sh -now -user <username>
   ```

   Make sure that you type a valid user name that has Administrator privileges.

For more information about kStop.sh, see *kStop.sh* on page 284. For information about how to stop servers in a cluster, see *Starting and Stopping Servers in a Cluster* on page 125.

# Configuring or Reconfiguring the Server

If you configured the PPM Server during installation, it is probably not necessary to reconfigure it, unless your environment or requirements have changed. If you did not configure the server during installation, configure it now.

You can perform most of the configuration using the configuration procedure described in the next section, *Standard Configuration*. In some cases, however, configuration requires custom parameters. For information about when and how to configure the server using custom parameters, see *Defining Custom and Special Parameters* on page 71.

The server configuration tool runs in both console and graphical modes. To run in graphical mode in a Windows environment, the tool requires an X Window session.

## Standard Configuration

This section provides the steps for standard configuration and all of the settings required for a typical installation.

To configure the PPM Server:

1. From a DOS or UNIX command line, run the kConfig.sh script (located in the <*PPM_Home*>/bin directory) as follows:

   - To run the script in graphical mode, type:

     ```
     sh kConfig.sh
     ```

   - To run the script in console mode (UNIX only), type:

     ```
     sh kConfig.sh -console
     ```

   ➤ Run this utility in an X Window session.

2. Follow the configuration wizard prompts to complete the configuration.

   Enter a value for every parameter required for your system environment. To determine the correct value to enter for a parameter, move your cursor over the parameter name and display the tooltip text. For more information, see *Server Configuration Parameters* on page 225.

All confidential information (such as passwords) is hidden and encrypted before it is stored.

Do not change default values unless you are sure that the default value does not meet the requirements of your organization.

➤ Always use forward slashes (/) as a path separator, regardless of your operating system environment. PPM Center automatically uses the correct path separators when communicating with Windows, but expects to read only forward slashes on the configuration file.

You specify any required parameters on the Custom Parameters page.

3. If you have no custom parameters to add, leave **Custom Parameters** empty. If you require custom parameters, see *Defining Custom and Special Parameters* on page 71 for instructions on how to specify them.

4. After you finish configuring the server, click **OK.**

   The configuration wizard writes the configuration parameters to the `server.conf` file and generates other files that the PPM Server requires (for example, `jboss-service.xml`).

5. Stop, and then restart, the server.

   For information about how to stop and start the server, see *Starting and Stopping the PPM Server* on page 66.

➤ You can also modify parameters directly in the server configuration file, which is described in Appendix A, *Server Configuration Parameters,* on page 225.

If you modify parameters directly, be sure to run the script `kUpdateHtml.sh` after you make your changes.

## Defining Custom and Special Parameters

In addition to the standard parameters that HP supplies, PPM Center supports two additional kinds of server parameters:

- You can define your own custom parameters.

  Custom parameter names must have the prefix `com.kintana.core.server`. For example, to add a custom parameter named `NEW_PARAMETER`, in the **Key** field, type the following:

  **`com.kintana.core.server.NEW_PARAMETER`**

  Parameters that you add to the custom parameters list are accessible as tokens from within the application. These tokens are in the format `[AS.parameter_name]`.

- HP has created configuration parameters that you can use in special situations after you add them to the custom parameters folder. *Table 4-1* lists these special parameters.

If you edit the `server.conf` file directly, you must then run the `kUpdateHtml.sh` script to rebuild the startup files. To implement your changes, you must stop, and then restart, the PPM Server. After you restart the server, you can run the Server Configuration Report to see the new or modified parameter values in the `server.conf` file.

Instead of modifying the `server.conf` file directly and then running the `kUpdateHtml.sh` script, you can run the `kConfig.sh` script (located in the `<`*`PPM_Home`*`>/bin` directory). The `kConfig.sh` calls the same Java code that the `kUpdateHtml.sh` does to rebuild the startup files. If you use the `kConfig.sh` script, you are not required to run the `kUpdateHtml.sh` script. However, to apply your changes, you must stop and restart the PPM Server.

For information about the `kConfig.sh` script, see *kConfig.sh* on page 280. For information about the `kUpdateHtml.sh` script, see *kUpdateHtml.sh* on page 286.

Table 4-1. Special configuration parameters

| Parameter | Description | Sample Value |
|---|---|---|
| com.kintana.core.server. DB_CONNECTION_STRING | If the `JDBC_URL` parameter is specified, then the security identifier (SID) of the database on which the PPM Center schema resides is requested. It is assumed that the connect string for this database is the same as the SID. However, this is not always the case.<br>If the connect string (for connecting to the database using SQL*Plus from the server machine) is different than the database SID, add this parameter and supply the correct connect string. | PROD |
| com.kintana.core.server. NON_DOMAIN_FTP_SERVICES | **Windows environment only:** To open an FTP session, FTP servers on Windows typically require the Windows domain name and user name (in the form `Domain\User name`). By default, PPM Center includes the domain name and user name in an FTP session to a Windows computer.<br>If you use an FTP server that does not require the domain name, you can use this parameter to override the default functionality.<br>For more information, contact HP-Mercury support (support.Mercury.com). | WAR-FTPD |
| com.kintana.core.server.TEMP_DIR | This parameter defines a PPM Center temporary directory. This defaults to a `temp` subdirectory of the `logs` directory.<br>If you use this parameter, make sure that you provide the full directory path. | |

## Enabling Secure RMI (Optional)

To enable SRMI (RMI over SSL):

1. Create a keystore for SSL to use.

   You can use the Java keytool application to create a keystore. For information about the keytool application, see churchillobjects.com/c/11201e.html.

   Use the keystore password that you use to run keytool to define the KEY_STORE_PASSWORD (see step 2).

2. In the server.conf file, specify values for the three parameters, as follows:

   - RMI_URL
   - Set the KEY_STORE_FILE parameter to point to the keystore file.
   - Set the KEY_STORE_PASSWORD to the keystore password you created in step 1. This password can be encrypted.

### Example

If you ran keytool to create the file security/keystore relative to the *<PPM_Home>* directory, and you used the password "welcome," ran on host "caboose," and listened on port 1099, your server.conf parameters would look as follows:

```
com.kintana.core.server.RMI_URL=rmis://caboose:1099/
KintanaServer
com.kintana.core.server.KEY_STORE_FILE=security/keystore
com.kintana.core.server.KEY_STORE_PASSWORD=welcome
```

You can create a self-signed certificate.

## Configuring Private Key Authentication with Secure Shell

This section provides information on how to configure private key authentication with secure shell (SSH). The procedure is based on the following assumptions:

- SSH is installed.
- The SSH server is configured for private key authorization.
- The ssh-keygen utility is part of the Cygwin installation. (To get this utility, you must enable the Open SSH components during Cygwin installation.)

Before you configure private key authentication, do the following:

- Verify that the PPM Center user account can be used to log on to the remote host through the SSH session.
- Add the RSA certificate information of the remote host to the ssh `known_hosts` file, which is located in the `<PPM_Home>` directory.

To add the RSA certificate of the remote SSH host to the PPM Server SSH `known_hosts` file:

1. Log on to the PPM Server as the PPM Center user.
2. From the command line, run the following:

   ```
   ssh <USER_ID>@<REMOTE_HOST>
   ```

   The first time you run this command, you are prompted to indicate whether you want to continue.
3. Type **`yes`**.
4. Terminate the SSH connection with the remote host.

To set up private key authentication with SSH:

1. Generate the private/public key pair on the PPM Server.
2. Add the generated public key to the remote `SSH Authorized_Key` file.
3. Configure the PPM Server.

The following sections provide the steps required to perform each of these tasks.

## Generating the Private and Public Keys

To generate the private/public key pair on the PPM Server:

1. Log on to the PPM Server machine as the PPM Center user.
2. Change directory to the home folder defined for the PPM Center user on the operating system.
3. Run the following SSH utility:

   ```
   ssh-keygen -t rsa -b 1024
   ```

   ➤ PPM Center only supports the RSA key type, and not the DSA key type.

   Do not provide the “passphrase.”

4. Press **`Enter`** twice.
5. Verify that the `<`*`PPM_Home`*`>/<`*`PPM_USER`*`>/.ssh` directory now contains the `id_rsa` (the private key) and `id_rsa.pub` (the public key) files.

## Adding the Public Key to the SSH authorized_keys File on the Remote Host

To append the public key to the remote SSH `authorized_keys` file (remote hosts):

1. Transfer the `id_rsa.pub` file to the remote SSH host machine, in the `/<`*`PPM_USER_HOME_FOLDER`*`>/.ssh` directory as `ppm_id_rsa.pub`.

   ➤ On the remote UNIX host, the `.ssh` folder is in the `/home/<`*`PPM_USER`*`>/` directory. On Windows, the folder location depends on the user home directory defined during Cygwin installation.

2. Log on to the remote host with the user ID that the PPM Server is to use to connect.
3. Change directory to “.ssh” directory under `<`*`PPM_Home`*`>/<`*`USER_ID`*`>/.ssh`.
4. Append the content of the `itg_id_rsa.pub` to the `authorized_keys` file.

   If the file does not exists, create it. If it exists, append the following to it:

   ```
   cat itg_id_rsa.pub > authorized_keys
   ```

5. Repeat these steps on the PPM Server to allow public key authentication from the PPM Server back to itself.

## Configuring the PPM Server

To configure the PPM Server:

1. Open the `server.conf` file in a text editor such as Notepad.
2. Add the following server directive to the file:

```
com.kintana.core.server.SSH_PRIVATE_IDENTITY_FILE=/<PPM_
Home>/<PPM_USER>/.ssh/id_rsa
```

3. Change to the `<PPM_Home>/bin` directory.
4. To update the required startup files, run the `kUpdateHtml.sh` script.
5. Restart the PPM Server.

### Verifying Server Configuration

To verify the configuration:

1. Open a command-line window outside of the PPM Server.
2. Log on to the PPM Server machine as the PPM Center user, as follows:

```
ssh <USER_ID>@<REMOTE_HOST>
```

You should not be prompted for the password. It should log on to the remote host using the RSA key file.

3. On the PPM Server, log on to PPM Center.
4. From the menu bar, select **Administration > Open Workbench.**

   The PPM Workbench opens.

5. From the shortcut bar, select **Environments > Environments.**

   The Environment Workbench page opens.

6. Click **New.**

   The Environment: Untitled window opens.

7. In the **Environment Name** field, type the name of the remote host.
8. In the **Server** section, do the following:

   a. In the **Name** field, type the remote server name.

   b. In the **Type** list, select the operating system type on the remote server.

   c. In the **Username** field, type the user ID you provided in step 2.

d. In the **Password** field, click the Enter or Change Password button.

   The Enter or Change Password dialog box opens.

   ➤ The PPM Workbench requires that you provide a password, regardless of whether the authentication uses RSA.

e. In the **Enter New Password** and **Confirm New Password** fields, type the password for the user ID you provided in step 2.

f. Click **OK.**

g. In the **Base Path** field, type the base path.

h. In the **Connection Protocol** list, select **SSH2.**

i. In the **Transfer Protocol** list, select **Secure Copy 2.**

9. Clear the **Enable Client** and **Enable Database** checkboxes.

   ➤ The user name specifies the user ID to be used to log on to the destination SSH server. The Environment Checker requires the password. Package line uses the public key file for authentication.

10. Click **Save.**

11. At the bottom left of the window, click **Check.**

    The Check Environment window opens.

12. In the left pane, expand the **Server** folder, and then click **SSH2 Server.**

13. Click **Check.**

In the left pane, an icon to the left of the selected server indicates whether the check succeeded or failed. The right pane displays the details.

## Generating Password Security (Optional)

For password security, PPM Center uses a client/server encryption model based on the ElGamal algorithm, which generates a public/private key pair. Passwords are encrypted using the server’s public key. Only the server can decrypt the data using the private key. The client application does not have access to decrypted data.

The public and private keys, which are generated during PPM Center installation, reside in `<PPM_Home>/security`. Generate the key pair only once, unless you think that server security has been breached. In that case, regenerate the key pair and reencrypt all passwords.

To regenerate the private and public key pair:

1. From a DOS or UNIX prompt, run the `kKeygen.sh` script, which is located in the `<PPM_Home>/bin` directory:

   ```
   sh kKeygen.sh
   ```

2. If information is not available in `server.conf`, you are prompted for the following information:

   `JDBC_URL` (for example, `jdbc:oracle:thin:@DBhost.domain.com:1521:SID`, which the server needs to communicate with the database)

   `DB_USERNAME` (the user name for the PPM Center database schema)

   `DB_PASSWORD` (the password for the PPM Center database schema)

   ‼ If you generate new public or private keys, users cannot log on. The old passwords stored in the database are encrypted using the old key. All of the passwords encrypted using the new keys do not match those stored in the database.

As the script run completes, the following two keys are placed in the `<PPM_Home>/security` directory:

```
public_key.txt
private_key.txt
```

On Windows, anyone can read these files. As the system administrator, make sure that non-trusted users do not have read privilege to the files.

On UNIX, the files are read-only for the user running the script. If the user running the script is not the user who started the server, the server cannot read the keys and cannot start.

For more information about the `kKengen.sh` script, see *kKeygen.sh* on page 283.

### Configuring Solaris and Linux Environments to Use HP Deployment Management

PPM Center can connect to a machine on which the environment variable TERM is set to dumb. To enable HP Deployment Management to work in Solaris and Linux environments, you must set this environment variable.

To set the TERM variable on Solaris, run the following:

```
.login:
if ("$TERM" == "dumb") ksh
```

To set the TERM variable on Linux, run the following:

```
.profile:
if [ "$TERM" = "dumb" ]
then
   EDITOR=null
   SHELL=/bin/ksh
   export EDITOR
   VISUAL=null
   export VISUAL
   stty erase '^H'
fi
```

To set the TERM environment variable on Linux 2.1, run the following:

```
.cshrc:
if ("$TERM" == "dumb") sh
```

## Verifying Client Access to the PPM Server

All PPM Center clients use the same URL to log on. To specify the URL for PPM Center, append `/itg/web/knta/global/.jsp` to the value of the `BASE_URL` server configuration parameter, as follows:

`<`*`server.mydomain.com`*`>:<`*`port`*`>/itg/web/knta/global/Logon.jsp`

To verify client access to the PPM Server after installation, log on to a client machine as administrator.

To log on to PPM Center as administrator:

1. On a client machine, start a supported browser, and then enter the URL for your PPM Center site.

   The PPM Center logon screen opens.

2. In the **Username** field, type **`admin`**.

3. In the **Password** field, type **`admin`**.

   PPM Center provides this default account for logging on the first time. HP recommends that you disable the admin account or change the password after you generate accounts for all of your users.

4. Click **Submit.**

The PPM Center standard interface opens.

For more information about how to configure licenses and user access, see the *Security Model Guide and Reference.*

## Accessing the JBoss JMX Console

The JMX console is an application that ships with JBoss and provides visibility into the JMX MBeans that make up the JBoss application server. You can use the console to view all registered services that are active in the application server and that can be accessed either through the JMX console or programmatically from Java code.

To access the JMX console, go to the following URL:

```
http://<localhost>:8080/jmx-console
```

The user name and password required to access the JMX console are encrypted to prevent unauthorized access to the information that the JMX console makes available. The user name is stored in the `jmx-console-users.properties` file, which you can find in the `<PPM_Home>\deploy\admin-jmx.war\WEB-INF\classes directory`. To change the user name, change the `ADMIN` parameter value. (You cannot change the password.)

# Configuring or Reconfiguring the Database

The settings described in this section are intended to serve as starting values only. Monitor the database and analyze performance data to fine-tune the settings for your system environment. Tuning an Oracle database involves an Oracle SYS DBA.

The recommendations provided in this section are based on the assumption that PPM Center is the only application that uses the database instance. If other applications share the database, adjust the recommended parameter values accordingly.

## Database Parameters

This section describes the key Oracle database parameters that can affect PPM Center system performance. It also provides parameter settings recommended for the PPM Center environment.

For detailed information about the Oracle parameters described in the following sections, see your Oracle database documentation.

### _B_TREE_BITMAP_PLANS

The `_B_TREE_BITMAP_PLANS` parameter enables creation of interim bitmap representation for tables in a query with only binary index(es).

#### Recommended Setting

Set the `_B_TREE_BITMAP_PLANS` parameter value to FALSE. HP recommends that you set this parameter at the *instance* level instead of at the system level. You can use the `ON LOGON` trigger so that the setting does not interfere with other application schemas that use the database.

To set the parameter to FALSE, execute the following:

```
ALTER SYSTEM SET "_B_TREE_BITMAP_PLANS"=FALSE scope=both
```

This prevents Oracle from unnecessary conversions between ROWID and BITMAPS when executing queries.

### _LIKE_WITH_BIND_AS_EQUALITY

In situations in which the `LIKE` pattern is expected to match very few rows, you can set the hidden parameter `_LIKE_WITH_BIND_AS_EQUALITY` to TRUE. The optimizer treats the predicate as though it were `COLUMN = :BIND`, and uses column density as the selectivity instead of a fixed five percent selectivity factor. The optimizer treats expressions in the format `[indexed-column like :b1]` as it does expressions in the format `[index-column = :b1]`.

Oracle uses some defaults to estimate column selectivity for the `LIKE` operator, but most of the time this estimate is not precise and can cause an index path access to be rejected.

Default selectivity has changed from earlier releases, as follows:

| Release | Selectivity |
|---|---|
| < 9.2.x | 25% |
| >= 9.2.x | 5% |

As Oracle 9i, this parameter also enabled equality costing for expressions in the following format:

```
function(column) LIKE function(:bind)
```

### Recommended Setting

Set the parameter value to TRUE.

## _SORT_ELIMINATION_COST_RATIO

For certain restrictive (with good filters specified) and limited (returns few records) searches, PPM Center uses the FIRST_ROWS_N optimization mode.

If a search such as this also uses SORT on one or more fields returned by the search, Oracle uses the INDEX on the sorted columns under the FIRST_ROW_N optimization, even if other indexes on supplied filters may yield to a better execution plan for a SQL statement. This often leads to a less desirable INDEX FULL SCAN on the index on sorted column.

### Recommended Setting

Set the parameter value to 5. This directs Oracle to consider an execution plan with `ORDER BY` sort elimination, as long as the plan is no more expensive than five times the cost of the best-known plan (that uses sort).

## DB_BLOCK_SIZE

The `DB_BLOCK_SIZE` parameter is used to specify the size (in bytes) of Oracle database blocks. After the database is created, you cannot change this parameter.

### Recommended Setting

Set the `DB_BLOCK_SIZE` parameter value to 8192 (8 KB).

## DB_CACHE_SIZE

The DB_CACHE_SIZE parameter value specifies the size (in KB or MB) of the default buffer pool for buffers with the primary block size (the block size defined by the DB_BLOCK_SIZE parameter).

### Recommended Setting

Specify a DB_CACHE_SIZE parameter value of at least 500 (expressed in MB).

HP recommends that you set a value for this parameter, even if SGA_TARGET is set.

## GLOBAL_NAMES

The GLOBAL_NAMES parameter value determines whether a database link must have the same name as the database to which it connects.

### Recommended Setting

Set GLOBAL_NAMES to FALSE. If you set the value to TRUE, loopback database link creation fails.

If multiple PPM Center test instances use the same database instance, you must set GLOBAL_NAMES to FALSE.

To create a loopback database link with this parameter set to TRUE:

```
create database link <user_name.oracle_sid.domain_name> connect
to <user_name> identified by <password> using <oracle_sid>
```

### Example 1

```
create database link kinadm.dlngrd02.world connect to kinadm
identified by <password> using 'dlngrd02'
```

To use the database link you created:

```
select * from <table_name>@<oracle_sid>
```

### Example 2

```
select * from clis_users@dlngrd02
```

## LOG_BUFFER

The `LOG_BUFFER` parameter value determines the size (in bytes) of the memory area used to save transaction change information. When data is committed, the log buffer is flushed to disk. Small log buffers cause more frequent flushes to disk.

### Recommended Setting

For systems with 50 or more concurrent users, set the parameter value to 1 (expressed in MB).

## MAX_COMMIT_PROPAGATION_DELAY (RAC Only)

The `MAX_COMMIT_PROPAGATION_DELAY` parameter value determines the time delay (in milliseconds) after a change committed on one instance is applied to other instances on the RAC (Real Application Clusters) system.

### Recommended Setting

Set the `MAX_COMMIT_PROPAGATION_DELAY` parameter value to 0.

## NLS_LENGTH_SEMANTICS

The `NLS_LENGTH_SEMANTICS` initialization parameter lets you create CHAR and VARCHAR2 columns using either byte or character length semantics.

### Recommended Setting

Set the `NLS_LENGTH_SEMANTICS` parameter to CHAR. After you do, the VARCHAR2 columns in tables use character length semantics. This means that if, for example, you declare a column as VARCHAR2(30), the column stores 30 characters, and not 30 bytes. In a multi-byte character set, this ensures that adequate space is available for multi-byte characters.

If you are using a single-byte character set, setting `NLS_LENGTH_SEMANTICS` to CHAR makes it easier to transition to a multi-byte character set later.

## OPEN_CURSORS

Oracle uses cursors to handle updates, inserts, deletes, and result sets that queries return. The `OPEN_CURSORS` parameter value determines the number of cursors one session can hold open at a given time.

### Recommended Setting

Set the `OPEN_CURSORS` parameter value to 1000 or higher.

## OPEN_LINKS

The OPEN_LINKS parameter setting affects only HP Deployment Management. It determines the number of open database link connections to other databases that can be active at a given time.

### Recommended Setting

Set the OPEN_LINKS parameter value to 20.

## OPTIMIZER_INDEX_CACHING

The OPTIMIZER_INDEX_CACHING parameter gives the optimizer an estimate of the percentage of indexes cached in the buffer cache. The default value is 0. At this setting, Oracle does not “expect” any index to be cached while deciding the best access plan for a query. If you set this parameter to a higher value, Oracle favors using an index in the ‘IN-list iterator’ and nested loop joins.

### Recommended Setting

For most PPM Center deployments, set the OPTIMIZER_INDEX_CACHING parameter value to 75.

## OPTIMIZER_INDEX_COST_ADJ

The OPTIMIZER_INDEX_COST_ADJ parameter is used to tune optimizer behavior for access path selection by making the optimizer more or less prone to selecting an index access path over a full table scan. The lower this parameter value, the likelier it is that the optimizer will choose index scan over table scan.

The default for this parameter is 100%. With this default value, the optimizer evaluates index access paths at regular cost. With any other value, the optimizer evaluates the access path at that percentage of the regular cost. For example, a setting of 50 makes the index access path look half as expensive as with the default value.

### Recommended Setting

Although there is no perfect OPTIMIZER_INDEX_COST_ADJ value for every PPM Center deployment, HP recommends that you first set this parameter to between 25 and 40, test the performance of the application at this setting, and, based on the results, adjust the value.

## PGA_AGGREGATE_TARGET

The PGA_AGGREGATE_TARGET parameter value determines the aggregate Program Global Area (PGA) memory available to all PPM Server processes attached to the instance. This parameter allows for the automatic sizing of SQL working areas used by memory-intensive SQL operators such as sort, group-by, hash-join, bitmap merge, and bitmap create.

PGA_AGGREGATE_TARGET replaces the traditional SORT_AREA_SIZE parameter. Use it with the WORKAREA_SIZE_POLICY parameter set to AUTO.

### Recommended Setting

Set the PGA_AGGREGATE_TARGET parameter value to a minimum of 450 MB. For very large databases, you can set the parameter to 1 GB.

## PROCESSES

The PROCESSES parameter value determines the maximum number of operating system user processes that can simultaneously connect to the Oracle database. PPM Center uses a pool of database connections. When database activity is required, connections are picked from the pool and the database activity is performed on this existing connection. This process saves the overhead of creating and cleaning up database connections.

### Recommended Setting

Set the PROCESSES parameter value to 20 plus the number of total connections that might be used.

Although concurrent usage and usage nature are factors used to determine the number of connections used, a PPM Server rarely uses more than 25 database connections. If a PPM Server cluster configuration is used, each PPM Server might use 25 database connections.

For single-server configurations, set the parameter value to 45 (the default). For a PPM Server cluster configuration running three servers, set the parameter value to (3 x 25) + 20 = 95.

## SGA_TARGET (Oracle 10G or Later)

The SGA_TARGET parameter value determines the maximum size of all System Global Area (SGA) components combined in the instance. If you specify SGA_TARGET, it is not necessary to specify individual values for SGA components such as SHARED_POOL_SIZE, JAVA_POOL_SIZE, LARGE_POOL_SIZE, and DB_CACHE_SIZE.

### Recommended Setting

Set the SGA_TARGET parameter value to 1.66 GB. If you also set the SGA_MAX_SIZE parameter, its value must be higher than the value set for SGA_TARGET.

## SHARED_POOL_RESERVED_SIZE

The SHARED_POOL_RESERVED_SIZE parameter helps to ensure that a portion of the shared pool (determined by the SHARED_POOL_SIZE parameter) is set aside for large objects. Reserving an area for large objects helps to make sure that requests for a large number of bytes will not fail as a result of shared pool fragmentation.

If you want to place an object in the reserved area, make sure that the object is larger than the SHARED_POOL_RESERVED_MIN_ALLOC value. HP recommends that you use the default value for the SHARED_POOL_RESERVED_MIN_ALLOC parameter.

### Recommended Setting

Set the SHARED_POOL_RESERVED_SIZE parameter value to 10 percent of the shared pool (as determined by the SHARED_POOL_SIZE parameter).

## SHARED_POOL_SIZE

The shared pool contains shared cursors and stored procedures. The SHARED_POOL_SIZE parameter value determines the size (in bytes) of the shared pool. Larger values can improve performance in multiuser systems, but they use more memory. Smaller values use less memory, but they can degrade the performance of multiuser systems.

### Recommended Setting

Set the SHARED_POOL_SIZE parameter value to at least 350 MB.

## WORKAREA_SIZE_POLICY

The WORKAREA_SIZE_POLICY parameter value determines whether work areas operate in automatic or manual mode. If the value is set to AUTO, work areas used by memory-intense operators are sized automatically based on the PGA memory that the system uses and the target PGA memory set for the PGA_AGGREGATE_TARGET parameter. If the value is set to MANUAL, work areas are set manually and based on the value of the *_AREA_SIZE parameter.

### Recommended Setting

Set the parameter value to AUTO.

## Granting Select Privileges to v_$session

If you want PPM Center to keep track of the open database sessions it uses, make sure that a public grant exists on the v_$session dynamic performance table. To do this, connect as SYS to the database that contains the PPM Center database schema, and then issue the following SQL statement:

```
SQL> grant select on v_$session to public
```

➤ You typically assign this grant during PPM Center installation or upgrade.

## Generating Database Links (Oracle Object Migration)

PPM Center can use database links to communicate with other databases. Usually a database link created and associated with a particular environment in PPM Center can be used in situations such as AutoCompleteSQL.

The following are examples of situations in which database links are used:

- Custom object types designed to provide parameter value lists directly from a source or destination database during HP Deployment Management activities
- Some HP Deployment Management Extensions, such as the Extension for Oracle E-Business Suite, to facilitate HP Deployment Management activities

You can define database links on an as-needed basis. For each database link you require (this probably includes a link to the PPM Center database), issue an SQL statement similar to the following in the PPM Center database schema:

```
SQL> create database link DEV_LINK
SQL> connect to APPS identified by APPS
SQL> using 'DEV'
```

For more information about database links, see:

- *HP Deployment Management Extension for Oracle E-Business Suite Guide*
- *HP Object Migrator Guide*
- *HP GL Migrator Guide*
- Oracle's reference document on the SQL language

# Configuring the PPM Workbench to Run as a Java Applet

This section provides the steps to follow to perform the following tasks:

- Enable the SOCKS proxy feature in PPM Center
- Run the PPM Workbench with secure RMI in place
- Provide users on client machines with the required version of the Java plug-in

## Enabling SOCKS Proxy (Optional)

Using the SOCKS proxy feature in PPM Center improves security. With SOCKS proxy enabled, all RMI connections are routed through a central server so that each and every PPM Workbench is not required to contact the application server directly. The SOCKS proxy feature also makes it easier to monitor RMI traffic.

➤ SOCKS proxy support is available for JRE 1.4.2_08 and later versions (including 1.5.x). Clients using JRE 1.4.2_07 or earlier versions cannot use this feature.

To enable the SOCKS proxy feature in PPM Center:

1. Open the `server.conf` file in a text editor.
2. Set the following two parameters:

   ```
   com.kintana.core.server.SOCKS_PROXY_HOST
   com.kintana.core.server.SOCKS_PROXY_PORT
   ```

   For the `com.kintana.core.server.SOCKS_PROXY_HOST` value, provide the hostname of the SOCKS proxy server.

   For the `com.kintana.core.server.SOCKS_PROXY_PORT` value, specify the port on the SOCKS proxy host that accepts proxy connections.

The PPM Server passes the SOCKS proxy configuration forward to the client applet launcher. Users are not required to configure anything.

To specify a different JRE version in the `server.conf` file, reset the `com.kintana.core.server.WORKBENCH_PLUGIN_VERSION` parameter.

For example:

```
com.kintana.core.server.WORKBENCH_PLUGIN_VERSION=1.5.0_02
```

## Running the PPM Workbench with Secure RMI (Optional)

To run the PPM Workbench as a Java applet with secure RMI:

- Specify the complete RMI URL, in the following format, when you start the PPM Workbench:

  ```
  java com.kintana.core.gui.LogonApplet rmis://<host>:<rmi_
  port>/<KintanaServer>
  ```

  You can type the RMI URL at the command line or, on Windows, specify it in a shortcut.

## Providing Users with the Java Plug-In

The Java plug-in is required to access the PPM Workbench interface. When a user starts the PPM Workbench, the system checks the client browser for the Java plug-in, and then determines whether the correct version is installed.

The supported Java plug-in version is specified by the `WORKBENCH_PLUGIN_VERSION` parameter in the `server.conf` file. If the system cannot find the required version, it directs the user to the Sun Microsystems site where the user can download the plug-in and follow the installer wizard prompts to install it.

➤ HP recommends that you leave the `WORKBENCH_PLUGIN_VERSION` parameter default value.

If users who access the PPM Workbench from client machines cannot access the Sun Microsystems Web site to download and install the Java plug-in, you must download the plug-in and make it available to users from within the firewall. You can obtain the plug-in directly from Sun Microsystems at `java.sun.com`.

➤ Consider restricting PPM Workbench access to users who must perform the kind of configuration and administration tasks performed through the PPM Workbench.

# Using the PPM Workbench: What Users Need to Know

This section provides the information that users require to start the PPM Workbench on client machines. It also includes information on how to address JVM-related problems that can arise on client machines. For information on how to set up your Web browser to access the PPM Workbench, see the *Getting Started* guide.

For information on how to access this and other PPM Center documents, see *Accessing PPM Center Documentation* on page 21. For more information about the PPM Workbench, see the *Getting Started* guide.

## Installing and Configuring the Java Plug-In on Client Machines

The server.conf contains one parameter that is associated with the Java plug-in. The `JAVA_PLUGIN_XPI_PATH` parameter specifies the Web location for downloading the cross-platform Java plug-in installer for Firefox browsers. The default setting for this parameter is java.com/en/download/windows_xpi.jsp.

For information about the Java plug-in supported for the current PPM Center version, see the *System Requirements and Compatibility Matrix.* For information about server configuration parameters, see *Server Configuration Parameters* on page 225.

## Setting the Default Web Browser

To run the PPM Workbench interface as an application, users must specify the default browser setting in their user profiles.

To set the default browser setting:

1. From the shortcut bar in the PPM Workbench, select **Edit > User Profiles.**
2. On the **General** tab, in the **Default Browser** field, enter the full path of the default Web browser.

If access to a URL is required, the PPM Workbench uses the default Web browser.

## Starting the PPM Workbench on a Client Machine

To start the PPM Workbench from the PPM Center standard (HTML) interface:

- On the menu bar, select **Administration > Open Workbench.**

➤ If a pop-up blocker is installed and enabled on the Web browser, the PPM Workbench cannot open. The user can configure the blocker to allow pop-ups from PPM Center.

## Troubleshooting Default JVM Problems on Client Machines

If the Java plug-in sets itself as the default JVM for the browser, users can encounter the following problems in the PPM Workbench:

- The PPM Workbench displays a "class not found" exception error.
- Problems occur because other applications you are using require different versions of the Java plug-in.

To resolve these issues, make sure that an installed Java plug-in is not specified as the default.

To remove the default browser association to the Java plug-in:

1. Open the Windows control panel.
2. Double-click the **Java Plug-in** icon.

   The Java Plug-in Control Panel window opens.
3. Click the **About** tab.

   This tab lists the Java plug-in that PPM Center uses, as well as any other Java plug-ins installed.
4. Click the **Browser** tab.
5. Under **Settings,** deselect the checkbox (or checkboxes) for the installed browser (or browsers).
6. Click **Apply.**

   The Java Control Panel displays a message to indicate that you must restart the browser(s) to apply your changes.
7. Click **OK.**

8. Close the Java Plug-in Control Panel window.

After you make this change, other applications can use the Java plug-in version they require, and the PPM Workbench functions correctly.

# What to Do Next

If you plan to perform any of the optional installations described in *Optional Installations* on page 63 (for example, if you are going to install an HP Deployment Management Extension), perform them now. If you have completed your installation tasks, test your system. As you do, be sure you understand the system maintenance tasks you must perform periodically. Those tasks are described in Chapter 6, *Maintaining the System,* on page 133.

# 5 Advanced System Configuration

In This Chapter:

## About this Chapter

The following chapter sections provide information about installations, integrations, and configurations ancillary to the core PPM Center setup. They includes information about installing optional products such as HP Deployment Management Extensions and Accelerators, and the service packs to be delivered after the main PPM Center version. You can also find much useful configuration and integration information.

## Integrating with an LDAP Server

You can integrate PPM Center with any LDAP v3-compliant server such as Microsoft Windows Active Directory. Integrating with an LDAP server helps minimize the setup and maintenance costs associated with user account management. With an LDAP server, the PPM Server authenticates users directly to the LDAP directory server, and does not store passwords in the PPM Center database.

➤ This section addresses LDAP directory server integration with a PPM Center. For information on how to import users from LDAP and on LDAP authentication, see the *Open Interface Guide and Reference.*

In an LDAP environment, the PPM Server authenticates users in the following way:

- The PPM Server binds to the LDAP server using the credentials supplied in the `KINTANA_LDAP_ID` and `KINTANA_LDAP_PASSWORD` server configuration parameters. If passwords are not supplied in the `server.conf` file, the PPM Server performs anonymous authentication.
- The PPM Server tries to obtain the user name by supplying a search filter to the LDAP server in the format `uid=user name`. The `uid` attribute can vary from one LDAP server to another, depending on the information supplied in the `server.conf` file.
- If the PPM Server obtains a name, it tries to rebind to the LDAP server using the name and the password supplied by the user.
- If more than one LDAP server has been specified in the `LDAP_URL` `server.conf` parameter, the PPM Server tries to authenticate against all LDAP servers until it succeeds. If the referral option is enabled, and the user is not logged on to the primary server, the PPM Server also checks the referral server for authentication.

To integrate PPM Center with an LDAP server:

1. Collect the following LDAP server information:

   - LDAP server URL (the default port is 389), in the following format:

     `Ldap://<LDAPSERVER>:PORT`

   - LDAP base distinguished name (DN) for PPM Center users, in the following format:

     `CN=Users,DC=PPMAD,DC=com`

   - LDAP user account and password. (The PPM Server uses this information to look up users.)

   - If you are integrating with SSL-enabled LDAP, collect the following additional information:

     - Entire certificate chain. That is, `root_certificate_authority/intermediate_certificate/host_certificate`, in the BASE-64 encoded X509 (`.cer`) file format.

     - LDAP SSL port number (the default is typically 636).

2. From `<PPM_Home>/bin` on the PPM Server, run the `kConfig.sh` script.

3. Provide the information that you collected in step 1 for the following server directives:

   - `AUTHENTICATION_MODE=PPM,LDAP`

     `LDAP_URL`. Specify the comma-delimited list of LDAP URLs that the PPM Server queries (in the order queried). If you do not specify a port number, the server uses port number 389.

     Example: `ldap://ldap.theurl.com:389`

   - `KINTANA_LDAP_PASSWORD`. Specify the PPM Center password on the LDAP server.

     Example: `#!#ghengis#!#`

     If you run the `kConfig.sh` script, the PPM Server configuration utility automatically encrypts this password. In this case, you must type the exact password string.

     If you modify the `server.conf` file manually, you must encrypt the password string by enclosing it with the `#!!#` character delimiters (as shown in the example), and then set this encrypted string as the `KINTANA_LDAP_PASSWORD` parameter value.

- `KINTANA_LDAP_ID`. Specify the PPM Center account on the LDAP server. The PPM Server uses this to bind to the LDAP server.

  Examples: `KINTANA_LDAP_ID=kintana`, or `\KINTANA_LDAP_ID=CN=kintana,CN=Users,DC=PPMAD,DC=com`.

- `LDAP_BASE_DN`. Specify the base in the LDAP server from which the search is to start. If you do not specify a value, the server queries the LDAP server to determine the base.

  Example: `LDAP_BASE_DN=CN=Users,DC=PPMAD,DC=com`

For an SSL-enabled LDAP server, provide the following additional information:

- `LDAP_SSL_PORT=636`
- `LDAP_KEYSTORE=<`*`JAVA_HOME`*`>/jre/lib/security/cacerts`
- `LDAP_KEYSTORE_PASSWORD=changeit`

The script run makes the required changes to the `server.conf` file, encrypts the LDAP password, and updates the required PPM Center startup files.

4. On the PPM Server, back up the existing `LdapAttribute.conf` file, which is located in the `<`*`PPM_Home`*`>/integration/ldap` directory.

   The `LdapAttribute.conf` file is required for user importation and authentication. The `<`*`PPM_Home`*`>/integration/ldap` directory contains LDAP attribute configuration files for different types of LDAP servers.

5. Copy over the `LdapAttribute.conf` file.

   If you are using Microsoft Active Directory, copy the `<`*`PPM_Home`*`>/integration/ldap/LdapAttribute_AD.conf` file to the `LdapAttribute.conf` file.

   If you are using an iPlanet LDAP Server or Sun Java System Active Server Pages LDAP server, copy the `<`*`PPM_Home`*`>/integration/ldap/LdapAttribute_Netscape.conf` file to the `LdapAttribute.conf` file.

6. If you are integrating with an SSL-enabled LDAP server, do the following:

   a. Get the entire trusted certificate chain of the LDAP server (`Root CA/Intermediate Certificate/host Certificate`, exported as Base-64 encoded `X509.cer` format) from your LDAP server administrator.

➤ If the certificate chain is not in the correct X509.cer format, you can import it to Internet Explorer, and then export it in the correct format.

b. Use the JDK Keytool utility (from jdk 1.4.2 or later) to import the certificate into the *<JAVA_HOME>*/jre/lib/security/cacerts keystore file.

➤ Your system administrator can help you use the JRE Keytool utility to import the LDAP server certificate chain into the JDK cacerts file.

c. Change to the *<JAVA_HOME>*/jre/lib/security directory.

d. Execute the following:

```
keytool -import -trustcacerts -alias <SSL_LDAP_HOST>
-file <SSL_LDAP_CERT.cer> -keystore cacerts
```

➤ The default cacerts keystore password is “changeit.” For tighter security, you may want to change this password.

7. To enable entity ownership and security, do the following:

a. Ensure that the PPM Server is running.

b. Use the Import Users report to import the LDAP user into the KNTA_USERS table on the PPM Server.

For instructions on how to run the Import Users report, see the *Open Interface Guide and Reference.*

If you are running the Import Users report for the first time, edit the LdapAttribute.conf file and comment out MANAGER_USERNAME, LOCATION_MEANING, and DEPARTMENT_MEANING. If you do not make these changes, the import fails and an error message such as “Unknown Manager,” “Unknown Location,” or “Unknown Department” is displayed. The error occurs because the import tries to validate the data before the data is imported. For information on how to address this issue, see the following HP Knowledge Base article:

kb-web.Mercury.com/top5/
kblinkExtern.asp?Conceptid=32339;Product=KINTANA

c. For the **LDAP Import?** option, click **Yes.**

## Authenticating Against Multiple LDAP Domains

PPM Center can handle multiple domains during LDAP authentication. To configure this feature, you add the server configuration parameter LDAP_URL_FULL to the server.conf file.

The values for the LDAP_URL_FULL parameter include a space-delimited (not comma-delimited) list of full LDAP URLs. Each LDAP URL must specify a base distinguished name (DN), which is used in place of the LDAP_BASE_DN server configuration parameter.

The following is an example of how to set the LDAP_URL_FULL parameter:

```
com.kintana.core.server.LDAP_URL_FULL=ldap://
<host>.<yourdomain>.com/CN=Users,DC=<yourdomain>,DC=com ldap://
<host>.<yourdomain>.com/OU=Users2,DC=<yourdomain>,DC=com
```

The LDAP_URL_FULL parameter supersedes the LDAP_URL parameter. That is, if both are specified in the server.conf file, PPM Center uses the value set for LDAP_URL_FULL. If the URLs provided for LDAP_URL_FULL do not have a DN value, PPM Center uses the value set for LDAP_BASE_DN.

➤ To specify a space character inside of an URL, use the URL-encoding scheme, and replace the space with %20. For example, if you have an organizational unit called My Org Unit, then specify My%20Org%20Unit in the LDAP URL.

For more information about server parameters related to LDAP integration, see *LDAP Attribute Parameters* on page 274.

## Validating LDAP Parameters

You can use any of several available GUI tools to validate and troubleshoot the LDAP configuration parameters. For example, Softerra provides Softerra LDAP Browser freeware, which you can download and install. You can then use the LDAP server information you collected in step 1 to create a new LDAP server profile. This will confirm that the information is correct. On the LDAP browser windows at the top, blue line, you can see the DN for a specific resource. Use this to determine the base DN as well as the search filter for the Import Users report.

To download the Softerra LDAP Browser software, go to the following Web site:

> ldapadministrator.com/download/index.php?PHPSESSID=793cd9e97a2be8f9cabcf7c148b14cf4

# Configuring an External Web Server

HP recommends that you use the Web server that is built into the PPM Server unless you have the special Web server requirements described in *Single-Server/External Web Server Configuration* on page 29 and *Server Cluster/External Web Server Configuration* on page 31. The following sections provide information about how to configure an external Web server to work with a PPM Center Server cluster.

PPM Server can run with external Web servers such as Sun Java System Web Server, Microsoft IIS, Apache HTTP Server, Apache-based Web Server (from HP), and IBM HTTP Server. For detailed information on which Web server versions PPM Center supports, and on which platforms, see the *System Requirements and Compatibility Matrix.*

### Using an External Web Server for Multiple PPM Center Instances

You cannot use a single Web server installation on a machine running Windows for multiple instances of PPM Center. The Windows registry imposes this limitation. Integration with an external server involves specifying the worker_file registry directive that points to the `workers.properties` file. The `workers.properties` file tells the redirector (`isapi_redirect.dll`) where to forward the request.

Redirecting to two different instances does not work because each instance requires different workers properties. However, a single Windows registry points to only a single `workers.properties` file.

If you must use an external Web server for multiple PPM Center instances, HP recommends that you either use a UNIX machine to host the Web server, or use a hardware load balancer.

## Overview of External Web Server Configuration

The next sections provide information about how to perform the following tasks, which are required to configure an external Web server.

1. Choose an external Web server.

   ➤ PPM Server can run with external Web servers such as Sun Java System Web Server, Microsoft IIS, Apache HTTP Server, Apache-based Web Server (from HP), and IBM HTTP Server. For detailed information on which Web server versions PPM Center supports, and on which platforms, see the *System Requirements and Compatibility Matrix.*

2. Choose an external Web port.

3. Configure a workers.properties file.
4. Configure a uriworkermap.properties file (Microsoft IIS and Apache-based servers only).
5. Configure the external Web server.
6. Integrate the external Web server with the PPM Server.
7. (Optional) Enable cookie logging on the external Web server.

## Choosing an External Web Port

Choose the port through which the external Web server and the PPM Server(s) are to communicate. Select a port that is not in use on the machine running PPM Center. Later, you identify this port in the PPM Center server.conf file and your workers.properties file.

If you are integrating with an external Web server, you must set the EXTERNAL_WEB_PORT parameter on the PPM Server. This port number is then specified in the workers.properties file that is used by the jakarta 1 redirector.

## Configuring the Workers Properties File

The workers.properties file stores information about the PPM Server(s), including the machine name, ports, and load balance. The external Web server uses this information to direct traffic to PPM Center applications, as required.

The following sections describe how to configure the workers.properties file for Sun Java System Web Server, Microsoft IIS, and Apache-based servers such as Apache HTTP Server, HP-UX, and IBM HTTP Server.

### Configuring the workers.properties File for a Single Server

The *Sample File* on page 102 shows the contents of a sample workers.properties file for a single-server configuration. Information that pertains to a clustered configuration is commented out.

As you edit the workers.properties file, keep the following two requirements in mind:

- The worker name must match the name of PPM Center instance defined for the KINTANA_SERVER_NAME parameter in the server.conf file.
- For Netscape-based Web servers such as Sun Java System Server, you must specify connection_pool_size, connection_pool_minsize and connection_pool_timeout (see comments in the sample file).

### Sample File

```
# JK 1.2.X configuration file. This file tells the external Web
# server how to connect to the PPM Servers.
# Place this file in the location you indicated in your Web
# server configuration.
# List of workers for handling incoming requests.
worker.list=load_balancer
# If "status" worker is defined (see below), then add it to the
# list of workers.
# worker.list=load_balancer,jkstatus

# Defines the PPM Server instances. The
# worker name is the value between the first and second period
# (server1, in this case). Copy this block for each additional
# server in the server cluster. Make sure the port number
# matches the port defined in the EXTERNAL_WEB_PORT parameter
# of the server.conf file, and that the worker name matches the
# PPM Center instance name defined by the
# KINTANA_SERVER_NAME parameter of the server.conf file. Please
# note that, for a server cluster setup, each HP PPM Center node
# has its own KINTANA_SERVER_NAME parameter.
worker.server1.host=localhost
worker.server1.port=8009
worker.server1.type=ajp13
worker.server1.lbfactor=1
# The following three parameters are required for
# Netscape-based Web servers such as Sun ONE Web Server. Set
# connection_pool_size equal to RqThrottle parameter in the Web
# server's magnus.conf file. Keep connection_pool_minsize at 1
# and connection_pool_timeout at 600.
# HP recommends that you not use these parameters with
# Apache-based servers, including IBM HTTP Server, HP Web
# Server, IIS, and Apache itself.
#worker.server1.connection_pool_size=128
#worker.server1.connection_pool_minsize=1
#worker.server1.connection_pool_timeout=600
# Clustered configurations only.
# Defines a second PPM Server instance.#
# worker.server2.host=localhost
# worker.server2.port=8010
# worker.server2.type=ajp13
# worker.server2.lbfactor=1
#See comments above regarding setting the following three
# parameters.
#worker.server2.connection_pool_size=128
#worker.server2.connection_pool_minsize=1
#worker.server2.connection_pool_timeout=600
# Defines the load balancer. Be sure to list all servers in the
# PPM cluster in the balanced_workers group.
worker.load_balancer.type=lb
worker.load_balancer.balanced_workrs=server1,server2
# Optional. Define a special "status" worker. It allows
# monitoring of jk plugin status. If enabled, add it to the list
# of available workers (see above).
#worker.jkstatus.type=status
```

For more information about how to configure a server cluster, see *Configuring a Server Cluster* on page 115.

### Configuring the workers.properties file

To configure a `workers.properties` file:

1. Go to the `<PPM_Home>/integration/webserverplugins/configuration` directory and open the `workers.properties` file in a text editor such as Notepad.
2. Set the `worker.list` parameter to `load_balancer`.
3. For the single server (or for each PPM Server in a cluster), configure the following values:

   a. Set `<worker.name>` to the name of PPM Center instance to which this worker connects. This is the name defined by the `KINTANA_SERVER_NAME` server configuration parameter in the `server.conf` file.

   ➤ For a clustered setup, each PPM Server has its own `KINTANA_SERVER_NAME` parameter.

   b. Set the `worker.server#.host` parameter to the network address of the machine on which PPM Center is installed.

   ➤ If the PPM Center instance runs on the same machine as the Web server, you can use `localhost`.

   c. Set the `worker.server#.port` parameter to the external Web port (`EXTERNAL_WEB_PORT` parameter) to use.

   d. Set the `worker.server#.type` parameter to `ajp13`, which is the protocol used to connect to the remote server.

   e. Set the `worker.server#.lbfactor` parameter to the load balancing factor used to distribute load to the PPM Servers.

   If all servers can handle approximately the same load, assign "1" to each server. If a server can handle twice as much load as another server, assign "2" to that more robust server and "1" to the other server.

4. Set the `worker.load_balancer.type` parameter to `lb`.
5. Set the `worker.load_balancer.balanced_workers` parameter to a comma-delimited list of all servers in the cluster (as configured in step 3).

   Example:

   ```
   worker.load_balancer.balance_workers=worker1,worker2,worker3
   ```

6. (Optional) To enable the JK status page, add a worker of special type “status” (worker.jkstatus.type=status), and then add this worker to the list of workers (worker.list).

### Configuring the uriworkermap.properties File (for Microsoft IIS and Apache-based servers only)

The uriworker.properties file is used to specify mappings between a given URL (or URL pattern) and worker name. The following shows the contents of a sample uriworker.properties file.

```
# /ppm/* must be mapped to one of the workers
/ppm/*=load_balancer
# You can access the JK status page at
# http://web_server_host:web_server_port/jkmanager.
# If you want to enable the JK status page, uncomment the
# following line.
#/jkmanager=jkstatus
```

Each line of uriworker.properties file represent a single mapping in the format *<URL_PATTERN>* = *<WORKER_NAME>*. If the Web server processes a URL that matches *<URL_PATTERN>*, then *<WORKER_NAME>* is used to serve this request. *<WORKER_NAME>* must be defined in the workers.properties file.

## Configuring the External Web Server

This section provides information about how to set up the following external PPM Center–supported Web servers:

- Sun Java System Web Server
- Microsoft IIS
- Apache HTTP Server
- HP-UX Apache-based Web Server
- IBM HTTP Server

For a list of supported versions, see *System Requirements and Compatibility Matrix.*

## Configuring the Sun Java System Web Server

To configure the Sun Java System Web Server to run as the external Web server for the PPM Server:

1. Connect to the Sun Java System administration server and create a new server named “PPM.”

   The `https`-PPM directory is created. This directory contains two files: `magnus.conf` and `obj.conf`.

2. Stop the PPM Server.

   For information about how to stop the PPM Server, see *Starting and Stopping the PPM Server* on page 66.

3. Place the configured `workers.properties` file (see *Configuring the Workers Properties File* on page 102) in the `<Sun_Home>/https-<webserver_name>/config` directory.

4. Copy the `nsapi_redirector.so` plug-in to any directory on the machine running the Sun Java System Web Server.

   The Web server must have permissions to read and execute this file.

5. Add the following two lines to the `magnus.conf` file (the text can wrap, but each “`init fn=`” must be a continuous line with no spaces):

   ```
   Init fn="load-modules" shlib="<path_to_nsapi_redirector>/
   nsapi_redirector.so" funcs="jk_init,jk_service"

   Init fn="jk_init" worker_file=<PPM_Home>/workers.properties
   log_level="error" log_file=<path_to_log_files>/ppm_
   server.log
   ```

6. Add the following line to `obj.conf` at the beginning of the “Object” section (that is, after `<Object name=default>`):

   ```
   NameTrans fn="assign-name" from="/ppm/*"name=<ppm_servlet>
   ```

7. Place the following text after the “Object” section (after `</Object>`):

   ```
   <Object name="ppm_servlet">
   Service fn="jk_service" worker=<load_balancer>
   </Object>
   ```

   The `<ppm_servlet>` strings must match.

➤ The `worker` attribute specifies the name of the JK worker used to serve requests with URLs that match the `path` attribute, which is `/ppm/*` in this case.

### Enabling Cookie Logging on the Sun Java System Web Server (Optional)

To enable cookie logging:

1. Stop the Sun Java System Web Server.
2. In the magnus.conf file, find the line that initializes flex. The line begins with the following text:

   ```
   Init fn=flex-init
   ```

3. Append the following string to the end of this line:

   ```
   %Req->headers.cookie.JSESSIONID%
   ```

   The line now looks as follows:

   ```
   Init fn=flex-init access="$accesslog" format.access=
   "%Ses->client.ip% - %Req->vars.auth-user%[%SYSDATE%]
   \"%Req->reqpb.clf-request%\" %Req->srvhdrs.clf-status%
   %Req->srvhdrs.content-length%"
   JSESSIONID=%Req->headers.cookie.JSESSIONID%
   ```

4. Restart the Web server.

## Configuring the Microsoft Internet Information Services 6.0 Web Server

To configure the Microsoft Internet Information Services (IIS) 6.0 Web server on Windows:

1. Create a virtual directory named jakarta that points to the IIS scripts directory, as follows:

   a. Select **Start > Control Panel > Administrative Tools > Internet Information Services Manager.**

   b. Copy the workers.properties file, the uriworkermap.properties file, and <*PPM_Home*>/integration/webserverplugins/iis/windows/x86-32/isapi_redirect.dll file to the c:\inetpub\scripts directory.

   c. In the IIS Services Manager, right-click the Web site you plan to integrate with the PPM Server, and then select **New > Virtual Directory** on the shortcut menu.

   d. On the first page of the Virtual Directory Creation Wizard, click **Next.**

   e. On the Virtual Directory Alias page, under **Alias,** type **jakarta.**

   f. Click **Next.**

g. On the Web Site Content Directory page, under **Directory**, type the full path to the `isapi_redirect.dll` file (`C:\inetpub\scripts\isapi_redirect.dll`).

   Depending on the IIS root directory configuration, the drive and directory may vary. This directory must have run permission.

h. Click **Next.**

i. On the Access Permission page, select **Read, Run scripts (such as ASP)**, and **Execute (such as ISAPI application or cgi)**, and then click **Next.**

j. Click **Finish.**

2. Change directory security to set up anonymous access, as follows:

   ➤ Anonymous access allows anyone to visit the public areas of your Web site but prevents unauthorized users from gaining access to your Web server's critical administrative features and restricted information.

   If you do not change directory security, you may encounter an "Access Denied" error message.

   a. From the Control Panel, open **Administrator Tools**, and then open the Windows Internet Information Services Manager.

   b. In IIS Manager, click **Web Site**, right-click the Web site instance you are integrating with PPM Server, and then select **Properties** on the shortcut menu.

   c. Click the **Directory Security** tab.

   d. In the **Authentication and Access Control** section, click **Edit.**

   e. Select the **Enable anonymous access** checkbox, and then browse for IUSR_<HOST>.

   f. Clear everything selected in the **Authenticated access** section.

   g. Click **OK.**

3. Configure a `workers.properties` file and a `uriworkermap.properties` file, as described in *Configuring the Workers Properties File* on page 102 and *Configuring the uriworkermap.properties File (for Microsoft IIS and Apache-based servers only)* on page 104.

4. Configure IIS to load `isapi_redirect.dll` as a filter, as follows:

   a. To define registry values for IIS with Apache Jakarta Tomcat Connector (JK):

      i. Add the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software
Foundation\Jakarta Isapi Redirector\1.0
```

      ii. Add a string value with the name `extension_uri` and set `/jakarta/isapi_redirect.dll` as its value.

      The `extension_uri` key in the Windows registry must point to the full path of the `isapi_redirect.dll` file.

      iii. Add a string value with the name `worker_file` and set its value as the full path to the `workers.properties` file. That is, `<`*`ISAPI_REDIRECTOR_HOME`*`>\workers.properties` (for example, `c:\inetpub\scripts\workers.properties`).

      iv. Add a string value with the name `log_level` and set `ERROR` as its value.

      For more verbose logging, use `DEBUG` or `INFO.`

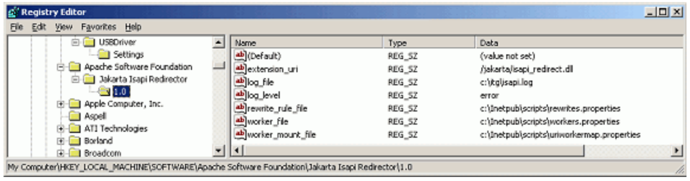
      v. Add a string value with the name `log_file` and specify the directory in which you want the log file to reside. (Include the log file name. For example, `c:\inetpub\scripts\isapi.log`.)

      vi. Add a string value with the name `worker_mount_file` and a value that is the full path to your `uriworkermap.properties` file (`c:\inetpub\scripts\uriworkermap.properties`).

      vii. Create an empty file named `rewrites.properties,` and save it to `c:\inetpub\scripts`. Add a string value with the name `rewrite_rule_file` and assign it a value that is the full path to the `rewrites.properties` file (`c:\inetpub\scripts\rewrites.properties`).

      The previous step is required as the result of a known issue in JK 1.2.18. For detailed information about this issue, go to the Web site http://issues.apache.org/bugzilla/show_bug.cgi?id=40384.

The following figure shows a correctly configured registry:



| Name | Type | Data |
|---|---|---|
| (Default) | REG_SZ | (value not set) |
| extension_uri | REG_SZ | /jakarta/isapi_redirect.dll |
| log_file | REG_SZ | c:\ktg\isapi.log |
| log_level | REG_SZ | error |
| rewrite_rule_file | REG_SZ | c:\Inetpub\scripts\rewrites.properties |
| worker_file | REG_SZ | c:\Inetpub\scripts\workers.properties |
| worker_mount_file | REG_SZ | c:\Inetpub\scripts\uriworkermap.properties |

My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software Foundation\Jakarta Isapi Redirector\1.0

b. Restart your Windows system.

c. Select **Start > Control Panel > Administrative Tools > Internet Information Services Manager.**

➤ Perform the following steps at the Web-sites level.

d. Right-click the Web site name, and then click **Properties** on the shortcut menu.

   The Properties dialog box opens.

e. Click the **ISAPI Filter** tab.

f. Click **Add.**

   The Filter Properties window opens.

g. In the **Filter Name** field, type **`jakarta`**.

h. In the **Executable** field, enter the full path to the `isapi_redirect.dll` file `(c:\inetpub\scripts\isapi_redirect.dll`).

i. Click **OK.**

5. Allow Tomcat's redirector DLL in Web service extensions, as follows:

   a. In the Windows management console, click **Web Services Extensions.**

   b. Select **Add a new Web service extension.**

   c. Type the extension name (for example, **`Jakarta-Tomcat`**).

   d. Type the path to the `isapi_redirect.dll` file `(c:\inetpub\scripts\isapi_redirect.dll).`

   e. Select **Set extension status to Allowed.**

   f. Click **Add.**

   g. Click **OK.**

6. Restart the IIS service.

➤ Restarting the Web site is not enough. You must restart World Wide Web Publishing Service from the Services management console.

7. Start the PPM Server(s).

### Enabling Cookie Logging on Microsoft IIS 6.0 (Optional)

To enable cookie logging on IIS 6.0:

1. Open IIS.
2. Select a Web or FTP site and open its property sheets.
3. Select **Enable Logging.**
4. Click **Properties.**
5. On the **Extended Properties** page, select **Cookies.**
6. Click **Apply.**

## Configuring an Apache-Based Web Server (Apache HTTP Server, HP-UX Apache-based Web Server, or IBM HTTP Server)

The following sections provide the steps you use to:

- If (and only if) a precompiled binary does not work on your system, compile a binary of JK.
- Configure Apache HTTP Server

### Compiling a Binary of JK

Configuring an Apache-based Web server on UNIX requires a dynamically linkable JK module binary named `mod_jk.so`. In most cases, the `<PPM_Home>/integration/webserverplugins/<Web_Server_Name>` directory contains precompiled binaries of JK for several operating systems. Before you try to compile the JK module, check this directory to see if it contains the binaries required for your system.

If a precompiled binary is unavailable, perform the following steps.

To compile a binary of JK:

1. Download and unpack a source code bundle from the following Web site:

   tomcat.apache.org/connectors-doc/index.html

2. Change to the following directory:

```
tomcat-connectors-<version>-src/native
```

3. Run the configuration script, as follows:

```
./configure --with-apxs=/<path_to_apache_bin>/apxs
```

   The configuration script generates the `make` files for the current machine environment. The `make` files are required to run the `make` command, as described in the next step.

4. Run the `make` command to build the Apache module that forwards requests from the Apache HTTP Server to the PPM Server using the AJP13 protocol.

➤ For more details on how to recompile the connector, go to the following Web site: http://tomcat.apache.org/connectors-doc/index.html

### Enabling Cookie Logging on Apache 2.0 (Optional)

To enable cookie logging on Apache 2.0:

1. Open the Apache `httpd.conf` file in a text editor.

2. Find the line of text that begins with the following string:

```
LogFormat "%h %l %u %t \"%r\"%>s %b
```

3. After "`%b,`" type the following:

```
%{Cookie}i"
```

   The log format and custom log lines now look as follows:

```
LogFormat "%h %l %u %t \"%r\"%>s %b %{Cookie}i" common
CustomLog logs/access_log common
```

4. Save and close the `httpd.conf` file.

## Enabling Secure Sockets Layer on an External Web Server

To enable Secure Sockets Layer (SSL) on the Web server:

1. Generate a certificate signing request (CSR) for the server on which you plan to install the SSL certificate.

   To do this, use the software that your external Web server provides. If you do not know what software your server uses, contact the Web server vendor for that information.

2. Submit the CSR to a certificate authority (such as VeriSign).

➤ It may take several days for the certificate authority to validate the company.

3. After you obtain the SSL certificate, install it on your Web server.
4. Contact your Web server administrator or Web server vendor to help you enable SSL on the Web server.
5. If your external Web server or hardware load balancer uses SSL, open the `server.conf` file and change the server configuration parameter `BASE_URL` to `https://<`*`Web_Server`*`>:<`*`Port`*`>`.

➤ By default, the https typically runs on port 443 on the Web server. If you use a port other than 443, you must specify the port number in the `BASE_URL`.

6. Restart the Web server.

# Integrating an External Web Server with a PPM Server

To integrate an external Web server with the PPM Server, perform the following tasks:

1. Stop the PPM Server.

   For information about how to do this, see *Starting and Stopping the PPM Server* on page 66.

2. Set the server configuration parameter values.
3. Validate the integration.

The following sections provide the steps you use to set the `server.conf` parameters and verify the integration.

## Setting the Server Configuration Parameters

To set the server configuration parameters:

1. Back up the `<PPM_Home>/server.conf` file.
2. Open the `server.conf` file in a text editor such as Notepad.
3. Add `com.kintana.core.server.EXTERNAL_WEB_PORT,` and set it to the port number in the `workers.properties` file.
4. Change `BASE_URL` to the base URL of the external Web server.

   ➤ If your external Web server or hardware load balancer uses Secure Sockets Layer (SSL), you must change the BASE_URL parameter value to `https://<Web_Server>:<Port_Number>`.

   By default, the https runs on port 443 on the Web server. If you use a port other than 443, you must specify that port number in the BASE_URL.

5. Do one of the following:
   - For IIS Web servers, add:

     ```
     com.kintana.core.server.WEB_SERVER=IIS
     ```

   - For Apache and all other Web servers, add:

     ```
     com.kintana.core.server.WEB_SERVER=APACHE
     ```

6. Save and close the `server.conf` file.
7. Run the `kUpdateHtml.sh` script.

For more information about the `BASE_URL` parameter, see Appendix A, *Server Configuration Parameters,* on page 225. For more information about the `kUpdateHtml.sh` script, see *kUpdateHtml.sh* on page 286.

## Verifying the Integration

To verify the integration between the external Web server and the PPM Server:

1. Start the external Web server and check for errors.
2. Start the PPM Server and check for errors.
3. In a supported browser, open the page `<BASE_URL>/ppm/dashboard/app/portal/PageView.jsp.`

   ➤ You must use the complete path. Specifying only `<BASE_URL>/ppm` does not work.

For information about how to start the PPM Server, see *Starting and Stopping the PPM Server* on page 66. For information about supported browsers, see the *System Requirements and Compatibility Matrix*.

# Configuring a Server Cluster

This section provides the following information about server clustering in the PPM Center environment:

- Server clustering overview
- Server clustering configuration
- Starting and stopping servers in a cluster
- Validating the cluster configuration

## Overview of Server Clustering

Before you begin to set up a PPM Server cluster, review the information provided in Chapter 2, *System Overview,* on page 23, particularly *Server Cluster Configurations* on page 30. The concepts described in this section are key to understanding configuring server clusters.

### KINTANA_SERVER_NAME and the <*PPM_Home*>/server directory

A PPM Server consists of the common code located in the `<PPM_Home>` directory, as well as the directory of files that make up the actual PPM Server. These are separate directories in the `<PPM_Home>/server` directory.

*Server nodes* are the individual PPM Servers that comprise the server cluster. Each node, or server, in a cluster requires a separate directory in the `<PPM_Home>/server` directory. The directory names are the server names, and they are configured in `server.conf` with the `KINTANA_SERVER_NAME` parameter. Each server directory in `<PPM_Home>/server` must have a corresponding `KINTANA_SERVER_NAME` defined in `server.conf`, all with the same assigned value.

➤ Server directories cannot contain spaces, commas, or other non-alphanumeric characters, except for hyphens (-) or underscores (_). For example, `server1_1` is a valid name, but `server 1,1` is not.

### @node Directive in the server.conf File

The `@node` directive in the `server.conf` file (that is, `@node` alone on a line) tells the PPM Server that the variables after `@node` are specific to one node in the cluster. You must specify one `@node` directive for each server in your cluster. Variables displayed above the first `@node` are common to all servers.

A common practice in single-server environments is to append new server configuration parameters to the bottom of the file. If you add a configuration parameter to the end of a file associated with a clustered environment, the parameter applies only to the last node defined in the file.

Make sure that you add variables that are common to all nodes in a cluster to the top of the `server.conf` file, before the first `@node` directive.

### Server Parameters Affected by Clustering

*Table 5-1* on page 116 shows which server configuration variables to define for each server in a server cluster, based on the type of clustering used. For more information about these parameters, see *Server Configuration Parameters* on page 225.

Table 5-1. Server configuration parameters affected by clustering (page 1 of 2)

| Parameter | External Web Server, Single Machine | External Web Server, Multiple Machines | Hardware Load Balancer, Multiple Machines |
|---|---|---|---|
| com.kintana.core.server. KINTANA_SERVER_ NAME | X | X | |
| com.kintana.core.server. ATTACHMENT_ DIRNAME | | X | X |
| com.kintana.core.server. BASE_PATH | | X<br>The `BASE_PATH` specified for the core server is inherited by all of the @node sections. Specify this in an individual @node only if the value is different for that specific instance. | X |
| com.kintana.core.server. ORACLE_HOME | | X | X |
| com.kintana.core.server. BASE_URL | X | X | X |

Table 5-1. Server configuration parameters affected by clustering (page 2 of 2)

| Parameter | External Web Server, Single Machine | External Web Server, Multiple Machines | Hardware Load Balancer, Multiple Machines |
|---|---|---|---|
| com.kintana.core.server.BASE_LOG_DIR | | X | |
| com.kintana.core.server.HTTP_PORT | X | X | X |
| com.kintana.core.server.EXTERNAL_WEB_PORT | X | X | |
| com.kintana.core.server.RMI_URL | X | X | X |
| com.kintana.core.server.TRANSFER_PATH | | X | X |
| com.kintana.core.server.PACKAGE_LOG_DIR | | X | X |
| com.kintana.core.server.REPORT_DIR | | X | X |
| com.kintana.core.server.REQUEST_LOG_DIR | | X | X |

➤ If the servers in a server cluster are running on different operating systems, then each @node section requires the `SERVER_NAME=<HOST_NAME> server.conf` directive.

### Overview of Server Cluster Configuration

To configure a server cluster, perform the following tasks:

1. If you are using an external Web server, set up your PPM Server for integration with an external Web server in single-server mode.
2. Stop the PPM Server.

   For information about how to stop the PPM Server, see *Starting and Stopping the PPM Server* on page 66.

3. If you are using an external Web server:

    a. Stop the external Web server.

    b. Configure the workers.properties file to include information for the multiple cluster nodes. Each node requires an external Web port defined (using the EXTERNAL_WEB_PORT configuration parameter).

       For information about how to configure the workers.properties file, see *Configuring the Workers Properties File* on page 102.

4. Configure the server nodes on the file system.

5. Configure the server nodes in the server.conf file.

## Server Cluster Configuration

This section provides the steps you use to configure the following server cluster setups (*Table 5-1* on page 116):

- External Web server, single machine
- External Web server, multiple machines
- Hardware load balancer, multiple machines

### External Web Server, Single Machine

To set up a cluster with an external Web server on a single machine:

1. Stop the PPM Server.

   For information about how to stop the PPM Server, see *Starting and Stopping the PPM Server* on page 66.

2. Stop the external Web server.

3. Add the new node and relevant information to the workers.properties file.

   For information about how to configure the workers.properties file, see *Configuring the Workers Properties File* on page 102.

Example for a Sun Java Web Server:

```
# node1, already defined when integrating with
# the external Web server
worker.server1.host=machine1
worker.server1.port=8009
worker.server1.type=ajp13
worker.server1.lbfactor=1

# node2, as part of a cluster
worker.server2.host=machine1
worker.server2.port=8010
worker.server2.type=ajp13
worker.server2.lbfactor=1

# Define the load balancer. Be sure to list all servers
# in the PPM Server cluster in the
# balanced_workers group. When adding new nodes,
# add them in the last line to ensure that the load
# is balanced.
worker.load_balancer.type=lb
worker.load_balancer.balanced_workers=server1,server2
```

4. Create the new `<PPM_Home>/server` directory.

   Make a copy of the first server directory (the entire directory) at the same level as the first.

   Example:

```
<PPM_Home>
+ server
      + node1
      + node2
```

➤ Use the value specified for the `KINTANA_SERVER_NAME` parameter in the `server.conf` file that corresponds to the subdirectory node for that system.

5. Configure `server.conf` to include the new node.

For a single-machine clustered environment, the following is a typical `server.conf` excerpt:

```
# Map the name of the first server to server/node1
# and set the Web port.
# These values should match the workers.properties file.
com.kintana.core.server.KINTANA_SERVER_NAME=node1
com.kintana.core.server.EXTERNAL_WEB_PORT=8009

@node
# Map the name of this node to server/node2
# If the instance, @node, is on different system than the
# primary, you must specify the host machine name:
# "com.kintana.core.server.SERVER_NAME=<HOST_NAME>"
com.kintana.core.server.KINTANA_SERVER_NAME=node2
com.kintana.core.server.EXTERNAL_WEB_PORT=8010
# Each node must have its own RMI_URL for the Workbench
com.kintana.core.server.RMI_URL=
rmi://machine1:21601/KintanaServer
# Each node must have its own internal Web port
com.kintana.core.server.HTTP_PORT=21700
```

6. To apply the changes to all the servers in the cluster, from `<PPM_Home>/bin`, run `kUpdateHtml.sh`.

7. If you have additional nodes in your cluster, repeat step 1 through step 6.

8. If the PPM Server is running in a Windows environment, start it using the Windows service called "`PPM Server_name`," where `server_name` is the value of the `KINTANA_SERVER_NAME` parameter for the node in the cluster.

9. Generate a new service for the new node, as follows:

   a. From `<PPM_Home>/bin`, run `kConfig.sh`.

      The configuration wizard starts up.

   b. Select **Configure Windows Services.**

   c. Follow the wizard prompts to create the service.

10. To validate the cluster, use the procedure provided in *Verifying Successful Cluster Configuration* on page 125.

## External Web Server, Multiple Machines

In a server cluster, an *<PPM_Home>* directory must reside on each machine, each with a server running against the same database.

To set up a cluster with an external Web server on multiple machines:

1. Install the PPM Server on the first machine in the cluster and configure it so that it is integrated with an external Web server.

   For information about how to configure a machine for integration with an external Web server, see *Configuring an External Web Server* on page 100.

2. Stop the PPM Server.

   For information about how to stop the PPM Server, see *Starting and Stopping the PPM Server* on page 66.

3. Stop the external Web server.

4. Make sure that the common directories that the servers use (`<PPM_Home>/logs`, `<PPM_Home>/reports`, `<PPM_Home>/attachments`, and `<PPM_Home>/transfers`) are shared.

   ➤ Set the permissions for the shared directories so that users of each machine in the cluster can read from and write to them.

5. Add the new node and relevant information to the `workers.properties` file.

   Example of a `workers.properties` file on Sun Java Web Server:

```
# node1, already defined when integrating with
# the external Web server
worker.server1.host=machine1
worker.server1.port=8009
worker.server1.type=ajp13
worker.server1.lbfactor=1

# node2, as part of a cluster on a different host
worker.server2.host=machine2
worker.server2.port=8010
worker.server2.type=ajp13
worker.server2.lbfactor=1

# Define the load balancer. Be sure to list all servers
# in the PPM Server cluster in the
# balanced_workers group. When adding new nodes,
# add them in the last line to make sure the load
# is balanced.
worker.load_balancer.type=lb
worker.load_balancer.balanced_workers=server1,server2
```

6. Configure `server.conf` to include the new node.

   The following is a typical `server.conf` excerpt for a multiple-machine clustered environment:

   If the servers in a server cluster are running on different operating systems, then each `@node` section requires the `SERVER_NAME=<HOST_NAME> server.conf` directive.

```
@node
#
# The name that identifies this server instance. If you are
# running multiple servers on the same machine, this name
# must be unique for each server. This value is also used as
# the Windows service name.
# NOTE: All HP PPM Server services are prefaced with the
# string 'HP PPM'.
#
com.kintana.core.server.KINTANA_SERVER_NAME=PPM710
#
# DNS name or IP address of the machine that hosts the
# PPM Center.
#
com.kintana.core.server.SERVER_NAME=servername

#
# The PPM Server uses the HTTP_PORT value
# to identify which port is used to communicate with the
# built-in HTTP server. This is a unique port number,
# distinct from Web server, SQL*Net, and RMI ports. The
# default HTTP port is 8080, but any port number above 1024
# is valid, depending upon system restrictions.
#
com.kintana.core.server.HTTP_PORT=8085

#
# Include pointers to shared log directories.
com.kintana.core.server.BASE_LOG_DIR=/shared/logs
com.kintana.core.server.PACKAGE_LOG_DIR=/shared/logs
com.kintana.core.server.REPORT_DIR=/shared/reports
com.kintana.core.server.REQUEST_LOG_DIR=/shared/logs
com.kintana.core.server.TRANSFER_PATH=/shared/transfers
```

You must specify the `BASE_LOG_DIR`, `REPORT_DIR`, `ATTACHMENT_DIRNAME`, and `TRANSFER_PATH`. The rest of the log directories are derived from these four directories. Consider specifying these above the first `@node` so that you do not have to specify them in each and every `@node` section.

On a Windows system, you must use the UNC format (`\\<hostname>\<drive_letter>\<path>`). You cannot use the local shared drive letter. Use forward slashes (for example, `//<host_name>/<drive_letter><path>`). To enable a node to share these directories, you must start the PPM Windows services using the PPM Center user account that has read and write permission on the shared host.

Example:

```
//com.kintana.core.server.TRANSFER_PATH=//kiwi/e$/PPM_Prod/
transferpath
```

On a UNIX system, you must NFS-mount the shared directories locally with the same directory structure.

```
# ORACLE_HOME of machine2
com.kintana.core.server.ORACLE_HOME=/opt/oracle

# <PPM_Home> for this node
com.kintana.core.server.BASE_PATH=/home/PPM

# Note that machine2 and 8010 should match
# the workers.properties file.
com.kintana.core.server.RMI_URL=
rmi://machine2:20001/KintanaServer
com.kintana.core.server.EXTERNAL_WEB_PORT=8010
com.kintana.core.server.KINTANA_SERVER_NAME=node2

#
# The absolute pathname to the directory where attachments
# are to be stored. Because this directory has the potential
# to grow very large, HP recommends that you not place
# it in the <PPM_Home> directory tree.
# NOTE: This directory must give read/write access to
# Web browsers.
#
com.kintana.core.server.ATTACHMENT_DIRNAME=C:/PPM710/
attachments
```

7. Repeat step 1 through step 6 for all nodes in the cluster.

8. After you configure the first server to include all additional nodes, copy the entire `<PPM_Home>/server` directory from machine1 to machine2, to the `BASE_PATH` defined in the `@node` directive.

   Zip the file, send it using FTP, and then unzip it at the destination.

9. After you copy the file, change the directory to `<PPM_Home>/server` on the new machine, and then rename the `node1` directory to `node2`.

   The server name must match the value set for the `KINTANA_SERVER_NAME` parameter.

For example, the directories on machine1 could be:

```
<PPM_Home>
   server/
      node1
```

The directories on machine2 could be:

```
<PPM_Home>
   server/
      node2
```

10. Put a new license on machine2, as required by the new IP address.
11. Run `kUpdateHtml.sh` on all servers to apply the `server.conf` changes.
12. Start the PPM Server using the Windows service.

    In a multiple-machine configuration, you must generate the services on all machines running Windows.

13. Generate a new service for the new node, as follows:

    a. From `<PPM_Home>/bin`, run `kConfig.sh`.

       The configuration wizard starts up.

    b. Select **Configure Windows Services.**

    c. Follow the prompts to create the service.

➤ The keys in the security directory are required to read encrypted values in `server.conf` and the database. The same keys must be present on all nodes in the cluster.

## Hardware Load Balancer, Multiple Machines

You can use a hardware load balancer as the front end of a PPM Server cluster configuration. A hardware load balancer is similar to an HTTP reverse-proxy server and forwards HTTP requests.

All PPM Servers in a server cluster must listen for HTTP requests on a unique port. Each server in the cluster must have its `HTTP_PORT` parameter set to a unique value that does not conflict with other external applications. You specify this parameter value for all servers in a cluster in the `@node` section of the `server.conf` file.

➤ Sticky sessions are required for hardware load balancing in the PPM Center environment.

## Starting and Stopping Servers in a Cluster

If you stop any server in a PPM Server cluster, the PPM Server cluster continues to operate as long as at least one server in the cluster is running. If a server stops, the HP PPM Web server module detects that the server is unavailable and stops sending it HTTP requests. When the server becomes available again, the HP PPM Web server module detects the server and sends it requests again.

The procedures used to start and stop the primary server in a cluster are identical to the procedures used to start and stop the server in a single-server configuration. For information, see *Starting and Stopping the PPM Server* on page 66.

To start a secondary server, use the `-name server-name` argument in the `kStart.sh` script, as follows:

```
sh ./kStart.sh -name=<secondary_server> -now -user <usr_name>
```

To stop a secondary server, use the `-name server-name=KINTANA_SERVER_NAME=server/server-name` argument in the `kStop.sh` script, as follows:

```
sh ./kStop.sh -name=<secondary_server> -now -user <usr_name>
```

On Windows, there is one service (called "`HP PPM <server-name>`") per server. If you prefer to use the Windows shell command line to start servers instead of using Windows Services, you can use the `kStarts.sh` script.

If you do not have a script to start or stop all servers in a cluster, you can write custom scripts to perform these tasks. For example, the following script for the UNIX environment starts all three servers in a cluster configuration (if all nodes are on the same machine):

```
#!/bin/sh
./kStart.sh -name serv1
./kStart.sh -name serv2
./kStart.sh -name serv3
```

The following script stops all three servers in a cluster configuration:

```
#!/bin/sh
./kStop.sh -name serv1
./kStop.sh -name serv2
./kStop.sh -name serv3
```

If you make a change to the `server.conf` file that affects more than one server in a cluster, you must:

- Stop and restart all the servers in the cluster.

- Update the server configuration file (`server.conf`) on all machines.

## Verifying Successful Cluster Configuration

To verify successful server cluster configuration:

1. If you are using an external Web server, start it and check for errors.

   If the server does not start, make sure that the values in the `workers.properties` file are correct. If you have already validated the external Web server configuration, the problem is likely in this file.

2. Start, and then try to connect to, one of the servers.

   If you cannot connect to the server, check the `server.conf` file and correct any errors you find.

3. Start the remaining servers in the cluster.

4. Use the `kStatus.sh` script to confirm that all server nodes are running.

   If a node is not running, check the server log files in `<PPM_Home>/server/<server name>/log` for errors.

   Example:

   ```
   > cd <PPM_Home>/bin
   > sh kStatus.sh
   delorean[6]bin: sh kStatus.sh
   JAVA_HOME = /usr/j2sdk1.4.2_06
   java version "1.4.2_06"
   Java(TM) 2 Runtime Environment, Standard Edition (build
   1.4.2_06-b03)
   Java HotSpot(TM) Client VM (build 1.4.2_06-b03, mixed mode)
   Checking rmi://machine1:28001/KintanaServer
   --> running (load: 0.0, mode: NORMAL)

   Checking rmi://machine2:29001/KintanaServer
   --> running (load: 1.0, mode: NORMAL)
   ```

In addition, ensure that:

- Multiple users logging on are automatically distributed to all servers. Use server reports to verify which users are logged on to which servers.

- If you shut down a server, users logged on to the other servers can continue to work. Users logged on to the shut down server can log on again and continue to work.

- If you shut down a server that was running services, those services automatically start on one of the other servers. You can use server reports to determine which server is running services.

For information about server reports and how to run them, see *Running Server Reports from the Admin Tools Window* on page 138 and *Running Server Reports from the Command Line* on page 141.

### Multicast Settings for Server Cluster Configurations

Multicast must be enabled on network components such as network cards, switches, and routers. To avoid conflicts between cluster environments, consider the following:

- Each cluster environment must have `MULTICAST_IP` and/or `MULTICAST_PORT` values specified both in the `server.conf` file, and in the `cache.conf` file.
- All members in a cluster, such as a production cluster, must use the same `MULTICAST_IP` and `MULTICAST_PORT` settings.
- If several cluster environments (for example, for Development, Test, and Production) are on the same network segment, you must change the `MULTICAST_IP` and `MULTICAST_PORT` parameters in the `server.conf` file, and the corresponding settings in `cache.conf` file.
- On Linux systems, if multiple clusters use the same `MULTICAST_IP` and `MULTICAST_PORT` parameter values, the PPM Server(s) may not start up.
- If clusters other than those related to PPM Center are set up, and these use the same multicast IP/port, the environment may also conflict.

# Implementing Single Sign-On with PPM Center

This section provides the steps to use to implement either Web remote single sign-on or generic single sign-on with PPM Center.

## Implementing Web Remote Single Sign-On with PPM Center

This implementation is based on NTLM authentication and requires that the PPM Server(s) be integrated with an external Web server running Microsoft IIS.

Web remote single sign-on works with PPM Center as follows:

1. A user logs in to a Windows desktop.

2. The user accesses PPM Center through the external (IIS) Web server.

3. The user is authenticated through the Windows user account to IIS and the user name is passed to the PPM Server by way of the `REMOTE_USER` HTTP header field.

4. If the user is a valid PPM Center user, the standard interface and PPM Dashboard will open.

### Requirements for Implementing Web Remote Single Sign-On

To implement Web remote single sign-on, your system must meet the following requirements:

- PPM Center must be set up with an external Microsoft IIS Web server. For information on how to do this, see *Integrating an External Web Server with a PPM Server* on page 113.

- Clients must use Microsoft Internet Explorer to log on to PPM Center. Logon credentials are not automatically passed from Web browsers other than Internet Explorer (for example, Firefox) when connecting to IIS.

## Setting Up Web Remote Single Sign-On with PPM Center

To configure Web remote single sign-on with PPM Center:

1. Integrate the external IIS Web server with the PPM Server(s).

   For information about how to integrate the external Web server with a PPM Server, see *Integrating an External Web Server with a PPM Server* on page 113.

2. On the PPM Server, do the following:

   a. Stop the PPM Server.

   b. Open the `server.conf` file in a text editor, and then add to it the following:

   ```
   com.kintana.core.server.SINGLE_SIGN_ON_PLUGIN
   =com.kintana.sc.security.auth.WebRemoteUserSingleSignOn
   ```

   ➤ For information on how to edit the `server.conf` file, see *Server Configuration Parameters* on page 227.

   c. Save and then close the `server.conf` file.

   d. Execute the `kUpdateHtml.sh` script.

   ➤ For information about the `kUpdateHtml.sh` script, see *kUpdateHtml.sh* on page 286.

3. On the IIS external Web server, do the following:

   a. From IIS Microsoft Management Console, select the default Web site.

   b. Right-click the default Web site, and then click **Properties** on the shortcut menu.

   c. Click the **Directory Securities** tab.

   d. Under **Anonymous access,** click **Edit.**

   e. Deselect the **Anonymous Access** checkbox.

   f. Leave the **Integrated Windows authentication** checkbox selected.

   g. Click **OK.**

   h. Click **OK.**

i. Stop, and then restart the IIS Windows service.

➤ Do not use Basic Authentication. If you do, the Web server does not force the browser to authenticate, and so does not result in a single sign-on solution.

4. Stop, and then restart the PPM Server.

For information on troubleshooting issues you may encounter with Web remote single sign-on, see *Troubleshooting Your Single Sign-On Implementation* on page 132.

## Implementing Generic Single Sign-On with PPM Center

This section provides information on how to configure PPM Center to use the generic single sign-on module to integrate with third-party authentication servers.

Single sign-on works as follows:

1. A user logs on to a portal that has been configured to use a third-party authentication application.
2. The user accesses the PPM Center standard interface through an external Web Server integration that is part of the logged-in domain.
3. The PPM Server receives the user information through the HTTP header specified in the `sso.conf` file.
4. If the user is a valid PPM Center user, he is granted access to the PPM Center standard interface and PPM Dashboard.

### Requirements for Implementing Generic Single Sign-On

To implement generic single sign-on with PPM Center, your PPM Center system be integrated with an external Web server (Sun Java System Web Server, an Apache-based server, or IIS).

### Setting Up Generic Single Sign-On with PPM Center

To implement generic single sign-on:

1. Regarding the third-party authentication application you plan to use:
   a. To configure the third-party application, follow the instructions provided with the application.
   b. Verify that the PPM Center user is also a valid single sign-on user and can be authenticated.

2. External Web server:

   a. Integrate PPM Center with the external Web server.

      For information on how to integrate an external Web server, see *Integrating an External Web Server with a PPM Server* on page 113.

   b. Configure the external Web server to integrate with the third-party authentication application. For information on how to do this, see the documentation provided with the with third-party authentication application.

   c. Ensure that the authenticated user's HTTP request is forwarded to the PPM Server with the user ID inserted into the HTTP header specified in the `sso.conf` file.

      ➤ You can find the `sso.conf` file in the `<PPM_Home>/integration/sso` directory.

3. PPM Server configuration

   a. Verify that the `sso.conf` file has the following setting:

      ```
      USERNAME=<AUTHENTICATED_USER_HEADER>
      ```

      where `<AUTHENTICATED_USER_HEADER>` is the header your single sign-on system uses to store the user ID of the authenticated user. For example, RSA ClearTrust uses `ct-remote-user`.

   b. Add the following line to the `server.conf` file:

      ```
      com.kintana.core.server.SINGLE_SIGN_ON_
      PLUGIN=com.kintana.sc.security.auth.GenericSingleSignOn
      ```

   c. Run the `kUpdateHtml.sh` script, which is located in the `<PPM_Home>/bin` directory.

4. Stop, and then restart the PPM Server.

For information on troubleshooting issues you may encounter with single sign-on, see *Troubleshooting Your Single Sign-On Implementation* on page 132.

## Troubleshooting Your Single Sign-On Implementation

Determine the header information that the single sign-on server is sending.

1. Check the timestamp as follows:

   a. Open the `server.conf` file in a text editor, and set the value of the `ENABLE_WEB_ACCESS_LOGGING` parameter to `true`.

   ➤ For information on how to edit the `server.conf` file, see *Server Configuration Parameters* on page 227.

   b. Run the `kUpdateHtml.sh` script.

   c. Restart the PPM Server.

   ➤ For details on how to stop and start the PPM Server, see *Starting and Stopping the PPM Server* on page 66.

   d. Log on to PPM Center.

   e. Check the timestamp on the PPM Server.

   f. Go to the `<PPM_Home>/server/<instance>/log` directory.

   g. Open the `<date>.access.log` file and check the timestamp.

2. Open the `logging.conf` file (located in the `<PPM_Home>/conf` directory) in a text editor, and add the following text:

```
com.kintana.core.logging.PRODUCT_FUNCTION_LOGGING_LEVEL =
com.kintana.web.filter.debug, DEBUG
com.kintana.core.logging.PRODUCT_FUNCTION_LOGGING_LEVEL =
com.kintana.sc.authentication, DEBUG
com.kintana.core.logging.SYSTEM_THRESHOLD = DEBUG
```

3. Restart the PPM Server using the `kStart.sh -debug` option.

   The information is written to the `<PPM_Home>/bin/serverLog_<debug-timestamp>.txt` file.

4. Enable logging on the single sign-on agent side, and then check the information passed back and forth. Check for any error messages reported.

After you check for problems and error messages, remove the debugging code you added to the `logging.conf` file in step 2.

# 6 Maintaining the System

In This Chapter:

# Overview of Administration Tools and System Maintenance

Two kinds of administration tools and facilities are available to PPM Center system administrators:

- Administration tools accessible from the standard interface

  These tools let you:

  - View and cancel running reports
  - View running executions
  - View interrupted executions

- Administration tools accessible from the PPM Workbench

  These tools include:

  - Admin Tools let you submit and view server reports
  - SQL Runner lets you submit SQL statements against the PPM Center database

The following sections provide information about these tools and facilities.

This chapter also provides information about how to:

- Access and use log files
- Periodically stop and restart the server
- Maintain the database
- Back up PPM Center instances

# Administration Tools in the Standard Interface

The PPM Center standard interface includes tools that you can use to:

- View running reports

➤ For information about viewing running reports, see the *Reports Guide and Reference.*

- View running executions
- View interrupted executions

You can access these tools in the standard interface through the **Administration** menu.

## Viewing Running Executions

To view running executions:

1. Log on to PPM Center.
2. From the menu bar, select **Administration > View Running Executions.**

   The View Running Executions page opens, and the **Summary** section lists any distributions, reports, requests, or packages that are running.
3. If any reports are listed as running, click **View Running Reports.**

## Viewing Interrupted Executions

This section provides the steps you use to view interrupted executions (including reports).

To view interrupted executions:

1. Log on to PPM Center.
2. From the menu bar, select **Administration > View Interrupted Executions.**

   The View Interrupted Executions page opens, and, if any interrupted executions exist, the page lists them.
3. In the list below **View Interrupted Executions for a Server Startup,** select the date of the interrupted execution you want to see.
4. Click **View** to see the details of the selected interrupted execution listed in the **Failed Executions** section.

# Server Tools In the PPM Workbench

You can use the server tools in the PPM Workbench to:

- View the technical status of the PPM Server in the Admin Tools window
- Access the database directly and run SQL statements from the SQL Runner window
- Edit server settings

## Access Grants Required to Use Server Tools

*Table 6-1* lists the names and descriptions of the three access grants that give users various levels of access to the Server Tools window.

Table 6-1. Server tools access grants

| Access Grant | Description |
|---|---|
| Sys Admin: View Server Tools | Lets the user view the Admin Tools and SQL Runner windows in read-only mode. |
| Sys Admin: Server Tools: Execute Admin Tools | Lets the user:<br>■ Run server reports in the Admin Tools window<br>■ View the SQL Runner window in read-only mode |
| Sys Admin: Server Tools: Execute SQL Runner | Lets the user:<br>■ Run SQL queries in the SQL Runner window<br>■ View the Admin Tools window in read-only mode |

For more information about security groups and access grants, see the *Security Model Guide and Reference.*

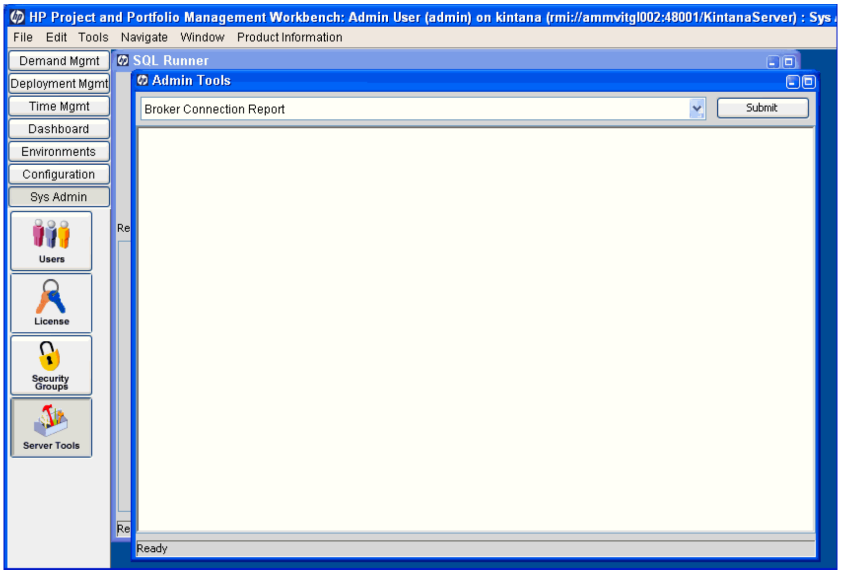
## Accessing and Using the PPM Workbench Server Tools

To access the server tools in the PPM Workbench:

1. Log on to PPM Center.
2. From the menu bar, select **Administration > Open Workbench.**

   The PPM Workbench opens.
3. From the shortcut bar, select **Sys Admin > Server Tools.**

   The Admin Tools and the SQL Runner windows open.

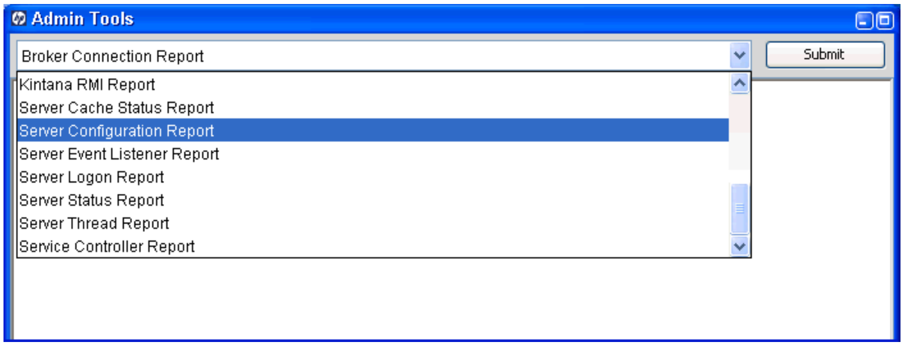


## Running Server Reports from the Admin Tools Window

Use the Admin Tools window to run server reports such as Server Status Report and Cache Manager Statistics. *Table 6-2* on page 139 contains descriptions of the server reports.

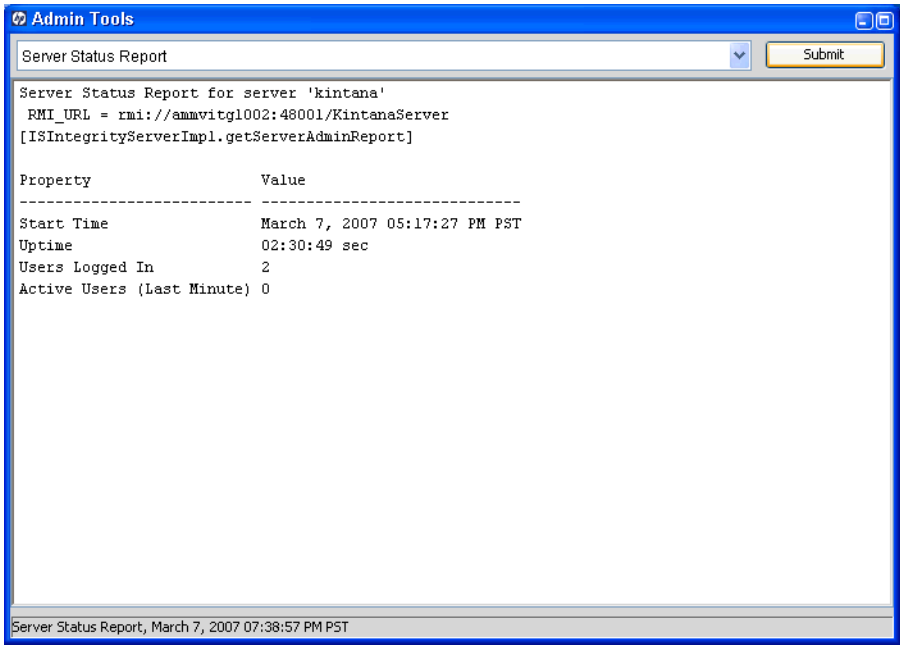
To select a report to run:

1. In the expanded report list, select a report.



2. Click **Submit.**

   The Admin Tools window displays the output of the selected report.



```
Server Status Report for server 'kintana'
 RMI_URL = rmi://ammvitgl002:48001/KintanaServer
[ISIntegrityServerImpl.getServerAdminReport]

Property                   Value
-------------------------- -----------------------------
Start Time                 March 7, 2007 05:17:27 PM PST
Uptime                     02:30:49 sec
Users Logged In            2
Active Users (Last Minute) 0
```

Server Status Report, March 7, 2007 07:38:57 PM PST

Table 6-2. Server reports (page 1 of 3)

| Report Name | Description |
|---|---|
| Broker Connection | Information about open database pool connections, organized by connection ID. |
| Broker In Use Sessions | Information about database pool connections in use, organized by user. If the server parameter `DB_SESSION_TRACKING` is set to TRUE, this report also shows stack traces of where the connection was allocated. |
| Broker Performance | Statistics on database connection usage in the connection pool, to help assess system performance.<br>For performance reasons, the PPM Server holds a connection pool to the database and reuses these connections for accessing the database. Prepared statements created within a connection are also held open in a cache.<br>If the PPM Server cannot allocate more connections, threads that need to access the database might need to wait for a connection.<br>This report also shows:<br>■ Number of threads waiting for connections<br>■ Average duration threads had to wait for connections<br>■ Percentage of threads that had to wait for connections<br>■ Total number of connection requests, and if JDBC logging is enabled<br>■ Statement cache hit rate percentage (over the last 100 statements) |
| CacheManager Sizes | Displays the number of objects in the cache of each entity, the total cache size (in KB), and the average size of each cached object type. |
| CacheManager Statistics | Displays useful statistics on the caching behavior of each cachable entity in PPM Center, including:<br>■ Hits, misses, and hit rate<br>■ Number of cache flushes (broken down by the categories "old," "idle," "reclaimed," and "max cache size reached")<br>■ Average load time<br>■ Cached object count and maximum idle time |
| Client Font | All supported fonts for the PPM Center installation. |
| Client Property | Details about the environment of the client computer currently running the PPM Workbench. |
| Client Time Zone | All time zones recognized by the client. |

Table 6-2. Server reports (page 2 of 3)

| Report Name | Description |
|---|---|
| Execution Dispatcher Manager | Batch executions in progress. |
| Execution Dispatcher Pending Batch | Batches pending execution due to the lack of available execution manager threads. |
| Execution Dispatcher Pending Group | Batches pending group execution (batches that are grouped together) due to the lack of available Execution Manager threads. |
| Installed Extensions | |
| JVM Memory | Free and total memory in the PPM Server JVM. |
| Kintana RMI | All RMI connection threads. |
| Server Cache Status | Shows the following cache information:<br>▪ Cached entities<br>▪ Number of units that can be cached<br>▪ Number of free units<br>▪ The number of hits and misses, and the miss rate<br>▪ Number of entities swapped<br>▪ Amount of memory taken up by the cache<br>**Note:** Although this report displays information that is similar to the that displayed in the CacheManagerStatistics report, the data is for a different set of cached objects. |
| Server Configuration | All server parameters in effect for each of the active servers. Includes parameters not specifically set in the `server.conf` file. |
| Server Event Listener | Events that the PPM Server can send to the client. |
| Server Logon | Information about all users logged on to the PPM Server(s) and logon information such as IP address and idle time.<br>This information is used to determine PPM Server load. If server clustering is used, this report provides a picture of load distribution. |

Table 6-2. Server reports (page 3 of 3)

| Report Name | Description |
|---|---|
| Server Status | Status information about PPM Server(s):<br>■ Whether the server is available and its start time<br>■ Length of time the server has been available<br>■ Number of users logged on to the server<br>■ Number of users active during the last minute |
| Server Thread | Information about running threads within a PPM Server(s).<br>This information is used to determine which services are running. If a server cluster is used, this report also provides information about which server is running these services. |
| Service Controller | Enabled services for the PPM Server(s), when services were last run, and when they are scheduled to run again. |

### Running Server Reports from the Command Line

You can also run server reports directly from a command line on the PPM Server using the `kRunServerAdminReport.sh` script, which is located in the `<`*`PPM_Home`*`>/bin` directory. For more information about the `kRunServerAdminReport.sh` script, see *kRunServerAdminReport.sh* on page 283.

## Running SQL Statements in the SQL Runner Window

You can use the SQL Runner window to run database queries directly against the PPM Center database schema using the PPM Workbench instead of using an external program such as SQL*Plus. One benefit of using SQL Runner is that you can gain access to the database directly, without having to submit the database password. Developers and administrators can also use the SQL Runner window to test custom validations and request rule SQL, among other things.

To run an SQL statement from the SQL Runner window:

1. If the Admin Tools window hides the SQL Runner window, minimize it.

2. In the **SQL Statement** field, type the SQL statement to run.

   Ensure that your SQL statement does not end with a semicolon (;).

3. To run the SQL statement, click **Run SQL.**

The SQL Runner window displays the list of results in the table below the SQL statement. It also displays timing information such as how long the statement took to run, and how much of that time was spent in the database.

4. To view the results as text, click **Open As Text.**

*Table 6-3* lists the controls in the SQL Runner window, along with a description of each.

Table 6-3. Controls in the SQL Runner window

| Control Name | Control Type | Description |
|---|---|---|
| SQL Statement | Text box | Use this box to type an SQL query for running and testing purposes.<br>**Note:** Make sure that you do not include a semicolon (;) at the end of your SQL statement. |
| Server Roundtrip | Read-only text box | Amount of time (in milliseconds) spent sending the SQL statement out to the network and back.<br>Used to show network latency and performance. |
| SQL execution | Read-only text box | Amount of time (in milliseconds) the database spent actually executing the SQL statement.<br>Use the displayed information to tune validations or write complex statements to address performance concerns. |
| ResultSet Extraction | Read-only text box | Amount of time (in milliseconds) that the server spent processing the SQL statement results. |
| Total time | Read-only text box | Total amount of time (in milliseconds) spent running the SQL statement. |
| Run SQL | Button | Runs the SQL statement displayed in the **SQL Statement** field. |
| Clear | Button | Clears the window. |
| Ping Server | Button | Tests the connection speed between the client and the PPM Server. |
| Ping DB | Button | Tests the connection speed between the client and the database (through the PPM Server). |
| Open As Text | Button | Opens results in a text window.<br>You can cut and paste information from this window. |

## Running an SQL Script with SQL*Plus on a Windows System

If your PPM Center instance is running on a Windows system, and you are using the SQL*Plus utility to run an SQL script, the utility “expects” to get the exact number of parameters defined in the script. Some versions of SQL*Plus ignore null command-line parameters and get hung up waiting for missing parameter values.

For example, in the following line, the second parameter is null. But, because SQL*Plus is a command-line utility, it hangs and waits for the user to input the second parameter value.

```
ppm/ppm@ppm10a @somescript.sql "Y" ""
```

To work around this problem, add the following to the `server.conf` file:

```
SQLPLUS_VERSION=<SQL_PLUS_VERSION_NUMBER>
```

A valid version number is `90101`.

## Setting Debugging and Tracing Parameters

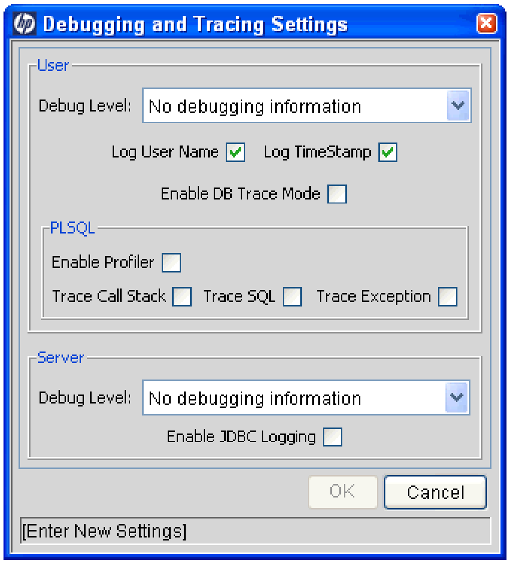
You use the Debugging and Tracing Settings dialog box to set debugging and tracing parameters at both the user and server levels.

To open the Debugging and Tracing Settings dialog box:

1. Log on to PPM Center.
2. From the menu bar, select **Administration > Open Workbench.**

   The PPM Workbench opens.

3. From the shortcut bar, select **Edit > Debug Settings.**



Debugging and Tracing Settings

User

Debug Level: No debugging information

Log User Name ☑ Log TimeStamp ☑

Enable DB Trace Mode □

PLSQL

Enable Profiler □

Trace Call Stack □ Trace SQL □ Trace Exception □

Server

Debug Level: No debugging information

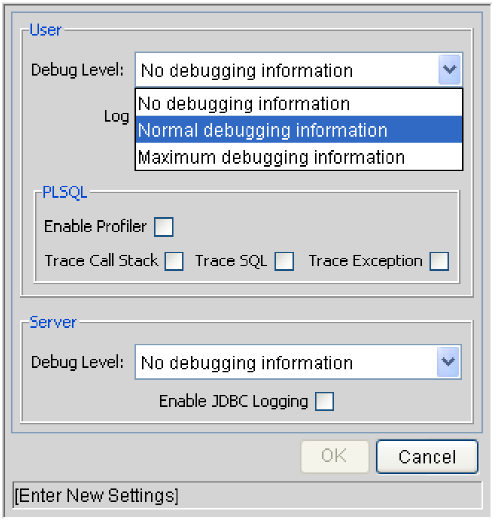
Enable JDBC Logging □

[Enter New Settings]

## User Settings

To override the default debug level set for your PPM Center sessions:

- From the **Debug Level** list in the **User** section, select a different value.



User

Debug Level: No debugging information

Log

No debugging information
Normal debugging information
Maximum debugging information

PLSQL

Enable Profiler □

Trace Call Stack □ Trace SQL □ Trace Exception □

Server

Debug Level: No debugging information

Enable JDBC Logging □

[Enter New Settings]

The **Debug Level** list values map to `DEFAULT_USER_LOGGING_LEVEL` values in the `server.conf` file as follows:

- **No debugging information** is equivalent to the parameter value `ERROR`. Only errors are logged.
- **Normal debugging information** is equivalent to the parameter value `INFO`. Errors and information that describes the normal tasks that the running server is performing are logged.
- **Maximum debugging information** is equivalent to the parameter value `DEBUG`. This setting provides the most logging information. In addition to the normal debugging information, information is also logged for various server functions.

  This additional debugging information can be useful for troubleshooting any problems you encounter in PPM Center. If a problem arises, you can set the debug level to **Maximum debugging information,** perform the problematic action again, and then check the server logs for information that can help resolve the issue.

  Make sure that you do not to leave the server running in debug mode for too long. A large volume of extra information is written to the logs, taking up disk space much more quickly than during normal operation. The extra logging overhead can affect system performance.

### Log User Name Setting

If you want your user name written into the log for each line of debugging text that corresponds to actions you have performed, select this checkbox. This can be helpful if you need to sift through the server logs to find information relevant to your user session. (The **Log User Name** checkbox corresponds to the `ENABLE_SQL_TRACE` configuration parameter.)

### Log TimeStamp Setting

If you want a timestamp written into the log for each line of debugging text that corresponds to actions you have performed, select this checkbox. The timestamp can help you locate information in the server log files about events that occurred at a specific time, or to determine how much time elapsed between specific logged statements.

Bear in mind that including the timestamp adds text to each logged statement. This bloats the log file and can make it more difficult to read. (The **Log TimeStamp** checkbox corresponds to the `ENABLE_TIMESTAMP_LOGGING` parameter in the `server.conf` file.)

### Enable DB Trace Mode Setting

To enable the SQL trace facility during your PPM Center session, select the **Enable DB Trace Mode** checkbox. This facility ensures that performance statistics for all SQL statements that you run are placed into a trace file. (The **Enable DB Trace Mode** checkbox corresponds to the `ENABLE_SQL_TRACE` server configuration parameter.)

### PL/SQL Settings

The **PLSQL** field contains the following Procedural Language/Structured Query Language (PL/SQL) options:

- Select the **Enable Profiler** checkbox to profile the run-time behavior of the PL/SQL code that PPM Center applications use by calling the Oracle-supplied PL/SQL package `DBMS_PROFILER`.

  ➤ You must set up the PL/SQL package. For an example of how to do this, see *Setting Up the Oracle Profiler* on page 147.

  The profiling information is logged in a JDBC log file in the PPM Center `log` directory. Enabling the profiler can help you to identify performance bottlenecks.

  ➤ Because running the `DBMS_PROFILER` package might slow system performance and reduce storage space, HP recommends that you use it only for debugging.

### Setting Up the Oracle Profiler

The following example illustrates how to set up the Oracle profiler:

```
CONNECT sys/password@service AS SYSDBA
@$ORACLE_HOME/rdbms/admin/profload.sql

CREATE USER profiler IDENTIFIED BY profiler DEFAULT TABLESPACE
users QUOTA UNLIMITED ON users;
GRANT connect TO profiler;

CREATE PUBLIC SYNONYM plsql_profiler_runs FOR profiler.plsql_
profiler_runs;
CREATE PUBLIC SYNONYM plsql_profiler_units FOR profiler.plsql_
profiler_units;
CREATE PUBLIC SYNONYM plsql_profiler_data FOR profiler.plsql_
profiler_data;
CREATE PUBLIC SYNONYM plsql_profiler_runnumber FOR
profiler.plsql_profiler_runnumber;

CONNECT profiler/profiler@service
@$ORACLE_HOME/rdbms/admin/proftab.sql
GRANT SELECT ON plsql_profiler_runnumber TO PUBLIC;
GRANT SELECT, INSERT, UPDATE, DELETE ON plsql_profiler_data TO
PUBLIC;
GRANT SELECT, INSERT, UPDATE, DELETE ON plsql_profiler_units TO
PUBLIC;
GRANT SELECT, INSERT, UPDATE, DELETE ON plsql_profiler_runs TO
PUBLIC;
```

### Trace Call Stack, Trace SQL, and Trace Exception

Select the **Trace Call Stack, Trace SQL,** and **Trace Exception** checkboxes to enable the Oracle DBMS_TRACE package functionality that the PL/SQL programs (used by PPM Center applications) use.

The output of the profiling information is saved to a JDBC log file in the PPM Center log directory.

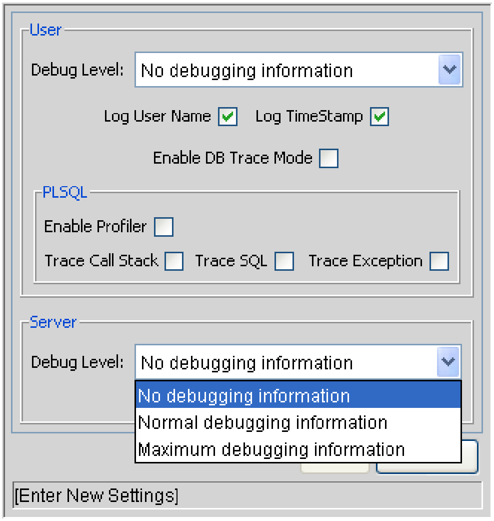
➤ Because running the DBMS_TRACE package can have a negative effect on system performance and storage space, use it only for debugging.

## Server Settings

To override the default logging level for the entire PPM Server, and not just your user session:

1. Under **Server,** in the **Debug Level** list, select one of the following:

   The following settings correspond to the settings for the DEFAULT_SERVER_LOGGING_LEVEL server configuration parameter. The value names, however, are different.



User

Debug Level: No debugging information

Log User Name ☑ Log TimeStamp ☑

Enable DB Trace Mode □

PLSQL

Enable Profiler □

Trace Call Stack □ Trace SQL □ Trace Exception □

Server

Debug Level: No debugging information

No debugging information
Normal debugging information
Maximum debugging information

[Enter New Settings]

- **No debugging information** is equivalent to the DEFAULT_SERVER_LOGGING_LEVEL parameter value ERROR. Only errors are logged.
- **Normal debugging information** is equivalent to the parameter value INFO. Errors and information that describes the normal tasks that the running server is performing are logged.
- **Maximum debugging information** is equivalent to the parameter value DEBUG. This setting provides the most logging information. In addition to the normal debugging information, information is also logged for various server functions.

  This additional debugging information can be useful when troubleshooting any problems you encounter in PPM Center. If a problem arises, you can set the debug level to **Maximum debugging information,** perform the problematic action again, and check the server logs for information that can help resolve the issue.

For more information about the DEFAULT_SERVER_LOGGING_LEVEL parameter, see *Server Configuration Parameters* on page 225.

2. To have the PPM Server(s) maintain a Java Database Connectivity (JDBC) log file, select the **Enable JDBC Logging** checkbox.

# Getting Information from Log Files

The PPM Server generates log files in the file system. Depending on the type of log file, certain maintenance practices should be employed to maintain the file system. The following sections provide maintenance recommendations for each type of log file.

## Server Log Files

Server log files are stored in the <*PPM_Home*>/server/<*server name*>/logs directory. Server log files are named serverLog.txt and serverLog_timestamp.txt. The timestamp variable uses the format YYYYMMDD_HHMMSS for the date and time the log was rotated.

Active PPM Servers log their output to the serverLog.txt file. The serverLog_timestamp files are archived versions of the serverLog.txt file. The size of these old log files are determined by the ROTATE_LOG_SIZE server parameter in the server.conf file. This parameter may be set to any value (in kilobytes) to control the rotation. A high value results in fewer but larger log files.

Generally, server log files are required only when contacting HP-Mercury support to resolve server issues. In most cases, it is safe to delete these log files on a regular basis.

The following parameters determine the data volume to be written to the logs by the server:

- DEFAULT_SERVER_LOGGING_LEVEL
- DEFAULT_USER_DEBUG_LEVEL
- RMI_DEBUGGING

In the server.conf file, set these parameters to their default values:

```
com.kintana.core.server.SERVER_DEBUG_LEVEL=NONE
com.kintana.core.server.DEFAULT_USER_DEBUG_LEVEL=NONE
com.kintana.core.server.RMI_DEBUGGING=FALSE
com.kintana.core.server.ENABLE_LOGGING=TRUE
```

By setting these parameters to their default settings, only critical error events are written to the server logs. This decreases the number of server logs generated in the file system, thereby improving system performance.

If the server experiences technical difficulties or server logs are required by HP-Mercury support, increase the debug level.

Unless instructed otherwise by HP-Mercury support, always set the RMI_DEBUGGING parameter to FALSE.

To change the USER_DEBUG_LEVEL parameter dynamically at runtime, change the DEFAULT_USER_DEBUG_LEVEL parameter in the **Edit > Debug Settings** screen group in the PPM Workbench interface. You can also retrieve current server settings by accessing the Server Tools window and running the Server Configuration report.

➤ Unless instructed by HP-Mercury support, do not run a production server with the debug levels set to Maximum. This can generate very large log files in the file system that could degrade system performance.

### Enabling HTTP Logging

To enable HTTP logging:

➤ Do not enable HTTP logging if you use an external Web server.

1. Stop the PPM Server.
2. Set the ENABLE_WEB_ACCESS_LOG server.conf parameter to TRUE.
3. Run the kUpdateHtml.sh script.
4. Start the server.

The internal Web log is saved in NCSA Common format:

```
host rfc931 username date:time request statuscode bytes
referrer user_agent cookie
```

Example:

```
127.0.0.1 - - [11/Dec/2005:1908:16 +0000] "GET/ppm/web/knta/
global/images/date_time.gif HTTP/1.1"200 155 "http://
localhost:8080/ppm/web/knta/crt/RequestCreateList.jsp"
"Mozilla/4.0 (compatible; MSIE 6.0; Windows; .NET CLR 1.0.3705;
.NET CLR 1.1.4322)" JSESSIONID=5pk1oof3fd65q
```

## Report Log Files

Report execution log files are stored in the <*PPM_Home*>/logs/reports directory. Report execution log files are named rep_log_ID.html. The ID variable corresponds to the report submission ID.

Use report execution log files to determine the cause when report executions failed or consumed considerable time to complete.

These log files are not purged automatically. Generally, report log files are required only to debug timely report requests. In most cases, it is safe to delete these log files on a regular basis.

## Execution Log Files

During normal package and request processing, execution log files are generated:

- For workflow steps running as EXECUTE_OBJECT_COMMANDS or EXECUTE_REQUEST_COMMANDS
- When resolving a validation defined using command execution logic

Execution log files from these executions are stored in the following directories:

- <*PPM_Home*>/logs/PKG_Package_ID
- <*PPM_Home*>/logs/REQ_Request_ID
- <*PPM_Home*>/logs/VAL_Validation_ID

If disk space becomes limited over time, you might need to purge or archive these log files. If the log files are deleted, the detailed execution logs are no longer available for a package or request.

## Execution Debug Log Files

If the USER_DEBUG_LEVEL or SERVER_DEBUG_LEVEL parameter is set to HIGH, additional execution debugging data is written to the execution debug log file. This file is named exe_debug_log.txt and is located in the <*PPM_Home*>/logs/ directory.

If the server is running with full debugging enabled, this file grows over time. Generally, execution debug log files are required only by HP-Mercury support to debug the execution engine. In most cases, it is safe to delete these log files on a regular basis.

### Temporary Log Files

Various other files generated in the <*PPM_Home*>/logs/temp directory are stored for temporary purposes. Unless requested otherwise by HP-Mercury support, you can delete these log files on a regular basis.

## Periodically Stopping and Restarting the Server

The PPM Server generally requires very little maintenance. To help make sure your system operates smoothly, HP recommends that you stop and restart the server once a month.

For information about starting and stopping the server, see *Starting and Stopping the PPM Server* on page 66.

## Maintaining the Database

Many IT departments have a policy of periodically changing the passwords of their database schemas. This section covers common topics related to maintaining the Oracle database that is part of PPM Center.

### Changing the Database Schema Passwords

If you must change the PPM Center database schema passwords, be sure to change them both in the database and in the `server.conf` file. Before you change all the database schema passwords, consider the following:

- Check your environment definitions to see if any contain a password that is to be changed. You can use the tool <*PPM_Home*>/bin/kEnvUpdatePassword.sh to automatically change all occurrences of a specific password for a particular host and user name.

  ➤ This functionality is also available from the **Environments** section of the PPM Workbench. (Open an environment on the Environment page, and then, on the menu bar, select **Environment > Update Password.**)

  - Check both server and client passwords, as well as database passwords.
  - Check passwords associated with application codes.
- Although it is not a recommended practice, you can hard-code passwords into commands in workflow steps, requests, and object types.

- There is no need to change commands that use tokens for passwords (that is, SOURCE_ENV.DB_PASSWORD), as long as the password was changed in the respective environment definitions.

To change the PPM Center database schema passwords:

1. Make sure that all users are logged off the system.
2. Stop the PPM Server.

   For information about how to stop PPM Servers, see *Stopping the PPM Server* on page 214.

3. Change the passwords you want to change in the database.
4. To change the passwords in the server.conf file, run the kConfig.sh script to set the DB_PASSWORD, CONC_REQUEST_PASSWORD, and RML_PASSWORD server parameters.

   ➤ When changing the passwords, do not edit the server.conf file directly. To encrypt password values correctly, use the kConfig.sh script.

5. Restart the PPM Server.

   For information about restarting PPM Servers, see *Starting and Stopping the PPM Server* on page 66.

## Maintaining Temporary Tables

The PPM Server uses several tables for temporary storage during processing (for example, during package migration) for:

- Logon attempts
- Debug messages
- Commands and parameters

PPM Server uses a set of services to monitor and clean up these temporary tables. Make sure the cleanup parameters (described in *Cleanup Parameters* on page 168 and in Appendix A, *Server Configuration Parameters,* on page 225) are set so that the temporary tables do not use too much database space.

### KNTA_LOGON_ATTEMPTS Table

The KNTA_LOGON_ATTEMPTS table contains information about attempts to log on to the PPM Server over the previous 14 days. This information includes:

- USER_ID of users who attempted to log on
- Status (success or failure) of each logon attempt
- Messages generated during the logon attempt

The KNTA_LOGON_ATTEMPTS table is only for auditing purposes. The PPM Server does not require the data to function.

If logon attempts succeed, the records for those most of those attempts are purged. However, the last successful logon based on a combination of USER_ID and IP address is retained.

If a logon attempt fails, the corresponding record remains in the table for future reference. You must delete the failed logon attempt records manually. the record of the last successful logon attempt also remains in the KNTA_LOGON_ATTEMPTS table.

The data is automatically purged after the time interval specified by the DAYS_TO_KEEP_LOGON_ATTEMPT_ROWS server parameter setting.

### KNTA_DEBUG_MESSAGES Table

The KNTA_DEBUG_MESSAGES table contains any debugging text that HP PL/SQL database packages generate. After you analyze this data, you can safely purge it. The PPM Server purges this data automatically at the frequency determined by the HOURS_TO_KEEP_MESSAGE_ROWS server configuration parameter setting.

## Backing Up PPM Center Instances

Backing up a PPM Center instance involves backing up both the file system and the database schema. HP stores all PPM Center configuration and transaction data in its associated database schema.

Because this information is so important, HP also recommends that you back up the database schema daily. You can use the Oracle export command to perform the backup, or use the hot backup procedure, which does not require that you shut down the PPM Server. For information about how to export a database schema, see your Oracle database documentation.

HP recommends that you back up the `<PPM_Home>/logs` directory daily. This directory contains transactional history files for each migrated package or request.

➤ Before you make critical changes to PPM Center, perform a full backup of the database schema and complete `<PPM_Home>` directory.

It is not necessary to back up registry settings.

# 7 Improving System Performance

In This Chapter:

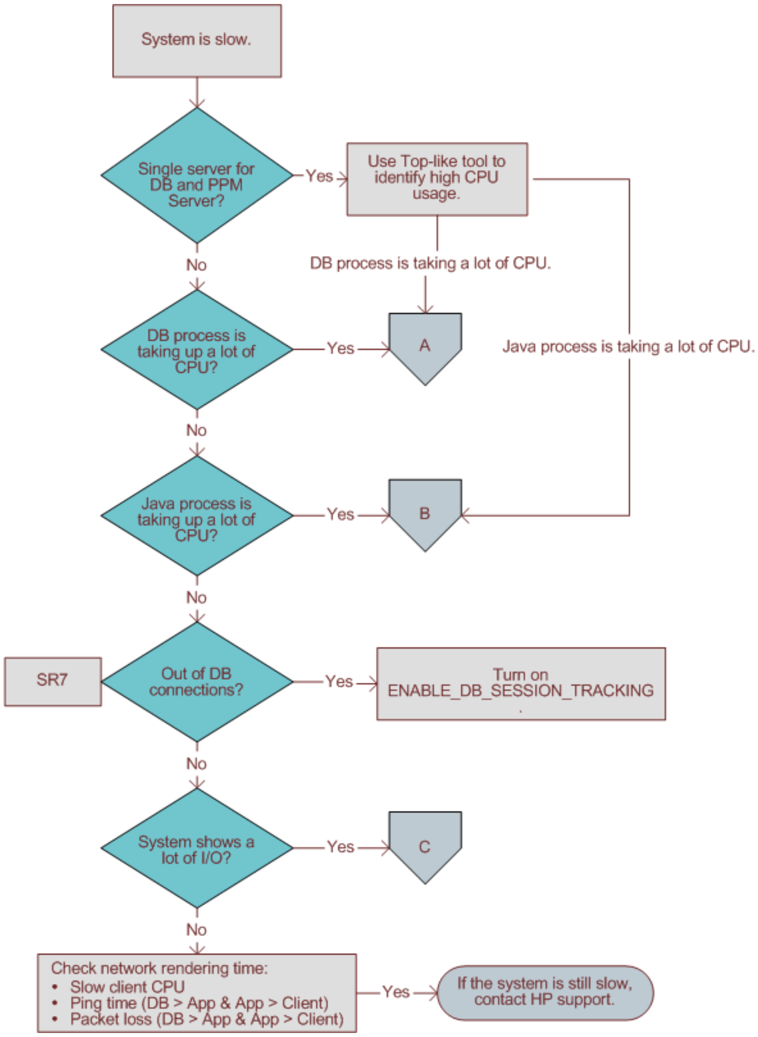
# Identifying Performance Problems

This chapter provides information about how to isolate performance problems, collect statistics about the database schema, and troubleshoot performance problems.

## Isolating Performance Problems

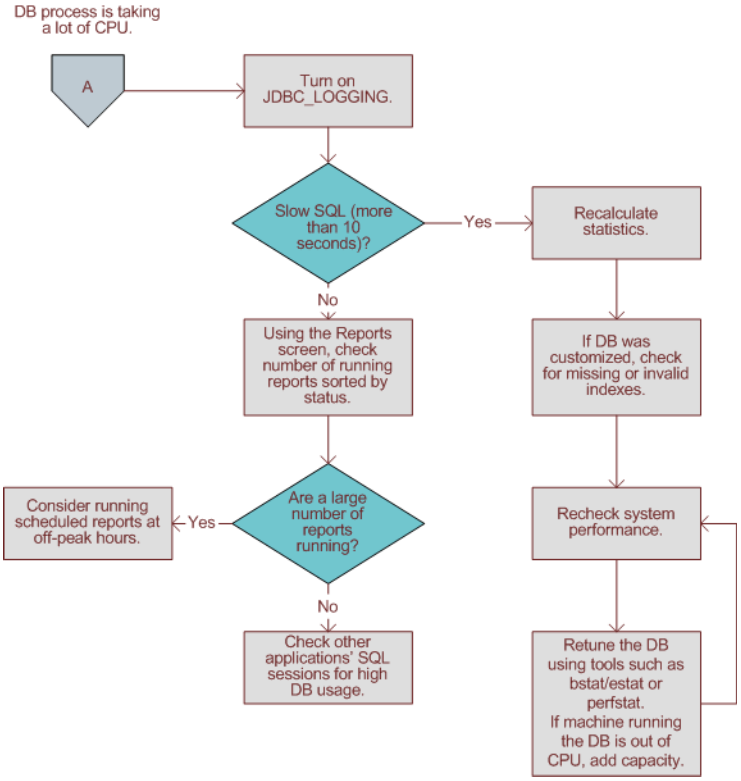
*Configuring or Reconfiguring the Database* on page 81 and Appendix A, *Server Configuration Parameters,* on page 225 contain information on the initial settings that HP recommends for the Oracle database and PPM Server. If PPM Center performance slows after these settings are in place, use the methodologies outlined in the flowcharts shown in *Figure 7-1* on page 159, *Figure 7-2* on page 160, and *Figure 7-3* on page 160 to isolate performance problems and determine how to fix them.

Figure 7-1. Identifying and addressing system performance problems



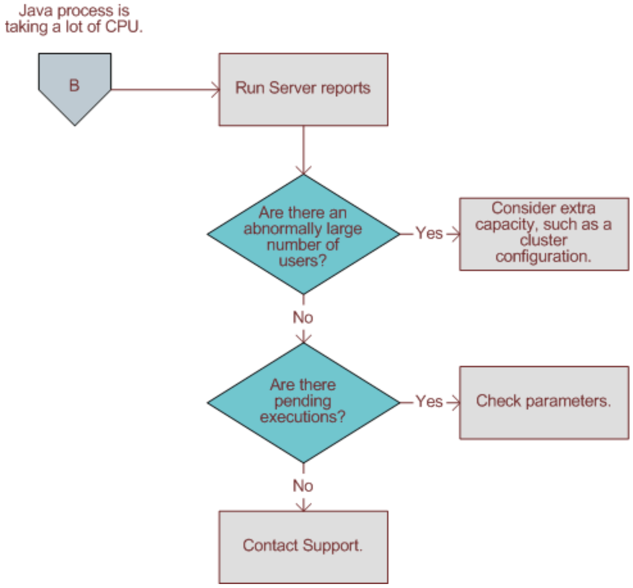
System is slow.
Single server for DB and PPM Server?
Yes
Use Top-like tool to identify high CPU usage.
DB process is taking a lot of CPU.
Java process is taking a lot of CPU.
No
DB process is taking up a lot of CPU?
Yes
A
No
Java process is taking up a lot of CPU?
Yes
B
No
SR7
Out of DB connections?
Yes
Turn on ENABLE_DB_SESSION_TRACKING
No
System shows a lot of I/O?
Yes
C
No
Check network rendering time:
• Slow client CPU
• Ping time (DB > App & App > Client)
• Packet loss (DB > App & App > Client)
Yes
If the system is still slow, contact HP support.

Figure 7-2. Identifying and addressing database performance problems (A)



DB process is taking a lot of CPU.
A
Turn on JDBC_LOGGING.
Slow SQL (more than 10 seconds)?
Yes
Recalculate statistics.
No
Using the Reports screen, check number of running reports sorted by status.
If DB was customized, check for missing or invalid indexes.
Consider running scheduled reports at off-peak hours.
Yes
Are a large number of reports running?
Recheck system performance.
No
Check other applications' SQL sessions for high DB usage.
Retune the DB using tools such as bstat/estat or perfstat. If machine running the DB is out of CPU, add capacity.

Figure 7-3. Identifying and addressing Java process performance problems (B)



Java process is taking a lot of CPU.
B
Run Server reports
Are there an abnormally large number of users?
Yes
Consider extra capacity, such as a cluster configuration.
No
Are there pending executions?
Yes
Check parameters.
No
Contact Support.

# Collecting Database Schema Statistics

This section provides information about collecting statistics about the Oracle database schema.

## Collecting Statistics by Setting Server Parameters

Collect statistics if you are:

- Applying field-level security to a request type with existing requests in the system
- Applying dynamic security to a workflow with existing instances in the system
- Adding field group(s) for Distributed Management Objects (DMO) or PMO
- Using Microsoft Project to import large projects or many projects

You can set a PPM Center service to collect this kind of data periodically about the PPM Center database schema. You can use the following parameters to collect database statistics on PPM Servers:

- `ENABLE_STATISTICS_CALCULATION` determines whether database statistics are collected automatically for the cost-based optimizer.
- `STATS_CALC_WAKE_UP_TIME` determines the hour of the day at which database statistics are to be calculated.
- `STATS_CALC_DAY_OF_WEEK` determines the day of the week on which database statistics are to be calculated.
- `STATS_CALC_WEEK_INTERVAL` controls the frequency with which statistics are calculated.

For a list of and descriptions for PPM Server parameters, see *Server Configuration Parameters* on page 225.

## Using the dbms_stats Package to Collect Additional Statistics

If statistics gathered using the PPM Center service are insufficient, you can use the `dbms_stats` package to gather additional statistics. Oracle provides this package as part of the database.

To gather additional statistics, execute the following script:

```
begin
dbms_stats.gather_schema_stats (ownname => <PPM_User>,
cascade => TRUE,
method_opt => 'FOR ALL COLUMNS SIZE SKEWONLY'
);
end;
/
```

You typically run the `dbms_stats` package as the SYSTEM user. To run it as a PPM Center user, grant the privilege to run the package by executing the following SQL statement as the SYSTEM user from an SQL*Plus session:

```
grant execute on dbms_stats to <PPM_User>;
```

The first time you run the `dbms_stats` package, use `method_opt => 'FOR ALL COLUMNS SIZE SKEWONLY’.` After the system has been up and running for a while, use `method_opt => 'FOR ALL COLUMNS SIZE AUTO’.`

### Sampling a Percentage of Data

With large databases, analysis can take up to three hours to run to completion. For large PPM Center installations, you can sample a percentage of data in each object instead of data from the entire PPM Center database schema.

Sampling a percentage of data may not be effective for small data sets. However, after the data set has grown, this method is almost as effective as calculating statistics for the entire database schema.

To calculate statistics on a percentage of the data, run the following script:

```
begin
dbms_stats.gather_schema_stats (ownname => <PPM_User>,
cascade => TRUE,
method_opt => 'FOR ALL COLUMNS SIZE SKEWONLY',
estimate_percent => <percentage_to_sample>
);
end;
/
```

# Troubleshooting Performance Problems

This section provides information about common performance problems and how you can correct them. If you are not using the default or recommended settings, reset your parameters to those values before you try other solutions to performance problems.

Consider upgrading to the latest PPM Center service pack. HP has a regular and well-established service pack release cycle. Much of the development effort that goes into these service packs is focused on resolving known performance issues. Review the *Release Notes* for the latest service pack to see if it addresses any of the performance problems you are dealing with.

## Scheduled Reports Do Not Run on Schedule

### Problem

Although the PPM Server has capacity available, the next scheduled tasks do not start.

### Possible source

This may be caused by a limitation specified in the `MAX_WORKER_THREADS` server parameter.

### Solution

To run more scheduled reports simultaneously, set the `MAX_WORKER_THREADS` parameter to a higher value. For more information about this parameter, see *Server Configuration Parameters* on page 225.

## Packages Do Not Execute

### Problem

Packages do not execute.

### Possible source

There are not enough execution managers available to service the packages that the system processed.

### Solution

Increase the `MAX_EXECUTION_MANAGERS` server configuration parameter value.

For more information about this parameter, see *Server Configuration Parameters* on page 225.

### Nightly Reports on Sunday Do Not Finish On Time, System Slows on Monday

#### Problem

By default, database server statistics are collected at 1:00 a.m. on Sundays. For large installations, collection take so long that it is not completed on time and system performance is slower on Monday.

#### Solution

Reschedule the statistics collection to a time that works better for your organization. Determine the most active system time by running the Server Logon report, which checks the number of active users.

Consider using the estimate method instead of the compute method for gathering statistics.

Monitor CPU use. If the system slows because of high peak load, you might require more hardware or faster hardware.

For more information about gathering statistics, see *Collecting Database Schema Statistics* on page 161.

# Improving System Performance

This section provides information on how you can improve system performance.

## Tuning Java Virtual Machine (JVM) Performance

Because the PPM Server uses JSP, a Java compiler must be available in the environment path where the server is started.

### Running in Interpreted Mode

To improve performance, the Java virtual machine (JVM) uses a just-in-time (JIT) compiler. For debugging purposes, you can disable the JIT compiler and run the JVM in interpreted mode. Exceptions that you encounter while running in interpreted mode contain line numbers that are helpful in debugging.

To run the JVM in interpreted mode, set a variable in the server environment, as follows (use the Bourne or K shell):

```
JAVA_COMPILER=None
export JAVA_COMPILER
```

To avoid performance degradation, do not run the JVM in interpreted mode for extended periods in a production environment.

### Debugging

The PPM Server startup script (`kStart.sh`) contains several parameters that you can use for debugging. The `kStart.sh` JVM debugging parameters are `-ms550m` and `-mx550m`. These specify that the JVM starts up with a heap size of 550 MB, and is limited to a maximum heap size of 550 MB.

These settings are usually sufficient. For sites with heavy usage, however, consider increasing the `-ms550m` and `-mx550m` values. Required memory depends on factors such as cache sizes and number of Oracle connections.

➤ After you first start the PPM Server following an installation or upgrade, the server occupies approximately 750 MB in memory. As you use the product, the cache fills up and the JSPs are loaded into memory. Over time, the system gradually uses more memory. This is normal, and memory usage levels out over time. In most cases, memory usage can increase to a maximum of 1 GB.

## Tuning Server Cluster Performance

High transaction volumes and a large number of concurrent users on a PPM Server can degrade server response time. If the PPM Server is running on a multiprocessor system, spare CPU may be available, but JVM limitations can prevent the system from using the spare CPU.

In this case, consider using a PPM Server cluster. In this system configuration, multiple PPM Servers point to the same database instance and can be started on one or more systems. In addition to added capacity, running on multiple systems increases availability.

To use your multiple-CPU system effectively, this may be necessary on a two-CPU system, and it is required on systems with more than two CPUs.

For information about how to set up a server cluster, see *Configuring a Server Cluster* on page 115.

## Improving Input/Output Throughput

The distribution of input and output across multiple disks is an important factor in database performance. If consistently high input/output (I/O) occurs on one or more disks housing the database, service time on that disk degrades. To address this problem, replan the database layout to improve application performance.

You can split the PPM Center database into the following segments:

- PPM Center tables
- PPM Center indexes
- Redo logs
- Rollback tablespaces
- Temporary tablespaces
- System tablespace
- Tablespace for management and related utilities

HP recommends that PPM Center database instances with moderate transaction volume (instances with more than 5,000 requests per month) have at least four discrete disks, divided as shown in *Table 7-1*.

Table 7-1. Database disk recommendations

| Disk | Recommendations for Data Placement |
|---|---|
| 1 | PPM Center tables |
| 2 | PPM Center indexes |
| 3 | Redo logs |
| 4 | ■ Rollback tablespaces<br>■ Temporary tablespaces<br>■ System tablespace<br>■ Tablespace for management and related utilities |

For PPM Center database instances that have higher transaction volumes (more than 10,000 requests per month), HP recommends that you do the following:

- Place each piece of the database on its own separate disk.
- Stripe the data and index tablespaces across multiple disks to provide adequate disk throughput.

For PPM Center database instances with an extremely high transaction volume (over 25,000 requests per month), move specific tables and indexes to separate tablespaces on separate disks. This provides better control and further increases available I/O throughput.

## Improving Advanced Searches

PPM Center users can search for requests based on custom fields defined in request types, request header types, and user data. Users can perform advanced searches to locate requests based on information that is defined as critical to business processes.

As the number of requests logged increases, users performing advanced searches can experience slower performance. To improve performance during advanced searches, use the following guidelines:

- Specify additional request header fields in the advanced searches. Header fields are automatically indexed by PPM Center, and therefore yield faster returns.
- Add indexes to a limited number of detail fields, preferably fields that are commonly used in advanced searches. Take care not to add too many indexes, since this can affect the performance of inserts and updates to the database.
- Set the `DEFAULT_REQUEST_SEARCH_ORDER_BY_ID` server configuration parameter value to `TRUE` to remove the sort order column on a request search. Record sorting slows performance.
- Change the value set for the `REQUEST_SEARCH_RESULTS_MAX_ROWS` server configuration parameter to restrict the maximum number of records retrieved.
- For portlet search queries, lower the value set for the `PORTLET_MAX_ROWS_RETURNED` server configuration parameter. For most portlets, 20 to 50 records is adequate. The default is 200.

# Adjusting Server Configuration Parameters

This section provides information about PPM Server parameters related to system performance and usage considerations for these parameters.

Parameter categories are:

- Cleanup parameters
- Debug parameters
- Timeout parameters
- Scheduler/services/thread parameters
- Database connection parameters
- Cache parameters

Most of the parameters are defined in the `server.conf` file. For a list of PPM Server parameters, see *Server Configuration Parameters* on page 225. The following sections provide descriptions of the parameters in each system performance parameter category.

## Cleanup Parameters

Cleanup parameters, which are all defined in the `server.conf` file, determine when the PPM Server invokes services to clean up database tables:

- `DAYS_TO_KEEP_INTERFACE_ROWS` determines how many days to keep records of all interfaces.
- `DAYS_TO_KEEP_LOGON_ATTEMPT_ROWS` determines how many days to keep records of all logon attempts.
- `ENABLE_INTERFACE_CLEANUP` periodically removes old records from the database open interface tables. You can use the associated parameter `INTERFACE_CLEANUP_INTERVAL` to specify the run frequency for this thread, and the parameter `DAYS_TO_KEEP_INTERFACE_ROWS` to specify how long to keep records in the interface tables.
- `HOURS_TO_KEEP_DEBUG_MESSAGE_ROWS` determines how long (in hours) to keep rows in the `KNTA_DEBUG_MESSAGES` table.
- `NOTIFICATIONS_CLEANUP_PERIOD` determines the cleanup interval (in days) for notifications sent previously.

If periodic slowdowns occur, check these parameters and the Service Controller report to check for a correlation between the times when cleanup services run and the slowdowns occur. If necessary, change these parameters to avoid running cleanup services during peak periods.

For information about the Service Controller report, see *Table 6-2* on page 139. For more information about the cleanup parameters, see *Server Configuration Parameters* on page 225.

## Debug Parameters

Debug parameters control the debug and log output from the PPM Server. Debug parameters are either high- or low-level.

### High-Level Debug Parameters

You can change high-level debug parameters without causing system downtime on the PPM Server. Users who have the required privileges can configure these parameters by selecting **Edit > Debug Settings** from the PPM Workbench.

The high-level debug parameters are:

- `DEFAULT_USER_DEBUG_LEVEL` (defined in the `logging.conf` file) control the debugging level.
- `ENABLE_JDBC_LOGGING` (defined in the `server.conf` file) determines whether the server maintains a JDBC log file. If it is enabled, JDBC logging records SQL runs against the database, the amount of time required to run the SQL, and the amount of time required to retrieve the results.
- `ENABLE_SQL_TRACE` (defined in the `server.conf` file) determines whether performance statistics for all SQL statements run are placed into a trace file.
- `SERVER_DEBUG_LEVEL` (defined in the `logging.conf` file) controls the verbosity of logs generated by independent server processes such as EmailNotificationAgent.

For more information about the high-level debug parameters, see *Server Configuration Parameters* on page 225 and *Logging Parameters* on page 271.

### Low-Level Debug Parameters

Enable the low-level debug parameters only if you require debugging information for a specific area. Enabling these parameters can degrade system performance because they consume additional CPU and generate large log files.

➤ HP strongly recommends that you consult HP-Mercury support before you enable low-level debug parameters.

The low-level debug parameters, which are all defined in the `logging.conf` file are:

- `ENABLE_DB_SESSION_TRACKING`
- `ENABLE_LOGGING`
- `ENABLE_TIMESTAMP_LOGGING`
- `EXECUTION_DEBUGGING`
- `JDBC_DEBUGGING`
- `WEB_SESSION_TRACKING`

For more information about low-level debug parameters, see *Logging Parameters* on page 271.

## Timeout Parameters

Timeout parameters determine how long the PPM Server waits before it times out. You can set timeout values for logon sessions, command runs, and workflows.

The timeout parameters, which are all defined in the `server.conf` file, are:

- `CLIENT_TIMEOUT` determines the interval (in minutes) at which PPM Workbench sessions send a message to inform the PPM Server that the client is active.
- `DB_LOGIN_TIMEOUT` determines the duration (in seconds) for the PPM Server to keep trying to log on to the database before reporting that the database is unavailable.
- `DEFAULT_COMMAND_TIMEOUT` determines the duration (in seconds) for the PPM Server to keep trying to run commands before timing out.

- PORTLET_EXEC_TIMEOUT determines the duration (in seconds) after which portlets time out.
- SEARCH_TIMEOUT determines the duration (in seconds) after which searches time out.

## Scheduler/Services/Thread Parameters

Scheduler/services/thread parameters, which are all defined in the server.conf file, control scheduling, services, and thread-related server activities.

The scheduler/services/thread parameters are:

- AUTOCOMPLETE_STATUS_REFRESH_RATE determines the frequency (in seconds) with which the command status is refreshed to provide a list of values in an auto-complete.
- EMAIL_NOTIFICATION_CHECK_INTERVAL determines the frequency (in seconds) with which the PPM Server checks for pending email notifications.
- ENABLE_EXCEPTION_ENGINE enables the exception engine, which runs a process to determine whether active projects are running on time.
- EXCEPTION_ENGINE_INTERVAL determines the frequency (in seconds) with which the exception engine process runs (if ENABLE_EXCEPTION_ENGINE = TRUE).
- EXCEPTION_ENGINE_WAKE_UP_CHECK_FREQUENCY determines the interval (in seconds) that elapses before a task is verified for exceptions (if ENABLE_EXCEPTION_ENGINE = TRUE).
- EXCEPTION_ENGINE_WAKE_UP_TIME determines the time at which the exception engine process runs (if ENABLE_EXCEPTION_ENGINE = TRUE).
- MAX_EXECUTION_MANAGERS determines the number of command executions that can run simultaneously. Organizations processing a high volume of packages may require a larger number of execution managers.
- MAX_RELEASE_EXECUTION_MANAGERS determines the number of command executions that can run in a release distribution simultaneously. Organizations that process a high package volume may require more release execution managers.
- MAX_WORKER_THREADS determines the number of threads that can run simultaneously to process scheduled tasks (for example, reports or request commands). If the PPM Server is heavily loaded, specify a lower value to

reduce the server workload. If there are many pending tasks, and additional capability is available on the server, set a higher value to improve performance.

- REPORTING_STATUS_REFRESH_RATE determines the frequency (in seconds) with which the report status is refreshed and displayed to the user.
- SCHEDULER_INTERVAL determines the number of seconds after which the scheduler checks for services to be run.
- THREAD_POOL_MAX_THREADS determines the maximum number of packages to run simultaneously within a release distribution. If a large number of packages in a distribution are processing, increase this value to improve performance.
- THREAD_POOL_MIN_THREADS determines the minimum number of packages to be run simultaneously within a release distribution.
- TURN_ON_WF_TIMEOUT_REAPER turns on the timeout reaper, which scans all active workflow steps to verify that they have timed out according to the settings for the step.
- TURN_ON_NOTIFICATIONS turns on the notification service. Use this to turn off notifications for copies of production instances being used for testing, and to turn them on again when the system goes to production.
- TURN_ON_SCHEDULER turns on the scheduler. Use this to improve performance. Turn off the scheduler in non-production instances.
- WF_SCHEDULED_TASK_INTERVAL establishes the frequency (in seconds) with which the PPM Server checks for pending scheduled tasks, and starts the tasks if worker threads are available.
- WF_SCHEDULED_TASK_PRIORITY determines the priority of scheduled tasks. Because scheduled tasks run in the background, it may be useful to run these tasks at a lower priority than the threads servicing user-oriented interactive tasks.
- WF_TIMEOUT_REAPER_INTERVAL determines the frequency (in seconds) with which the service checks for information (if TURN_ON_WF_TIMEOUT_REAPER = TRUE).

Database connection parameters relate to the management of the database connection pool that the PPM Server maintains. After the PPM Server starts, one database connection is established. Increased usage spawns additional database connections.

These parameters, which are all defined in the `server.conf` file, are:

- `MAX_DB_CONNECTION_IDLE_TIME` determines the amount of time (in minutes) that an unused database connection is held open before it is closed and removed from the pool.

- `MAX_DB_CONNECTION_LIFE_TIME` determines the duration (in minutes) that a database session is held open before it is closed and removed from the pool. Some Oracle cleanup operations that should be run periodically occur only at the end of database sessions. Do not keep database sessions open for the life of the PPM Server.

- `MAX_DB_CONNECTIONS` determines the number of database connections to hold open. In a server cluster configuration, this is the number of database connections for each PPM Server. Once this number is reached, user sessions queue for the next available database connection.

- `MAX_STATEMENT_CACHE_SIZE` determines the maximum number of prepared statements cached per database connection.

### Logging Parameters

The logging parameters are in the `logging.conf` file. For more information, see *Logging Parameters* on page 271.

# 8 Migrating Entities

In This Chapter:

# About Entity Migration

Entity migrators are HP Deployment Management object types. Each migrator is designed to migrate a specific kind of PPM Center entity and all of its dependent objects from one PPM Center instance to another.

You can use HP Deployment Management to manage configuration changes to PPM Center. HP Deployment Management comes with an out-of-the-box set of object types, or *entity migrators,* that you can use to move PPM Center configuration entities (workflows, request types, and so on) between PPM Center instances. If you maintain scratch instances for developing and testing PPM Center configurations before you deploy them into your production instance, you must use these entity migrators, and develop a workflow that drives configuration changes through your source configuration management deployment process.

Migrating configurations using entity migrators and workflows lets you automate and standardize a change-control process for your PPM Center implementation. You can build a workflow for every migrator object type, or create a single generic workflow for all migrator object types.

➤ You can only migrate entities between PPM Center instances of the same version.

You can migrate the following PPM Center entities:

- Special commands
- Object types
- Portlet definitions
- PPM Dashboard modules
- PPM Dashboard data sources
- Project types
- Work plan templates
- Report types
- Request header types
- Request types
- User data contexts
- Validations
- Workflows

## Migration Order

If you plan to migrate request type, workflow, project type, and work plan template configurations that are related to each other, you must perform the migration in the following order:

1. Request type
2. Workflow
3. Request type again (if circular references exist between request type and workflow)
4. Work plan template
5. Project type

# Overview of Entity Migration

Consider a scenario in which you want to migrate configuration entities between your “QA” and “Production” instances of PPM Center. You can automate and track the migration using either the source instance (QA) or the destination instance (Production). In the example that follows, you are using the destination instance to control the migration.

You migrate PPM Center entities in the same way that you perform any other deployment management process. To prepare for the entity migration you do the following:

- Set up the environment definitions for your “QA” and “Production” instances.

- Configure a workflow that directs the migration process (necessary approvals, and an automated execution step that specifies your “QA” and “Production” environments as source and destination, respectively).

After you perform these tasks, you can use HP Deployment Management packages to specify the entities to migrate. Create a package, specify your migration workflow, and add package lines using the entity migratory object types for each PPM Center configuration entity that you want to migrate.

When the automated migration execution workflow step is run, the following events occur (remember that, in this example, you are running the migration in the destination, or Production, environment):

1. The Production server connects to the QA server using Telnet or SSH, and then submits a request for the specified configuration data.

2. The QA server extracts the requested configuration data from its database and generates an XML representation of the data.

3. The QA server writes the extracted XML data into a set of temporary XML files, and packages that set of files together in a Zip file.

4. The Production server copies the Zip file that contains the bundled XML data from QA to Production.

   If you want to perform version control on changes to PPM Center configuration entities as they are migrated, you can version the compressed file that is extracted from the source instance.

   HP recommends that you not extract this file manually, except for debugging purposes.

5. The Production server unpacks the migrated compressed file into temporary storage, and reads the associated XML files.

6. The Production server imports the configuration data to its database, and then generates an execution log.

## Example Migration: Extracting a Request Type

The following example illustrates a procedure that you can use to migrate a request type from a QA instance of PPM Center to a production instance.

➤ To create, submit, and process migrations, you must have the required licenses and access grants. For more information, see the *Security Model Guide and Reference*.

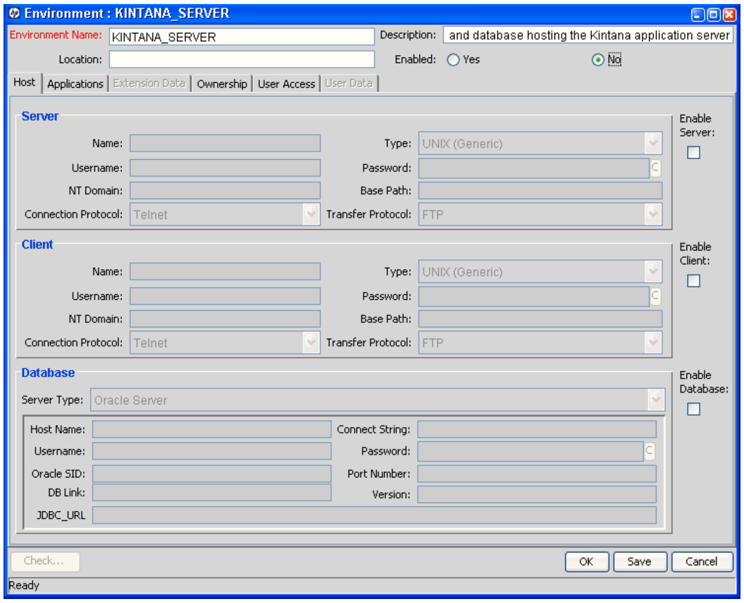
Before you perform the following steps, make sure that you have a valid user account in both the source and destination instances, and that these accounts have the same user name. When the migrator extracts an entity from the source instance, and then imports it into the destination instance, it provides your security information.

To migrate a request type:

1. If the environment definition for the PPM Server is not configured, then you must first create the `KINTANA_SERVER` environment, as follows:

   ➤ Because you control this migration from the production instance, the environment you define represents the destination for entity migrations.

   a. In the Environment Workbench, open the `KINTANA_SERVER` environment.



Environment : KINTANA_SERVER

Environment Name: KINTANA_SERVER
Description: and database hosting the Kintana application server
Location:
Enabled: ○ Yes ● No

Server: Name: Username: NT Domain: Connection Protocol: Telnet; Type: UNIX (Generic); Password: Base Path: Transfer Protocol: FTP; Enable Server: □

Client: Name: Username: NT Domain: Connection Protocol: Telnet; Type: UNIX (Generic); Password: Base Path: Transfer Protocol: FTP; Enable Client: □

Database: Server Type: Oracle Server; Host Name: Username: Oracle SID: DB Link: JDBC_URL; Connect String: Password: Port Number: Version:; Enable Database: □

Ready

   b. To the right of the **Server** section, select the **Enable Server** checkbox.

c. Provide the server information.

Because this environment definition represents the PPM Server that you are using to run the migration, there is no need to specify connection information for it. The migrator performs the required actions locally, without opening a separate Telnet or SSH session.

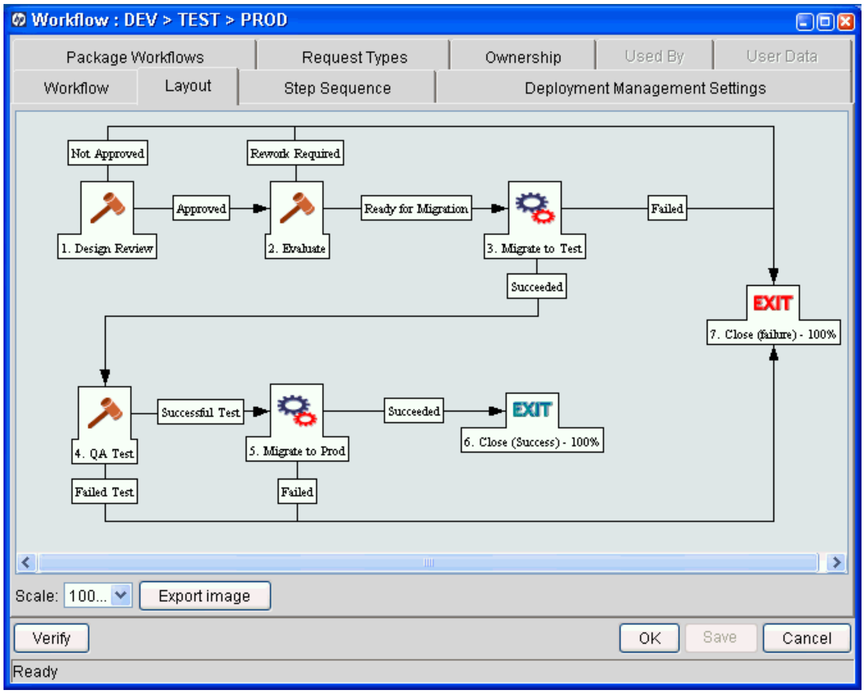
d. Define and enable the source environment.

You must specify connection information for the source environment, including the user name and password, base path, and connection and transfer protocols.

2. Create a deployment management workflow.

For information about how to create a workflow, see the *HP Deployment Management Configuration Guide.*

Specify the QA environment as the source, and the production environment (`KINTANA_SERVER`) as the destination of the execution step.



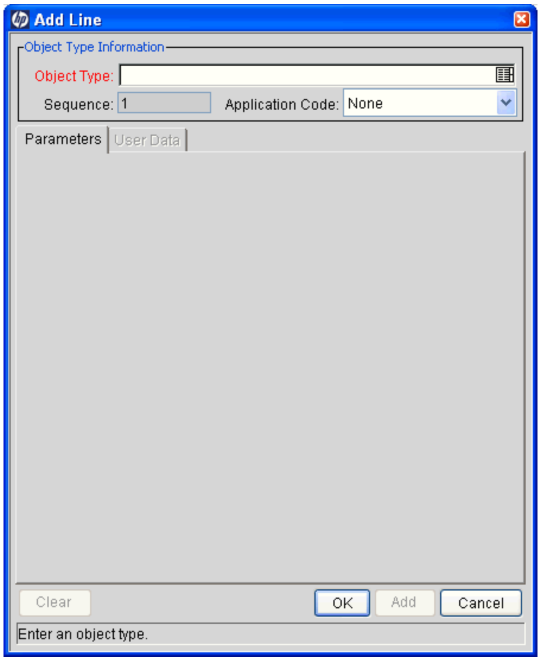
3. Create a package.

For information about packages and how to create a package, see the *HP Deployment Management User's Guide.*

4. In the Package: <*Package Name*> window, in the **Workflow** field, enter the workflow you created.

5. Click **New Line.**

   The Add Line dialog box opens.



Sequence: 1

Application Code: None

Enter an object type.

6. In the **Object Type** field, type **`PPM Request Type Migrator`**.

7. Enter the following required information:

   - In the **HP PPM source password** field, type the password for your PPM Center account in the source instance.

   - In the **HP PPM dest password** field, type the password for your PPM Center account in the destination instance.

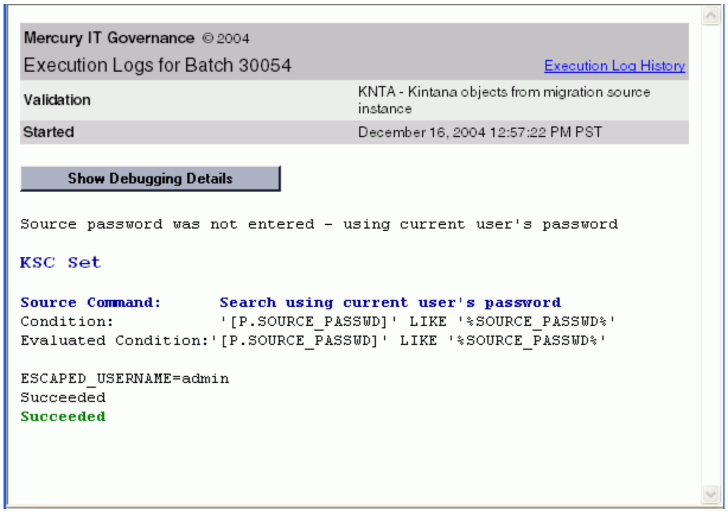
   - In the **Request type** field, type the name of the request type that you want to migrate.

8. Click **OK.**

9. Submit the workflow.

10. Process the workflow.

11. Check the execution log to verify that the migration was successful.



Mercury IT Governance © 2004

Execution Logs for Batch 30054

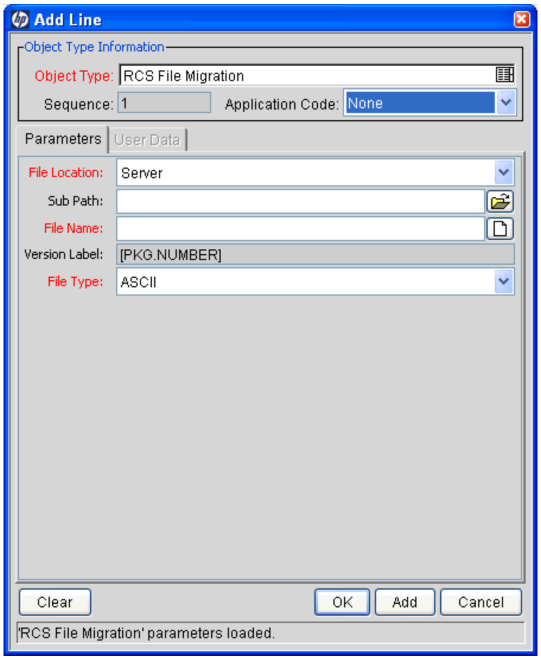
| | |
|---|---|
| Validation | KNTA - Kintana objects from migration source instance |
| Started | December 16, 2004 12:57:22 PM PST |

```
Source password was not entered - using current user's password

KSC Set

Source Command:      Search using current user's password
Condition:           '[P.SOURCE_PASSWD]' LIKE '%SOURCE_PASSWD%'
Evaluated Condition:'[P.SOURCE_PASSWD]' LIKE '%SOURCE_PASSWD%'

ESCAPED_USERNAME=admin
Succeeded
Succeeded
```

# Defining Entity Migrators

Each object type for the PPM Center entity migrators has a set of parameters similar to those described in this section (and as illustrated in the previous example). The RCS File Migrator shown in *Figure 8-1* is an example.

Figure 8-1. Add Line dialog box for the RCS File Migrator



Object Type: RCS File Migration

Sequence: 1    Application Code: None

File Location: Server

Sub Path:

File Name:

Version Label: [PKG.NUMBER]

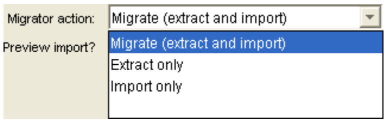
File Type: ASCII

'RCS File Migration' parameters loaded.

## Migrator Action List

To control how extensive a migration to perform, use the **Migrator action** list on the **Parameters** tab of the Add Line dialog box. *Figure 8-2* shows the **Migrator action** list.

Figure 8-2. Migrator action list



Migrator action: Migrate (extract and import)

In the **Migrator action** list, you can select one of the following actions:

- **Migrate (extract and import)**
- **Extract only**
- **Import only**

*Table 8-1* lists the controls in the Add Line dialog box that are affected by the migrator action you select, and provides information about how each control is affected.

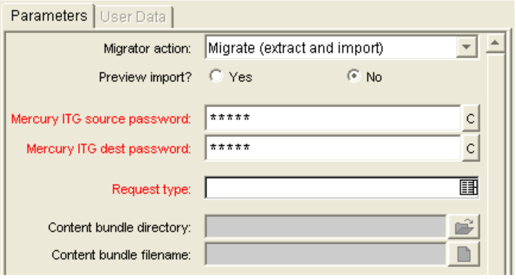
Table 8-1. Migrator action list dependencies

| Control and Control Set Names | Extract and Import | Extract Only | Import Only |
|---|---|---|---|
| Preview Import | Enabled | Disabled | Enabled |
| Target entity field | Required | Required | Disabled |
| Content bundle fields | Disabled | Enabled | Required |
| Import behavior fields | Enabled | Disabled | Enabled |
| Source password | Required | Required | Disabled |
| Destination password | Required | Disabled | Required |

## Basic Parameters

Whether the basic parameters are required or simply available depends on the migrator action you select. In *Figure 8-3*, the parameters are the entity name (in this case, the request type), content bundle directory, and content bundle filename.

Figure 8-3. Basic parameters



Migrator action: Migrate (extract and import)

Preview import? ○ Yes ● No

Mercury ITG source password: *****

Mercury ITG dest password: *****

Request type:

Content bundle directory:

Content bundle filename:

### Content Bundle Controls

The behavior of controls related to the content bundle depends on the migrator action you select, as follows:

- If you select **Migrate (extract and import)**, the migrator maintains its own internal scheme for naming and locating the temporary bundled XML data. This content bundle is extracted from the source, migrated to the destination, imported, and then cleaned up, all as part of the same execution step. The user cannot edit the content bundle information.

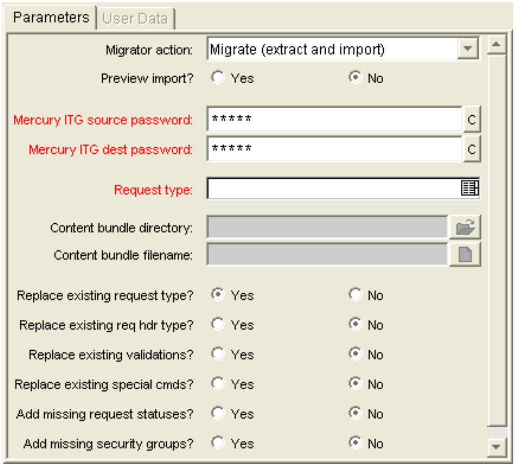
- If you select **Extract only**, you can specify the content bundle location and filename, or accept the default values. This lets you specify a location and naming convention that is easier to remember so that you can locate the extracted content bundle and use it as necessary (for example, check it into your version control system). By default, the migrator creates the bundle in the file system of the source PPM Server under the `<`*`PPM_Home`*`>/transfers` directory. The filename is based on the type of entity migrated, its package number, and its package line number.

- If you select **Import only**, you must enter the name and location of an existing content bundle file to import. You can select the file by browsing the file system of the destination PPM Server.

## Import Flags

Use the import flags listed in the lower portion of the **Parameters** tab (shown in *Figure 8-4*) to control migrator behavior.

Figure 8-4. Import flags



Parameters | User Data

| | | |
|---|---|---|
| Migrator action: | Migrate (extract and import) | |
| Preview import? | ○ Yes | ● No |
| Mercury ITG source password: | ***** | |
| Mercury ITG dest password: | ***** | |
| Request type: | | |
| Content bundle directory: | | |
| Content bundle filename: | | |
| Replace existing request type? | ● Yes | ○ No |
| Replace existing req hdr type? | ○ Yes | ● No |
| Replace existing validations? | ○ Yes | ● No |
| Replace existing special cmds? | ○ Yes | ● No |
| Add missing request statuses? | ○ Yes | ● No |
| Add missing security groups? | ○ Yes | ● No |

The available import flags vary with object type.

### Preview Import Option

If you set **Preview Import?** to **Yes,** the migrator does not actually import the migrated entity into the destination instance, but instead, simulates the migration and generates an execution log.

### Import Behavior Controls

The following settings modify the specific import behavior for the entity to migrate.

- **Replace existing request type?** If the entity to migrate already exists in the target PPM Center instance, you can decide whether or not to replace it. The default selection is **Yes.**

  If the entity does not exist in the destination instance, it is created.

- **Replace existing req hdr type?** If the request type to be migrated references a request header type that already exists in the target PPM Center instance, you can decide whether or not to replace it. The default value is **No.**

- **Replace existing validations?** If the target entity references validations that already exist in the target PPM Center instance, you can decide whether or not to overwrite them. The default value is **No.**

Regardless of the value, any validations that are missing from the destination instance are automatically created.

- **Replace existing special cmds?** If the validation to be migrated references PPM Center special commands (including parent and child special commands) that exist in the target PPM Center instance, you can decide whether or not to replace them. The default value is **No.**

- **Add missing request statuses?** If the request type to be migrated references request statuses that do not exist in the target PPM Center instance, you can decide whether or not to create them. The default value is **No.**

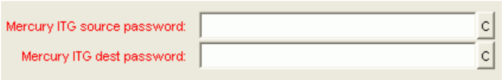
- **Add missing security groups?** If the entity to be migrated references security groups that are not included in the target instance, you can add those security groups. The default value is **No.**

  Only the list of associated access grants, but not associated users, is transferred.

## Password Controls

If the **Migrator action** list displays **Migrate (extract and import)**, then the **HP PPM source password** and **HP PPM dest password** fields (*Figure 8-5*) are enabled.

Figure 8-5. Password fields



Mercury ITG source password: C

Mercury ITG dest password: C

### Source Password Field

When the migrator contacts the source server, it uses the credentials of the current PPM Center user to authorize the entity extraction. This user must be part of a security group that contains the access grant “System Admin: Migrate HP PPM Objects.” Confirm the user password for the source server in the **HP PPM source password** field.

### Destination Password Field

When the migrator contacts the destination server, it uses the credentials of the current PPM Center user to authorize the entity import. This user must be part of a security group that has the “Sys Admin: Migrate HP PPM Objects” access grant. Confirm the user password for the destination server in the **HP PPM dest password** field.

## Internationalization List

Typically, in an environment in which you are managing configuration across multiple PPM Servers, all of the PPM Center databases involved have the same localization settings. However, if you must migrate configuration entities between PPM Center databases that have different localization settings, you can change the localization-checking behavior of the migrator by changing the value of the **Internationalization** list.

By default, the **Internationalization** list is invisible to users on migrator object types. But the control is enabled and set to **Same language and character set.** To change this setting:

1. Log on to PPM Center.
2. From the menu bar, select **Administration > Open Workbench.**

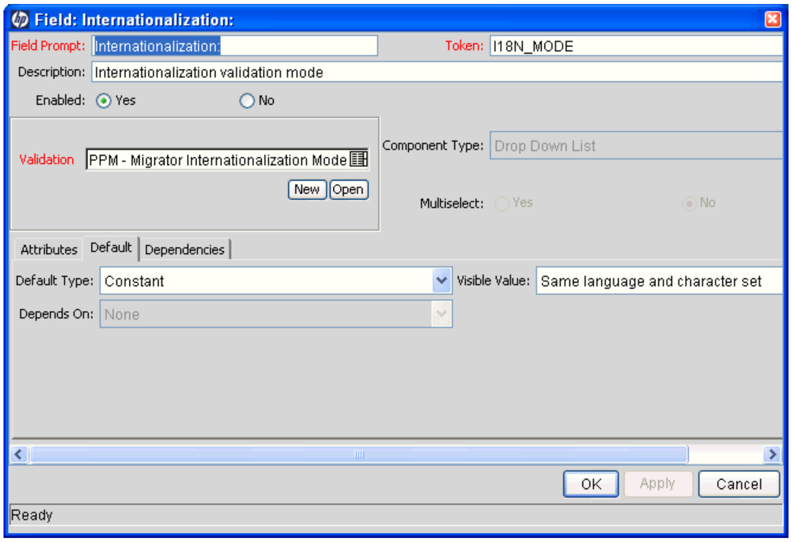
   The PPM Workbench opens.
3. From the shortcut bar, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench window opens.
4. Click **List.**
5. In the **Object Name** column on the **Results** tab, double-click **PPM Request Type Migrator.**

   The Object Type: PPM Request Type Migrator window opens.
6. In the **Prompt** column on the **Fields** tab, double-click **Internationalization.**

   The Field: Internationalization window opens.

7. Click the **Default** tab.



Field: Internationalization:

Field Prompt: Internationalization:

Token: I18N_MODE

Description: Internationalization validation mode

Enabled: ● Yes ○ No

Validation: PPM - Migrator Internationalization Mode

Component Type: Drop Down List

Multiselect: ○ Yes ● No

Default Type: Constant

Visible Value: Same language and character set

Depends On: None

Ready

8. From the **Visible Value** list, select one of the following:

- **Same language and character set.** This is the default option for migrating entities between PPM Center instances running under the same language and character set configuration. It is the most conservative option; any difference in locale, language, or character set between the source and destination servers is flagged as an error and the migration fails.

- **Different language or character set.** This option lets you override character set or language incompatibilities within the same localization. Use this option if you know that the language or character set settings are different across the source and destination servers, but you want to run the migration anyway and you do not anticipate the differences to cause problems with the entity data you want to migrate. For example, if the destination character set is a superset of the source character set, then you know that data extracted from the source will be valid in the destination.

- **Different localization.** This option lets you migrate content between instances belonging to different localizations (for example, English to German, or German to English). This is the least restrictive option for migrating configuration data across PPM Servers that have different locale settings. Selecting this value could potentially result in invalid data (unsupported characters, and so on) in the destination instance. Be sure to examine (and possibly update) the migrated entity data to ensure that it is valid in the destination.

9. Click **OK.**

# Environment Considerations

When migrating entities, HP Deployment Management logs on to remote machines in the same way another user would (that is, using FTP, SCP, SSH, or Telnet). HP Deployment Management can log to a remote server using any existing operating system user name and password.

HP recommends that you generate a new user (for example, PPM Center) on every machine to which HP Deployment Management has access. A user you create for this purpose must have full access to the `<PPM_Home>` directory on the PPM Server, and read and write permissions on other required directories.

## Environment Connection Protocol

The environment definition must include information about the communication protocol (for example, Telnet) to be used to connect to the server or client. For information about connection protocols that PPM Center supports, see the *System Requirements and Compatibility Matrix* and the *HP Deployment Management Configuration Guide*.

## Environment Transfer Protocol

The environment definition must include information about the transfer protocol to be used to transfer files to or from machines specified in the environment definition. Choose the transfer protocol that best suits your business and technology needs. Consider factors related to security and performance when selecting the transfer protocol. Work with the application administrator to determine which connection protocols are supported for the machines housing the deployment environments.

For information about transfer protocols, see the *HP Deployment Management Configuration Guide.*

## Setting the SERVER_ENV_NAME Parameter

The PPM Center migrators depend on the `SERVER_ENV_NAME` server configuration parameter. This parameter specifies the name of an environment definition in the PPM Center system that describes the host server running that PPM Center instance.

When you installed PPM Center, the `KINTANA_SERVER` environment definition was automatically created on your system. This name is set as the default value of the `SERVER_ENV_NAME` server parameter. PPM Center often refers to this server parameter to find the environment definition that contains information about the computer[s] that host the PPM Server and database. For this reason, it is important that you keep this server parameter synchronized with the name of the corresponding environment definition, as follows:

```
SERVER_ENV_NAME=KINTANA_SERVER
```

# Security Considerations

This section provides information about security considerations related to ownership and entity restrictions.

## Migration and Ownership

Different groups of PPM Center users have ownership and control over different PPM Center entities. These groups are called ownership groups. Unless a global permission has been designated to all users for an entity, members of ownership groups are the only users who have the right to edit, delete, or copy that entity. The ownership groups must also have the proper access grant for the entity in order to complete those tasks.

Application administrators can assign multiple ownership groups to entities. The ownership groups will have sole control over the entity, providing greater security. Ownership groups are defined in the Security Groups window. Security groups become ownership groups when used in the ownership configuration.

Ownership applies to PPM Center entities during migrations in the following ways:

- If no ownership security is configured for the entity, any user who has permission to perform migrations can migrate it.
- If entity ownership is configured and the user migrating is not in the ownership group, the migration fails.
- If entity ownership is configured and the user migrating is in the ownership group, the migration succeeds.
- If entity ownership is configured and the user migrating is not in the ownership group but has the Ownership Override access grant, the migration succeeds.

➤ These conditions apply to entity import, but not to entity export.

## Migrations and Entity Restrictions

A report type might refer to security groups through entity restrictions. The Report Type migrator transfers references to security groups, but does not create any new security groups in the destination instance of PPM Center. If the referenced security group does not exist in the destination instance, the reference is discarded in transit. A message to that effect is displayed in the migration execution log.

If the source instance contains security groups that do not exist in the destination instance during migration, the entity restrictions for the migrated report type might be inaccurate. Therefore, after migration, manually verify report types that contain entity restrictions in the destination instance.

# Entity Migrators

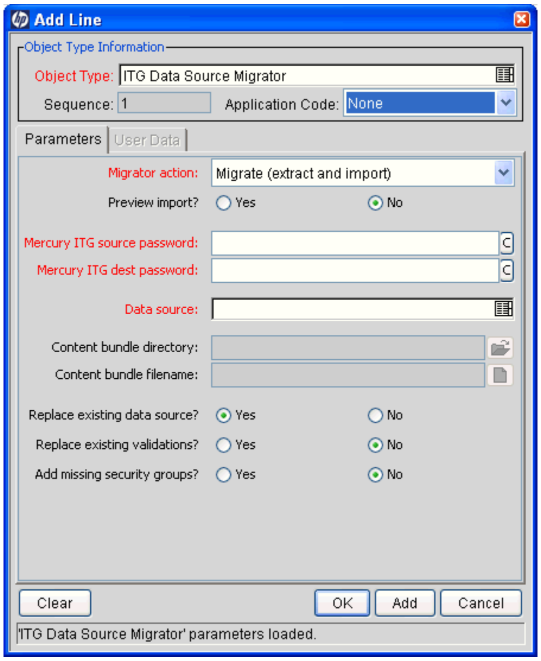
This section provides descriptions of PPM Center entity migrators.

## Data Source Migrator

You can use the Data Source Migrator to move a data source that you created in the Data Source Workbench between the PPM Center instances. (Data sources provide data displayed in PPM Dashboard portlets.)

*Figure 8-6* on page 192 shows the parameters for the Data Source migrator as they are displayed during package line creation.

Figure 8-6. Data Source Migrator



Object Type: ITG Data Source Migrator

Sequence: 1 Application Code: None

Migrator action: Migrate (extract and import)

Preview import? ○ Yes ● No

Mercury ITG source password:

Mercury ITG dest password:

Data source:

Content bundle directory:

Content bundle filename:

Replace existing data source? ● Yes ○ No

Replace existing validations? ○ Yes ● No

Add missing security groups? ○ Yes ● No

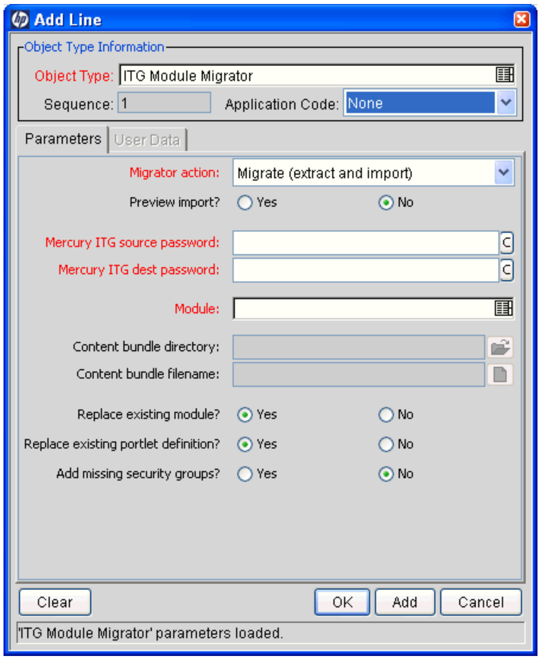
'ITG Data Source Migrator' parameters loaded.

For information about the fields in this migrator, see *Defining Entity Migrators* on page 182. For information about how to create a portlet data source, see the *Creating Portlets and Modules* guide.

## Module Migrator

In the PPM Center standard interface, a module is the set of pages that an administrator sets up for users to view and navigate in the PPM Dashboard. You can use the Module Migrator to move PPM Center modules from one PPM Center environment to another.

Figure 8-7. Module Migrator



Add Line

Object Type Information

Object Type: ITG Module Migrator

Sequence: 1 Application Code: None

Parameters | User Data

Migrator action: Migrate (extract and import)

Preview import? ○ Yes ● No

Mercury ITG source password:

Mercury ITG dest password:

Module:

Content bundle directory:

Content bundle filename:

Replace existing module? ● Yes ○ No

Replace existing portlet definition? ● Yes ○ No

Add missing security groups? ○ Yes ● No

'ITG Module Migrator' parameters loaded.

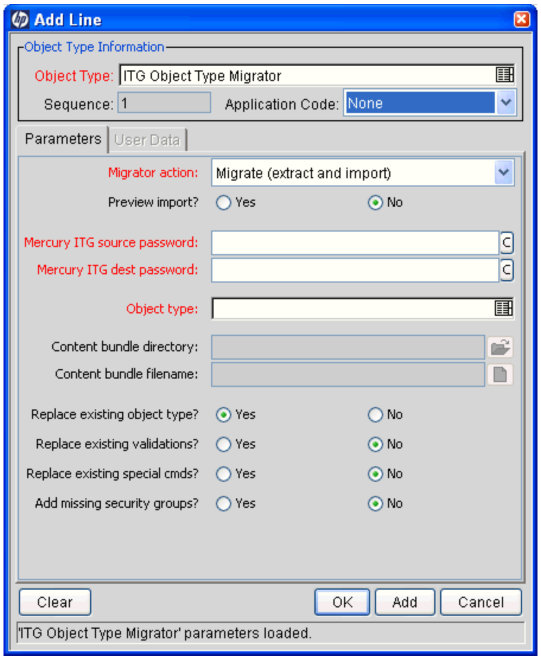
For information about the fields in this migrator, see *Defining Entity Migrators* on page 182. For information about how to create modules, see the *Creating Portlets and Modules* guide.

## Object Type Migrator

The Object Type Migrator (*Figure 8-8* on page 194) contains the additional option **Replace existing special cmds?** If the validation to be migrated references PPM Center special commands (including parent and child special commands) that exist in the target PPM Center instance, you can decide whether or not to replace them. The default value is **No.**

Regardless of the migrator settings, special commands missing from the destination instance are created automatically.

Figure 8-8. Object Type Migrator



Object Type: ITG Object Type Migrator
Sequence: 1
Application Code: None

Migrator action: Migrate (extract and import)
Preview import? ○ Yes ● No
Mercury ITG source password:
Mercury ITG dest password:
Object type:
Content bundle directory:
Content bundle filename:
Replace existing object type? ● Yes ○ No
Replace existing validations? ○ Yes ● No
Replace existing special cmds? ○ Yes ● No
Add missing security groups? ○ Yes ● No

'ITG Object Type Migrator' parameters loaded.

For information about most of the controls in this migrator window, see *Defining Entity Migrators* on page 182.

### Configuration Considerations

The PPM Object Type Migrator also transfers the following information:

- Special commands referenced by command steps
- Validations referenced by fields
- Environments referenced by validations
- Special commands referenced by validations
- Special commands referenced by other special commands

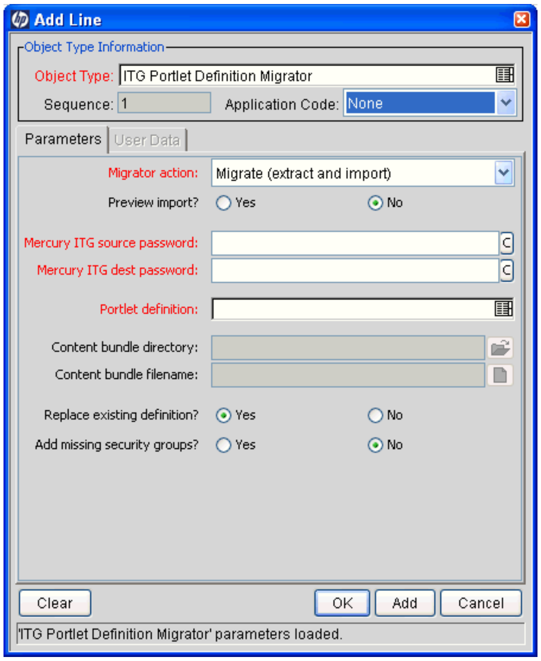
- Ownership group information for the entity

The migrator transfers references to environments from validations, but does not create any new environments. If the referenced environment does not exist in the destination instance, the migration fails. If this happens, create the missing environment manually in the destination instance.

## Portlet Definition Migrator

The Portlet Definition Migrator (*Figure 8-9*) contains all standard entity migrator object type fields. If you migrate a portlet definition to replace an existing enabled portlet definition the destination instance of PPM Center, the migrated changes are applied to all users who have added the same portlet to their PPM Dashboard pages.

Figure 8-9. Portlet Definition Migrator



Object Type: ITG Portlet Definition Migrator

Sequence: 1

Application Code: None

Migrator action: Migrate (extract and import)

Preview import? ○ Yes ● No

Mercury ITG source password:

Mercury ITG dest password:

Portlet definition:

Content bundle directory:

Content bundle filename:

Replace existing definition? ● Yes ○ No

Add missing security groups? ○ Yes ● No

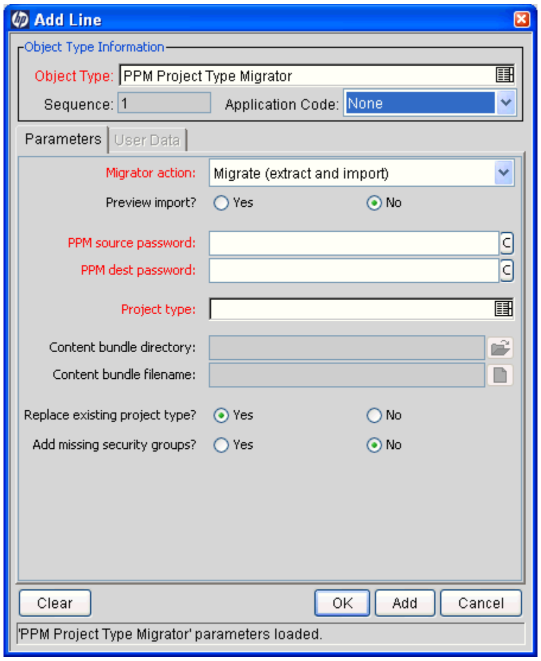
'ITG Portlet Definition Migrator' parameters loaded.

For information about the fields in this migrator, see *Defining Entity Migrators* on page 182.

## Project Type Migrator

You can define project types in a development or testing instance of PPM Center, and then use the Project Type Migrator (*Figure 8-10*) to migrate them to production after testing.

Figure 8-10. Project Type Migrator



Object Type Information

Object Type: PPM Project Type Migrator

Sequence: 1

Application Code: None

Migrator action: Migrate (extract and import)

Preview import? ○ Yes ● No

PPM source password:

PPM dest password:

Project type:

Content bundle directory:

Content bundle filename:

Replace existing project type? ● Yes ○ No

Add missing security groups? ○ Yes ● No

'PPM Project Type Migrator' parameters loaded.

The Project Type Migrator migrates the following:

- Header information such as name and enabled flag
- All policies (including all attributes)
- References to request types for project, issue, and so on

If the migrator cannot locate these objects in the destination instance, then the references are dropped and a warning message is written into the migrator log file. The migrator report contains information about how each entity association was resolved (or lost).

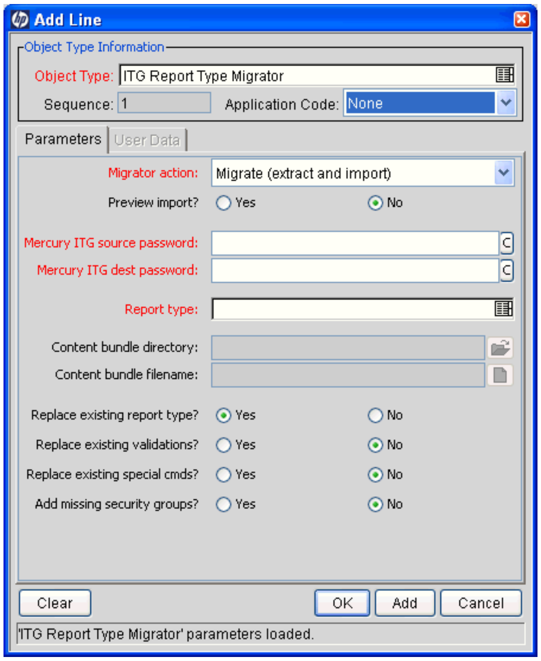
Project types are connected to work plan templates, resource pools, project requests, and issue requests. None of these entities are migrated with project types. However, if these entities exist in the destination instance, the connection to them is maintained (the migrators identify entities by name). Because project types are useless without an associated project request, you must either migrate the associated request type first, so that the link to the project type is resolved when you migrate the project type is migrated, or edit the project type after you migrate it.

The Project Type Migrator does not transport secondary objects as dependencies.

## Report Type Migrator

The Report Type Migrator (*Figure 8-11*) contains the additional option **Replace Existing special cmds?** If the validation to be migrated references PPM Center special commands (including parent and child special commands) that already exist in the target PPM Center instance, you can choose to replace them (or not). (The default value is **No.**) Regardless of their values, PPM Center automatically re-creates special commands that are missing from the destination instance.

Figure 8-11. Report Type Migrator



Add Line

Object Type Information

Object Type: ITG Report Type Migrator

Sequence: 1 Application Code: None

Parameters | User Data

Migrator action: Migrate (extract and import)

Preview import? ○ Yes ● No

Mercury ITG source password:

Mercury ITG dest password:

Report type:

Content bundle directory:

Content bundle filename:

Replace existing report type? ● Yes ○ No

Replace existing validations? ○ Yes ● No

Replace existing special cmds? ○ Yes ● No

Add missing security groups? ○ Yes ● No

'ITG Report Type Migrator' parameters loaded.

For information about most of the fields in this migrator, see *Defining Entity Migrators* on page 182.

### Configuration Considerations

The Report Type Migrator also transfers the following information:

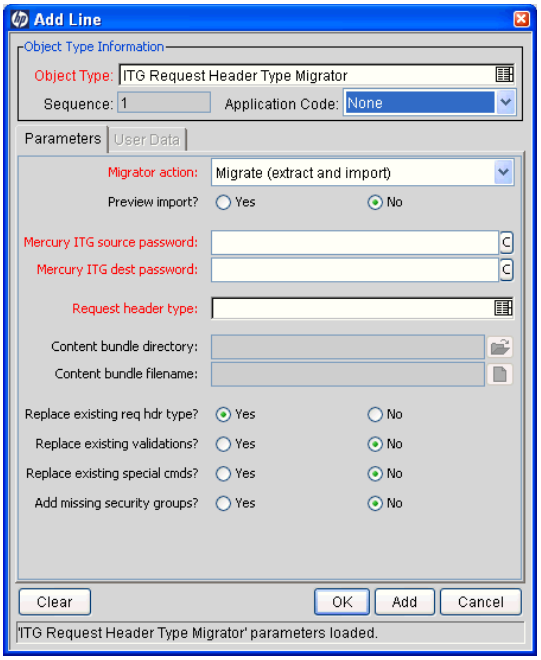
- Special commands referenced by command steps
- Validations referenced by fields
- Environments referenced by validations
- Special commands referenced by validations
- Special commands referenced by other special commands
- Ownership group information for the report type

➤ The Report Type Migrator transfers references to environments from validations, but does not create an environment. If the referenced environment does not exist in the destination instance, the migration fails. If this occurs, you must create the missing environment manually in the destination instance.

## Request Header Type Migrator

The Request Header Type Migrator (*Figure 8-12*) contains the additional option **Replace Existing special cmds?** If the validation to be migrated references PPM Center special commands that already exist in the target PPM Center instance, you can decide whether or not to replace them. This includes both parent and children special commands. (The default value is **No.**) Regardless of their values, PPM Center automatically re-creates special commands that are missing from the destination instance.

Figure 8-12. Request Header Type Migrator



Object Type Information

Object Type: ITG Request Header Type Migrator

Sequence: 1 Application Code: None

Migrator action: Migrate (extract and import)

Preview import? ○ Yes ● No

Mercury ITG source password:

Mercury ITG dest password:

Request header type:

Content bundle directory:

Content bundle filename:

Replace existing req hdr type? ● Yes ○ No

Replace existing validations? ○ Yes ● No

Replace existing special cmds? ○ Yes ● No

Add missing security groups? ○ Yes ● No

'ITG Request Header Type Migrator' parameters loaded.

For information about most of the fields in this migrator, see *Defining Entity Migrators* on page 182.

### Configuration Considerations

The Request Header Type Migrator also transfers the following information:

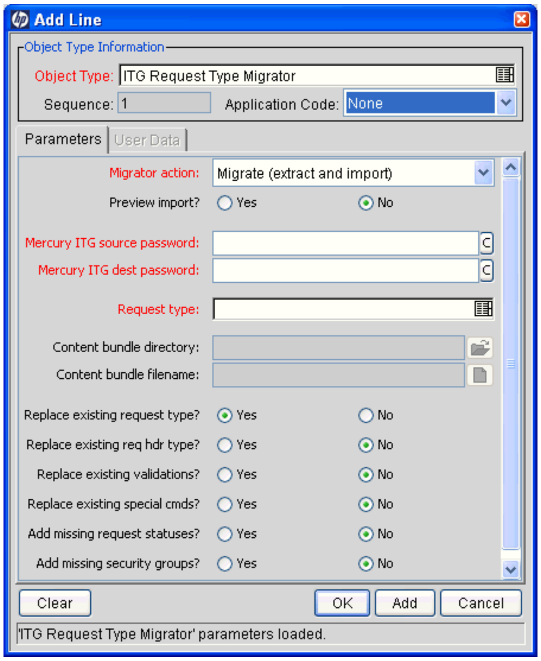
- Validations referenced by fields
- Environments referenced by validations
- Special commands referenced by validations
- Special commands referenced by other special commands
- Ownership group information for the request header type

The Request Header Type Migrator transfers references to environments from validations, but does not create an environment. If the referenced environment does not exist in the destination instance, the migration fails. In this case, you must create the missing environment manually in the destination instance.

## Request Type Migrator

The Request Type Migrator (*Figure 8-13*) has additional import behavior options from which to choose.

Figure 8-13. Request Type Migrator



Object Type Information

Object Type: ITG Request Type Migrator

Sequence: 1 Application Code: None

Migrator action: Migrate (extract and import)

Preview import? ○ Yes ● No

Mercury ITG source password:

Mercury ITG dest password:

Request type:

Content bundle directory:

Content bundle filename:

Replace existing request type? ● Yes ○ No

Replace existing req hdr type? ○ Yes ● No

Replace existing validations? ○ Yes ● No

Replace existing special cmds? ○ Yes ● No

Add missing request statuses? ○ Yes ● No

Add missing security groups? ○ Yes ● No

'ITG Request Type Migrator' parameters loaded.

The additional import behavior options are as follows:

- **Replace existing req hdr type?** If the request type to be migrated references a request header type that already exists in the target PPM Center instance, you can decide whether or not to replace it. The default value is **No.**

- **Replace Existing special cmds?** If the validation to be migrated references PPM Center special commands that already exist in the target PPM Center instance, you can decide whether or not to replace them. This includes both parent and children special commands. The default value is **No.**

  Regardless of their values, PPM Center automatically re-creates special commands that are missing from the destination instance.

- **Add missing request statuses?** If the request type to be migrated references request statuses that do not exist in the target PPM Center instance, you can decide whether or not to create them. The default value is **No.**

  In the execution log, a message is displayed for each referenced request status that is not created.

➤ If this option is set to **No,** and one of the missing request statuses is the initial status of the request type, the migration fails. In this case, you must create the request status for the initial status manually.

### Configuration Considerations

The Request Type Migrator also transfers the following information:

- Request header types referenced by the request type
- Special commands referenced by command steps
- Validations referenced by fields of the request type or request header type
- Environments referenced by validations
- Special commands referenced by validations
- Special commands referenced by other special commands already referenced elsewhere
- Request statuses referenced by the request type
- Security groups referenced by the request type (on the **Access** tab)
- Workflows referenced by the request type
- Notifications referenced by the request type
- Ownership group information for the request type

The Request Type Migrator transfers references to environments from validations, but does not create an environment. If the referenced environment does not exist in the destination instance, the migration fails. In this case, you must create the missing environment manually in the destination instance.

Simple default rules, defined in the request type **Rules** tab, might reference users, workflows, or other objects. The Request Type Migrator transfers these references, but does not create a missing user or workflow. If the referenced user or workflow does not exist in the destination instance, the reference is discarded in transit, and a message to that effect appears in the migration's execution log. You must manually reconfirm advanced default rules after migration.

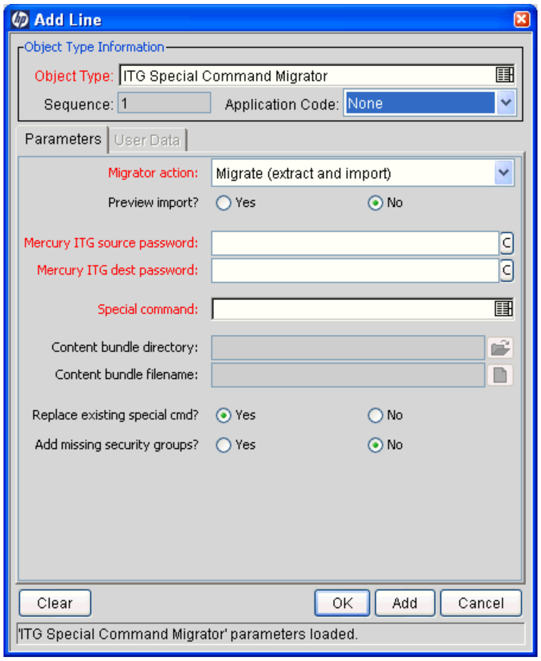
Circular references between request types and workflows could make it necessary to migrate either a request type or workflow twice:

- A new request type referring to a new workflow is migrated. Because the new workflow does not exist in the destination instance, not all references to that workflow are included in the new instance destination.
- The new workflow is migrated.
- The new request type is migrated again. This time, since the workflow it refers to exists, the references are included in the destination instance.

## Special Command Migrator

If you migrate a workflow step, request type, or object type that contains special commands, the special commands are not migrated along with the entities. You must use the Special Command Migrator (*Figure 8-14*) to move the special commands between instances of PPM Center separately.

Figure 8-14. Special Command Migrator



Add Line

Object Type Information

Object Type: ITG Special Command Migrator

Sequence: 1 Application Code: None

Parameters | User Data

Migrator action: Migrate (extract and import)

Preview import? ○ Yes ● No

Mercury ITG source password:

Mercury ITG dest password:

Special command:

Content bundle directory:

Content bundle filename:

Replace existing special cmd? ● Yes ○ No

Add missing security groups? ○ Yes ● No

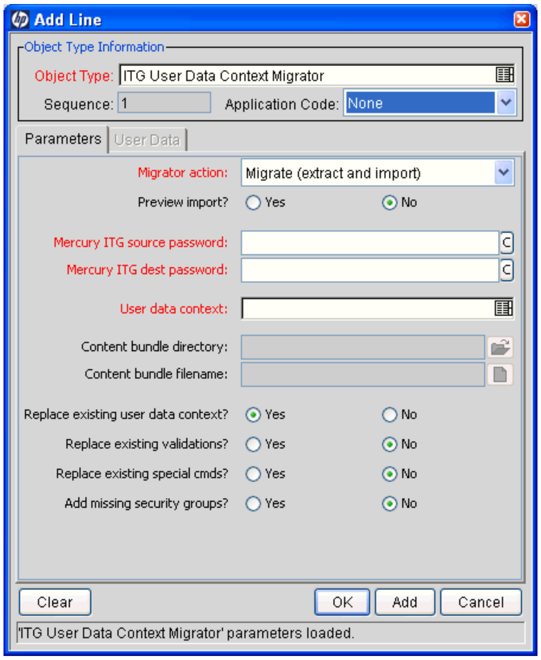
'ITG Special Command Migrator' parameters loaded.

For information about the fields in this migrator, see *Defining Entity Migrators* on page 182.

## User Data Context Migrator

The User Data Context Migrator (*Figure 8-15*) contains the additional option **Replace Existing special cmds?** If the validation to be migrated references PPM Center special commands that already exist in the target PPM Center instance, you can decide whether or not to replace them. This includes both parent and child special commands. (The default value is **No.**) Regardless of their values, PPM Center automatically re-creates special commands that are missing from the destination instance.

Figure 8-15. User Data Context Migrator



Object Type: ITG User Data Context Migrator

Sequence: 1

Application Code: None

Migrator action: Migrate (extract and import)

Preview import? ○ Yes ● No

Mercury ITG source password:

Mercury ITG dest password:

User data context:

Content bundle directory:

Content bundle filename:

Replace existing user data context? ● Yes ○ No

Replace existing validations? ○ Yes ● No

Replace existing special cmds? ○ Yes ● No

Add missing security groups? ○ Yes ● No

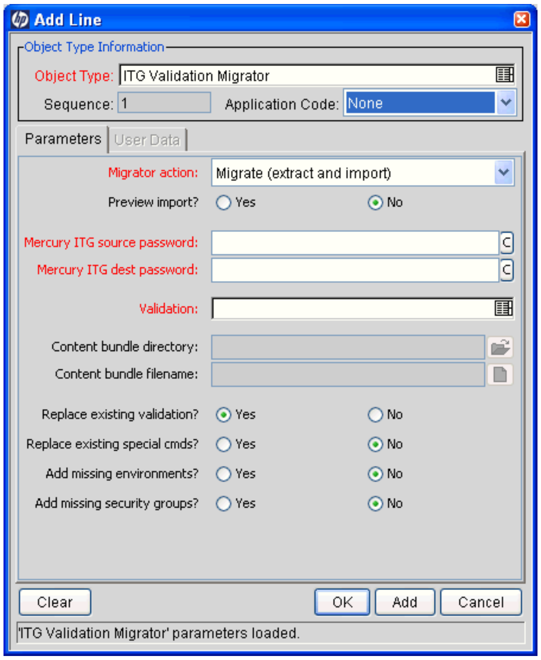
'ITG User Data Context Migrator' parameters loaded.

For information about most of the fields in the User Data Context Migrator, see *Defining Entity Migrators* on page 182.

## Validation Migrator

The Validation Migrator is shown in *Figure 8-16*.

Figure 8-16. Validation Migrator



Object Type: ITG Validation Migrator

Sequence: 1 Application Code: None

Migrator action: Migrate (extract and import)

Preview import? ○ Yes ● No

Mercury ITG source password:

Mercury ITG dest password:

Validation:

Content bundle directory:

Content bundle filename:

Replace existing validation? ● Yes ○ No

Replace existing special cmds? ○ Yes ● No

Add missing environments? ○ Yes ● No

Add missing security groups? ○ Yes ● No

'ITG Validation Migrator' parameters loaded.

This migrator contains the following two additional import behavior options:

- **Replace existing special cmds?** If the validation to be migrated references PPM Center special commands that already exist in the target PPM Center instance, you can decide whether or not to replace them. This includes both special commands directly referenced by the validation, and also special commands referenced by these special commands. (The default value is **No.**) Regardless of their values, PPM Center automatically re-creates special commands that are missing from the destination instance.

- **Add missing environments?** If the validation to be migrated references environments or environment groups that do not exist in the target PPM Center instance, you can decide whether or not to create them (assuming that the option has been marked Yes). However, only the environment header information and user data are transferred. Application codes and extension-specific environment tabs are not transferred. The default value is **No.**

  Similarly, environment group application code information is not transferred. If an environment group already exists in the destination instance, it is not updated with environments that were added in the source instance. After migration is complete, if the migrator has created any environments, confirm and complete environment data manually.

For information about the controls in this migrator, see *Defining Entity Migrators* on page 182.

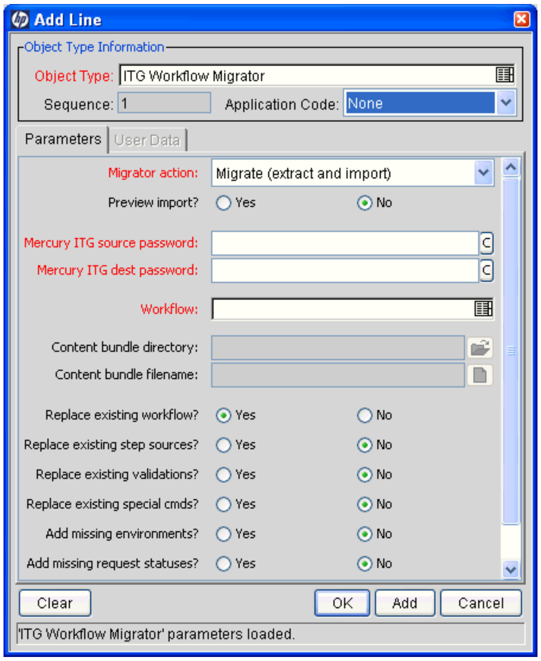
### Configuration Considerations

Validation values can also carry context-sensitive user data. When migrating validation values that have such fields, you should manually set up the user data configuration in the destination instance before migration begins.

## Workflow Migrator

The Workflow Migrator is shown in *Figure 8-17*.

Figure 8-17. Workflow Migrator



Object Type Information

Object Type: ITG Workflow Migrator

Sequence: 1 Application Code: None

Migrator action: Migrate (extract and import)

Preview import? ○ Yes ● No

Mercury ITG source password:

Mercury ITG dest password:

Workflow:

Content bundle directory:

Content bundle filename:

Replace existing workflow? ● Yes ○ No

Replace existing step sources? ○ Yes ● No

Replace existing validations? ○ Yes ● No

Replace existing special cmds? ○ Yes ● No

Add missing environments? ○ Yes ● No

Add missing request statuses? ○ Yes ● No

'ITG Workflow Migrator' parameters loaded.

This migrator provides the following additional import behavior options:

- **Replace existing special cmds?** If the workflow to be migrated references PPM Center special commands that already exist in the target PPM Center instance, you can replace them. This includes special commands that the workflow references directly, as well as special commands that these special commands reference. Special commands in validations that the workflow references are also migrated.

The default value is **No.** Regardless of the value, any special commands missing from the destination instance are created automatically.

- **Replace existing step sources?** If the workflow to be migrated references workflow decision and execution step sources that exist in the target PPM Center instance, you can choose to replace them or leave them in place. However, if workflows in the destination instance are using the existing step sources, you cannot change certain options (such as **Workflow Scope, Validation,** and **Decision Type**), even if you set **Replace Existing Step Sources?** to **Yes.**

- **Add missing environments?** If the workflow to be migrated references environments or environment groups that do not exist in the target PPM Center instance, you can create the environments or environment groups. However, only the environment header information and user data are transferred. Application codes and extension-specific **Environment** tabs are not transferred. The default value is **No.**

  Similarly, environment group application code information is not transferred. If an environment group exists in the destination instance, it is not updated with environments added to the source instance. If the migrator has created environments, then after migration, make sure that you confirm and complete the environment data manually.

- **Add missing request statuses?** If the workflow to be migrated references request status values that do not exist in the target PPM Center instance, you can create the status values. The default value is **No.**

For information about controls in this migrator, see *Defining Entity Migrators* on page 182.

### Configuration Considerations

The Workflow Migrator also transfers the following information:

- Subworkflows that the workflow steps reference
- Special commands that the command steps reference
- Workflow step sources that the workflow steps reference
- Validations that the parameters or workflow step sources reference
- Environments and environment groups that the workflow steps reference
- Environments that the environment groups referenced by workflow steps reference

- Environments that validations reference
- Special commands that validations reference
- Special commands that the workflow step sources reference
- Special commands referenced by other special commands referenced elsewhere
- Security groups that the workflow steps reference
- Request statuses that the workflow steps reference
- Notifications that the workflow steps reference
- Notification intervals that notifications reference
- Security groups that notifications reference
- Ownership group information for the workflow and workflow steps

If a notification in a workflow uses a notification interval that does not exist in the destination instance, the migrator creates this notification interval. The workflow migrator does not replace existing notification intervals in the destination instance.

The Workflow Migrator transfers entity restriction references to object types, but does not create an object type. If the referenced object type does not exist in the destination instance, the migrator discards the reference and records the event in its execution log.

The Workflow Migrator transfers references to request types, but does not create request types. If the referenced request type does not exist in the destination instance, the migrator discards the reference and records the event in its execution log.

If there are circular references between workflows and request types, you may have to migrate either a workflow or request type twice:

- A new request type referring to a new workflow is migrated. Because the new workflow does not exist in the destination instance, all references to that workflow are dropped in transit.
- The new workflow is migrated.
- The new request type is migrated again. This time, because the referenced workflow exists, the references are preserved.

### Replacing an Existing Workflow

There are some restrictions on using the Workflow Migrator to make changes to a process that is already in use (by requests or package lines). These restrictions help to ensure that migration does not damage these existing requests or package lines.

Specifically, workflow migration cannot succeed unless the migrator logic finds a workflow step that corresponds to each step in the existing workflow. The following conditions are used to match workflow steps between instances:

- The step source (the particular decision, execution, or condition) of a workflow step is used to match workflow steps. If the step source is not identical, then two workflow steps do not match.

- If both the incoming and existing workflows assign a unique name to each workflow step, these workflow step names are used in combination with the step source to assess the match.

- If a workflow step name is repeated within either workflow, the step sequence is used instead, in combination with the step source, to assess the match.

  The Workflow Migrator cannot handle a single change in which both the names of existing workflow steps and the step sequence of existing workflow steps have changed.

  To change both the names and step sequences of a workflow:

  - Change step names, but do not change any step sequences. Migrate the changed workflow.

  - Change step sequences, but do not change any step names. Migrate the changed workflow a second time.

  Because of this matching restriction, each open request is on the same process step following the migration as it was before the migration. The migration might have changed the name of this step, but it has not transitioned request workflows.

It is important to note that the migrator does not prevent the removal of outgoing transitions from workflow steps. Therefore, avoid “stranding” open requests at a workflow step that will be deprecated. When deprecating a process step, remove incoming transitions, but leave at least one outgoing transition from the step. This lets open requests move forward. The execution log for the migration contains a table that lists old and new workflow steps.

HP recommends that you use the **Preview import** mode first when you replace an existing workflow, and inspect this table of matched workflow steps before you run the workflow migration in non-preview mode.

### Deprecating a Workflow

When the changes to a workflow are extensive, you can deprecate the existing workflow and bring the changes into the production instance as a new workflow. One advantage of implementing the changes as a new workflow is simplicity, since the new workflow is not required to contain all of the steps of the old workflow for backward compatibility.

To bring a new workflow into a production instance:

1. Rename the existing workflow and disable it in production.

   Disabling the workflow removes it from lists of workflow options when new requests are created. Requests that are in process continue to follow the old workflow until they close, unless each is manually shifted to the new process and transitioned to an appropriate point in the process. Existing defaulting rules and other configurations also continue to refer to the old workflow, regardless of the name change.

2. Migrate the new version of the workflow into the production instance, under the original name.

   Because the production instance no longer contains a workflow by this name, the migrator treats it as a new workflow.

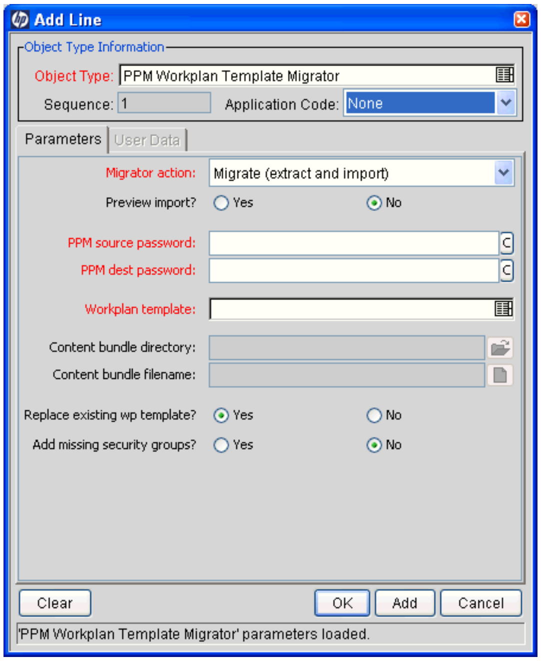
3. After the migration, you can update defaulting rules in request types to reference this new workflow.

   You can do this manually, or by migrating in versions of the request types that refer to the new workflow by its original name.

## Work Plan Template Migrator

You can define work plan templates in a development or testing instance of HP Project Management, and then use the Work plan Template Migrator (*Figure 8-18*) to migrate them to production after testing is completed.

Figure 8-18. Work Plan Template Migrator



Add Line

Object Type Information

Object Type: PPM Workplan Template Migrator

Sequence: 1 Application Code: None

Parameters | User Data

Migrator action: Migrate (extract and import)

Preview import? ○ Yes ● No

PPM source password:

PPM dest password:

Workplan template:

Content bundle directory:

Content bundle filename:

Replace existing wp template? ● Yes ○ No

Add missing security groups? ○ Yes ● No

'PPM Workplan Template Migrator' parameters loaded.

The Work Plan Template Migrator migrates the following:

- Header information such as work plan template name and list of owners (users)
- Work plan (hierarchy of tasks and task information)
- References to assigned resource groups or users (by reference only—security groups are not treated as dependent objects)

The Work Plan Template Migrator does not transport secondary objects (for example, validations) as dependencies.

# 9 Migrating Instances

In This Chapter:

# Overview of Instance Migration

Each PPM Center instance consists of a file system and an Oracle database, which can exist on Windows or UNIX machines. You can migrate PPM Center using one of the following methods:

- Copy an entire PPM Center instance (server file system and database schemas) and move it to another location. If you are moving the copied instance to a different machine, you must have a new license key for it.
- Migrate the PPM Server to a different machine, but maintain the existing database schemas. Migrating the server requires a new license key.
- Migrate the database schemas, but maintain the existing PPM Server. Migrating only the database schema does not require a new license key.

Enterprise environments typically have multiple PPM Center instances (for example, development, test, and production). The following sections address the simplest multiple-instance configuration, which consists of a development instance (DEV) and a production instance (PROD). Each is set up on a different machine. You can extend the migration steps to support all of the instances used at your site.

## Copying an Instance to Create a New Instance

To create additional PPM Center instances from an existing production (PROD) instance, clone the PROD instance.

To move from a single active instance to multiple instances:

1. Copy the PROD instance to DEV.

   This includes the file system, database, and license information.

2. Configure any changes to HP products in the DEV instance.

   This includes creating or modifying entities such as workflows, object types, request types, validations, security groups, and environments.

3. From the PROD instance, configure a package workflow to import the configuration data from the DEV instance.

4. Migrate data from the DEV instance into the PROD instance.

### Running the Installation Script Twice to Create Two Instances

You can set up multiple instances as you first install and set up PPM Center. Configure one instance as the DEV instance, and the other as the PROD instance. This saves you from having to copy data from one instance into another later.

### Migrating Document Management (Optional)

If your source machine has document management installed and integrated with PPM Center, see the *Document Management Guide and Reference* for information about how to migrate document management.

## Preparing to Migrate

Before you can begin to migrate an entire instance to a different machine, you must obtain a new license key and stop the PPM Server, as described in the following sections.

### Obtaining a New License Key

PPM Center is licensed based on the computer that hosts the PPM Server. If you plan to migrate the PPM Server to a different machine, you must obtain a new license key for the target machine. If you plan to migrate only the database schema, you do not need a new license key.

To obtain a new license key:

1. Gather the following information:

   - PPM Center version number
   - Machine IP address
   - Operating system (Windows or UNIX)
   - Server purpose (development, test, or production)

2. Go to the HP-Mercury support site (support.Mercury.com).

3. In the right panel of the HP Customer Support page, click **Submit a License Key Request.**

   The License Request home page opens.

4. In the list of products, to the right of **PPM**, click **Get License.**

5. Enter the required information, and then click **Submit.**

## Stopping the PPM Server

To ensure that you do not lose transactions, reports, or logs, stop the PPM Server before you migrate any part of a PPM Center instance. For information about how to stop the server, see *Starting and Stopping the PPM Server* on page 66.

# Migrating the PPM Server

Before you migrate the PPM Server, make sure that the target machine meets the requirements described in the document *System Requirements and Compatibility Matrix.*

## Migrating to a Windows Machine

To migrate the PPM Server to a Windows machine:

1. Obtain a new license key for the target server, as described in *Obtaining a New License Key* on page 213.

2. Stop the PPM Server.

   For information on how to stop the server, see *Starting and Stopping the PPM Server* on page 66.

3. Migrate the PPM Center file system:

   a. Make a compressed file of the entire `<PPM_Home>` directory.

   b. Copy the compressed file to the target machine, and then extract the file contents.

4. Migrate the PPM Center database schema.

   For information about how to migrate the database schema, see *Migrating the Database Schemas* on page 219.

5. Reconfigure the PPM Server in the target location, as follows:

   a. Run the `kConfig.sh` script, which is located in the `<PPM_Home>/bin` directory.

      The `kConfig.sh` script starts the server configuration utility, which then displays the values for each server parameter from the previous server configuration.

   b. Browse through all server configuration parameters, and make the following updates:

      - Update all parameters that refer to the DNS name or IP address of the old server to instead refer to the DNS name or IP address of the new server.

        - `BASE_URL` specifies the Web location (top directory name) of the PPM Server.

        - `RMI_URL` specifies the port on which the PPM Server listens to initiate RMI client/server communication. (This must be a unique port, distinct from the Web server, SQL*Net, and the HTTP or HTTPS ports.)

      - Update all parameters that reference a specific directory on the old server to instead reference the corresponding directory on the new server. These parameter include:

        - `ORACLE_HOME` specifies the home directory for the Oracle client tools on the PPM Server machine.

        - `BASE_PATH` specifies the full path to the directory where the PPM Server is installed.

        - `ATTACHMENT_DIRNAME` specifies the absolute pathname of the directory where attached documents are to be stored. This directory must give read/write access to Web browsers and, if the system includes an external Web server, exist outside the directory tree.

        - `SERVER_TYPE_CODE` specifies the operating system on which the PPM Server is installed. Because you are placing the server on a computer running Windows, make sure you update the value to `Windows`.

        - `SERVER_NAME` specifies the name of the PPM Server instance. If multiple PPM Servers are running on the same machine, this name must be unique for each server. If the server is running

Windows, this name must match the name of the Windows service name.

   c. To implement your changes, run the kUpdateHtml.sh script from the <*PPM_Home*>/bin directory.

6. Create a Windows service for the new PPM Center instance, as follows:

   a. Navigate to the <*PPM_Home*>/bin directory.

   b. Run kConfig.sh as follows:

      i. Select **Configure Windows services.**

      ii. Select **Change service parameters and refresh the services.**

      iii. Specify a value for the JAVA_HOME parameter.

      iv. Click **Finish.**

7. Start the new PPM Server.

For information about how to start the server, see *Starting and Stopping the PPM Server* on page 66.

## Migrating to a UNIX Machine

To migrate the PPM Server to a UNIX machine:

1. Obtain a new license key, as described in *Obtaining a New License Key* on page 213.

2. Stop the PPM Server.

   For information about how to stop the PPM Server, see *Starting and Stopping the PPM Server* on page 66.

3. Migrate the PPM Center file system as follows:

   a. On the PPM Server host machine, navigate to the parent of the <*PPM_Home*> directory.

   b. Using an archiving utility (such as Tar or Zip), create an archive file of the entire <*PPM_Home*> directory.

      For example, if the <*PPM_Home*> directory is named "PPM," run the following TAR command:

```
$ tar cf mitg71.tar PPM
```

c. Using FTP in binary mode, copy the archive file to the target machine. Put the archive file in the parent of the new *<PPM_Home>* directory.

d. Extract the archive file as follows:

```
$ tar xf mitg71.tar
```

This creates the new PPM Server directory structure. A directory named PPM is created automatically.

4. Migrate the PPM Center database schema.

   For information about how to migrate the database schema, see *Migrating the Database Schemas* on page 219.

5. Reconfigure the PPM Server in the target location as follows:

   a. Run the `kConfig.sh` script, which is located in the *<PPM_Home>*`/bin` directory.

      The `kConfig.sh` script starts the server configuration utility, which then displays the values for each server parameter from the previous server configuration.

   b. Browse through all server configuration parameters, and make the following updates:

      - Update all parameters that refer to the DNS name or IP address of the old server to instead refer to the DNS name or IP address of the new server.

        - `BASE_URL` specifies the Web location (top directory name) of the PPM Server.

        - `RMI_URL` specifies the port on which the PPM Server listens to initiate RMI client/server communication. (This must be a unique port, distinct from the Web server, SQL*Net, and the HTTP or HTTPS ports.)

- Update all parameters that reference a specific directory on the old server to instead reference the corresponding directory on the new server. These parameters include:

    - `ORACLE_HOME` specifies the home directory for the Oracle client tools on the PPM Server machine.

    - `BASE_PATH` specifies the full path to the directory where the PPM Server is installed.

    - `ATTACHMENT_DIRNAME` specifies the absolute pathname of the directory where attached documents are to be stored. This directory must give read/write access to Web browsers and, if the system includes an external Web server, exist outside the directory tree.

- `SERVER_TYPE_CODE` specifies the operating system on which the PPM Server is installed. Because you are placing the server on a computer running UNIX, make sure you update the value to `UNIX`.

- `SERVER_NAME` specifies the name of the PPM Server instance. If multiple PPM Servers are running on the same machine, this name must be unique for each server.

c. To implement your changes, run the `kUpdateHtml.sh` script from the `<`*`PPM_Home`*`>/bin` directory.

6. Place the new `license.conf` file into `<`*`PPM_Home`*`>/conf`.

7. Start the new PPM Server.

   For information on how to start the server, see *Starting and Stopping the PPM Server* on page 66.

# Migrating the Database Schemas

This section provides the procedures used to migrate the PPM Center database schemas from one database to another.

### Export and Import Tools

Using incompatible versions of export and import tools causes errors in instance migration. Make sure that the export and import tools you use are either the same version, or the export tool version is earlier than the import tool version.

### If You Use the Extension for Oracle E-Business Suite

If you have HP Deployment Management Extension for Oracle E-Business Suite, you must consider the location of your Primary Object Migrator Host when migrating the PPM Center database schema, because HP Object Migrator might reside in the same database, or even the same schema, as PPM Center.

Migrating the schema does not require migrating the HP Object Migrator instance because the integration method in PPM Center can be refreshed to use the existing HP Object Migrator installation. If Object Migrator shares a database with PPM Center, and you intend to migrate it as well as PPM Center, the destination database must support Object Migrator. (For more information, see the *HP Object Migrator Guide.*)

Unless PPM Center and HP Object Migrator share the same schema, the migration of Object Migrator is completely separate from the migration of PPM Center, and should be completed before you migrate the PPM Center database. Contact HP-Mercury support (support.Mercury.com) for instructions on how to perform this migration.

If PPM Center and HP Object Migrator share the same schema and you want to migrate both, you must coordinate the migration activities. Contact HP-Mercury support (support.Mercury.com) for instructions.

Regardless of the configuration, refresh the integration definition after you migrate the PPM Center schemas.

To migrate the database schemas:

Exporting and importing the database schemas involves using the `exp` and `imp` commands. The variables for these commands are described in *Export Command Variables* on page 223 and *Import Command Variables* on page 224.

1. Stop the PPM Server.

   For information about how to stop the PPM Server, see *Starting and Stopping the PPM Server* on page 66.

2. Export the PPM Center database schema to a file by running the `exp` command as shown in the following example:

   ```
   $ORACLE_HOME/bin/exp USERID=system/password@db FILE=<Export_
   Filename> OWNER=<PPM_Username> LOG=c:/export_knta_710.log
   ```

3. Export the RML schema.

4. Create the new PPM Center database schema, as follows:

   a. Run the `CreateKintanaUser.sql` script (located in the `<PPM_Home>/install_710/mitg710/system` directory) from SQL*PLUS as the SYSTEM user.

      Example:

      ```
      SQL> @CreateKintanaUser.sql PPM_User PPM_Password Data_
      Tablespace Index_Tablespace TEMP_Tablespace Clob_
      Tablespace
      ```

   b. Run the `GrantSysPrivs.sql` script (located in the `mitg710/sys` directory) from SQL*PLUS as the SYS DBA user.

   For more information, see *Preliminary Database Tasks* on page 293.

5. Create the new PPM Center RML database schema.

   To create a new, empty RML database schema in the target database, run the `CreateRMLUser.sql` script (located in the `mitg710/sys` directory) from SQL*PLUS as the SYSTEM user.

   Example:

   ```
   SQL> @CreateRMLUser.sql Rml_User Rml_Password Rml_data_
   tablespace Rml_temp_tablespace
   ```

6. To import data from the export file that you created earlier into the new empty PPM Center database schema, run the `imp` command, as illustrated in the following example:

```
$ ORACLE_HOME/bin/imp USERID=<system>/<Password>@<DB>
FILE=<Export_Filename> IGNORE=Y TOUSER=<New_PPM_Username>
FROMUSER=<PPM_Username> LOG=c:/import_knta_710.log
```

7. Import the RML export file.

8. Create the RML-related packages in the RML schema:

```
sqlplus <rml_user>/<rml_password>@<SID> @rmlpackages
```

9. Grant privileges to the PPM Center RML database schema:

➤ You can find the following scripts in the `<PPM_Home>/install_710/rml directory.`

   - To set up the permissions between the two:

   ```
   sqlplus <ppm_user>/<ppm_password>@SID
   @RMLSetupInPPMSchema.sql <rml_user>
   ```

   - To create synonyms to PPM Center objects in the RML schema:

   ```
   sqlplus <rml_user>/<rml_password>@SID
   @RMLSetupInRMLSchema.sql <ppm_user>
   ```

10. Configure the database schema to allow appropriate access to rebuild optimizer statistics.

➤ If PPM Center and HP Object Migrator share the same database schema, the PPM Center database schema is referred to as the PPM Center account, and the HP Object Migrator schema is referred to as the HP Object Migrator account.

To provide the necessary grants and permissions to the PPM Center user, run the `GrantSysPrivs.sql` script, as follows:

As SYS DBA:

```
SQL> @GrantSysPrivs.sql <ppm_user>
```

11. If the Extension for Oracle E-Business Suite is in use and HP Object Migrator resides in the same schema as PPM Center, complete the HP Object Migrator migration.

For assistance, contact HP-Mercury support (support.Mercury.com).

12. If you are using the Extension for Oracle E-Business Suite, refresh the Primary Object Migrator Host definition.

If you encounter problems, recompile invalid objects, as follows:

To validate any invalid PPM Center database objects generated during link regeneration, run the `RecompileInvalid.sql` script, which is located in the `<PPM_Home>/install_710` directory. Run this script from SQL*PLUS connected as the new PPM Center database schema account.

13. Reconfigure the PPM Server to connect to the new database schema as follows:

    a. Start the configuration utility by running the `kConfig.sh` script located in the `<PPM_Home>/bin` directory.

    b. Update the server configuration parameters, which are described in *Server Configuration Parameters* on page 225.

    If you edit the `server.conf` files manually, be sure to run the `kUpdateHTML.sh` script after you complete the edit.

14. Start the PPM Server (see *Starting and Stopping the PPM Server* on page 66).

# Troubleshooting Instance Migrations

This section describes common problems that you might encounter as you migrate PPM Center instances.

## PPM Server Does Not Start

If you cannot start the PPM Server, check the `serverLog.txt` file (located in the `<PPM_Home>/server/<server_name>/logs` directory) for error messages. If the `serverLog.txt` file contains no error messages, increase the server debug level to see if any additional helpful information is written to the log.

To increase the server debug level:

1. Open the `logging.conf` file (located in the `<PPM_Home>/conf` directory) in a text editor such as Notepad.

2. Set the value of the `SERVER_DEBUG_LEVEL` parameter to `HIGH,` and then save and close the `logging.conf` file.

3. Run the `kUpdateHtml.sh` script.

4. Rerun the `kStart.sh` script, and then recheck the `serverLog.txt` file to see if it contains any additional information.

5. Open the `logging.conf` file.
6. Restore the default value of the `SERVER_DEBUG_LEVEL` parameter.

   Restoring the default value ensures that the file system does fill up with unnecessary information recorded in the `serverLog.txt` file(s).

7. Run the `kUpdateHtml.sh` script.

## Server Starts, but You Cannot Access Applications

If the Web browser accessing the PPM Center URL generates a “Not Found” or an “Access Denied” error, check the `server.conf` file and the external Web server (if one exists) to ensure that the PPM Server installation directory is specified correctly.

If the PPM Server has recently been upgraded and the URL has changed, make sure that any saved links to the previous PPM Center URL (for example, existing requests) are updated to point to the new URL.

# Export Command Variables

*Table 9-1* provides descriptions of the variables in the following export (`exp`) command example:

```
$ ORACLE_HOME/bin/exp USERID=<system>/<password>@<db>
FILE=<Export_Filename> OWNER=<PPM_Username> LOG=c:/export_knta_
710.log
```

The `exp` command might have a different name on Windows.

Table 9-1. Export command variables

| Variable | Description |
|---|---|
| password | Password of the system user on the Oracle database. |
| db | Database connect string. |
| Export_Filename | Name of the file that is to contain the export. The filename must use the `dmp` extension (for example, `kntaExport.dmp`). |
| PPM_Username | Name of the PPM Center database schema to export. |

# Import Command Variables

*Table 9-2* provides descriptions of the variables in the following import (`imp`) command example:

```
$ ORACLE_HOME/bin/imp USERID=<system>/<Password>@<DB>
FILE=<Export_Filename> IGNORE=Y TOUSER=<New_PPM_Username>
FROMUSER=<PPM_Username> LOG=c:/import_knta_710.log
```

➤ The `imp` command might have a different name on Windows.

Table 9-2. Import command variables

| Variable | Description |
|---|---|
| Password | Password for the SYSTEM user on the database. |
| DB | Database connect string. |
| Export_Filename | Name of the file that contains the export file. The filename must use the `dmp` file extension (for example, `kntaExport.dmp`). |
| New_PPM_ Username | Name of the new PPM Center database schema. |
| PPM_Username | Name of the database schema that was previously exported. |

# A Server Configuration Parameters

In This Appendix:

# Overview of Configuration Parameters

This appendix lists and describes the PPM Server configuration parameters located in three files in the `<PPM_Home>` directory:

- `server.conf`
- `logging.conf`
- `LdapAttribute.conf`

For more information about the PPM Server directory structure, see Appendix B, *Server Directory Structure and Server Tools,* on page 277.

## Determining the Correct Parameter Settings

For most PPM Center installations, the default parameter values are optimal. Considerations detailed in the parameter descriptions can help you determine under what circumstances you might want to change the parameter settings.

## Required Parameters

In the tables in this appendix, an asterisk in the **Parameter** column indicates that the parameter is required to set up a PPM Server. Two asterisks in this column indicates that the parameter is required based on the condition of another parameter. For example, the `KINTANA_LDAP_ID` parameter is only required if the `AUTHENTICATION_MODE` parameter is set to `LDAP`.

In a server cluster configuration, required parameters must be set for the primary server. Secondary servers inherit the parameter values from the primary server. To override the inherited value, set the parameter to the value you want in the appropriate secondary server section of the `server.conf` file. For more information about setting up PPM Servers in a server cluster configuration, see *Configuring a Server Cluster* on page 115.

For information about how to specify your own parameters, see *Defining Custom and Special Parameters* on page 71.

## Directory Path Names

Use forward slashes (/) to enter directory paths in the `server.conf` file, regardless of the operating system used. PPM Center automatically uses the appropriate path separators to communicate with Microsoft Windows. HP recommends that you not use backslashes (\) to enter directory paths in the `server.conf` file.

## Categories of Performance-Related Parameters

Some parameters are labeled with category names (for example, `DAYS_TO_KEEP_INTERFACE_ROWS` is labeled as a cleanup parameter). For information about these performance-related categories, see *Adjusting Server Configuration Parameters* on page 168.

# Server Configuration Parameters

The `server.conf` file contains the values of all of the server parameters applied when the server configuration utility (`kConfig.sh` script) was last run.

HP recommends that you *not* modify the `server.conf` file directly. Instead, use the `kconfig.sh` utility, which provides a graphical interface that you can use to change the server configuration parameter values.

To edit the `server.conf` file:

1. Stop the PPM Server.
2. Run the `kConfig.sh` script.

   After you finish specifying configuration parameter values, the `kConfig.sh` script automatically runs the `kUpdateHtml.sh` script to regenerate the server.conf file and apply your changes.

   For information about the `kUpdateHtml.sh script, see` *kUpdateHtml.sh* on page 286 .

3. Restart the PPM Server.

   To see a list of the `server.conf` parameter values on an active PPM Server, run the Server Configuration report. For information about how to run the Server Configuration report, see *Running Server Reports from the Admin Tools Window* on page 138 and *Running Server Reports from the Command Line* on page 141.

*Table A-1* provides descriptions of the configuration parameters in the `server.conf` file. The parameter names listed in the table are shortened versions of the actual names, all of which start with the string `com.kintana.core.server`. For example, the full name of the `CLIENT_TIMEOUT` parameter is `com.kintana.core.server.CLIENT_TIMEOUT`.

Table A-1. Server configuration parameters (page 1 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| ALLOW_SAVE_REQUEST_DRAFT | Allows requests to be saved without automatically submitting them in the standard interface. | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |
| APPLET_KEY_CLEANUP_INTERVAL | The frequency with which the *ENABLE_APPLET_KEY_CLEANUP* thread runs.<br>See also *DAYS_TO_KEEP_APPLET_KEYS* on page 231. | Default: 21600<br>(seconds) |
| *ATTACHMENT_DIRNAME | Absolute pathname of the directory where attached documents are to be stored. This directory must:<br>■ Give read/write access to Web browsers<br>■ Be outside the directory tree if the system includes an external Web server<br>In a server cluster, all servers must be able to access and share the specified directory. | Example:<br>`c:\ppm\eon\attachments` |
| AUTHENTICATE_REPORTS | If set to `TRUE,` access to all reports requires user authentication. (A user must provide a PPM Center user login ID). | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |
| *AUTHENTICATION_MODE | User authentication method.<br>To specify multiple modes, use a comma-delimited list of valid values. | Default: `PPM`<br>Valid values: `PPM, LDAP, NTLM, SITEMINDER` |
| AUTO_COMPLETE_SHORT_TYPE_MAX_ROWS | Determines the maximum number of rows to retrieve from the database for short type auto-completion lists. | Default: 500 |

Table A-1. Server configuration parameters (page 2 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| AUTOCOMPLETE_ STATUS_ REFRESH_RATE<br>Category: Scheduler/ services/thread | Interval at which the command status is refreshed to provide a list of values in an auto-complete. | Default: 5 (seconds) |
| BASE_LOG_ DIRECTORY | Points to the “logs” directory directly under the directory specified by the *BASE_PATH parameter.<br>In a server cluster, all servers must be able to access and share the specified directory. | Example: `com.kintana.core.server.BASE_LOG_DIR=C:\PPM\eon\logs` |
| *BASE_PATH | Full path to the directory where the PPM Server is installed. | The default value depends on the operating system platform.<br>Example:<br>`C:\PPM\eon\` |
| *BASE_URL | Web location (top directory name) of the PPM Server. | Example: `http://www.mydomain.com:8080` |
| BUDGET_IN_ WHOLE_DOLLARS | Determines whether budget values are expressed in whole dollars. | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |
| CLIENT_TIMEOUT<br>Category: Timeout | The value of this parameter determines the frequency (in minutes) with which the PPM Workbench interface sessions sends a message to the PPM Server that indicates the client is still active.<br>Under normal operation, do not change this value. | Default: 5 |
| CLOSE_BROWSER_ ON_APPLET_EXIT | This parameter determines whether the client browser closes after the user quits the PPM Workbench. | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |

Table A-1. Server configuration parameters (page 3 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| COLLAPSE_ SINGLE_PAGE_ MODULE | If set to `TRUE,` a module that consists of only one page is displayed with the name: "page_name (module_name)," and not as a menu item. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |
| COMMANDS_ CLEANUP_ INTERVAL | The value of this parameter determines the frequency with which the *ENABLE_APPLET_KEY_ CLEANUP* thread (page 244) runs. See also *DAYS_TO_KEEP_ COMMANDS_ROWS* on page 231. | Default: 16200 |
| **CONC_LOG_ TRANSFER_ PROTOCOL<br>Required if *ORACLE_ APPS_ENABLED* = TRUE | Transfer protocol used to transfer concurrent request logs and patching README files. | Default: FTP<br>Valid values: FTP, SCP |
| **CONC_REQUEST_ PASSWORD<br>Required if *ORACLE_ APPS_ENABLED* = TRUE | Encrypted password of the concurrent request user. | Encrypted example: `fnd` |
| **CONC_REQUEST_ USER<br>Required if *ORACLE_ APPS_ENABLED* = TRUE | Valid user on the Oracle system that can be used to retrieve concurrent request output files.<br>Set the retrieval method (FTP or SCP). *See **CONC_LOG_ TRANSFER_PROTOCOL* on page 230. | Example: `applmgr` |
| CONCURRENT_ REQUEST_WATCH_ DOG_INTERVAL | The value of this parameter determines the frequency with which the *TURN_ON_CONCURRENT_ REQUEST_WATCH_DOG* thread (page 277) runs. | Default: 30 |
| COST_ CAPITALIZATION_ ENABLED | Determines whether cost capitalization is enabled. | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |

Table A-1. Server configuration parameters (page 4 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| COST_RATE_RULE_ UPDATE_ INTERVAL_ MINUTES | This service updates the planned and actual costs of open projects when new cost rate rules are added or existing cost rate rules are modified. | Default: 60 (minutes) |
| COST_ROLLUP_ INTERVAL_ MINUTES | The Cost Rollup Service asynchronously recalculates and rolls up cost (project and program budget costs) asynchronously as part of a service. To set up the service, set the *ENABLE_COST_ROLLUP_SERVICE* parameter to TRUE and use this parameter to specify the delay between consecutive runs of the service. | Default: 300 (minutes)<br>Valid values: any positive integer |
| COST_UPDATE_ SERVICE_ INTERVAL | The cost update service is used to update cost information with modified cost rate rules or currency exchange rates. This parameter determines the frequency with which the service is invoked. | Default: 3600 (seconds) |
| DASHBOARD_DB_ CONNECTION_ PERCENTAGE | The percentage of *MAX_DB_ CONNECTIONS* that the PPM Dashboard module can use for database connections. | Default: 25<br>Valid values: Integer between 0 and 100 |
| DASHBOARD_ PAGE_AUTO_ REFRESH_ DISABLED | When set to TRUE, disables the PPM Dashboard auto-refresh feature (in the Personalize Page view). | Default: FALSE<br>Valid values: TRUE, FALSE |
| DATE_ NOTIFICATION_ INTERVAL | Interval at which the PPM Server is to check to determine whether date-based notifications are pending, and to send them. | Default: 60 (minutes) |
| DAYS_TO_KEEP_ APPLET_KEYS | The value of this parameter determines the number of days applet keys are retained in the KNTA_ APPLET_KEYS table. | Default: 1 |
| DAYS_TO_KEEP_ COMMANDS_ROWS | The value of this parameter determines how many days records are kept in the prepared commands tables before they are cleaned up. | Default: 1 |

Table A-1. Server configuration parameters (page 5 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| DAYS_TO_KEEP_ INTERFACE_ROWS<br>Category: Open Interface | The value of this parameter determines the number of days to keep records of all interfaces. | Default: 5 |
| DAYS_TO_KEEP_ LOGON_ATTEMPT_ ROWS<br>Category: Cleanup | Number of days to keep records of all logon attempts. | Default: 14 |
| **DB_ CONNECTION_ STRING<br>(Required if RAC is used) | Oracle RAC (Real Application Clusters) service name. | Example:<br>`K92RAC` |
| DB_LOGIN_ TIMEOUT<br>Category: Timeout | The amount of time that the PPM Server is to continue to try to log on to the database (acquire the JDBC connections that make up the connection pool) before reporting that the database is unavailable. | Default: 30000<br>(milliseconds) |
| *DB_PASSWORD | Password for the database schema that contains the PPM Center tables. | Example:<br>`#!#`*`<password>`*`#!#` |
| *DB_USERNAME | Name of the database schema that contains the PPM Center tables. | Example:<br>`knta` |
| DEBUG_MESSAGE_ CLEANUP_ INTERVAL | Use this parameter to specify the run frequency for the *ENABLE_DEBUG_ MESSAGE_CLEANUP* thread. | Default: 21600 |
| DEFAULT_ COMMAND_ TIMEOUT<br>Category: Timeout | Determines the number of seconds the PPM Server tries to run commands before it times out. | Default: 90 |

Table A-1. Server configuration parameters (page 6 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| DEFAULT_PAGE_ SIZE | The default number of work plan lines that can be loaded into the Work Plan page for all new users. This setting indicates whether to use the fast setting or the slow setting (rather than indicating a specific size).<br>In new installations, this defaults to the slow connection setting. HP recommends that the system administrator review this setting after installation.<br>If your system has mostly LAN users (fast connections), set this to use the fast setting. If your system has mostly WAN/VPN users (slow connections) or mixed usage, set this to use the slower setting. | Default: 50 |
| DEFAULT_PAGE_ SIZE_OPTION | Controls the initial type of page size (low, high, or custom) selected for the Edit My Profile page. | Default: LOW_ PAGE_SIZE<br>Valid values: |
| DEFAULT_ REQUEST_ SEARCH_ORDER_ BY_ID | Affects the **Sort By** field on the Search Requests page. The default value is TRUE, which sorts the search results based on Request ID. When set to FALSE, search results are returned unsorted. | Default: TRUE |
| DEMAND_FIELDS_ CACHE_SIZE | Specifies the size of the demand set fields cache in number of demand set. | Default: 10 |
| DEMAND_FIELDS_ CACHE_TIMEOUT | The timeout for the demand set fields cache, expressed in seconds. | Default: 360000 (seconds) |
| DEPLOY_BASE_ PATH | Specifies the deployment destination.<br>**Note:** HP recommends that you leave the default value unless the PPM Server directory is renamed. | Default: server |

Table A-1. Server configuration parameters (page 7 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| DIST_ENGINE_MONITOR_SLEEP_TIME | Used in release distribution. Specifies the number of milliseconds the monitor waits between checking existing result listener. Use this parameter to adjust the amount of time the monitor sleeps between checks.<br>**Note:** HP recommends that you not change this value. It does not affect performance. | Default: 5000 (milliseconds) |
| DOCUMENT_CLEANUP_SERVICE_DELAY | Determines the run frequency of a server thread that checks for documents no longer attached to a PPM Center entity, and removes them from the PPM Center file system.<br>This parameter is associated with the parameter *ENABLE_DOCUMENT_CLEANUP_SERVICE* on page 236. | Default: 1440 (minutes) |
| EMAIL_NOTIFICATION_CHECK_INTERVAL<br>Category: Scheduler/services/thread | Determines the frequency (in seconds) with which the PPM Server checks for pending email notifications. | Default: 20 |
| EMAIL_NOTIFICATION_SENDER | Email address of the default sender of email notifications.<br>This sender receives any error messages associated with email notifications. | Example: `sender@ppm.com` |
| ENABLE_APPLET_KEY_CLEANUP | Periodically removes old records from the database table `KNTA_APPLET_KEYS`. (These are temporary, system-generated keys used for one-time access to the system—for example, if a user wants to open the PPM Workbench.) This parameter is associated with the frequency parameter *APPLET_KEY_CLEANUP_INTERVAL*. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |

Table A-1. Server configuration parameters (page 8 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| ENABLE_ COMMANDS_ CLEANUP | If set to TRUE, a service periodically removes old records from the KNTA_ PREPARED_COMMANDS and KNTA_ PREPARED_COMMAND_STEPS database tables. These tables contain temporary data used during command processing.<br>This parameter is associated with the *COMMANDS_CLEANUP_INTERVAL* frequency parameter and the *DAYS_ TO_KEEP_COMMANDS_ROWS* parameter. | Default: TRUE<br>Valid values: TRUE, FALSE |
| ENABLE_ CONCURRENT_ REQUEST_ UPDATES | This parameter is related to requests in HP Demand Management. When this is set to true, multiple users can change the same request simultaneously. Request data such as notes, new references and new table entries are always saved. Conflicting changes that cannot be saved are displayed to the user as differences. | Default: TRUE<br>Valid values: TRUE, FALSE |
| ENABLE_COST_ RATE_RULE_ UPDATE_SERVICE | This service updates the planned and actual costs of open projects when new cost rate rules are added or existing cost rate rules are modified. | Default: TRUE<br>Valid values: TRUE, FALSE |
| ENABLE_COST_ ROLLUP_SERVICE | PPM Center recalculates and rolls up cost (project and program budget costs) asynchronously as part of a service. To set up the service, set the this parameter to TRUE, and then use the *COST_ROLLUP_INTERVAL_ MINUTES* parameter to specify the frequency with which the service performs its calculations. | Default: FALSE<br>Valid values: TRUE, FALSE |
| ENABLE_COST_ UPDATE_SERVICE | If set to TRUE, updates cost information with modified cost rate rules or currency exchange rates. The *COST_UPDATE_SERVICE_ INTERVAL* parameter setting determines how often the service is invoked. | Default: FALSE<br>Valid values: TRUE, FALSE |

Table A-1. Server configuration parameters (page 9 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| ENABLE_ DASHBOARD_ LOADING_ MESSAGE | If set to TRUE, the PPM Dashboard displays a message as it loads a page. | Default: FALSE<br>Valid values: TRUE, FALSE |
| ENABLE_DB_ SESSION_ TRACKING<br>Category: Low-level debug | If set to TRUE, enables a stack trace to be reported in the PPM Center DB Server Reports, which you can use to track the exact line of code used to request a database connection. | Default: FALSE<br>Valid values: TRUE, FALSE |
| ENABLE_DEBUG_ MESSAGE_ CLEANUP | Periodically removes old records from the KNTA_DEBUG_MESSAGES database table, which can collect a lot of temporary data. Use the *DEBUG_ MESSAGE_CLEANUP_INTERVAL* parameter to specify the run frequency for this thread. Use the **HOURS_TO_ KEEP_DEBUG_MESSAGE_ROWS* parameter to specify how long records stay in the debug table before they are cleaned up. | Default: TRUE<br>Valid values: TRUE, FALSE |
| ENABLE_ DIRECTORY_ CLEANUP | Determines whether the Directory Cleanup Service is enabled. | Default: TRUE<br>Valid values: TRUE, FALSE |
| ENABLE_ DOCUMENT_ CLEANUP_SERVICE | Enables a server thread that periodically checks for documents that are no longer attached to a PPM Center entity, and removes those it finds from the PPM Center file system. This parameter is associated with the parameter DOCUMENT_CLEANUP_ SERVICE_DELAY, which determines the frequency with which this thread runs. | Default: FALSE<br>Valid values: TRUE, FALSE |
| ENABLE_ EXCEPTION_ ENGINE<br>Category: Scheduler/ services/thread | If set to TRUE, enables the exception engine, which runs a process to determine whether active projects are running on time.<br>Set the exception engine interval with ***EXCEPTION_ENGINE_WAKE_UP_ TIME* on page 242. | Default: TRUE<br>Valid values: TRUE, FALSE |

Table A-1. Server configuration parameters (page 10 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| ENABLE_FINANCIAL_METRICS_UPDATE_SERVICE | Determines whether the financial metrics update service is enabled. This service calculates net present value (NPV) and return on investment (ROI) for HP Portfolio Management. | Default: `TRUE`<br>Valid values:<br>`TRUE, FALSE` |
| ENABLE_FLS_PENDING_DENORM | Managing field-level security is very computationally expensive, so whenever the security settings at the field level are updated, the PPM Server performs a number of calculations that allow live security checks in performance. The server performs these calculations asynchronously, by a separate server thread.<br>This parameter enables the thread that performs the calculations. You can use the following associated parameters to specify the time at which this thread runs:<br>■ *FLS_PENDING_DENORM_WAKE_UP_TIME*<br>■ *FLS_PENDING_DENORM_DAY_OF_WEEK*<br>■ *FLS_PENDING_DENORM_WEEK_INTERVAL* | `Default: TRUE`<br>Valid values:<br>`TRUE, FALSE` |
| ENABLE_FX_RATE_UPDATE_SERVICE | Recalculates cost after financial exchange (FX) rates change. | Default: `TRUE`<br>Valid values:<br>`TRUE, FALSE` |
| ENABLE_INTERFACE_CLEANUP | Periodically removes old records from the database open interface tables. You can use the associated parameter *INTERFACE_CLEANUP_INTERVAL* to specify the run frequency for this thread, and the parameter *DAYS_TO_KEEP_INTERFACE_ROWS* to specify how long to keep records in the interface tables. | Default: `TRUE`<br>Valid values:<br>`TRUE, FALSE` |

Table A-1. Server configuration parameters (page 11 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| ENABLE_JDBC_ LOGGING<br>Category: High-level debug | Determines whether to enable JDBC logging, which records SQL run against the database, the time required to run the SQL, and the time to retrieve the results. This information is recorded in `jdbc.System_ Name.log` in the server log directory.<br>This parameter is useful in debugging system performance problems.<br>You can set this parameter in the PPM Workbench interface without stopping the system (**Edit > Settings**). | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |
| ENABLE_LOGIN_ COOKIE | If set to `TRUE,` the **Remember my logon** checkbox options are displayed on the logon page, and a cookie is placed on the client browser to maintain a record of the user logon information.<br>**Remember my logon** sets a cookie on the local machine that lets a user log on to PPM Center later, without entering logon information. You can also view reports through notification links, and so on, without logging on. This cookie is removed only if the user clicks **Sign Out** (or clears cookies, or the cookie expires). If a user closes the browser window without signing off, the cookie is not cleared.<br>To disable this function, change the parameter value to `FALSE`. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |

Table A-1. Server configuration parameters (page 12 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| ENABLE_LOGON_ATTEMPTS_CLEANUP | Periodically removes old records from the KNTA_LOGON_ATTEMPTS database table, which contains records of all logon attempts. You can use the *LOGON_ATTEMPTS_CLEANUP_INTERVAL* parameter to specify the run frequency of this thread. Use the *DAYS_TO_KEEP_LOGON_ATTEMPT_ROWS* parameter to specify how long records stay in the logon table before they are removed. | Default: TRUE<br>Valid values:<br>TRUE, FALSE |
| ENABLE_OVERVIEW_PAGE_BUILDER | This parameter is provided for backward compatibility if you have customized "overview pages." If you do not have customized "overview pages," leave the default value (FALSE). | Default: FALSE<br>Valid values:<br>TRUE, FALSE |
| ENABLE_PENDING_ASSIGNMENTS_CLEANUP | Periodically checks for duplicate rows in the KNTA_PENDING_ASSIGNMENTS table. This parameter is related to the "work item breakdown" service. If a work item is updated more than once between runs of the work item breakdown service, the KNTA_PENDING_ASSIGNMENTS table contains duplicate rows. This thread removes the duplicates. Use the *PENDING_ASSIGNMENTS_CLEANUP_INTERVAL* parameter to specify the run frequency for this thread. | Default: TRUE<br>Valid values:<br>TRUE, FALSE |
| ENABLE_PENDING_EV_UPDATES_CLEANUP | If set to TRUE, removes duplicate rows in the Pending EV Updates table. Use this parameter in conjunction with PENDING_COST_EV_UPDATES_SERVICE. | Default: TRUE<br>Valid values:<br>TRUE, FALSE |

Table A-1. Server configuration parameters (page 13 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| ENABLE_PORTLET_ FULL_RESULTS_ SORTING | Lets you optimize portlet results for either accuracy or performance. Typically, portlet results are limited to 200 items. If this parameter is set to `FALSE`, the first 200 results are sorted and displayed. (This improves performance but may not yield the most accurate results). If the parameter is set to `TRUE`, all results are sorted and the top 200 displayed (yielding accurate results, with poor performance). | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |
| ENABLE_ PROGRAM_ SUMMARY_ CONDITION_ ENGINE | If set to `TRUE`, enables the automatic update of program health indicators. | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |
| ENABLE_PROJECT_ LAUNCH_FROM_ ACTION_MENU | If set to `TRUE`, allows users with the required permission to open the PPM Workbench as a stand-alone application. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |
| ENABLE_PROJECT_ PV_UPDATE_ SERVICE | If set to `TRUE`, updates planned values for tasks. The interval at which the service runs is determined by the *PROJECT_PV_UPDATE_ INTERVAL_MINUTES* parameter setting. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |
| ENABLE_QUALITY_ CENTER_METRICS_ SYNC | If set to `TRUE`, enables a service that synchronizes PPM Center with HP Quality Center. | Valid values: `TRUE, FALSE` |
| ENABLE_QUERY_ BUILDER | If set to `TRUE`, enables the advanced “query builder” capability for searching HP Demand Management requests. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |
| ENABLE_ QUICKLIST_ UPDATE | Controls the visibility of the **Update** button on the Quick List. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |

Table A-1. Server configuration parameters (page 14 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| ENABLE_REFERENCE_UPDATE_SERVICE | When set the TRUE, enables the service that updates references between entities. See also *REFERENCE_UPDATE_INTERVAL*. | Default: TRUE<br>Valid values:<br>TRUE, FALSE |
| ENABLE_RESOURCE_COST_UPDATE_SERVICE | Determines whether costs are recalculated. If set to TRUE, the *RESOURCE_COST_UPDATE_SERVICE_DELAY* parameter determines how frequently costs are recalculated. | Valid values:<br>TRUE, FALSE |
| ENABLE_RESOURCE_POOL_ROLLUP_SERVICE | If set to TRUE, enables resource pool rollup (between child resource pool and parent resource pool). | Default: TRUE<br>Valid values:<br>TRUE, FALSE |
| ENABLE_SHARED_LOCK_CLEANUP | If set to TRUE, enables the shared lock cleanup service, which cleans up any entries left in the shared lock table after a server crash. | Default: TRUE<br>Valid values:<br>TRUE, FALSE |
| ENABLE_SQL_TRACE<br>Category: High-level debug | Determines whether performance statistics for all SQL statements run are placed into a trace file.<br>The SQL trace facility generates the following statistics for each SQL statement:<br>■ Parse, run, and fetch counts<br>■ CPU and elapsed times<br>■ Physical reads and logical reads<br>■ Number of rows processed<br>■ Misses on the library cache<br>■ User name under which each parse occurred<br>■ Each commit and rollback<br>This parameter corresponds to the **Enable DB Trace Mode** checkbox in the Server Settings dialog box. | Default: FALSE<br>Valid values:<br>TRUE, FALSE |

Table A-1. Server configuration parameters (page 15 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| ENABLE_ STATISTICS_ CALCULATION<br>Category: Database statistics | Whether to automatically collect statistics for the cost-based optimizer.<br>By default, statistics are rebuilt every Sunday at 1 a.m. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |
| ENABLE_TASK_ ACTUAL_ROLLUP_ SERVICE | Enables a service that asynchronously rolls up actuals entered through HP Time Management or the My Tasks portlet. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |
| ENABLE_TIME_ SHEET_ NOTIFICATIONS_ SERVICE | If set to `TRUE,` enables notification on time sheets. | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |
| ENABLE_ TIMESTAMP_ LOGGING | If set to `TRUE,` specifies that a timestamp is written into the log for each line of debugging text that corresponds to actions you have performed. The timestamp can help you locate information in the server log files about events that occurred at a specific time, or to determine how much time elapsed between specific logged statements.<br>**Note:** Including the timestamp adds text to each logged statement, which bloats the log file and can make it more difficult to read. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |
| ENABLE_UNICODE_ FOR_ TELNETCLIENT | If set to `TRUE,` enables Unicode support in Telnet. | Valid values: `TRUE, FALSE` |
| ENABLE_WEB_ ACCESS_LOGGING | If set to `TRUE,` tells Tomcat (the Web server provided with JBoss) to log all http requests received. This parameter has no default.<br>**Note:** If enabled on a busy system, Web access logging generates many log files. | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |
| ENABLE_WEB_ SERVICES | To use the PPM Center Web services interface, set this to `TRUE.` | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |

Table A-1. Server configuration parameters (page 16 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| **EXCEPTION_ENGINE_WAKE_UP_TIME<br>Required if *ENABLE_EXCEPTION_ENGINE* = TRUE<br>Category: Scheduler/services/thread | Time at which the exception engine process runs. | Default: 1 (that is, 1:00 a.m.)<br>Valid values: 1 through 24 |
| EXTERNAL_WEB_PORT | If you are using an external Web server to serve PPM Center clients, you must configure this parameter as an available port that can communicate with the PPM Server. This port receives AJP (Apache JServ Protocol) requests from the external Web server. AJP is the standard protocol used for communication between a Web server and an application server.<br>**Note:** If you are using an external Web server, you must still configure the standard PPM Center **HTTP_PORT*. This port is used internally by PPM Center reports. There is no need to make it accessible to the network. | Valid value: Any available port number |
| FAIL_EXECUTIONS_ON_STARTUP | If the PPM Server stops while command executions are running, those executions are interrupted and the parent entities (Package Lines, Releases, Requests, and so on) are assigned the status "in progress." This parameter tells the server that, after it restarts, it must check for any entities that have "in progress" status and that have no executions running (that is, executions that were interrupted). The server sets the internal status of those entities to FAILED, with a visible status of "Failed (Interrupted)." | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |

Table A-1. Server configuration parameters (page 17 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| FINANCIAL_ METRICS_UPDATE_ INTERVAL | Determines how often financial metrics are updated. Financial metrics calculates the net present value (NPV) and ROI. | Default: 1440 (minutes) |
| FLS_PENDING_ DENORM_DAY_OF_ WEEK | Determines the day of the week to run the fls_pending_denorm service. | Default: 7<br>Valid values: An integer between 1 and 7 (inclusive), where 1 represents Sunday and 7 represents Saturday |
| FLS_PENDING_ DENORM_WAKE_ UP_TIME | Determines the time of day the fls pending denorm service is run. | Default: 21<br>Valid values: Number between 1 and 24, inclusive |
| FLS_PENDING_ DENORM_WEEK_ INTERVAL | Determines the number of weeks between each fls_pending_denorm service run. | Default: 4<br>Valid values: Number between 1 and 4, inclusive |
| FX_RATE_UPDATE_ SERVICE_ INTERVAL_ MINUTES | This service updates the planned and actual costs of open projects, budgets, and benefits when new currency exchange rates rules are added or existing exchange rates are modified. | Default: 120 (minutes) |
| GRAPHICAL_WF_ ENABLE | If set to TRUE, makes links to view graphical workflow available on submitted requests. | Default: TRUE<br>Valid values: TRUE, FALSE |
| GROUP_PRIVATE_ PUBLIC_PAGES | If set to TRUE, after a user selects **Dashboard > Personalize Dashboard,** the PPM Dashboard displays two sections: **Private** and **Shared.** The **Private** section contains single pages and groups, the **Shared** section contains modules. | Default: FALSE<br>Valid values: TRUE, FALSE |

Table A-1. Server configuration parameters (page 18 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| GZIP_ENCODING_ ENABLED | Determines whether HTTP responses are compressed before they are sent to PPM Center HTML clients. If set to TRUE, then textual HTTP responses are compressed using GZIP compression (if the requesting browser supports GZIP).<br>By default, this is set to TRUE to improve the responsiveness of the PPM Center standard (HTML) interface, because less overall data is carried across the Internet between the client and the PPM Server.<br>If all PPM Center clients have fast network access to the PPM Server, then consider setting this parameter to FALSE to reduce the overhead of compressing and decompressing responses. | Default: TRUE<br>Valid values: TRUE, FALSE |
| HIGH_PAGE_SIZE | The recommended number of work plan lines to load into the Work Plan page if the user is connected through a fast connection such as a LAN. | Default: 100 |
| *HOURS_TO_KEEP_ DEBUG_MESSAGE_ ROWS<br>Category: Cleanup | The number of hours that rows in the KNTA_DEBUG_MESSAGES table are to be kept.<br>For high-volume PPM Center installations, a large number of rows may be generated in this table. For such installations, decrease this value accordingly.<br>See also *ENABLE_DEBUG_ MESSAGE_CLEANUP* on page 236. | Default: 48 |

Table A-1. Server configuration parameters (page 19 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| *HTTP_PORT | Port to use to communicate with the built-in HTTP server.<br>If PPM Center is in stand-alone mode (that is, it is not integrated with an external Web server), then PPM Center clients must have access to the `HTTP_PORT`.<br>If PPM Center is integrated with an external Web server, then client HTTP traffic is routed through the *EXTERNAL_WEB_PORT*. However, even in that case, the PPM Server still uses the **HTTP_PORT* internally to run reports. However, in this case, it is not necessary to make the **HTTP_PORT* externally accessible to PPM Center clients (and thus, the port need not be exposed outside of the PPM Server). | Default: 8080<br>Valid values: Unique port greater than 1024 and distinct from the Web server, SQL*Net, and RMI ports. |
| I18N_CARET_ DIRECTION | Caret position on input fields (for example, text fields).<br>If unspecified, same as *I18N_SECTION_DIRECTION*. | Valid values: `ltr`, `rtl`<br>(left to right, right to left) |
| I18N_ENCODING | Character encoding to be used on all HTML pages in the PPM Center standard interface. | Default: UTF-8 |
| I18N_LAYOUT DIRECTION | Default layout direction of HTML pages in the PPM Center standard interface. | Default: ltr<br>Valid values: `ltr`, `rtl`<br>(left to right, right to left) |
| I18N_REPORT_ HTML_CHARSET | HTML character set to use in PPM Center reports.<br>Must map to the character set specified in *I18N_REPORTS_ENCODING*. | Default: `WINDOWS-1252`<br>Valid values (Windows): windows-hebrew |

Table A-1. Server configuration parameters (page 20 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| I18N_REPORTS_ENCODING | Character encoding to use to generate reports in PPM Center.<br>Recommended for Windows systems: IW8MSWIN1255 | Valid values: Any encoding algorithm that Oracle can interpret. |
| I18N_SECTION_DIRECTION | Layout direction of custom sections (for example, request detail sections).<br>If unspecified, same as *I18N_LAYOUT DIRECTION*. | Valid values: ltr, rtl |
| INSTALLATION_CURRENCY | Determines the currency symbol displayed. | Default: 93 |
| *INSTALLATION_LOCALE | Language and country code of the PPM Center installation.<br>The language code must match the PPM Center installation language. | Default: en_US<br>Example:<br>`de_DE` |
| INTERFACE_CLEANUP_INTERVAL | The value of this parameter determines the frequency with which the *ENABLE_INTERFACE_CLEANUP* thread runs. | Default: 11700 |
| JAVA_CLASSES_LOC | Specifies the location of the JRE classes. | Example:<br>`C:/Java/j2sdk1.4.2_08/jre/lib/classes.zip` |
| JAVA_COMPILER | The server sets the (read-only) value of this parameter at runtime. | Default: internal |
| JAVA_PLUGIN_XPI_PATH | Specifies the Web location for downloading the cross-platform Java plug-in installer for Firefox browsers. | Example:<br>`http://java.sun.com/update/1.4.2/j2re-1_4_2_06-windows-i586.xpi` |
| JDBC_DEBUGGING | Specifies the SQL_DEBUG property on the PPM Dashboard. | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |

Table A-1. Server configuration parameters (page 21 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| *JDBC_URL<br>**Note:** For Oracle RAC (Real Application Clusters), this parameter must contain the host and port information for all databases to which the PPM Server will connect. | Locator for the database containing the PPM Center database schema.<br>Must be specified correctly for PPM Server to communicate with the database.<br>Format:<br>`jdbc:oracle.thin:@<hostname>:<port>:<SID>`<br>Where:<br>`<hostname>` is the DNS name or IP address of the system running the database.<br>`<port>` is the port used by SQL*Net to connect to the database. Refer to the database entry in the `tnsnames.ora` file. Default is 1521.<br>`<SID>` is the database system ID. | Default: 1521<br>Example:<br>`jdbc:oracle:thin:@DBhost.domain.com:1521:SID` |
| JVM_OPTIONS | This parameter is for HP internal use only. Do not alter its value unless directed to do so by HP-Mercury support for PPM Center. | |
| JSP_RECOMPILE_ENABLED | Determines whether changes to JSP files are picked up on a running server, thereby quickly making them visible.<br>If set to `FALSE`, JSP files are checked for changes only the first time they are accessed, with the result that changes are visible only after the server is restarted.<br>If you expect JSP pages to be updated regularly, set to `TRUE`. The PPM Server detects JSP changes without restarting. | Default: `FALSE` on production systems, `TRUE` on development systems<br>Valid values:<br>`TRUE, FALSE` |
| **KINTANA_LDAP_ID<br>Required if *AUTHENTICATION_MODE* = LDAP | PPM Center account on the LDAP server.<br>Used by the PPM Server to bind to the LDAP server. | Example:<br>`uid=admin, ou=dev` |

Table A-1. Server configuration parameters (page 22 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| **KINTANA_LDAP_PASSWORD<br>Required if *AUTHENTICATION_MODE = LDAP | PPM Center password on the LDAP server.<br>The PPM Server configuration utility automatically encrypts this password. To manually edit this value, surround the encrypted password with `#!#` delimiters. | Default: `#!##!#`<br>Example:<br>`#!#<password>#!#` |
| KINTANA_LOGON_FILENAME | Used in non-HTML notification, this parameter value is specified with the filename (to be appended to the URL), which points to the logon page.<br>**Note:** HP recommends that you not reset this parameter. | Example:<br>`kintanaHome.html` |
| KINTANA_SERVER_DIRECTORY | Specifies the server directory location. You define this value if you are using a multiple-server (clustered) setup. | Default:<br>`/server/kintana/` |
| KINTANA_SERVER_LIST | The server sets the (read-only) value of this parameter at runtime. | Example:<br>`aeon!rmi://ice:27099/KintanaServer` |
| *KINTANA_SERVER_NAME | Name of the PPM Server instance.<br>If multiple PPM Servers are running on the same machine, this name must be unique for each server.<br>If the server is running Windows, this name must match the name of the Windows service name. | Default: kintana |
| *KINTANA_SESSION_TIMEOUT | The time to elapse before the PPM Server terminates a user session (in the PPM Workbench or standard interface) because of inactivity. A value of 0 denotes no timeout. | Default: 120<br>(minutes)<br>Valid values: 10 through 720 |

Table A-1. Server configuration parameters (page 23 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| LDAP_BASE_DN | The base distinguished name on the LDAP server. Each LDAP URL must specify a base distinguished name (DN), which is used in place of the LDAP_BASE_DN server configuration parameter.If the URLs provided for *LDAP_URL_FULL* do not have a DN value, PPM Center uses the value set for LDAP_BASE_DN. | Default:<br>Valid values: |
| **LDAP_GROUP_ RECURSION_LIMIT<br>Required if **AUTHENTICATION_ MODE* = LDAP | Number of levels of subgroups to traverse when importing users from groups. | Default: 15 |
| **LDAP_SSL_PORT<br>Required if **AUTHENTICATION_ MODE* = LDAP | SSL port number on the LDAP server.<br>If not specified, all transactions are carried over the port specified by the ***LDAP_URL* parameter. | Default: 636 |
| **LDAP_URL<br>Required if **AUTHENTICATION_ MODE* = LDAP | Comma-delimited list of LDAP URLs, which the PPM Server queries in the order specified.<br>If no port number is specified, the default port number 389 is used.<br>**NOTE:** The `LDAP_URL_FULL` parameter supersedes the `LDAP_URL` parameter. That is, if a value is set for both in the `server.conf` file, `LDAP_ URL_FULL` is used. If the URLs specified for `LDAP_URL_FULL` do not have a DN value, the value set for `LDAP_BASE_DN` is used. | Example:<br>`ldap:// ldap.theurl.com:389`<br>Example:<br>`ldap:// 10.100.102.199: 389` |

Table A-1. Server configuration parameters (page 24 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| LDAP_URL_FULL | PPM Center uses this parameter to handle multiple domains during LDAP authentication. The values for the parameter include a space-separated (not comma-separated) list of full LDAP URLs. Each LDAP URL must specify a base DN.<br>**NOTES:**<br>■ To specify a space character inside a URL, use the URL encoding scheme, and replace the space with ”%20.” For example, if you have an organizational unit called ”My Org Unit,” then specify ”My%20Org%20Unit” in the LDAP URL.<br>■ The `LDAP_URL_FULL` parameter supersedes the `LDAP_URL` parameter. That is, if a value is set for both in the `server.conf` file, `LDAP_URL_FULL` is used. If URLs specified for `LDAP_URL_FULL` do not have a DN value, the value set for `LDAP_BASE_DN` is used. | Example:<br>`com.kintana.core.server.LDAP_URL_FULL=ldap://<host.yourdomain.com/CN=Users,DC=yourdomain,DC=com ldap://host.yourdomain.com/OU=Users2,DC=yourdomain,DC=com` |

Table A-1. Server configuration parameters (page 25 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| LOCAL_IP<br>**Note:** Setting this parameter resolves the following potential problems:<br>- If the parameter is set to the IP address of the machine running the firewall, clients inside the firewall can connect, but clients outside cannot, because they have no route to the host.<br>- If the parameter is set to the name of the machine running the firewall, clients inside the firewall can connect, but clients outside cannot, because they cannot resolve the hostname.<br>- If the parameter is set to an IP address that is different from the machine running the firewall, clients outside the firewall can connect, but clients inside the firewall cannot, because the address is not translated between a different IP address to the IP address on the machine running the firewall. | Name of the machine running the firewall. This parameter applies only to RMI traffic for the PPM Workbench.<br>Before you set this parameter, register the external IP address on the external DNS server, and then specify the name of the machine running the firewall as the LOCAL_IP value.<br>If you set this up correctly:<br>- Client A running inside the firewall connects to the internal DNS server and the machine name resolves to an IP address.<br>- Client B running outside the firewall connects to an external DNS server and the machine name resolves to a different IP address.<br>Both clients can then connect, each to a different IP address. | Example:<br>`10.1.101.64` |

Table A-1. Server configuration parameters (page 26 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| LOGON_ ATTEMPTS_ CLEANUP_ INTERVAL | The value of this parameter determines the run frequency of the *ENABLE_LOGON_ATTEMPTS_ CLEANUP* thread. | Default: 18000 |
| *LOGON_TRIES_ INTERVAL | Time interval during which logon attempts are monitored. | Default: 1<br>(minutes) |
| LOW_PAGE_SIZE | The recommended number of work plan lines to load into the Work Plan page if the user is connected through a slow connection such as a WAN. | Default: 50 |
| MAINFRAME_JOB_ WATCH_DOG_ ENABLED | If you are using HP Deployment Management to integrate with a mainframe system, then you must enable this "watch dog" thread. WhenHP Deployment Management submits a job to the mainframe, this thread polls the mainframe system to determine what state the job is in, and when it is completed. This parameter is associated with the frequency parameter *MAINFRAME_JOB_ WATCH_DOG_INTERVAL*. | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |
| MAINFRAME_JOB_ WATCH_DOG_ INTERVAL | This parameter determines the frequency with which the *MAINFRAME_JOB_WATCH_DOG_ INTERVAL* thread runs. | Default: 30<br>(minutes) |
| MAX_DB_ CONNECTION_ IDLE_TIME<br>Category: Database connection | Amount of time that an unused database connection stays open before it is closed and removed from the pool. | Default: 60<br>(minutes) |
| MAX_DB_ CONNECTION_ LIFE_TIME<br>Category: Database connection | Amount of time that a database session is held open before it is closed and removed from the pool.<br>Some Oracle cleanup operations that should be run periodically occur only at the end of database sessions. Therefore, do not keep database sessions open for the life of the PPM Server. | Default: 1440<br>(minutes) |

Table A-1. Server configuration parameters (page 27 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| MAX_DB_ CONNECTIONS<br>Category: Database connection | The number of connections the PPM Server has to the database. Each user does not get their own connection. The server uses connection pooling, so it only opens a new database connection if there are no connections available in the pool.<br>After this number is reached, user sessions queue for the next available database connection.<br>The PPM Server rarely requires more than 25 database connections. | Default: 60 |
| *MAX_EXECUTION_ MANAGERS<br>Category: Scheduler/ services/thread | Maximum number of concurrent executions allowed to run on the server. If your system is heavily loaded, decreasing this may help reduce load, but may also delay execution of tasks.<br>If your organization processes a high volume of packages, you may require more execution managers. | Default: 15 |
| MAX_PPM_DB_ CONNECTIONS | Determines the maximum number of connections that the Database Pool is to maintain. When this number is reached, subsequent requests for database connection must wait until a database becomes available. | Default: 45 |
| *MAX_LOGON_ TRIES | Maximum number of logon attempts in the time interval specified by **LOGON_TRIES_INTERVAL*. | Default: 0 |
| MAX_PAGE_SIZE | The absolute maximum number of work plan lines that can be loaded into the Work Plan page. Use this parameter to prevent excessive load on the server from excessive queries, and to prevent users from getting themselves into low performance situations. | Default: 500 |

Table A-1. Server configuration parameters (page 28 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| *MAX_RELEASE_ EXECUTION_ MANAGERS<br>Category: Scheduler/ services/thread | Number of command executions that can run in a release distribution simultaneously.<br>Organizations processing a high volume of packages may require a larger number of release execution managers. | Default: 15<br>Valid values: Number greater than 1 |
| MAX_STATEMENT_ CACHE_SIZE | Maximum number of prepared statements cached per database connection.<br>Part of the database connection pool settings. | Default: 50<br>Valid values: Integer greater than 0 |
| *MAX_WORKER_ THREADS<br>Category: Scheduler/ services/thread | Worker threads are spawned by the scheduler to run scheduled tasks. This specifies the maximum number of scheduled tasks (for example, reports or request commands) that can be simultaneously active on the server.<br>If the PPM Server is heavily loaded, specify a lower value to reduce the server workload. If there are many pending tasks, and additional capability is available on the server, set a higher value to improve performance. | Default: 10 |

Table A-1. Server configuration parameters (page 29 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| MSP_PROJECT_<br>CUSTOM_FIELD | Microsoft Project includes a set of pre-defined text fields (Text1, Text2, Text3, and so on) that users can use to store whatever they want. By default, PPM Center uses the Text30 field at the project level in Microsoft Project to store information about the PPM Center project with which the Microsoft Project file is associated.<br>If you already use the Text30 field in Microsoft Project, you can use this parameter to specify the Microsoft Project field for PPM Center to use.<br>**Warning:** Do not change this parameter value once the Microsoft Project integration has been put into use. | Default: `Text30` |
| MULTICAST_<br>CLUSTER_NAME | Unique name of a PPM Server cluster.<br>Do not configure two clusters with the same name running on the same subnet. | Example:<br>`http://wwwserver.mydomain.com/ppm` |
| MULTICAST_DEBUG | Whether or not incoming and outgoing multicast messages are to be logged to the PPM Server log. | Default: `FALSE`<br>Valid values:<br>`TRUE, FALSE` |
| MULTICAST_IP | Multicast IP address. | Default:<br>225.39.39.244<br>Valid values:<br>224.0.0.0 through 239.255.255.255 |
| MULTICAST_<br>LEASE_MILLIS | Interval at which the PPM Server sends out heartbeats. | Default: 20000 (milliseconds) |
| MULTICAST_PORT | Multicast IP port. | Default: 9000 |
| NOTIFICATIONS_<br>CLEANUP_PERIOD<br>Category: Cleanup | Interval to clean up previously-sent notifications. | Default: 7<br>(days) |

Table A-1. Server configuration parameters (page 30 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| OPTIMIZATION_ ITERATION_ MULTIPLIER | The number of algorithmic iterations that the optimization engine is to run. The more iterations, the more time is given to finding an optimal portfolio. Although the default is adequate in most instances, complex cases can benefit from more iterations.<br>**Note:** This parameter also affects generation of the Efficient Frontier curve. | Default: 100 (iterations) |
| OPTIMIZER_ NUMBER_OF_ TIMESHIFTS | Maximum number of periods the optimizer can shift start dates forward. This does not affect manually-shifted HP Portfolio Management entities.<br>If you allow a new start date for a project, the optimizer can start the project any time between the original start date and six months beyond that date. | Default: 6 (months) |
| ORACLE_APPS_ ENABLED | Determines whether PPM Center is to be integrated with Oracle applications.<br>You must set this parameter to `TRUE` for installations running HP Object Migrator or HP GL Migrator. | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |
| ORACLE_APPS_ VERSION | The version of Oracle applications used. | Default: R11 |
| ORACLE_DB_ VERSION | The server sets this read-only parameter value during startup. | Example:<br>`10.1.0.3.0` |
| *ORACLE_HOME | Full path to the Oracle home directory on the PPM Server.<br>The Oracle_Home/network/admin directory must contain the correct TNS names (or a file containing the names: tnsnames.ora) required to connect to the PPM Center database schema. | Example:<br>`d:/orant` |

Table A-1. Server configuration parameters (page 31 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| PACKAGE_LOG_DIR | In a server cluster, If you have overridden the default value for this parameter to refer to a different directory, then all servers in the cluster must be able to access and share the directory. | Default: Same default value as the *BASE_LOG_DIRECTORY* parameter |
| PAGE_PDF_EXPORT_DISABLED | Lets you enable (or disable) the functionality to export PPM Dashboard pages as PDF files. | Default: FALSE<br>Valid values: TRUE, FALSE |
| *PASSWORD_EXPIRATION_DAYS | Default expiration period of passwords for new users.<br>A value of 0 indicates no expiration. | Default: 0<br>(days)<br>Valid values:<br>0 through 366 |
| *PASSWORD_REUSE_RESTRICTION_DAYS | The number of days to restrict the use of an old password after a new password is set.<br>The value 0 indicates no restriction. | Default: 0<br>Valid values: 0 through 2192 |
| PENDING_ASSIGNMENTS_CLEANUP_INTERVAL | Determines the frequency with which the *ENABLE_PENDING_ASSIGNMENTS_CLEANUP* thread runs. | Default: 14400 |
| PENDING_COST_EV_UPDATE_SERVICE_DELAY | The number of seconds to wait after completion of the Pending Cost EV Update service before restarting the service. | Default: 30<br>Valid values: Number greater than 0 |
| PENDING_COST_EV_UPDATE_SERVICE_ENABLED | Enables a service that asynchronously applies external updates to the Pending Cost EV Updates service when updates cannot be made immediately. | Default: FALSE<br>Valid values: TRUE, FALSE |
| PENDING_EV_UPDATES_CLEANUP_INTERVAL | Specifies the interval at which to run pending earned value updates. | Default: 3600 (seconds) |

Table A-1. Server configuration parameters (page 32 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| PGA_AGGREGATE_TARGET | Determines the maximum physical memory Oracle can use for working areas for all processes together.<br>See also *WORKAREA_SIZE_POLICY* on page 269. | Maximum number of MB that can be dedicated to working Oracle processes. |
| PM_CAN_ROLLUP_ACTUALS_ON_SAVE | If set to `TRUE`, lets the user change the mode on HP Project Management's Enter Actuals page. The resulting options are either rollups calculated during save processing, or rollups are deferred (and rolled up using the associated service). | Valid values: `TRUE, FALSE` |
| PM_NUM_EDIT_ASGMTS | Specifies the maximum number of assignments that can be displayed for editing on HP Project Management's Enter Actuals page. | Default: 200 |
| PM_NUM_EDIT_TASKS | Specifies the maximum number of tasks that can be edited on HP Project Management's Quick Edit page. | Default: 200 |
| PORTLET_EXEC_TIMEOUT<br>Category: Timeout | The amount of time (in seconds) after which portlets time out.<br>This parameter is used to limit long-running queries in portlets, which may be caused by adding portlets without filtering criteria. Used to avoid excessive database CPU processing when users end their sessions before processing has completed. | Default: 20 (seconds) |
| PORTLET_MAX_ROWS_RETURNED | Determines the maximum number of rows to display in portlets. | Default: 200 |
| PROGRAM_SUMMARY_CONDITION_INTERVAL | The interval between summary condition updates. | Default: 4000 (seconds) |
| PROJECT_PV_UPDATE_INTERVAL_MINUTES | The interval at which the Project Planned Value Update Service runs. The service is enable and disabled using the *ENABLE_PROJECT_PV_UPDATE_SERVICE* parameter. | Default: 1440<br>(24 hours) |

Table A-1. Server configuration parameters (page 33 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| REFERENCE_ UPDATE_INTERVAL | Controls the interval for checking whether references between entities should be updated. See also *ENABLE_REFERENCE_UPDATE_ SERVICE*. | Default: 5 (minutes) |
| REMOTE_ADMIN_ REQUIRE_AUTH | Determines whether user authentication is required for remote administration.<br>If set to `TRUE`, users running `kStop.sh` to shut down the PPM Server are required to supply a valid PPM Center user name and password.<br>If set to `FALSE`, any user with access to `kStop.sh` can shut down the server. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |
| REPORT_DIR | Default directory to which report output is written. If you require report output to be written to a location other than the default directory (outside of the PPM Server directory structure), use this parameter to specify an alternate directory here. Make sure that the PPM Server has access to the directory so that the report output HTML files can be written here. | Example:<br>`D:/<PPM_ Home>/710/ aeon/reports/` |
| REPORT_LOG_DIR | Directory in which the PPM Center report logs are stored.<br>**Note:** In a server cluster, If you have overridden the default value for this parameters to refer to a different directory, then all servers in the cluster must be able to access and share the directory. | Same default value as the *BASE_LOG_ DIRECTORY* parameter<br>Example:<br>`D:/<PPM_ Home>/710/ aeon/logs/ reports/` |
| REPORTING_ STATUS_ REFRESH_RATE<br>Category: Scheduler/ services/thread | The frequency with which report status is refreshed and displayed to the user. | Default: 5<br>(seconds) |

Table A-1. Server configuration parameters (page 34 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| REQUEST_LOG_DIR | Specifies the location for Request execution log outputs.<br>**Note:** In a server cluster, If you have overridden the default value for this parameters to refer to a different directory, then all servers in the cluster must be able to access and share the directory. | Same default value as the *BASE_LOG_DIRECTORY* parameter<br>Example:<br>`D:/PPM/710/aeon/logs/` |
| REQUEST_SEARCH_RESULTS_MAX_ROWS | Determines the maximum number of results returned by a search. The value is displayed as the default in the **Limit Rows Returned To** field. | Default: 1000<br>Valid values: |
| REQUEST_TYPE_CACHE_TIMEOUT | Determines the stale check timeout for the cache that maintains mappings between parameters and tokens for Request Type and Request Header Type.<br>**Note:** HP strongly recommends that you not change the value of this parameter. | Default: 3600<br>(seconds) |
| RESOURCE_COST_UPDATE_SERVICE_DELAY | If *ENABLE_RESOURCE_COST_UPDATE_SERVICE* is set to `TRUE`, use this parameter to determine how often costs are recalculated. | Valid values: `TRUE, FALSE` |
| RESOURCE_FINDER_ROLE_WEIGHT | The value of this parameter is used to calculate the suitability score for items returned on the resource finder results page. | Default: `25`<br>Valid values: `0` through `100` |
| RESOURCE_FINDER_SKILL_WEIGHT | The value of this parameter is used to calculate the suitability score for items returned on the resource finder results page. | Default: `25`<br>Valid values: `0` through `100` |

Table A-1. Server configuration parameters (page 35 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| RESTRICT_ BYPASS_ EXECUTION_TO_ MANAGERS | Determines whether bypass execution of workflow steps in packages is restricted to managers.<br>If set to `TRUE`, only users with an access grant of Package Manager or Request Manager access can bypass executions.<br>If set to `FALSE`, all users eligible to act on executions can bypass them. | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |
| RESTRICT_ BYPASS_REQ_ EXEC_TO_ MANAGERS | Setting this parameter to `TRUE` restricts bypass execution to request managers. When set to `TRUE`, only a user with the Manage Request access grant can bypass an execution step on a request | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |
| RM_ALLOWED_ EFFORT_TYPES | Setting used to specify the allowed effort types for HP Resource Management staffing profile and other modules. You can specify a combination of up to three comma-delimited values.<br>Examples:<br>■ `fte, person_days, hours`<br>■ `fte, person_days`<br>■ `fte, hours`<br>■ `person_days, hours`<br>The order does not matter. | Default: `fte, person_days`<br>Valid values: `hours, fte, person_days` |
| RM_DEFAULT_ EFFORT_TYPE | Setting used to determine the default effort type, in hours or full-time equivalents (fte) used to display staffing profiles and resource pool information. | Default: `fte`<br>Valid values: `hours, fte, person_days` |
| RM_DEFAULT_ PERIOD_TYPE | Setting used to determine the default period type used to display staffing profiles and resource pool information. | Default: `month`<br>Valid values: `quarter, month, week, year` |

Table A-1. Server configuration parameters (page 36 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| *RMI_URL | Port on which the PPM Server listens to initiate RMI client/server communication.<br>Must be a unique port, distinct from the Web server, SQL*Net, and the HTTP or HTTPS ports.<br>Format:<br>`rmi://<hostname>:<port>/KintanaServer` | Default:1099<br>Valid values: Port numbers higher than 1024<br>Example:<br>`rmi://gold.ppm.com:1099/PPMServer` |
| RMI_VALIDATE_SERVER_CERTIFICATE | This parameter is used if PPM Server is running in secure RMI mode.<br>If set to `TRUE`, the client PPM Workbench validates the server certificate against the Certificate Authorizer's to verify server identity. If set to `FALSE`, the certificate is not validated. | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |
| *RML_PASSWORD | Password of the Oracle schema name specified in *RML_USERNAME. | Valid values: [encrypted password] |
| *RML_USERNAME | Oracle schema name for the meta layer schema.<br>Must be the same as the database schema name used during installation. | Valid values:<br>Any user name format that Oracle supports |
| *SCHEDULER_INTERVAL<br>Category: Scheduler/services/thread | Number of seconds after which the scheduler checks for services to be run. | Default: 60 |
| SCPCLIENT_TIMEOUT | Amount of time after which SCP clients must provide feedback after a file transfer has initiated, else a timeout occurs.<br>Set to the maximum expected time for file transfer. | Default: 10000<br>(milliseconds) |

Table A-1. Server configuration parameters (page 37 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| SEARCH_TIMEOUT<br>Category: Timeout | The number of seconds after which searches time out.<br>Used to limit long-running queries in searches, which may be caused by submitting a search without entering selective data. Avoids taking up database CPU when users end their sessions before the search is completed. | Default: 60<br>(seconds) |
| SECURE_RMI | If set to `TRUE`, RMI network traffic between PPM Workbench clients and the PPM Server is encrypted. | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |
| SERVER_ENV_ NAME | Name of the PPM Center environment containing information PPM Server machine (for example, host name, user name, and password).<br>Must be set before PPM Center entity migrators or commands involving secure copy can run. | Default: `KINTANA_ SERVER` |
| SERVER_MAX_ PERM_SIZE | This parameter is for HP internal use only. Do not change its value unless directed to do so by HP-Mercury support for PPM Center. | |
| SERVER_MODE | Specifies the server mode to use in case you want exclusive access to a running server. | Default: `NORMAL`<br>Valid values: `Normal, Restricted, Disabled` |
| *SERVER_NAME | DNS name or IP address of the machine hosting the PPM Server. | Default: kintana<br>Valid values: [any valid machine name] |
| SERVER_TYPE_ CODE | Operating system on which the PPM Server is installed. | Valid values: `UNIX, WINDOWS` |
| SERVICE_ PROVIDER_ SECURITY_GROUP | This parameter is for HP internal use only. Do not change its value unless directed to do so by HP-Mercury support for PPM Center. | |

Table A-1. Server configuration parameters (page 38 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| SHOW_BASE_URL_ ON_NOTIFICATIONS | Determines whether the URL for the PPM Center logon window is displayed at the top of each email notification. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |
| SHOW_ PERSONALIZE_ FIRST | If set to `TRUE`, **Personalize Dashboard** is the first PPM Dashboard menu item listed on the menu bar. | Default: `FALSE`<br>Valid values: `TRUE, FALSE` |
| **SMTP_SERVER<br>Required if notifications are used | Host name of the SMTP-compliant mail server that acts as the gateway for email notifications. | Example: `mailserver.mydomain.com` |
| SOCKS_PROXY_ HOST | Host name of the SOCKS proxy server. | Host name of the SOCKS proxy server. |
| SOCKS_PROXY_ PORT | The port on the SOCKS proxy host that accepts proxy connections. | Any available port on the SOCKS proxy host. |
| *SQLPLUS | Name of the command-line SQL*Plus executable, which must be in the `<`*`Oracle_Home`*`>/bin` directory. | Default: `sqlplus.exe` |
| SQLPLUS_VERSION | The Oracle SQL*Plus version installed on the machine that hosts the PPM Server. You must set this for some PPM Center reports that run from command-line SQL*Plus calls.<br>If you encounter problems running PL/ SQL-based reports in PPM Center, set this parameter. | Example:<br>`com.kintana.core.server.SQLPLUS_ VERSION=10.1.0.2` |
| **STATS_CALC_ DAY_OF_WEEK<br>Required if *ENABLE_ STATISTICS_ CALCULATION* = TRUE<br>Category: Database statistics | Day of the week on which to calculate Oracle database statistics. | Default:<br>1 (designates Sunday)<br>Valid values:<br>1 through 7 |

Table A-1. Server configuration parameters (page 39 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| **STATS_CALC_ WAKE_UP_TIME<br>Required if *ENABLE_ STATISTICS_ CALCULATION* = TRUE<br>Category: Database statistics | Hour of the day (using 24-hour clock) at which statistics are to be calculated. | Default:<br>1 (designates 1 a.m. or 01:00)<br>Valid values:<br>0 (midnight) through 23 (11 p.m. or 23:00) |
| **STATS_CALC_ WEEK_INTERVAL<br>Required if *ENABLE_ STATISTICS_ CALCULATION* = TRUE<br>Category: Database statistics | Frequency (in weeks) with which statistics are calculated. | Default:<br>1 (designates weekly calculation)<br>Valid values:<br>1 through 52<br>Example:<br>2 (designates every other week) |
| SYNC_EXEC_INIT_ WAIT_TIME | Duration after which the intermediate Request Working page opens. | Default: 4<br>(seconds) |
| SYNC_EXEC_MAX_ POLL_TRIES | Number of times to poll for completion of a request until a final message is returned to the user. | Default: 4 |
| SYNC_EXEC_POLL_ INTERVAL | Time interval (in minutes) at which to poll for completion of a request after the intermediate Request Working page opens. | Default: 15 |
| TASK_ACTUAL_ ROLLUP_INTERVAL | This parameter determines the delay between consecutive runs of the Task Actual Rollup Service, which asynchronously rolls up actuals entered through HP Time Management or the My Tasks portlet. | Default:<br>(minutes)<br>Valid values: |
| THREAD_POOL_ MAX_THREADS<br>Category: Scheduler/ services/thread | Maximum number of packages to run simultaneously within a release distribution.<br>If a large number of packages in a distribution are processing, increasing this value can improve performance. | Default: 10 |

Table A-1. Server configuration parameters (page 40 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| THREAD_POOL_ MIN_THREADS<br>Category: Scheduler/ services/thread | Minimum number of packages to be run simultaneously within a release distribution.<br>See also *THREAD_POOL_MAX_ THREADS* on page 265. | Default: 5 |
| **TIME_ZONE<br>Required if the PPM Server and the Oracle database are in different time zones | Use this parameter to display to users to a time zone other than the one in which the PPM Server and the associated Oracle database server reside. For example, even if your PPM Server is in New York, you can set the time zone so that users see US/ Central (Central Standard Time) time.<br>This setting affects users instance-wide; you cannot display different time zones depending on the locale of users in different time zones.<br>You must specify a fully-qualified time zone name. For a list of fully qualified names, run the Client Time Zone Report in the Admin Tools window of the PPM Workbench. (For details on how to run the report, see *Running Server Reports from the Admin Tools Window* on page 138).<br>If you do not a specify a value for this parameter, the value defaults to the time zone in which the server is running. | Valid values: Any fully-qualified time zone designation such as "America/ Los_Angeles" or "Australia/LHI."<br>Do not use daylight savings-modified time zones such as "EDT" or "PDT." |
| TMG_DATE_ NOTIFICATION_ INTERVAL | This value determines how often the system sends delinquent time sheet notifications (see *ENABLE_TIME_ SHEET_NOTIFICATIONS_SERVICE* on page 242). | Default: 120 (minutes)<br>Valid values: integer > 0 |
| TMG_FUTURE_ PERIODS_TO_ ALLOW | Specifies the number future periods for which users can enter time. | Default: 10 |
| TMG_PAST_ PERIODS_TO_ ALLOW | Specifies the number of previous periods for which users can enter time. | Default: 10 |

Table A-1. Server configuration parameters (page 41 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| TRANSFER_PATH | This specifies the default temporary directory that PPM Center uses. The main purpose of this directory is to temporarily hold files as they are migrated from a source environment to a destination environment with HP Deployment Management.<br>In a server cluster, all servers must be able to access and share the specified directory. | Example:<br>`D:/<PPM_Home>/710/ionia/transfers/` |
| TURN_ON_CONCURRENT_REQUEST_WATCH_DOG | If you are using HP Deployment Management to integrate with Oracle applications (via Object Migrator), then you must enable this "watch dog" thread. When HP Deployment Management submits a concurrent request (job) to Oracle Apps, this thread polls Oracle to determine what state the job is in, and when it has completed. This parameter is associated with the frequency parameter *CONCURRENT_REQUEST_WATCH_DOG_INTERVAL*. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |
| TURN_ON_NOTIFICATIONS<br>Category: Scheduler/services/thread | Turns on the notification service. Usage: Turn off notifications for copies of production instances being used for testing. Turn them on again when the system goes to production. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |
| TURN_ON_SCHEDULER<br>Category: Scheduler/services/thread | Turns on the scheduler.<br>Usage: To improve performance, turn off the scheduler in non-production instances. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |
| TURN_ON_WF_TIMEOUT_REAPER<br>Category: Scheduler/services/thread | Turns on the timeout reaper, which scans all active workflow steps to verify that they have timed out according to the settings for the step.<br>Use the ***WF_TIMEOUT_REAPER_INTERVAL* parameter to set the frequency with which the service checks for information. | Default: `TRUE`<br>Valid values: `TRUE, FALSE` |

Table A-1. Server configuration parameters (page 42 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| USER_PASSWORD_ MAX_LENGTH | Maximum number of characters in user passwords. | Default: 16 |
| USER_PASSWORD_ MIN_DIGITS | Minimum number of digits in user passwords. | Default: 0 |
| USER_PASSWORD_ MIN_LENGTH | Minimum number of characters in a user password. | Default: 4 |
| USER_PASSWORD_ MIN_SPECIAL | Determines the minimum number of non-alphanumeric (special) characters that user passwords must contain. | Default: 0 |
| VALIDATION_LOG_ DIR | In a server cluster, If you have overridden the default value for this parameters to refer to a different directory, then all servers in the cluster must be able to access and share the directory. | Same default value as the *BASE_LOG_ DIRECTORY* parameter<br>Example:<br>`D:/<PPM_ Home>/710/ aeon/logs/ reports/` |
| VISUALIZATION_ EXEC_TIMEOUT | Length of time (in seconds) that resource management visualizations can run before they time out. | Default: 180 |
| WF_SCHEDULED_ TASK_INTERVAL<br>Category: Scheduler/ services/thread | Time interval at which the PPM Server checks for pending scheduled tasks, and starts the tasks if worker threads are available. | Default: 60<br>(seconds) |
| WF_SCHEDULED_ TASK_PRIORITY<br>Category: Scheduler/ services/thread | Determines the priority of scheduled tasks.<br>Because scheduled tasks run in the background, it may be useful to run them at a lower priority than the threads servicing user-oriented interactive tasks. | Default: 10 |

Table A-1. Server configuration parameters (page 43 of 43)

| Parameter (*Required, **Required If) | Description, Usage | Default and Valid Values |
|---|---|---|
| **WF_TIMEOUT_ REAPER_INTERVAL<br>Required if *TURN_ ON_WF_TIMEOUT_ REAPER* = TRUE<br>Category: Scheduler/ services/thread | If *TURN_ON_WF_TIMEOUT_ REAPER* is set to TRUE, this parameter setting determines the frequency with which the service checks for information.<br>Example: If you set a timeout value of 86400 (seconds), which is 24 hours, on Monday at 10 a.m., then all active workflow steps would time out immediately at 10 a.m. on Tuesday. | Default: 900<br>(seconds) |
| WORKAREA_SIZE_ POLICY | Controls how the memory for SQL working areas is allocated for intensive operations as sort, group by, hash join, and so on. If this parameter is set to AUTO (the default), Oracle manages the allocation and de-allocation of the memory area for each process, and these need not be set separately or manually. Oracle calculates memory allocation based on the load and characteristic of the system.<br>**Note:** HP strongly recommends setting WORKAREA_SIZE_POLICY to AUTO. This parameter must be set concurrently with *PGA_ AGGREGATE_TARGET*. | Default: If you are using Oracle 10g, the default is AUTO.<br>In Oracle 9i, the default is AUTO only if *PGA_ AGGREGATE_ TARGET* is set. |
| WORKBENCH_ PLUGIN_VERSION | Specifies the Java plug-in version used to access the PPM Workbench interface.<br>Use this parameter to configure PPM Center to use a specific version (other than the default version) of the Java plug-in to open the PPM Workbench. | Example:<br>com.kintana.core.server.WORKBENCH_ PLUGIN_ VERSION=1.5.0_02 |
| WS_UPDATE_ CLOSED_AND_ CANCELED_ REQUESTS | If set to TRUE, lets Web services update closed and canceled requests. | Default: FALSE<br>Valid values: TRUE, FALSE |

## Logging Parameters

*Table A-2* lists the PPM Server configuration parameters located in the `logging.conf` file, and provides a description of each. The `logging.conf` file is located in the <*PPM_Home*>/`conf` directory.

➤ Changes to `logging.conf` are picked up dynamically by the application (it takes about one minute) so there is no need to restart the application.

Table A-2. Logging parameters (page 1 of 3)

| Parameter (*Required) | Description, Usage | Default, Valid Values, Example |
|---|---|---|
| CATCH_SYSTEM_ERR | Used to determine whether to redirect `System.err` to the server log. | Default: `TRUE`<br>Valid values:<br>`TRUE, FALSE` |
| CATCH_SYSTEM_OUT | Used to determine whether to redirect `System.out` to the server log. | Default: `TRUE`<br>Valid values:<br>`TRUE, FALSE` |
| DEFAULT_SERVER_ LOGGING_LEVEL | Default debug level of the PPM Server.<br>Controls the verbosity of logs generated by the PPM Server.<br>The values, which can also be set dynamically at runtime in the Workbench Server Settings window, map as follows:<br>■ `ERROR` maps to None in the Server Settings window<br>■ `INFO` maps to Normal<br>■ `DEBUG` maps to Max<br>For more information about the Server Settings window, see *Setting Debugging and Tracing Parameters* on page 143. | Valid values:<br>■ `NONE` - No information, (including errors) is logged<br>■ `ERROR` - Only errors are logged<br>■ `INFO` - Errors and additional information is logged<br>■ `DEBUG` - Includes verbose debugging messages<br>■ `ALL` - Displays all log messages generated |

Table A-2. Logging parameters (page 2 of 3)

| Parameter (*Required) | Description, Usage | Default, Valid Values, Example |
|---|---|---|
| DEFAULT_USER_ DEBUG_LEVEL<br>Category: High-level debug | Specifies the default debug level of a user's client session.<br>Controls the verbosity of users' logs on the client, application server, and database. Can be different for different client sessions, and can be changed in the standard interface as a user preference.<br>The values, which can also be set in the Workbench Server Settings window dynamically at runtime, map as follows:<br>- ERROR maps to None in the Server Settings window<br>- INFO maps to Normal<br>- DEBUG maps to Max<br>For more information about the Server Settings window, see *Setting Debugging and Tracing Parameters* on page 143. | Valid values:<br>- `NONE` - No information, (including errors) is logged<br>- `ERROR` - Only errors are logged<br>- `INFO` - Errors and additional information is logged<br>- `DEBUG` - Includes verbose debugging messages<br>- `ALL` - Displays all log messages generated |
| ENABLE_CONSOLE_ LOGGING | Enables logging by the PPM Server to the console. | Valid values:<br>`TRUE, FALSE` |
| ENABLE_WEB_ ACCESS_LOGGING | Whether or not to log information sent to the internal PPM Center Web server (Tomcat). | Valid values:<br>`TRUE, FALSE` |
| FILE_RECHECK_ INTERVAL | Time interval (in seconds) at which the `logging.conf` file is checked for changes.<br>The file keeps being checked as long as the PPM Server is running. | Default: 30 |
| LOG_LAYOUT | Layout format of the log files. | Default: `TEXT`<br>Valid values:<br>`TEXT, XML` |

Table A-2. Logging parameters (page 3 of 3)

| Parameter (*Required) | Description, Usage | Default, Valid Values, Example |
|---|---|---|
| MAX_BACKUP_INDEX | Limits the number of backup logs kept in the system. | Default 20 |
| ROTATE_LOG_SIZE | As the PPM Server logs information into the `serverLog.txt` file, the file can grow quite large. This parameter determines how large (in KB) it can grow before the server creates a new log file. When the `serverLog.txt` file reaches the size specified by this parameter, the PPM Server renames it (to `serverLog_<`*`timestamp`*`>.txt`), and starts a new serverLog.txt file. | Default: 250 |
| SERVER_DEBUG_LEVEL<br>Category: High-level debug | Debug level of the PPM Server.<br>Controls the verbosity of logs generated by independent server processes (for example, EmailNotificationAgent).<br>Corresponds to the **Debug Level** list in the **Server** section of the Server Settings page. | Valid values: `NONE, LOW, HIGH` |

# LDAP Attribute Parameters

*Table A-3* lists and provides descriptions of the PPM Server configuration parameters in the LdapAttribute.conf file, which is located in the <*PPM_Home*>/conf directory. Use the LdapAttribute.conf file to map the attributes of the LDAP server with the attributes used by the PPM Server.

The default mapping uses the standard LDAP attributes. All values are case-sensitive. Do not add spaces between tokens.

Do not map the ORG_UNIT_NAME and PARENT_ORG_UNIT_NAME parameters in LdapAttribute.conf. These attributes are specified in the KRSC_ORG_UNITS_INT table.

Table A-3. LDAP attribute parameters (page 1 of 2)

| Parameter (*Required) | Description, Usage | Default, Valid Values, Example |
|---|---|---|
| KNTA_USERS_INT | Target table for the import. Can be mapped to any LDAP attribute.<br>Always map both VISIBLE_USER_DATA and USER_DATA.<br>To disable default mapping, either comment out or delete the mapping line.<br>Mappings:<br>■ USERNAME = sAMAccountName<br>■ FIRST_NAME = givenName<br>■ LAST_NAME = sn<br>■ EMAIL_ADDRESS = mail<br>■ PHONE_NUMBER = telephoneNumber<br>■ DEPARTMENT_MEANING = departmentNumber<br>■ LOCATION_MEANING = locality<br>■ MANAGER_USERNAME = manager<br>■ USER_DATA1 = mail<br>■ VISIBLE_USER_DATA1 = mail | Format:<br>ColumnName = LDAPAttribute |

Table A-3. LDAP attribute parameters (page 2 of 2)

| Parameter (*Required) | Description, Usage | Default, Valid Values, Example |
|---|---|---|
| LDAP_TIME_ FORMAT | Attribute that keeps track of the time format that the LDAP server uses. | Format for Active Directory servers:<br>`yyyyMMddHHmmss'.0Z'`<br>Format for Netscape LDAP servers:<br>`yyyyMMddHHmmss'Z'` |
| LDAP_USER_ OBJECTCLASS | Objectclass attribute for a user on the LDAP server. | Default: person |

# B Server Directory Structure and Server Tools

In This Appendix:

# Overview of Directory Structure

This appendix addresses the `mitg710` and `<PPM_Home>` directories and the scripts and tools they contain. The `mitg710` directory (the installation directory) contains two subdirectories that relate to the Oracle database schemas: `mitg710/sys` and `mitg710/system`.

The `<PPM_Home>` directory (the install directory for PPM Center) holds several subdirectories (`bin, docs, logs, reports`, and so on) that contain server- and system-oriented information, and administrative tools that perform tasks such as starting, stopping, and reporting on the PPM Server or system.

## mitg710/system Directory

The `mitg710/system` directory contains the `CreateKintanaUser.sql` and `CreateRMLUser.sql scripts`. *Table B-1* lists and describes the `CreateKintanaUser.sql` script variables.

Table B-1. CreateKintanaUser.sql variables

| Variable | Description |
|---|---|
| PPM_User | User name of the new database schema. |
| PPM_Password | Password of the new database schema. |
| Data_Tablespace | Tablespace used to store PPM Center tables. |
| Index_Tablespace | Tablespace used to store PPM Center indexes. |
| Temp_Tablespace | Temporary tablespace. |
| Clob_Tablespace | Tablespace used to store large data (CLOB). |

*Table B-2* lists the `CreateRMLUser.sql` script variables.

Table B-2. CreateRMLUser.sql variables

| Variable | Description |
|---|---|
| Rml_User | User name for the new RML database schema. |
| Rml_Password | Password for the new RML database schema. |
| Rml_data_tablespace | Tablespace used to store PPM Center database tables. |
| Rml_temp_tablepace | Temporary tablespace. |

# *<PPM_Home>*/bin Directory

The bin subdirectory of *<PPM_Home>* contains all of the scripts required to configure and administer the server. This section provides descriptive information about these scripts.

## kBuildStats.sh

The kBuildStats.sh script instructs Oracle to gather statistics about the PPM Center database schema. This information can be very important in improving the overall performance of PPM Center. For information about how to use this script, see *Using the dbms_stats Package to Collect Additional Statistics* on page 162.

## kCancelStop.sh

If a command such as kStop.sh-delay is being used to stop the server, you can run kCancelStop.sh to cancel the stop request. Authentication may be required for this, which works in the same way as for kStop.sh. Use the -user user name flag.

## kConvertToLog4j.sh

The kConvertToLog4j.sh script converts the JDBC log, Web log, or server log to the log4j XML format. You can view logs in this format with a tool such as Chainsaw (a GUI-based log viewer available at the Web site logging.apache.org/log4j/docs/chainsaw.html).

### Examples

To convert a Web log to the log4j XML format:

```
sh kConvertToLog4j.sh -webLog apacheLog.txt
```

To convert a JDBC log to the log4j XML format:

```
sh.kConvertToLog4j.sh -jdbcLog jdbc.kintana.log
```

To convert a serverLog.txt file in text format to the log4j XML format:

```
sh kConvertToLog4j.sh -serverLog serverLog.txt
```

To convert a server log, JDBC log, and Web log, and then concatenate them in a result log:

```
sh kConvertToLog4j.sh -serverLog serverLog.txt -jdbcLog
jdbc.kintana.log -webLogiisLog.txt
```

For information about usage type:

```
sh kConvertToLog4j.sh -help
```

## kConfig.sh

The `kConfig.sh` script launches the server configuration interface. Because `kConfig.sh` cannot update variables in a cluster node (that is, anything that comes after an `@node`), HP recommends that, for a server cluster environment, you edit (or add) parameter values directly in the `server.conf` file using a text editor. After you do, be sure to run the `kUpdate.sh` script to implement your changes. For more information about how to set the server mode, see *Setting the Server Mode* on page 66.

## kDeploy.sh

The `kDeploy.sh` script is a command-line tool used to install HP Deployment Management Extensions, PPM Center Best Practices, and PPM Center product service packs. This software is distributed as a deployment (a software bundle that contains files) in the following format:

```
mitg-<ver>-<id>[.#'].jar
```

where:

- `mitg` is the product code
- `<ver>` is the PPM Center version for which you can install the service pack.
- `<id>` is the unique identifier for service pack.
- (Optional) `.#'` is the revision number for the deployment

For example, to install a product service pack SP1:

1. Extract the deployment JAR file.

   This file must be in the *<PPM_Home>* directory. There is no need to extract anything. The script does that.

2. To apply the SP1 service pack, run the following:

   ```
   sh kDeploy.sh -i SP1
   ```

*Table B-3* displays the key command-line parameters for `kDeploy.sh`. To generate a list of parameters, run the following command:

```
sh kDeploy.sh -h
```

Table B-3. Key command-line parameters for kDeploy.sh

| Parameter | Description |
|---|---|
| -i | Installs deployments.<br>For example, the command to install a PPM Center service pack (SP) could be:<br>**`sh kDeploy.sh -i SP14`** |
| -l | Lists the deployments installed on an instance.<br>For example:<br>**`sh kDeploy.sh -l`**<br>results in:<br>`JAVA_HOME = /u1/java/j2sdk1_3_1_07`<br>`java version "1.3.1_07"`<br>`Java(TM) 2 Runtime Environment, Standard Edition`<br>`(build 1.3.1_07-b02)`<br>`Java HotSpot(TM) Client VM (build 1.3.1_07-b02,`<br>`mixed mode)` |
| -D | Searches for bundles in a given directory.<br>For example, to search for a file in the `DIR` directory, run the following:<br>`sh kDeploy.sh -D DIR` |
| -h | Provides help for `kDeploy.sh`. Lists all the command-line options. |
| -f | Reinstalls an existing deployment. |
| -k | Includes the PPM Center database schema password in the command. Automates command execution but may be a security risk. |
| -u | Includes the PPM Center user name in the command. |
| -p | Includes the password for the PPM Center user name in the command. Automates command execution but may be a security risk. |
| -tidy | Cleans up unnecessary deployment files. |
| -skip -database | Specifies that database changes are not to be applied if they already exist. |
| –update–deploy | Extracts the new `kDeploy.sh`, if it exists. |

## kEncrypt.sh

In some cases it may be necessary to generate encrypted strings in accordance with the encryption scheme of your PPM Server installation. The `kEncrypt.sh` script provides a convenient way to do this.

Run the script as follows:

```
kEncrypt.sh <string to encrypt>
```

## kGenPeriods.sh

Use the `kGenPeriods.sh` script to generate the period information and populate the database tables that contain `knta_periods` and `knta_period_groups`. This script generates the monthly periods and period groups from the start year through the end year based on the start year and end year parameters.

The `kGenPeriods.sh` script does not regenerate periods that already exist between the specified years. It only creates periods between the minimum of the specified start year and the existing minimum period year—and the maximum of the existing maximum Period Year and the specified end Year.

## kGenTimeMgmtPeriods.sh

The kGenTimeMgmtPeriods.sh script is used in HP Time Management to populate the `KTMG_PERIODS` table with data. The script takes the number of periods to be populated and the start date from which the periods are to be populated.

Run the script as follows:

```
kGenTimePeriods.sh <num> <start_date>
```

The <*num*> value is the number of time periods required. The <*start_date*> value is the date from which the periods are to be populated. For a new installation, running this script is optional. Running `kGenTimePeriods.sh` with no arguments defaults the number of time periods to 24.

## kJSPCompiler.sh

The first time a user requests a page in the PPM Center standard interface, the server must compile the page. To eliminate this initial performance drag, run the `kJSPCompiler.sh` script to precompile all of the JSP pages before users request them. This gives first-time users faster access to the standard PPM Center interface.

## kKeygen.sh

The kKeygen.sh script generates new security keys.

## kMigratorExtract.sh

The script kMigratorExtract.sh is used in PPM Center entity migration.

## kMigratorImport.sh

Use the kMigratorImport.sh script to migrate PPM Center entities. Make sure that you only type **Y** or **N** for the 17 flags listed. For example, to import a file, run the following command:

```
sh kMigratorImport.sh -username <username> -password
<password> -action import -filename <full file path> -i18n
none -refdata nochange -flags NNNNNNNNNNYYNNNNN
```

Be sure to place the full file path in single quotes.

## kRunCacheManager.sh

Use the kRunCacheManager.sh script to clear your cache without having to restart the server. You can script this to execute after your DB changes have been committed.

## kRunServerAdminReport.sh

You can use the kRunServerAdminReport.sh script to run diagnostic reports on the PPM Server. This utility provides a summary of current activity on the system and the number of database connections made.

➤ You can also access this functionality through the PPM Workbench. To access and run these diagnostic reports from the PPM Workbench, on the shortcut bar, select **Sys Admin > Server Tools.**

The reports listed in the Admin Tools window are the same reports you can use the kRunServerAdminReport.sh script to run.

## kStart.sh

The kStart.sh script is used only on UNIX systems to start the PPM Server as a background process. For more details about starting the server, see *Starting and Stopping the PPM Server* on page 66.

## kStatus.sh

Run the `kStatus.sh` script to check the state of the PPM Server. This script returns the server status whether the server is running or not. If it is running, the script returns the current load value, which refers to the number of active user sessions.

## kStop.sh

Use the `kStop.sh` script to stop the PPM Server. This script requires some arguments. You can use the `-now` flag to quickly stop the server, or use the `-delay <#minutes>` flag to stop it after a delay of a specified number of minutes.

If you are using the `-delay` option, you can use the `kCancelStop.sh` script to cancel the stop request.

Using the `-delay` option automatically issues a message to advise all connected PPM Center users that the server will stop after the specified delay. This script requires authentication if the server parameter `REMOTE_ADMIN_REQUIRE_AUTH` is set to `TRUE`. In this case, you must also specify the flag `-user <username>`.

For more information on available flags, run `kStop.sh` without any options. For information about how to stop the server, see *Starting and Stopping the PPM Server* on page 66.

## kSupport.sh

Use the `kSupport.sh` script to gather information useful to HP-Mercury support in diagnosing system problems, and create a Zip file with a timestamp in the `support/zipfiles` directory.

The `kSupport.sh` script gathers information from the following:

- Install logs
- Server logs (with the option for a date range)
- JDBC logs
- Deploy logs (for the installation of patches and HP Deployment Management Extensions)
- Configuration files
- Server reports
- Database information
- File system information

As it collects server logs or JDBC logs, the script concatenates all the files into one server `Log.txt` file.

You can run `kSupport.sh` in GUI, console, or silent mode. Silent mode automatically captures a default set of information without prompting for user input.

To run in GUI mode:

```
sh kSupport.sh
```

To run in console mode:

```
sh kSupport.sh -console
```

To run in silent mode:

```
sh kSupport.sh -silent -k <password> -customer <company_
name> -sr <service_request_number>
```

## kUpdateHtml.sh

The `kUpdateHtml.sh` script is a key script used to update the PPM Server configuration. Run the `kUpdateHtml.sh` script any time a server configuration is updated in the `server.conf` file, regardless of whether you use the `kConfig.sh` script to change parameter values, or use a text editor to make the changes directly.

## kWall.sh

Use the `kWall.sh` script to send a message to all users logged on to the PPM Workbench. When you run the script, it prompts you for your PPM Center user name and password, and for the message text.

## setServerMode.sh

The `setServerMode.sh` script, located in the `<`*`PPM_Home`*`>/bin` directory, sets the server mode in case you want exclusive access to a running server.

The following are valid server mode values:

- **Normal.** In normal mode, all enabled users can log on, and all services are available, subject to restrictions set in `server.conf` parameters.
- **Restricted.** In restricted mode, the server lets users with Administrator access grant log on. The server cannot run scheduled executions, notifications, or the concurrent request manager while in this mode.

  Before you can install an HP Deployment Management Extension, you must set the server to restricted mode.
- **Disabled.** Disabled mode prevents server startup. A server enters disabled mode only after a PPM Center upgrade exits before the upgrade is completed.

To set the server mode using the `setServerMode.sh` script:

1. On the desktop, select **Start > Run.**

   The Run dialog box opens.

2. In the **Open** field, type the following:

   ```
   sh setServerMode.sh <MODE VALUE>
   ```

3. Click **OK.**

For more information about server modes, see *Setting the Server Mode* on page 66.

# <*PPM_Home*>/pdf Directory

The docs subdirectory contains all documentation files for PPM Center (to view them, you need Adobe Reader).

You can also access product documentation:

- From **Product Information > Documentation** in either the PPM Center standard interface or the PPM Workbench interface
- The PPM Download Center

For information on how to access PPM Center documentation, see *Accessing PPM Center Documentation* on page 21.

# <*PPM_Home*>/integration Subdirectory

The integration subdirectory contains information or examples for various common integrations between the PPM Server and external systems. For example, the <*PPM_Home*>/integration/webserver directory contains information about each external Web server that you can integrate with the PPM Server. Files used to perform the integration are located in these folders. For more information on using the folders and files in the integration subdirectory, see the relevant document that pertains to the integration involved.

## *<PPM_Home>*/logs Directory

The server directory structure has two log directories. The <*PPM_Home*>/logs directory contains the reports subdirectory, which contains a log file for each PPM Server report that is run, and directories named PKG_number and REQ_ number. These subdirectories contain execution logs for HP Deployment Management packages and HP Demand Management requests. The <*number*> variable in the directory name corresponds to the ID of the package or request being run.

The other log directory, <*PPM_Home*>/server/<*server_name*>/log contains all PPM Server-generated logs. As the server runs, it generates logging messages and writes them to the serverLog.txt file. When this file reaches the size indicated by the ROTATE_LOG_SIZE server parameter, it is renamed to serverLog_timestamp.txt, and a new serverLog.txt is started.

The Java servlets used to serve the Web pages generate their own log files, named servletLog.txt. The amount of information in the server log files depends on the debugging level set in the server configuration. The server parameters SERVER_DEBUG_LEVEL and DEFAULT_USER_DEBUG_LEVEL control the debugging level. If a problem arises and you require more information in the logs, log on to the PPM Workbench as Administrator and reset the server debug level to Maximum debugging information (select **Edit > Debug Settings**).

## *<PPM_Home>*/reports Directory

The reports subdirectory contains the HTML files for all reports that PPM Center clients have run.

## *<PPM_Home>*/server Directory

The <*PPM_Home*>/server directory contains the deployed PPM Server. Typically, administrators are not required to make any changes in this directory. Server configurations are handled through the provided admin scripts in the <*PPM_Home*>/bin directory.

## <*PPM_Home*>/sql Directory

The sql subdirectory contains source code for the built-in PPM Center reports and core PL/SQL packages. This is provided for convenience and for customization needs.

## <*PPM_Home*>/transfers Directory

The transfers subdirectory serves as temporary storage for files transferred between the server and remote computers. For more information about how the transfers directory is used in entity migration, see *Basic Parameters* on page 184.

## Other Directories

Other directories contain reference files, as indicated by their names. You are not likely to require access to these directories.

# C Preinstallation Checklists

In This Appendix:

## Preliminary Tasks

Before you can install PPM Center, you must perform a number of tasks on various system components to prepare for the installation. This appendix provides information to help ensure that your systems meet the technical requirements for installing PPM Center. It contains checklists for the preliminary tasks to perform on the application server (or servers), database server, client machines, and the network.

As you finish each task listed in the checklists, mark it as completed and make a note of the date and time you completed it. After you finish all of the required tasks, return this document to your HP Professional Services representative. The checklist will help your representative make the necessary preparations before installation and speed up the installation process. If you have questions or concerns, contact the Professional Services representative or log a service request on our support site at support.Mercury.com.

➤ The tables in the following sections describe some system requirements. For a complete list of requirements, see the *System Requirements and Compatibility Matrix*.

## Preliminary Database Tasks

*Table C-1* lists the Oracle database-related tasks to perform before you install PPM Center.

Table C-1. Preinstall checklist for database tasks

| Done | Database Task | Information | Date and Time |
|---|---|---|---|
| | Identify the name and IP address of the database server. | | |
| | Install an Oracle database to house PPM Center solutions. | The database server can reside on the same machine as the PPM Server, or on a different machine. | |
| | Create the two required database schemas, and then set up access grants for them.<br>For information about how to run the script to create the database schemas, see *Creating the Database Schemas* on page 52.<br>For information about how to run the script that sets up the required access grants for the schemas, see *Key Decisions* on page 41.<br>**Note:** Setting up the schemas before installation is optional. You can create the schemas and set up access grants during installation. | Set up the following grants for the schema:<br>■ `GRANT SELECT ON v_$parameter to <`*`PPM_Schema`*`>`<br>■ `GRANT SELECT ON v_$mystat to <`*`PPM_Schema`*`>`<br>■ `GRANT SELECT ON v_$process` to `<`*`PPM_Schema`*`>`<br>■ `GRANT SELECT ON v_$session to <`*`PPM_Schema`*`>`<br>■ `GRANT EXECUTE ON dbms_stats to <`*`PPM_Schema`*`>`<br>To set up these grants before (or during) installation, run the `GrantSysPrivs.sql` script (located in the `mitg710/sys` directory).<br>**Note:** HP recommends that you create these users during installation. If (and only if) your DBA does not provide you with the SYSTEM password for the database to use during installation, and does not type the password for you, then you can create the users ahead of time and use them during installation. | |

## Preliminary Application Server Tasks

*Table C-2* lists the tasks to perform on every machine you plan to use as a PPM Center application server.

Table C-2. Preinstall checklist for application server tasks (page 1 of 2)

| Done | Application Server Task | Information | Date and Time |
|---|---|---|---|
|  | Identify the operating system (UNIX or Windows) running on each machine on which you plan to install the PPM Center application server. |  |  |
|  | Identify the name and IP address of each application server. |  |  |
|  | Identify the installation directory. |  |  |
|  | For software installation, set aside the amount of disk space specified in the *System Requirements and Compatibility Matrix.* |  |  |
|  | Create a system (mitg) user for PPM Center installation and future system maintenance activities on this server.<br>Create an email account for this system user. | Specify a user name that is consistent with your corporate naming standards. |  |
|  | PPM Center requires that you set `JAVA_HOME` in the system environment of the user account to be used to start the PPM Server. | For information about how to verify that the `JAVA_HOME` parameter is set or about how to set it, see *Verifying that the JAVA_HOME Parameter Is Set* on page 49. |  |
|  | After you install Oracle client on the server machine, set the `ORACLE_HOME` parameter. |  |  |

Table C-2. Preinstall checklist for application server tasks (page 2 of 2)

| Done | Application Server Task | Information | Date and Time |
|---|---|---|---|
| | Install the Sun Java Software Development Kit (SDK) for your operating system. | The SDK version you install on the server depends on the operating system the server is running.<br>For the exact version and operating system requirements, see the *System Requirements and Compatibility Matrix.*<br>For information about how to install the SDK, see *Installing the Java Software Developer Kit (SDK)* on page 48. | |
| | Each PPM Server requires the Oracle client library. After you install the Oracle database and client libraries, make sure that you can connect to that instance from the command line by running:<br>`sqlplus <username>/<password>@<SID>` | PPM Center must be able to log in to the database instance in non-interactive mode. This step uncovers possible configuration issues with the database and client libraries. | |

*Table C-3* lists the tasks to perform on Windows server that is to interact with PPM Center application servers.

Table C-3. Preinstall checklist for Windows servers that interact with PPM Servers

| Done | Task | Information | Date and Time |
|---|---|---|---|
| | Make sure that the mitg user has administrator-level access to the machine. | | |
| | Make sure that the regional setting on the server is English (United States). | | |
| | Make sure that FTP is installed and enabled, and that Bourne shell (bash) is installed. | Product support for Windows:<br>Van Dyke (VShell Server)<br>OpenSSH<br>Cygwin provides a complete UNIX-like environment. For information how to download and install Cygwin UNIX Emulator, go to cygwin.com.<br>**Note:** HP recommends Van Dyke vShell. | |
| | If you plan to use PPM Center to perform deployments to other Windows machines in your environment, make sure that each Windows server with which PPM Center is to interact has the following:<br>■ UNIX Bourne shell emulator<br>■ FTP, SSH, SSH2, or Telnet server | Cygwin provides a complete UNIX-like environment. For information how to download and install Cygwin UNIX Emulator, go to cygwin.com. | |

## Preliminary Network Tasks

Use *Table C-4* to keep track of the network tasks you perform before you install or upgrade toPPM Center.

Table C-4. Preinstall checklist for network tasks

| Done | Network Task | Information | Date and Time |
|---|---|---|---|
|  | If you plan to use PPM Center to perform deployments to other Windows machines in your environment, make sure that each Windows server that is to interact with PPM Center has the following:<br>- UNIX Bourne shell emulator<br>- FTP, SSH, SSH2, or Telnet server | HP-Mercury supports the following UNIX Bourne shell emulators:<br>- Van Dyke (VShell Server)<br>- OpenSSH (included in Cygwin)<br>- Telnet server from Microsoft Windows Services for UNIX (SFU) (a supported remote command processor)<br>**Note:** HP recommends Van Dyke vShell.<br>Cygwin provides a complete UNIX-like environment. For information how to download and install Cygwin UNIX Emulator, go to cygwin.com. |  |
|  | Although PPM Server comes with its own HTTP server, you may want to use an industry-standard external Web server to serve PPM Center clients. If you plan to use HTTPS or a server cluster configuration, you must install and configure an external Web server. | PPM Center supports the following external Web servers:<br>- Microsoft IIS<br>- Microsoft Windows Server Sun Java System Web Server<br>- Apache HTTP Server<br>- IBM HTTP Server (IHS)<br>For information on supported versions, see the*System Requirements and Compatibility Matrix.*<br>PPM Server cluster uses an external Web server to load balance Web traffic across multiple application servers. For the most current configuration information, see the *System Requirements and Compatibility Matrix.* |  |

## Preliminary Client Tasks

Use *Table C-5* to keep track of the tasks that you must perform on client machines to be used to access the PPM Dashboard and PPM Workbench.

Table C-5. Preinstall checklist for client machine tasks

| Done | Client Task | Information | Date and Time |
|---|---|---|---|
| | Make sure that the client machine has a supported Web browser installed. | For information on which Web browsers are supported, see the *System Requirements and Compatibility Matrix.* | |
| | Make sure that the client machine has sufficient RAM for the part of PPM Center it must access on PPM Workbench. | Client machines that access the PPM Workbench must have at least 256 MB of RAM. Client machines that are not intended to access the PPM Workbench must have at least 128 MB of RAM.<br>For client machines that access HP Project Management, HP recommends 512 MB of RAM. | |
| | Make sure that the client machine has sufficient disk space. | 100 MB free disk space is required | |
| | Make sure that the client processor is adequate. | 600 MHz is required | |

As with most applications, greater memory and higher processor speeds result in higher application and user interface performance. However, in most cases, the minimum requirements shown in *Table C-5* provide adequate performance.

# Index