Mercury IT Governance Center™

System Administration Guide and Reference

Version: 7.0

Mercury
379 North Whisman Road
Mountain View, CA 94043
http://www.mercury.com

If you have any comments or suggestions regarding this document, please send email to documentation@mercury.com.

Publication Number: ITGSysAdmin7.0/02

Table of Contents

List of Figures

List of Tables

Chapter

Introduction

In This Chapter:

- *Administering the Mercury IT Governance Center™ System*
- *Related Documents*
- *Accessing Documentation from the Mercury IT Governance Download Center*

Administering the Mercury IT Governance Center System

This document provides information about how to install, configure, and maintain the Mercury IT Governance Center™ system, including:

- The Mercury IT Governance Server or server cluster
- The Oracle database and database schema used with Mercury IT Governance Center
- Other system components

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

- Overview of Mercury IT Governance Center system architecture and of single-server and server cluster system configuration (Chapter 2, *System Overview,* on page 19)
- Information about product licensing and optional programs that you can install (Chapter 3, *Installation Overview,* on page 33)
- Instructions on how to create the required database schemas, verify installation, and install service packs and Mercury Deployment Management Extensions and Accelerators (Chapter 4, *Installing Mercury IT Governance Center,* on page 41)
- Details on how to configure all components of the Mercury IT Governance Center system and to start and stop the Mercury IT Governance Server. (Chapter 5, *Configuring the System,* on page 67)
- Information that Mercury IT Governance Center users need to know in order to use the Workbench (Chapter 5, *Configuring the System,* on page 67)
- Advanced configuration information, including details on how to configure an external Web server and Mercury IT Governance Server clusters (Chapter 6, *Advanced System Configuration,* on page 105)
- Information on how to integrate Mercury IT Governance Center with an LDAP server (Chapter 6, *Advanced System Configuration,* on page 105)
- Details on how to maintain the Mercury IT Governance Center and the database after installation and configuration (Chapter 7, *Maintaining the System,* on page 139)

- Information about the kinds of performance issues that can arise, and how to identify and resolve them (Chapter 8, *Improving System Performance,* on page 163)
- Information on how to migrate entire instances of Mercury IT Governance Center, and on how to migrate just the database schemas (Chapter 9, *Migrating Instances,* on page 183)
- Details on how to use the Mercury entity migrators to migrate specific kinds of Mercury IT Governance Center entities and associated objects between instances of Mercury IT Governance Center (Chapter 10, *Migrating Entities,* on page 199)
- Mercury IT Governance Server configuration parameters (Appendix A, *Server Configuration Parameters,* on page 237)
- Details about Mercury IT Governance Center directories and the scripts and tools they contain (Appendix B, *Server Directory Structure and Server Tools,* on page 289)
- Checklists of the tasks to perform on the application server (or servers), database server, client machines, and the network before you install and configure Mercury IT Governance Center for your organization (Appendix C, *Preinstallation Checklists,* on page 303)

This document is written for:

- Application developers and configurators
- System and instance administrators
- Database administrators

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

Related Documents

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

- *System Requirements and Compatibility Matrix*

  Before you install Mercury IT Governance Center, check the *System Requirements and Compatibility Matrix* to make sure that your operating environment meets *all* of the minimum system requirements.

- *Release Notes*

  The *Release Notes* provide product information that is not included in the regular documentation set.

- For general information about Mercury IT Governance Center, see *Configuring the Standard Interface.*

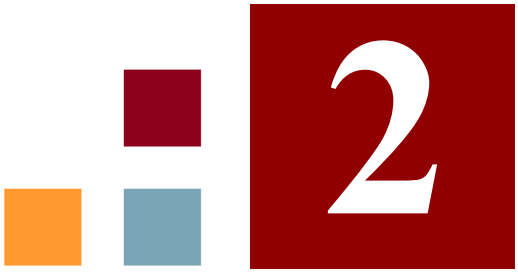
Additional documents that you might find useful as you configure or maintain Mercury IT Governance Center include:

- *Commands, Tokens, and Validations Guide and Reference*
- *Open Interface Guide and Reference*
- *Reports Guide and Reference*
- *Security Model Guide and Reference*
- *Mercury-Supplied Entities Guide* (includes descriptions of all portlets, request types, and workflows in Mercury IT Governance Center)

Accessing Documentation from the Mercury IT Governance Download Center

At the Mercury IT Governance Download Center, you have access to the same PDF files that are available through the standard interface after Mercury IT Governance Center installation, and to documents that are only available at that location. To get to the login page for the Mercury IT Governance Download Center, go to itg.merc-int.com/support/download/login.jsp.

Chapter



System Overview

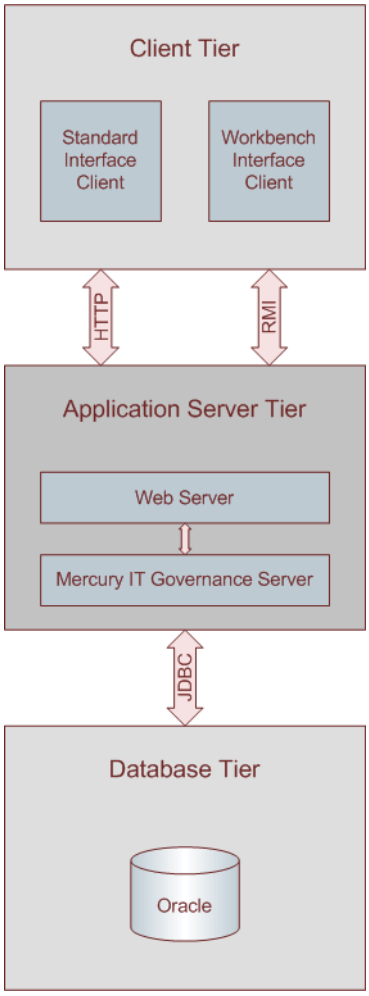
In This Chapter:

Overview of Mercury IT Governance Center Architecture

Mercury IT Governance Center employs a three-tier architecture composed of:

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

Figure 2-1. Mercury IT Governance Center architecture



Browser clients use HTTP or HTTPS to communicate with the Mercury IT Governance Center Web and application server. Mercury IT Governance Center 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 Mercury IT Governance Center standard interface. The standard interface is rendered using Java Server Pages (JSP) and is accessed using a Web browser.
- The Mercury IT Governance 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.
- For the 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 Mercury IT Governance 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.

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 (formerly Sun ONE Web Server and iPlanet), Microsoft IIS, and Apache

- 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 Mercury IT Governance Server and the Mercury IT Governance 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 110.

Database Tier

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

Note

The database consists of the following two database schemas:

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

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

Mercury IT Governance 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 Mercury IT Governance Center supports a variety of system configurations. You can deploy Mercury IT Governance Servers in a single-server configuration or a server cluster configuration. The following sections provide detailed information about these configurations.

Single-Server Configurations

Mercury IT Governance Center configurations are typically single-server configurations that consist of one Mercury IT Governance Server and one Oracle database. The single Mercury IT Governance 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.

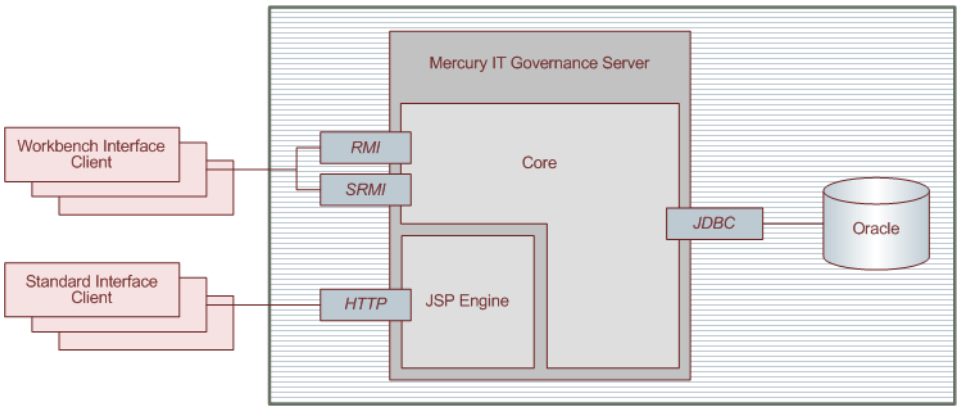
You can set up single-server configurations in the following ways:

- 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, which is illustrated in *Figure 2-2*, consists of one machine that hosts both the Mercury IT Governance Server and the Oracle database.

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



Standard interface clients communicate with the Mercury IT Governance Server using HTTP, or, for secure communication, HTTPS. Workbench interface clients communicate with the Mercury IT Governance Server using RMI, or, for secure communication, SRMI.

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

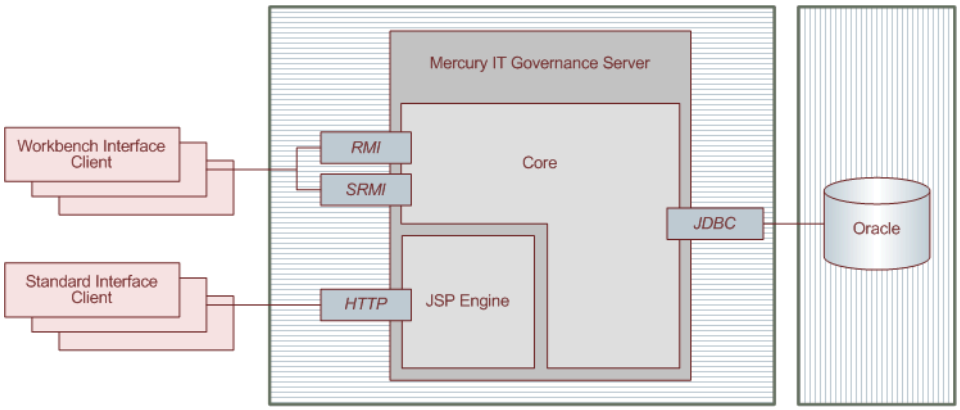
An organization typically uses this configuration if it requires a dedicated machine for all Mercury IT Governance Center services and database operations. User load, transaction capacity, and system performance depend on the available resources on a machine. This configuration does not support load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

For information about how to set up a single-server/single-machine configuration, see Chapter 4, *Installing Mercury IT Governance Center,* on page 41.

Single-Server/Multiple-Machine Configuration

In the single-server/multiple-machine configuration (illustrated in *Figure 2-3*) the Mercury IT Governance 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 Mercury IT Governance Server using HTTP, or HTTPS for secure communication. Workbench interface clients communicate with the Mercury IT Governance Server using

RMI, or SRMI for secure communication. The Mercury IT Governance Server and Oracle database use JDBC to communicate.

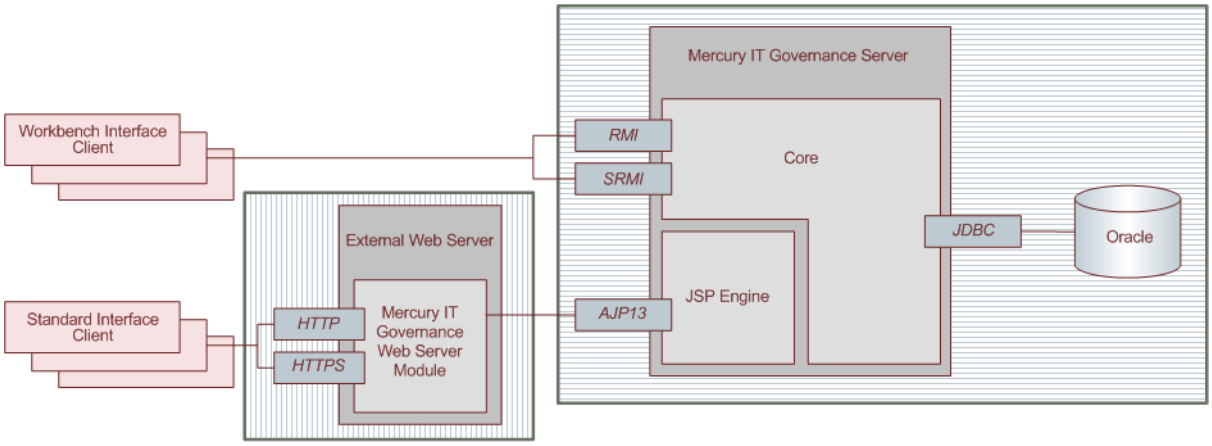
An organization typically uses the single-server/multiple-machine configuration if it requires a separate machine for database operations. User load, transaction capacity, and system performance depend on the resources available on the Mercury IT Governance Server machine. This configuration does not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

For information about how to set up a single-server/multiple-machine configuration, see Chapter 4, *Installing Mercury IT Governance Center,* on page 41.

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 Mercury IT Governance Center. The external Web server and the Mercury IT Governance 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 Mercury IT Governance Servers use AJP13 to communicate.
- Workbench interface clients communicate directly with the Mercury IT Governance Server using RMI, or, for secure communication, SRMI.
- The machine that houses the Mercury IT Governance Server also contains the Oracle database. The Mercury IT Governance Server communicates with the Oracle database using JDBC.

- The Mercury IT Governance 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 Mercury IT Governance Server.

IT departments often have standards for the Web server used for HTTP traffic. Running the HTTP listener allows for Mercury IT Governance 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 below 1024. Because Mercury does not recommend that the Mercury IT Governance Server run as root, it recommends integration with an external Web server in this case.

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

Note

Mercury recommends using the internal Web server built into the Mercury IT Governance 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 4, *Installing Mercury IT Governance Center,* on page 41 and Chapter 6, *Advanced System Configuration,* on page 105.

For a list of supported Web servers, see the *System Requirements and Compatibility Matrix* document, which is available at the Mercury IT Governance Download Center.

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 Mercury IT Governance Servers. If a Mercury IT Governance Server shuts down, the activities running on that server are automatically transferred to an available Mercury IT Governance Server in the cluster. This server failover feature helps ensure that Mercury IT Governance Center system services such as email notifications and scheduled executions remain operational.

Server cluster configurations contain two or more Mercury IT Governance Servers and an Oracle database. The first Mercury IT Governance 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.

Note

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 Mercury IT Governance 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 Mercury IT Governance 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.

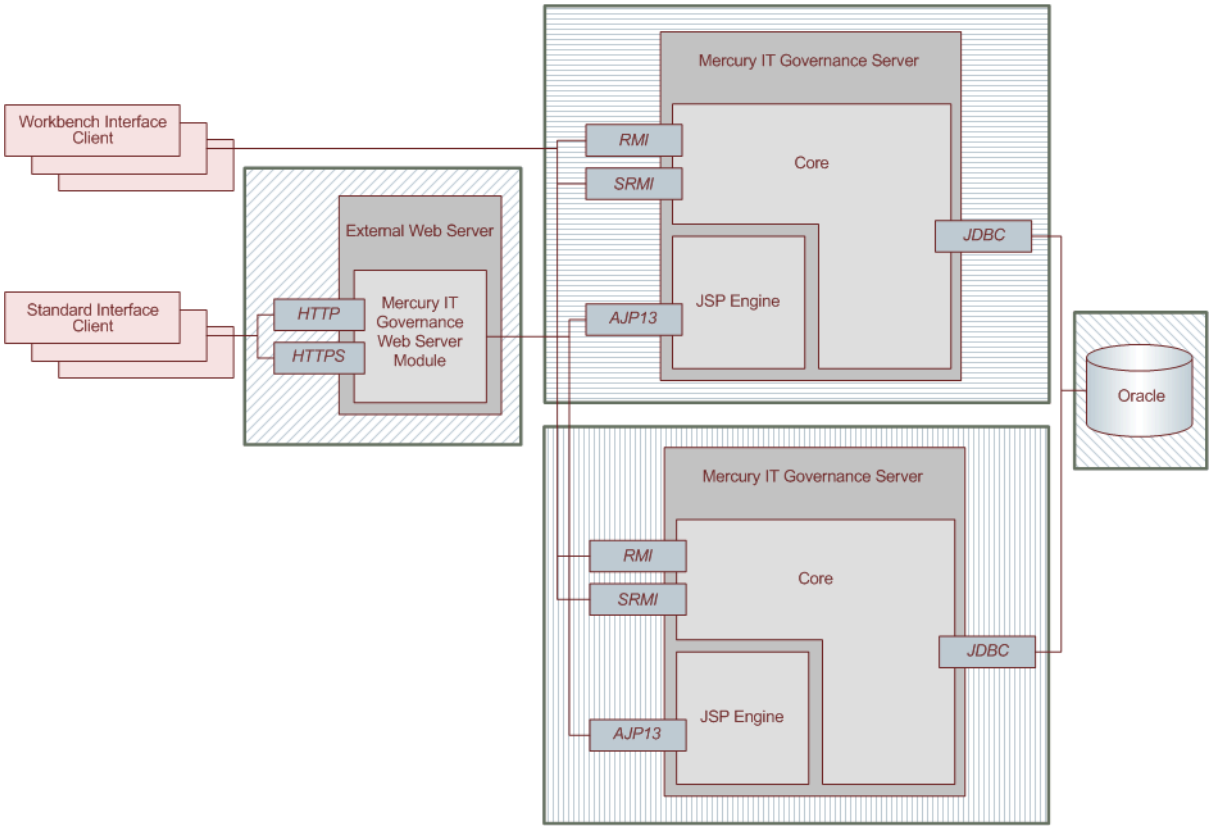
Server Cluster/External Web Server Configuration

The server cluster/external Web server configuration (see *Figure 2-5 on page 28*) distributes client connections evenly among any number of Mercury IT Governance Servers, based on Web traffic and server load. This configuration is typically used for organizations that need to load-balance Web traffic across multiple Mercury IT Governance 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 Mercury IT Governance 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. Mercury IT Governance Servers run in the background and are transparent to users. Users see only the URL to the external Web server.

The Mercury IT Governance Web server module forwards HTTP or HTTPS requests to one of the Mercury IT Governance Servers. The Mercury IT Governance Web server module and the Mercury IT Governance Servers communicate using AJP13.

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

Note

You cannot use a single Web server installation on a Windows-based system for multiple Mercury IT Governance Center instances. If you must use an external Web server for multiple Mercury IT Governance Center instances, Mercury 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 Mercury IT Governance Center Web server module as the software load balancer for a Mercury IT Governance Server cluster configuration. In this configuration, the Mercury IT Governance 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 Mercury IT Governance Web server module.
3. The Mercury IT Governance Web server module sends the request to a Mercury IT Governance 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. You can find further details about how to set up single sign-on in documentation at the Mercury IT Governance Download Center.

Using SSL Accelerators

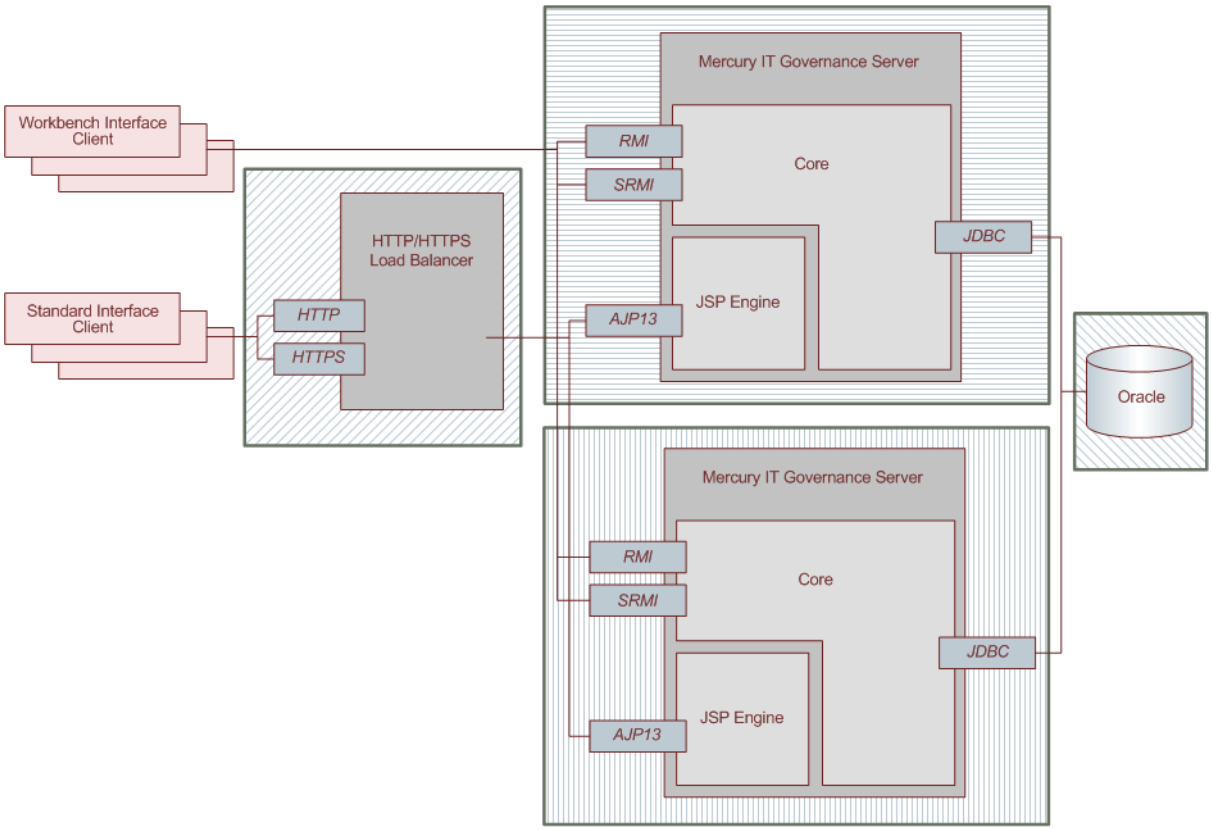
For Mercury IT Governance Server cluster configurations running HTTPS, you must integrate an external Web server that supports the appropriate accelerator to leverage a hardware-based SSL accelerator.

The external Web server and Mercury IT Governance Servers communicate using AJP13, a proprietary protocol that can be more efficient than HTTP for communicating with Mercury IT Governance Servers using an external Web server. For information about how to set up a server cluster with an external Web server, see Chapter 6, *Advanced System Configuration,* on page 105.

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 Mercury IT Governance Servers. This configuration enables the even distribution of client connections among Mercury IT Governance Servers based on server load and availability.

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



In this configuration:

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

Note

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 Mercury IT Governance Servers perform other tasks. Setting up your system this way can improve system performance.

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

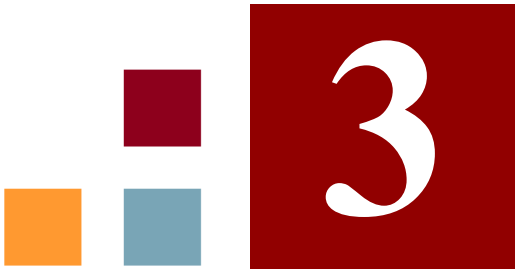
Note

Although *Figure 2-6* on page 30 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 Mercury IT Governance Servers.)

Using this configuration improves user load distribution, transaction capacity, and system performance. The degree of improvement depends on the number of Mercury IT Governance 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 6, *Advanced System Configuration,* on page 105.

Chapter



Installation Overview

In This Chapter:

Key Considerations

To prepare to install Mercury IT Governance Center, review the key considerations addressed in this section.

Installing for the First Time

If you are installing Mercury IT Governance Center for the first time, perform the following tasks:

1. Read the rest of this chapter.
2. Read the *System Requirements and Compatibility Matrix* document, which is described in *Related Documents* on page 18.
3. Read the *Release Notes,* which are described in *Related Documents* on page 18.
4. If you plan to install Mercury Object Migrator™, Mercury GL Migrator, one of the Mercury Deployment Management Extensions, or a Mercury Accelerator, see the documentation for the product.
5. Make certain that you have the valid licenses required for all of the products you plan to install.
6. For instructions on how to install Mercury IT Governance Center, see Chapter 4, *Installing Mercury IT Governance Center,* on page 41.

   Chapter 4 provides information on how to:

   - Prepare to install the product
   - Install the product
   - Verify the installation
7. Configure the Mercury IT Governance Server and system environment.

   For information about how to configure Mercury IT Governance Center, see Chapter 5, *Configuring the System,* on page 67.

8. Install and configure optional products you have purchased to work with Mercury IT Governance Center.

Note

After you install and configure Mercury IT Governance Center, you can install Extensions, Accelerators or Migrators in any order you choose. For information about how to install and configure optional products, see Chapter 4, *Optional Installations,* on page 65.

Installing the Document Management Module

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

Installing Mercury Object Migrator or GL Migrator

If you are running Mercury IT Governance Center in the Oracle environment, and have purchased the corresponding release of Mercury Object Migrator or Mercury GL Migrator, you must consult not only the installation instructions in this document, but also the instructions in the Mercury Object Migrator or Mercury GL Migrator documentation.

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

Installing a Mercury Deployment Management Extension

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

Obtaining License Keys

Check to make sure that you have purchased the Mercury 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 release. Mercury IT Governance Center license keys are delivered

in the `license.conf` file, which you can find in the `<ITG_Home>/conf` directory after installation.

After you purchase Mercury Deployment Management Extensions or Mercury Migrators, you receive a user name and password that you can use to download product code and documentation from the Mercury IT Governance Download Center. To go to the login page for the Mercury IT Governance Download Center, open a Web browser window and type itg.merc-int.com/support/download/login.jsp.

Checking System Requirements

Before you start to install Mercury IT Governance Center, check to make sure that your system environment meets all the requirements. For information about the system requirements, see the *System Requirements and Compatibility Matrix.* This document is available from the Mercury IT Governance Download Center. To go to the login page for the Mercury IT Governance Download Center, open a Web browser window and enter itg.merc-int.com/support/download/login.jsp.

Installing a UNIX Emulator and Telnet Server (Windows)

To run Mercury IT Governance 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 document *System Requirements and Compatibility Matrix.*

Note

To configure private key authentication with secure shell (see *Configuring Private Key Authentication with Secure Shell* on page 76), 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.

Key Decisions

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

When to Configure the Server

Before you can start the Mercury IT Governance Server, you must configure it. The installer prompts you to provide the values for several server parameters. If you choose not to 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 saved parameter values.

Note

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 information on how to configure the server, see *Configuring or Reconfiguring the Server* on page 71.

When to Set Up Grants to the Database Schema

To improve Mercury IT Governance Center performance, the installer rebuilds statistics for the Oracle optimizer during installation. To rebuild the statistics, the Mercury IT Governance Center database schema user must be granted the following privileges (as the SYS user, or SYSTEM on Oracle 9i):

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

If you have access to SQL*PLUS, you can run the script `sys/GrantSysPrivs.sql` (located in the mitg700/sys directory), which grants all required privileges for you. You can either run the script before installation (as the SYS user, or SYSTEM on Oracle 9i) or during installation.

Note

You cannot successfully complete the installation until you grant privileges and rebuild the statistics.

When to Create the Database Schemas

The Mercury IT Governance Server requires two database schemas to store application data. You can create them before you install Mercury IT Governance Center, or you can create the schemas automatically during installation.

To create the schemas before installation, follow the instructions provided in *Creating the Database Schemas* on page 50. If you set up the schemas before installation, the installer populates them with the entities and data required to run the Mercury IT Governance Server.

Running in Graphic (Swing) or Console Mode (UNIX)

On Windows platforms, you can only install the Mercury IT Governance Server in graphic (or *swing*) mode. On UNIX platforms, you can either install the Mercury IT Governance Server in graphic mode or in console mode (from the command line).

What's Installed

The Mercury IT Governance Center installer places the following products on your system:

- **Mercury IT Governance Dashboard.** The Mercury IT Governance Dashboard™ displays the information you need to understand and act on to make the key decisions required to govern IT resources, projects, and processes. At all levels of your organization, the Dashboard provides role-based, exception-oriented visibility into IT trends, status, and deliverables. If items such as demands, projects, or resources require your attention, you can drill down for details in the Dashboard before you act.
- **Demand Management.** Use Mercury Demand Management™ to prioritize and manage all the demand placed on IT.
- **Portfolio Management.** Use Mercury Portfolio Management™ to manage your portfolio of current applications, projects in progress, and proposed investments to align IT with business priorities.
- **Program Management.** Mercury Program Management™ provides a single location from which to initiate, operate, and manage your entire portfolio of programs and projects.

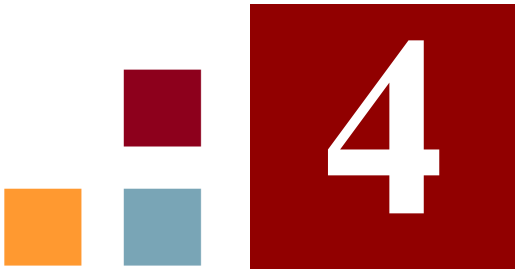
- **Project Management.** Mercury Project Management™ enables collaborative project management for repeating tasks, such as installing new releases of your HRMS applications, and one-time projects, such as developing a new e-commerce capability. Project Management helps you accelerate project delivery and at the same time, reduce project costs. Project Management uses task-level workflows to integrate project and process control. It works seamlessly with Mercury Program Management so that you can manage projects by exception and track project-to-project dependencies.
- **Financial Management.** Use Mercury Financial Management™ to monitor and manage your organization's IT portfolio. This component offers automatic real-time calculations of costs and variances to provide detailed comparisons of project health. It provides real-time visibility into budgets, costs (labor and non-labor), programs, projects, and overall IT demand.
- **Resource Management.** Mercury Resource Management™ lets you effectively monitor and manage resource capacity and allocation. It can help you balance your resource supply, including staffing levels and skill base, with incoming demand to provide full visibility and control over project demand.
- **Deployment Management.** Use Mercury Deployment Management™ to digitize the deployment process to support compliance initiatives, reduce application downtime, lower total costs, and minimize risk.
- **Time Management.** Use Mercury Time Management™ to get your company focused on value-added activities by streamlining time collection and improving accuracy across the wide range of work that IT performs.
- **Mercury IT Governance Foundation.** The Mercury IT Governance Foundation™ helps you to efficiently implement, protect, scale, and administer Mercury IT Governance Center by providing an integrated transaction-processing architecture with shared services available across all Mercury IT Governance Center applications.
- **Mercury IT Governance Best Practices.** Mercury IT Governance Center Best Practices provides customers with experience-derived information and advice about configuring and using Mercury Portfolio Management, and Mercury Program Management. Best Practices installation places various entities (for example, workflows and request types) on your system.

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

- You are logged on to your system as the database administrator.
- You have licenses for both the Portfolio Management and 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 have database administrator access.

If these conditions are not met during Mercury IT Governance Center installation, you can install Best Practices later, if you have the Portfolio Management and Program Management licenses, and if you log on as the database administrator. For detailed instructions on how to install Best Practices separately, see *Installing Mercury IT Governance Center Best Practices* on page 65.

Chapter



Installing Mercury IT Governance Center

In This Chapter:

- *What to Do Next*

Preparing to Install Mercury IT Governance Center

Before you start to install Mercury IT Governance Center, complete the following tasks:

1. Check the document *System Requirements and Compatibility Matrix* to make sure that your system meets *all* of the minimum requirements.
2. Collect the information that is required for installation.
3. Download the installation bundle (`mitg-700-install.zip`) from the Mercury IT Governance Download Center (itg.merc-int.com/support/download/login.jsp).

Note

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

4. Extract the installation files to a temporary directory.
5. Install the SDK.
6. Verify that the `JAVA_HOME` parameter is set.

Note

Mercury also recommends that you set the ORACLE_HOME parameter. To do this, you must install Oracle client on your server machine.

7. Create a Mercury IT Governance Center user.

Note

To create Mercury IT Governance Center users, you must have the Demand Management User Administration License.

8. Create the four Oracle tablespaces required to create the schemas and database objects.

9. Create the database schemas.

Note

> You can create the database schemas either before or during Mercury IT Governance Center installation.

10. Verify that required ports are open through the firewall and are not in use.

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

Note

> The variable `<ITG_Home>`, which is used throughout this document, refers to the root directory where Mercury IT Governance Center is installed. The name of this directory and its location are up to you.
>
> Do not unzip the installation files in your `<ITG_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 Mercury IT Governance Server(s), the database server, and Mercury IT Governance Center clients, see *Preinstallation Checklists* on page 303. Mercury recommends that you use these checklists to track the tasks that you must perform before you start to install Mercury IT Governance Center.

Tip

> After you complete the checklists, give them to your Mercury Product Support Organization (PSO) representative. The checklists will help your PSO representative make the necessary preparations and speed up installation.

Collecting Required Information

The Mercury IT Governance Center installer prompts you to enter several parameters values that are used to create and configure the Mercury IT Governance Server. The installer validates each value you enter before it continues the installation. *Table 4-1* lists the information required for installation.

Table 4-1. Required installation information (page 1 of 3)

Prompt	Description
Install Location	Directory in which the Mercury IT Governance Server is to be installed and configured. If the directory does not exist, the installer creates it. The directory path cannot contain spaces.
License Configuration File	This file contains valid Mercury IT Governance Center license keys. The Mercury IT Governance 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 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 Mercury IT Governance Center. Windows example: `C:\j2sdk1.4.2_08`
ORACLE_ HOME	The Mercury IT Governance Server machine must have Oracle client 9.2.0.7, or later, installed to communicate with the Mercury IT Governance Center database schema. Specify the home directory for the Oracle client tools on the Mercury IT Governance Server machine. The directory path cannot contain spaces.
SQL*PLUS	Location of the SQL*Plus utility. SQL*Plus is not required for installation, but is required for the Mercury IT Governance Server. Example: `C:\Oracle\bin\sqlplus.exe` 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 4-1. Required installation information (page 2 of 3)

Prompt	Description
Database Access Information	In addition to installing the Mercury IT Governance 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. 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 Mercury IT Governance Center database schema user name and password. The Mercury IT Governance Server uses the JDBC URL to connect to the Oracle database. The URL format is `jdbc:oracle:thin:@<hostname>:<port>:<SID>` where: ■ `<hostname>` is the DNS name or IP address of the computer running the database ■ `<port>` is the port that SQL*Net uses to connect to the database. Its value is usually 1521. To obtain the actual value, look at the corresponding entry in `tnsnames.ora`. ■ `<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. For Oracle Real Application Clusters (RAC), the JDBC_URL parameter must contain the host and port information for all databases to which the Mercury IT Governance Server is to connect. Following is an example of database access information used to allow the Mercury IT Governance Server to communicate with databases on two servers named Jaguar1 and Jaguar2: `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)))`
Mercury ITG Schema	If you create the database schema during installation, supply the user name and password for the Mercury IT Governance Center database schema.
Reporting Meta Layer Schema	User name and password of the Mercury IT Governance Center Reporting Meta Layer (RML) schema.
Tablespaces	Table, index, character large object data type (CLOB), and temporary tablespaces of the Oracle database that are required to create schemas and database objects.

Table 4-1. Required installation information (page 3 of 3)

Prompt	Description
Windows Service Name	Name of the Windows service for the Mercury IT Governance Server. The installer prefixes the service name with “Mercury ITG” to identify it. The installer also uses the service name to create the **Start** menu item.
Holiday Schedule	Holiday schedule on which the Mercury IT Governance 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. The system default is US dollars (USD). **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. 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. *Table A-1* on page 240 lists the server configuration parameters. Required parameters are marked with an asterisk.

Downloading the Installation Files

The installation files for Mercury IT Governance Center and Mercury Deployment Management Extensions and Migrators are distributed from the Mercury IT Governance Download Center (itg.merc-int.com/support/download/login.jsp). To access the files, you must have a user name and password to gain access to the Download Center. Mercury provides these when you purchase the software.

Download the Mercury IT Governance Center installation file (`mitg-700-install.zip`). If you are also installing one or more Mercury Deployment Management Extensions or Migrators, see the corresponding Mercury product documentation for specific download and install instructions.

Unzipping the Installation Files

Before you run the installation driver script, extract the installation files for the Mercury IT Governance 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 `mitg700/`. Run the command in a directory other than the `<ITG_Home>` directory.

Verifying that the JAVA_HOME Parameter is Set

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

Determining the Path in DOS

To determine the `JAVA_HOME` path in DOS:

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

Determining the 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 Parameter in Windows

To set the value of `JAVA_HOME` in Windows:

1. Open 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 Mercury IT Governance Center User

To install Mercury IT Governance 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 Mercury IT Governance Server maintenance—for example, stopping and restarting the Mercury IT Governance 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 Mercury IT Governance 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, Mercury IT Governance Center does not require root access for installation. Do not install the server as the root user.

Installing the Software Developer Kit (SDK)

Because the Mercury IT Governance 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 Mercury IT Governance Center are available from either Sun Microsystems or from the operating system vendor.

Note

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* document, which is available from the Mercury IT Governance Download Center.

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.

   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`.

   Note

   Note the following:

   The directory path name cannot contain spaces.

   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.

3. To verify that the user that Mercury IT Governance Center will be run under has the Java executable in its path, log on, and then run the following command:

   ```
   java -version
   ```

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

4. To ensure that the `JAVA_HOME` environment variable is set correctly, run the following command:

   ```
   echo %JAVA_HOME%
   ```

   If this does not echo the correct path to Java, set it to the correct value.

   For information about how to set the `JAVA_HOME` variable, see *Verifying that the JAVA_HOME Parameter is Set* on page 47.

Creating the Database Schemas

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

1. Generate at least one rollback segment for each of your tablespaces. For Oracle 9i or later, use an undo tablespace.

   These rollback segments should reside in a separate tablespace reserved for rollback segments. They should be generated with the OPTIMAL size constraint so the rollback segments automatically deallocate space as it becomes free.

2. Generate an additional tablespace to be used as the temporary tablespace for the Mercury IT Governance Center database schema.

   Be sure to specify this tablespace during the Mercury IT Governance Center database schema installation.

3. Generate unlimited quota on the data, index, temporary tablespaces, and CLOB for Mercury IT Governance Center.

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

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

1. Unpack the Mercury IT Governance Center installation bundle as outlined in *Installing Mercury IT Governance Center* on page 53.

   The `mitg700` directory is created. The `mitg700/sys` and `mitg700/system` directories contain the scripts required to create the database schemas.

2. Run the script `CreateKintanaUser.sql` (located in `mitg700/system`) against the database into which you plan to install Mercury IT Governance Center.

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

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

3. Run the `CreateRMLUser.sql` script (located in `mitg700/system`).

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

Note

> 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> \
<temporary_tablespace>
```

4. As the SYS user, run the `GrantSysPrivs.sql` script, which is located in the `mitg700/system` directory.

   This script grants the privileges that the Mercury IT Governance 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 *<Mercury_ITG_username>* and *<RML_username>*).

Verifying Port Availability

To successfully install and configure Mercury IT Governance Center, specific ports must be available through the firewall. To expedite installation, check to make sure that the ports are available before you start to install the product. *Table 4-2* contains summary information about the ports and protocols that Mercury IT Governance Center system components use to communicate.

Note

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

Table 4-2. Summary of Mercury IT Governance 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. ■ On UNIX systems, only processes started by the root user can be assigned a port number that is less than 1024.	
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 4-2. Summary of Mercury IT Governance 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 Mercury IT Governance Center

Perform the following steps to install the database objects and data to be used by the Mercury IT Governance Server. You can perform these steps on any UNIX or Windows computer with SQL*Net connectivity to the database on which the Mercury IT Governance Center database objects are to be installed.

Installing Mercury IT Governance 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 onto the server:

- Mercury IT Governance Center program files
- Mercury IT Governance Center database objects
- Start menu item
- Windows service

Warning

see *When to Set Up Grants to the Database Schema* on page 37

To install the Mercury IT Governance Server on Windows:

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

   Note

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

   To configure private key authentication with secure shell (see *Configuring Private Key Authentication with Secure Shell* on page 76), 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-700-install.zip` to the file system.

   The extraction creates the following in the `mitg700` directory:

   - `install.exe` file
   - JAR files
   - `system` directory
   - `sys` directory

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

   The installer prompts for the directory for the software installation (the `<ITG_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 44).

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

   After successful installation, Mercury IT Governance 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 Control Panel.

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

   c. Double-click **Services.**

d. Right-click name of the Mercury IT Governance Center service, and then click **Start** on the shortcut menu.

   Mercury recommends that you set the startup type to **Automatic** so that the Mercury IT Governance Server restarts automatically after the computer is restarted. If you have generated a custom Mercury IT Governance 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 Mercury IT Governance Center documentation and an uninstall program.

If you did not configure the Mercury IT Governance Server during installation, see *Configuring or Reconfiguring the Server* on page 71.

Note

Do not map the `<ITG_Home>` directory so that it is accessible from an external Web server. This introduces a potential security risk. Mercury recommends that you use the Mercury-supplied Web server, unless you have the special requirements described in *Single-Server/External Web Server Configuration* on page 25.

Installing Mercury IT Governance Center on UNIX

To install the Mercury IT Governance 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-700-install.zip`

   Alternatively,

   `jar xvf mitg-700-install.zip`

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

   The `mitg700` 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 SYS user) and specify the installation mode.

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

Table 4-3. 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 44).
- 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_$parameter to <Mercury_ITG_Schema>
grant select on v_$mystat to <Mercury_ITG_Schema>
grant select on v_$process to <Mercury_ITG_Schema>
grant select on v_$session to <Mercury_ITG_Schema>
grant execute on dbms_stats to <Mercury_ITG_Schema>
```

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

Warning

To run this script, you must have system database administrator privileges. You cannot run Mercury IT Governance Center until the privileges are granted and the statistics are rebuilt.

If you did not run this script before you started installation, do it now (as the SYS user, or SYSTEM on Oracle 9i).

Note

Mercury recommends that, after you install Mercury IT Governance Center, you change the password for the administrator user.

Configuring the FTP Server on Windows

Mercury IT Governance 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 Mercury IT Governance Center.

Before you configure the FTP server on a machine, check to make sure that the Windows user account (which Mercury IT Governance 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).

Warning

On Windows, most FTP servers, including Microsoft IIS, do not support drive letters. If you use FTP in Mercury IT Governance Center, the drive letter is removed from the base path. If your base path is d:\itg700, 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 /itg700 to D:\itg700.) 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 Mercury IT Governance 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 Mercury IT Governance Center.
3. Start the Mercury IT Governance 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 Mercury Support (support.mercury.com).

Contacting Mercury Support

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

1. Open the `mitg_install.txt` file (located in the `<ITG_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.txt` file for an error message that is specific to installation failure.

3. Place all of the files in the `<ITG_Home>/Install_700/logs` directory in a Zip 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 Mercury IT Governance Server has failed. The log file contains information about which failed action it attempted.

Installing the Microsoft Project Plug-In

To integrate Mercury IT Governance Center with Microsoft Project, you must install the Microsoft Project plug-in. You can install the plug-in at the server level, and then use a technology such as Microsoft SMS to push it to client machines. Alternatively, project managers can install the plug-in on client machines.

Note

To install the plug-in, you must have administrative rights.

This section provides the steps you perform to install the plug-in. Project managers can access information on how to install the plug-in in the document *Mercury Project Management User's Guide.*

Warning

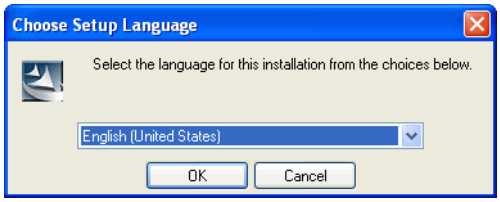
To install the Mercury plug-in for Microsoft Project, you must have Microsoft Internet Explorer 6.0 (SP2 or later) installed.

To install the plug-in:

1. After you install and configure Mercury IT Governance Center, log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Download Microsoft Project Plug-in.**

   The File Download dialog box opens and prompts you to indicate whether you want to open or save the `setup.exe` file.
3. Click **Open.**

   The Choose Setup Language dialog box opens.

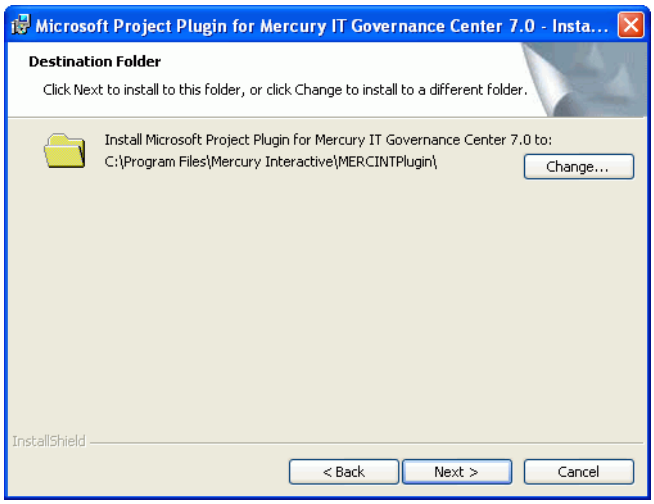


4. In the list, select a language.
5. Click **OK.**

   The Microsoft Project Plug-in for Mercury IT Governance Center InstallShield wizard starts up.

6. On the Welcome page, click **Next.**

   The Destination Folder page displays the default directory for the plug-in installation.



7. To accept the default directory, click **Next.** Otherwise, change the install directory, as follows:

   a. Click **Change.**

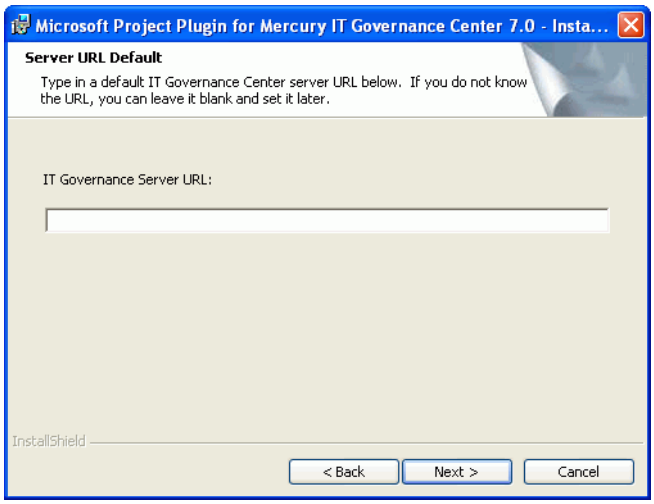
      The Change Current Destination Folder page opens.

   b. Browse to, and then select a destination installation folder.

   c. Click **OK.**

8. Click **Next.**

The Server URL Default page opens.



9. In the **IT Governance Server URL** field, type the URL for the Mercury IT Governance Server.

Note

If you do not know the default URL, you can provide this information later, after you install the plug-in.

10. Click **Next.**
11. Click **Install.**
12. After installation is completed, on the Install Shield Wizard Completed page, click **Finish.**

The next time you start Microsoft Project, the menu bar includes the **Mercury** menu.

Changing the Mercury IT Governance Server URL Setting

If, after installing the Mercury plug-in for Microsoft Project, you must change the server URL, make the change in the `ITGovernance.ini` file.

To change the server URL setting:

1. Go to the install directory for the Mercury plug-in for Microsoft Project, and then open the `ITGovernance.ini` file in a text editor.
2. Change the `Default URL` value.
3. Save and close the `ITGovernance.ini` file.

Installing Service Packs

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

Mercury IT Governance 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 Mercury IT Governance Center version on top of which you can install the service pack

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

`[.#]` is an optional revision number for the deployment (may or may not be included in the deployment name)

`.md` stands for Mercury deployment

For example, to install Service Pack 1:

1. Issue the following command:

   ```
   cp mitg-700-SP1.md ITG-home
   ```

2. Stop the Mercury IT Governance Server.

   For information about how to start and stop the server, see *Starting and Stopping the Mercury IT Governance Server* on page 68.

3. Issue the following command:

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

   As the script runs, follow the prompts.

4. Start the Mercury IT Governance Server.

For more information about the `kDeploy.sh` script, see *kDeploy.sh* on page 293.

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 Mercury Support

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

- Zip all the files in `<ITG_Home>/logs/deploy/700/directory/<SP#>`.

where `<SP#>` is the service pack version you are installing. For example, for Mercury IT Governance Center Release 7.0, Service Pack 2, you would zip the directory `<ITG_Home>/logs/deploy/700/directory/SP2`.

Optional Installations

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

Installing Mercury IT Governance Center Best Practices

Mercury IT Governance Center Best Practices provides customers with experience-derived information and advice about configuring and using Mercury Portfolio Management and Mercury Program Management. Best Practices installation places various entities (for example, workflows and request types) on your system that help optimize your use of Program Management and Portfolio Management.

If you did not, or could not, install Best Practices during Mercury IT Governance Center installation (for more information, see *What's Installed* on page 38), you can install it separately.

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

- You have installed and configured Mercury IT Governance Center.
- You are logged on to your system as the database administrator.
- You have licenses for both the Portfolio Management and 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 have system database administrator access.
- You have created the user name (if required) that is required for Best Practices installation. (You can access Best Practices with the admin user but not the user name you created for the database administrator.)

To install Best Practices:

1. Set the Mercury IT Governance Server to restricted mode.

   For information about how to set the server to restricted mode, and how to start and stop the server, see *Starting and Stopping the Mercury IT Governance Server* on page 68.

2. Start the Mercury IT Governance Server.

3. Run the `kDeploy.sh` script, as follows:

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

Note

For more information about the `kDeploy.sh` script, see *kDeploy.sh* on page 293.

Verifying Mercury IT Governance Center 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 Mercury Accelerators and Mercury Deployment Management Extensions

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

You are not required to stop the Mercury IT Governance Server(s) before you install an Extension. However, Mercury 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 a Mercury Accelerator or a Mercury Deployment Management Extension, see the documentation for the Extension or Accelerator you purchased.

What to Do Next

After you have successfully installed Mercury IT Governance Center, delete all subdirectories of the `install_700` directory, except for the `logs` subdirectory.

Proceed to Chapter 5, *Configuring the System,* on page 67.

Chapter

Configuring the System

In This Chapter:

Starting and Stopping the Mercury IT Governance Server

This section provides information about how to start the Mercury IT Governance Server on a single-server system. For information about configuring and running a clustered configuration, see *Server Cluster Configurations* on page 26 and *Configuring a Server Cluster* on page 126.

Note

Unless otherwise indicated, “the server” refers to the Mercury IT Governance Server or Mercury IT Governance Application Server, and not the server machine.

Setting the Server Mode

Mercury IT Governance 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 a Mercury Deployment Management Extension, you must set the server to restricted mode.
- **Disabled.** Disabled mode prevents server start-up. A server enters disabled mode only after a Mercury IT Governance Center upgrade exits before the upgrade is completed.

Setting the Server Mode with setServerMode.sh

The `setServerMode.sh` script, located in the `<ITG_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 `<ITG_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 68. For more information about the `kConfig.sh` script, see *kConfig.sh* on page 292.

Starting and Stopping the Server on Windows

To start the server on a Windows system:

1. If you are installing one of the Mercury 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 68.

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

   The service name starts with "Mercury ITG."
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 68.

Note

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 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 Mercury IT Governance Center service, and then click **Stop** on the shortcut menu.

   The service name starts with "Mercury ITG."

Starting and Stopping the Server on UNIX

To start the server on UNIX:

1. If you are installing a Mercury 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 68.
2. Change to the `<ITG_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 297. For information about how to start servers in a cluster, see *Starting and Stopping Servers in a Cluster* on page 135.

To stop the server on UNIX:

1. Navigate to the `<ITG_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 297. For information about how to stop servers in a cluster, see *Starting and Stopping Servers in a Cluster* on page 135.

Configuring or Reconfiguring the Server

If you configured the Mercury IT Governance 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 73.

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 Mercury IT Governance Server:

1. From a DOS or UNIX command line, run the `kConfig.sh` script (located in the `<ITG_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
     ```

Note

> Run this utility in an X Window session.

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

   Enter a value for every parameter that is 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 237.

   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.

Note

> Always use forward slashes (/) as a path separator, regardless of your operating system environment. Mercury IT Governance 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 73 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 Mercury IT Governance 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 Mercury IT Governance Server* on page 68.

Note

> You can also modify parameters directly in the server configuration file, which is described in Appendix A, *Server Configuration Parameters,* on page 237.
>
> 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 Mercury supplies, Mercury IT Governance 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]`.

- Mercury has created configuration parameters that you can use in special situations after you add them to the custom parameters folder. *Table 5-1* 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 Mercury IT Governance 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 `<ITG_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 Mercury IT Governance Server.

For information about the `kConfig.sh` script, see *kConfig.sh* on page 292. For information about the `kUpdateHtml.sh` script, see *kUpdateHtml.sh* on page 299.

Table 5-1. Special configuration parameters

Parameter	Description	Sample Value
com.kintana.core. server. DB_ CONNECTION_ STRING	When the JDBC_URL parameter is specified, the security identifier (SID) of the database on which the Mercury IT Governance 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. 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, Mercury IT Governance Center includes the domain name and user name in an FTP session to a Windows computer. If you use an FTP server that does not require the domain name, you can use this parameter to override the default functionality. For more information, contact Mercury Support.	WAR-FTPD
com.kintana.core. server.TEMP_DIR	This parameter defines a Mercury IT Governance Center temporary directory. This defaults to a `temp` subdirectory of the `logs` directory. If you use this parameter, make sure that you include 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 `<ITG_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
```

Tip

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 Mercury IT Governance Center user account can be used to log on to the remote host via the SSH session.
- Add the RSA certificate information of the remote host to the ssh `known_hosts` file, which is located in the `<ITG_Home>` directory.

To add the remote SSH host's RSA certificate to the Mercury IT Governance Server SSH `known_hosts` file:

1. Log on to the Mercury IT Governance Server as the Mercury IT Governance 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 Mercury IT Governance Server.
2. Add the generated public key to the remote `SSH Authorized_Key` file.
3. Configure the Mercury IT Governance 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 Mercury IT Governance Server:

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

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

   Note: Mercury IT Governance 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 `<ITG_Home>/<ITG_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 `/<ITG_USER_HOME_FOLDER>/.ssh` directory as `itg_id_rsa.pub`.

   Note: On the remote UNIX host, the `.ssh` folder is in the `/home/<ITG_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 Mercury IT Governance Server is to use to connect.
3. Change directory to ".ssh" directory under `<ITG_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 Mercury IT Governance Server to allow public key authentication from the Mercury IT Governance Server back to itself.

Configuring the Mercury IT Governance Server

To configure Mercury IT Governance 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=/<ITG_Home>/<ITG_USER>/.ssh/id_rsa`

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

Verifying Server Configuration

To verify the configuration:

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

   `ssh <USER_ID>@<REMOTE_HOST>`

   Note

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

3. On the Mercury IT Governance Server, start the Workbench.
4. On the shortcut bar, select **Environments > Environments.**

   The Environment Workbench page opens.

5. Click **New Environment.**

   The Environment: Untitled window opens.

6. In the **Environment Name** field, type the name of the remote host.

7. 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.

Warning

> If the RSA key authentication is configured correctly and working from the command line, the **Password** field can display a bogus password because it is not using the password to login. The Kintana Environment profile requires that the Password field contain a value.

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

Note

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

9. Click **Save.**
10. At the bottom left of the window, click **Check.**

    The Check Environment window opens.
11. In the left pane, expand the **Server** folder, and then click **SSH2 Server.**
12. Click **Check.**

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

Generating Password Security (Optional)

For password security, Mercury IT Governance 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 Mercury IT Governance Center installation, reside in `<ITG_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 `<ITG_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 Mercury IT Governance Center database schema)

   `DB_PASSWORD` (the password for the Mercury IT Governance Center database schema)

Warning

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 `<ITG_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 296.

Configuring Solaris and Linux Environments to Use Deployment Management

Mercury IT Governance Center can connect to a machine on which the environment variable TERM is set to dumb. To enable Mercury 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 Server

All Mercury IT Governance Center clients use the same URL to log on. To specify the URL for Mercury IT Governance 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 Mercury IT Governance Server after installation, log on to a client machine as administrator.

To log on to Mercury IT Governance Center as administrator:

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

   The Mercury IT Governance Center logon screen opens.

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

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

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

4. Click **Submit.**

The Mercury IT Governance Center standard interface opens.

For more information about configuring licenses and user access, see the *Security Model Guide and Reference* manual.

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 database administrator.

The recommendations provided in this section are based on the assumption that Mercury IT Governance Center is the only application using 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 Mercury IT Governance Center system performance. It also provides parameter settings recommended for the Mercury IT Governance 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`. Mercury recommends that you to 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.

_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.

Note

Default selectivity has changed from earlier release, 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, Mercury IT Governance 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 `8` (expressed in 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).

The value must be at least the size of one granule (KB or MB). Smaller values are automatically rounded up to the value of a granule size.

You cannot specify a value of zero (0) for this parameter. Zero is the size of the default pool for the standard block size, which is the block size for the system tablespace.

Recommended Setting

Specify a DB_CACHE_SIZE parameter value of at least 300 (expressed in MB).

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.

Note

If multiple Mercury IT Governance 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

Set the LOG_BUFFER parameter value based on the number of concurrent users, according to the following guidelines:

- For systems with fewer than 50 concurrent users, set the parameter value to 512 (expressed in KB).
- 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 zero (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.

For more information about the `NLS_LENGTH_SEMANTICS` initialization parameter, see the following Web pages:

- www.lc.leidenuniv.nl/awcourse/oracle/server.920/a96529/ch2.htm#104327
- www.lc.leidenuniv.nl/awcourse/oracle/server.920/a96529/ch3.htm#54128

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 value 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_MODE

The `OPTIMIZER_MODE` parameter value determines the default behavior for choosing an optimization approach for executing a query.

Recommended Settings

For Oracle 9i databases, set the `OPTIMIZER_MODE` parameter value to `CHOOSE`.

For Oracle 10G (or later) databases, set the `OPTIMIZER_MODE` parameter value to `ALL_ROWS`. This is the default Oracle setting.

Database statistics gathering is required. For information about collecting database statistics, see *Collecting Database Schema Statistics* on page 167.

PGA_AGGREGATE_TARGET

The PGA_AGGREGATE_TARGET parameter value determines the aggregate Program Global Area (PGA) memory available to all Mercury IT Governance 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 based on the total amount of memory available for the Oracle instance. You can then fine-tune the value at the instance level.

Calculate the initial value for the parameter as follows:

```
PGA_AGGREGATE_TARGET = (total_mem * 80%) * 40%.
```

Where total_mem is the total amount of physical memory available on the system for the Oracle instance.

PROCESSES

The PROCESSES parameter value determines the maximum number of operating system user processes that can simultaneously connect to the Oracle database. Mercury IT Governance 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 Mercury IT Governance Server rarely uses more than 25 database connections. If a Mercury IT Governance Server cluster configuration is used, each Mercury IT Governance Server might use 25 database connections.

For single-server configurations, set the parameter value to 45 (the default). For a Mercury IT Governance 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`.

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. Mercury 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 `600 MB`.

TIMED_STATISTICS

The TIMED_STATISTICS parameter value determines whether time-related statistics are collected. Setting this parameter helps to ensure that information about the database and timing of internal activities is available. The overhead of enabling this function is minimal, and the data obtained can be extremely helpful.

Recommended Setting

Set the TIMED_STATISTICS parameter value to TRUE.

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.

Oracle Database Configuration Examples

This section provides configuration examples for Oracle 9i and Oracle 10G. *Table 5-2* lists example parameters for Oracle 91.

Table 5-2. Example parameters for Oracle 9i (page 1 of 3)

Category/Parameter	Value
Cache and I/O	
db_block_size	8192
db_cache_size	2G
db_file_multiblock_read_count	16
Cursors and Library Cache	
open_cursors	1000
Database Identification	
db_domain	koka.com
db_name	ardent
Diagnostics and Statistics	
background_dump_dest	/opt/oracle/app/oracle/admin/ardent/bdump
core_dump_dest	/opt/oracle/app/oracle/admin/ardent/cdump
timed_statistics	TRUE
user_dump_dest	/opt/oracle/app/oracle/admin/ardent/udump
File Configuration	
control_files	(“/oramisc/oradata/ardent/control01.ctl”, “/oramisc/oradata/ardent/control02.ctl”, “/oramisc/oradata/ardent/control03.ctl”)
Instance Identification	
instance_name	ardent
Job Queues	
job_queue_processes	10
MTS	
dispatchers	“(PROTOCOL=TCP)(SERVICE=ardentXDB)”

Table 5-2. Example parameters for Oracle 9i (page 2 of 3)

Category/Parameter	Value
Miscellaneous	
aq_tm_processes	1
compatible	9.2.0
Optimizer	
hash_join_enabled	TRUE
query_rewrite_enabled	FALSE
star_transformation_enabled	FALSE
Pools	
java_pool_size	33554432
large_pool_size	8388608
shared_pool_size	1G
Processes and Sessions	
processes	300
Redo Log and Recovery	
fast_start_mttr_target	300
log_buffer	1048576
Security and Auditing	
remote_login_passwordfile	EXCLUSIVE
Sort, Hash Joins, Bitmap Indexes[a]	
#pga_aggregate_target	25165824
pga_aggregate_target	1500M
workarea_size_policy	auto
#sort_area_size	1500000
#sort_area_retained_size	1000000

a. Mercury recommends that, instead of setting these separately, you instead use the WORKAREA_SIZE_POLICY parameter. For information about this parameter, see *Table A-1* on page 240.

Table 5-2. Example parameters for Oracle 9i (page 3 of 3)

Category/Parameter	Value
System Managed Undo and Rollback Segments	
undo_management	AUTO
undo_retention	10800
undo_tablespace	UNDOTBS1
open_links	20
timed_statistics	true
optimizer_features_enable	9.2.0
Archive Log Parameters	
log_archive_start	true
log_archive_dest	“/oraarch/archive/ardent”
#log_archive_dest_1	“location=/oraarch/archive/ardent”
log_archive_format	%t_%s_ardent.arc

Oracle 10G: Example

Table 5-3 lists example parameters for Oracle 10G.

Table 5-3. Example parameters for Oracle 10G (page 1 of 3)

Category/Parameter	Value
Cache and I/O	
db_block_size	8192
db_file_multiblock_read_count	16
Cursors and Library Cache	
open_cursors	1000
Database Identification	
db_domain	koka.com
db_name	ardent

Table 5-3. Example parameters for Oracle 10G (page 2 of 3)

Category/Parameter	Value
Diagnostics and Statistics	
background_dump_dest	/opt/oracle/app/oracle/admin/ardent/bdump
core_dump_dest	/opt/oracle/app/oracle/admin/ardent/cdump
timed_statistics	TRUE
user_dump_dest	/opt/oracle/app/oracle/admin/ardent/udump
File Configuration	
control_files	(“/oramisc/oradata/ardent/control01.ctl”, “/oramisc/oradata/ardent/control02.ctl”, “/oramisc/oradata/ardent/control03.ctl”)
Instance Identification	
instance_name	ardent
Job Queues	
job_queue_processes	10
MTS	
dispatchers	“(PROTOCOL=TCP)(SERVICE=ardentXDB)”
Miscellaneous	
aq_tm_processes	1
compatible	10.0.0
Optimizer	
hash_join_enabled	TRUE
query_rewrite_enabled	FALSE
star_transformation_enabled	FALSE
Pools	
sga_target	3G
Processes and Sessions	
processes	300

Table 5-3. Example parameters for Oracle 10G (page 3 of 3)

Category/Parameter	Value
Redo Log and Recovery	
fast_start_mttr_target	300
log_buffer	1048576
Security and Auditing	
remote_login_passwordfile	EXCLUSIVE
Sort, Hash Joins, Bitmap Indexes (Oracle does not recommend sort_area_size; use pga_aggregate_target instead.)	
pga_aggregate_target	1500M
workarea_size_policy	auto
#sort_area_size	1500000
#sort_area_retained_size	1000000
System Managed Undo and Rollback Segments	
undo_management	AUTO
undo_retention	10800
undo_tablespace	UNDOTBS1
open_links	20
timed_statistics	true
optimizer_features_enable	10.0.0
Archive Log Parameters	
log_archive_start	true
log_archive_dest	“/oraarch/archive/ardent”
#log_archive_dest_1	“location=/oraarch/archive/ardent”
log_archive_format	%t_%s_ardent.arc

Granting Select Privileges to v_$session

If you want Mercury IT Governance 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 Mercury IT Governance Center database schema, and then issue the following SQL statement:

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

Note

You typically assign this grant during Mercury IT Governance Center installation or upgrade.

Generating Database Links (Oracle Object Migration)

Mercury IT Governance Center can use database links to communicate with other databases. Usually a database link created and associated with a particular environment in Mercury IT Governance 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 Mercury Deployment Management activities
- Some Mercury Deployment Management Extensions, such as the Extension for Oracle E-Business Suite, to facilitate 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 Mercury IT Governance Center database), issue an SQL statement similar to the following in the Mercury IT Governance 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:

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

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

Note

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 Mercury IT Governance 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 Mercury IT Governance 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 Workbench with Secure RMI (Optional)

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

- Specify the complete RMI URL, in the following format, when you start the 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 Mercury IT Governance Workbench interface. When a user starts the Mercury IT Governance 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.

Note

Mercury recommends that you leave the `WORKBENCH_PLUGIN_VERSION` parameter default value.

If users who must access the 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`.

Note

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

For information about how to configure the Workbench as a Java application, see the following section, *Configuring the Workbench as a Java Application*.

Configuring the Workbench as a Java Application

In most Mercury IT Governance Center installations, the Mercury IT Governance Workbench interface runs in the Java Virtual Machine (JVM) using a supported Web browser.

Organizations running on UNIX platforms that do not provide Java support in their available Web browsers (but do support JVM on their native operating system) can run the Workbench interface as a Java application.

Note

If you plan to run the Workbench interface as an application, check to make sure that client files are deployed correctly. If you upgrade Mercury IT Governance Center or install a service pack on the Mercury IT Governance Server, the client files might also require patching.

Copying the JAR Files

To run the Workbench interface as an application, copy the following JAR files to a single directory to which the client machine has access:

```
<ITG_Home>/server/kintana/deploy/itg.war/WEB-INF/lib/knta_
classes.jar
<ITG_Home>/server/kintana/deploy/itg.war/WEB-INF/lib/
libraries.jar
<ITG_Home>/server/kintana/deploy/itg.war/WEB-INF/lib/
oracle-jdbc.jar
```

Creating the Batch File

After you copy the required JAR files to a directory that the client machine can access, create a script to run the Workbench.

Creating kintana.bat for Windows

To create and run the batch file for use on a Windows client:

1. Create a batch file named `itg_workbench.bat` with the following content:

   ```
   @ECHO OFF

   REM
   REM Change to your client install directory.
   REM
   cd /D e:\Programs\Kintana
   set classpath=.
   set classpath=%classpath%;.\knta_classes.jar
   set classpath=%classpath%;.\libraries.jar
   set classpath=%classpath%;.\oracle-jdbc.jar

   REM
   REM Change to the host and RMI port of your primary Mercury
   ITG Server.
   REM
   java com.kintana.core.gui.LogonApplet
   <your_company.domain.com>:1200
   ```

2. Edit the `cd` command in the batch file to use the directory that contains the JAR files.

3. Edit the `java` command to reflect the hostname and RMI port of the primary server.

4. If you have any Mercury Deployment Management Extensions installed, edit the file to include the `extension_name.jar` files in the `<ITG_Home>/server/kintana/deploy/itg.war/html/client` directory.

5. Save the file.
6. Run the `itg_workbench.bat` file that you created.

Creating and Running kintana.sh for UNIX

To create and run the batch file for use on a UNIX client for SDK:

1. Create a batch file named `itg_workbench.sh` with the following content:

```
#!/bin/sh
#
# Change to your client install directory.
#
cd /usr/local/Kintana

CLASPATH=.
CLASSPATH=$CLASSPATH:./knta_classes.jar
CLASSPATH=$CLASSPATH:./libraries.jar
CLASSPATH=$CLASSPATH:./oracle-jdbc.jar
export CLASSPATH
#
# Change to the host and RMI port of your primary Mercury IT
# Governance Server.
#
java com.kintana.core.gui.LogonApplet
   <company.domain.com>:1200
```

2. Edit the `cd` command in the batch file to use the directory where the JAR files are located.
3. Edit the `java` command to reflect the hostname and RMI port of the primary server.
4. If you have any Mercury Deployment Management Extensions installed, edit the file to include the `extension_name.jar` files in the `<ITG_Home>/server/kintana/deploy/itg.war/html/client` directory.
5. Save the file.
6. Run the `itg_workbench.sh` script that you created.

Using the Workbench: What Users Need to Know

This section provides the information that users require to start the Workbench on client machines. It also includes information on how to address JVM-related problems that can arise on client machines.

Installing and Configuring the Java Plug-In on Client Machines

Table 5-4 lists the default settings for the server configuration parameters related to the Java plug-in.

Table 5-4. Server parameters related to the Java plug-in

Parameter	Description and Default Value
JAVA_PLUGIN_XPI_PATH	Specifies the Web location for downloading the cross-platform Java plug-in installer for Firefox browsers. Default: java.com/en/download/windows_xpi.jsp

For information about the Java plug-in supported for the current Mercury IT Governance Center release, see the *System Requirements and Compatibility Matrix* document. For more information about the server parameters in *Table 5-4*, see *Server Configuration Parameters* on page 237.

Setting the Default Web Browser

To run the 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 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 Workbench uses the default Web browser.

Starting the Workbench on a Client Machine

To start the Workbench from the Mercury IT Governance Center standard (HTML) interface:

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

Note

If a pop-up blocker is installed and enabled on the Web browser, the Workbench cannot open. The user can configure the blocker to allow pop-ups from Mercury IT Governance 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 Workbench:

- The 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, check to 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 Mercury IT Governance 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 Workbench functions correctly.

What to Do Next

If you plan to perform any of the optional installations described in *Optional Installations* on page 65 (for example, if you are going to install a Mercury 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 7, *Maintaining the System,* on page 139.

Chapter

Advanced System Configuration

In This Chapter:

About this Chapter

The sections in this chapter provide information about installations, integrations and configurations ancillary to the Mercury IT Governance Center setup. It includes information about installing optional products such as Mercury Deployment Management Extensions and Accelerators, and the service packs to be delivered occasionally after the main Mercury IT Governance Center release. You can also find much useful configuration and integration information.

Integrating with an LDAP Server

You can integrate Mercury IT Governance 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 Mercury IT Governance Server authenticates users directly to the LDAP directory server, and does not store passwords in the Mercury IT Governance Center database.

Note

This section addresses LDAP directory server integration with a Mercury IT Governance 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 Mercury IT Governance Server authenticates users in the following way:

- The Mercury IT Governance 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 Mercury IT Governance Server performs an anonymous authentication.
- The Mercury IT Governance 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 Mercury IT Governance 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 Mercury IT Governance Server tries to authenticate against all LDAP servers until it succeeds. If the referral option has been enabled, and the user is not logged on to the primary server, the Mercury IT Governance Server also checks the referral server for authentication.

To integrate Mercury IT Governance 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 Mercury IT Governance Center users, in the following format:

     `CN=Users,DC=ITGAD,DC=com`
   - LDAP user account and password (The Mercury IT Governance Server uses this information to look up users.)
2. From `<ITG_Home>/bin` on the Mercury IT Governance Server, run the `kConfig.sh` script.
3. Provide the information that you collected in step 1 for the following server directives for an LDAP server that is not SSL-enabled:

   `AUTHENTICATION_MODE=ITG,LDAP`

   `LDAP_URL`. Specify the comma-delimited list of LDAP URLs that the Mercury IT Governance 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 Mercury IT Governance Center password on the LDAP server.

   Example: `#!#ghengis#!#.`

Warning

If you run the `kConfig.sh` script, the Mercury IT Governance Server configuration utility automatically encrypts this password. In this case, you must type the exact password string without the "`#!!#`.

If you modify the `server.conf` file manually, then you must surround the encrypted password with the `#!!#` characters.To edit the password manually, surround the encrypted password with `#!!#` delimiters.

`KINTANA_LDAP_ID`. Specify the Mercury IT Governance Center account on the LDAP server. The Mercury IT Governance Server uses this to bind to the LDAP server.

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

The script that runs makes the required changes to the `server.conf` file, encrypts the LDAP password, and updates the required Mercury IT Governance startup files.

4. On the Mercury IT Governance Server, back up the existing `LdapAttribute.conf` file, which is located in the `<ITG_Home>/integration/ldap` directory.

   The `LdapAttribute.conf` file is required for importation and for user authentication. The `<ITG_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 `<ITG_Home>/integration/ldap/LdapAttribute_AD.conf` file to the `LdapAttribute.conf` file.

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

6. To enable entity ownership and security, import the LDAP user into the `KNTA_USERS` table on the Mercury IT Governance Server, as follows:

   a. Use the Import Users report to import the LDAP user into Mercury IT Governance Server.

      For instructions on how to run the Import Users report, see the document *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 Mercury Knowledge Base article:

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

b. Next to the **LDAP Import?** option, click **Yes.**

For more information about server parameters related to LDAP integration, see *LDAP Attribute Parameters* on page 286.

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

Enabling LDAP Authentication over SSL Using Passwords

To enable LDAP authentication over SSL using passwords:

1. Set the following `server.conf` parameters:

- `LDAP_SSL_PORT`
- `LDAP_KEYSTORE`
- `LDAP_KEYSTORE_PASSWORD`

1. Install the server's certificate in the JRE database of trusted certificates.
2. Check to make sure that the parameters in the `LdapAttribute.conf` file are set correctly.

For more information about `server.conf` parameters, see *Table A-1* on page 240. For more information about `LdapAttribute.conf` parameters, see *Table A-3* on page 286.

Configuring an External Web Server

Mercury recommends that you use the internal Web server built into the Mercury IT Governance Server unless you have the special Web server requirements described in *Single-Server/External Web Server Configuration* on page 25 and *Server Cluster/External Web Server Configuration* on page 27. The following sections provide information about how to configure an external Web server to work with a Mercury IT Governance Center server cluster.

For a list of external Web servers that Mercury IT Governance Center supports, see the *System Requirements and Compatibility Matrix* document.

Using an External Web Server for Multiple Mercury IT Governance Center Instances

You cannot use a single Web server installation on a machine running Windows for multiple instances of Mercury IT Governance Center. The Windows NT 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_redirector.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 Mercury IT Governance Center instances, Mercury 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 such as Sun ONE Web Server, Microsoft Internet Information Services (IIS), or Apache.
2. Choose an external Web port.
3. Configure a `workers.properties` file.
4. Configure a `uriworkermap.properties` file.
5. Configure the external Web server.
6. Integrate the external Web server with the Mercury IT Governance Server.
7. Enable cookie logging on the external Web server. (This step is optional.)

Choosing an External Web Port

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

Note

If you are integrating with an external Web server, you must set the `EXTERNAL_WEB_PORT` parameter on the Mercury IT Governance 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 Mercury IT Governance Server(s), including the machine name, ports, and load balance. The external Web server uses this information to direct traffic to Mercury IT Governance Center applications, as required.

The following sections describe how to configure workers properties files for:

- Sun ONE Web Server (`workers.properties` file)
- Microsoft IIS 6.0 (`workers.properties` file)
- Apache-based servers such as Apache 2.0, HP Web Server, and IBM HTTP Server (`workers.properties` file)

Configuring the workers.properties File for a Single Server

The following example shows the contents of a sample `workers.properties` file for a single-server configuration.

Bear in mind the following two conditions:

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

Sample File

```
# Defines a load balancer to handle requests to the Mercury IT
# Governance Server.
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 ITG server instance on k1.acme.com. The worker
# name is the value between the first and second period
# (server1, in this case).
# This value must match the ITG instance name defined in the
# KINTANA_SERVER_NAME parameter of the server.conf file. Please
# note that for a clustered setup, each ITG node has it's own
# KINTANA_SERVER_NAME parameter.
# Add the worker name to the balanced_workers list below.
worker.server1.host=k1.acme.com
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
# the connection_pool_size equal to the value for the
# RqThrottle parameter in Web server's magnus.conf file.
# Keep connection_pool_minsize at 1 and connection_pool_timeout
# at 600. Mercury recommends that you not use these parameters
# with Apache-based servers, including IBM HTTP Server and
# HP Web Server, or with 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 ITG server instance on k2.acme.com.
# worker.server2.host=k2.acme.com
# worker.server2.port=8010
# worker.server2.type=ajp13
# worker.server2.lbfactor=1
# The following three parameters are required for
# Netscape-based Web servers such as Sun ONE web server. Set
# the connection_pool_size equal to the value for the
# RqThrottle parameter in Web server's magnus.conf file.
# Keep connection_pool_minsize at 1 and
# connection_pool_timeout at 600. Mercury recommends that you
# not use these parameters with Apache-based servers, including
# IBM HTTP Server, HP Web Server, or with Apache itself.
#worker.server1.connection_pool_size=128
#worker.server1.connection_pool_minsize=1
#worker.server1.connection_pool_timeout=600

# Defines a load balancer. Be sure to list all servers in the
# Mercury ITG server cluster in the balanced_workers group.
worker.load_balancer.type=lb
worker.load_balancer.balanced_workers=server1
# worker.load_balancer.balanced_workers=server1, server2a
# List all servers in the server cluster in the
# balanced_workers group.
```

```
# Optional. Define a special "status" worker to allow
# monitoring of jk plugin status. If enabled, add it to the
# list of available workers
# worker.jkstatus.type=status
```

For more information about how to configure a server cluster, see *Configuring a Server Cluster* on page 126.

Configuration

To configure a `workers.properties` file:

1. Open the `workers.properties` file in a text editor such as Notepad.

   You can find the `workers.properties` file in the `<ITG_Home>/integration/webserverplugins/configuration` directory.

2. Set the `worker.list` parameter to `load_balancer`.

3. For the single server (or for each Mercury IT Governance Server in a cluster), configure the following values:

   a. Set `<worker.name>` to the name of Mercury IT Governance 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.

   Note: For a clustered setup, each Mercury IT Governance Server has its own `KINTANA_SERVER_NAME` parameter.

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

   Note: If the Mercury IT Governance 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 Mercury IT Governance 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).
6. If you want to enable the JK status page (optional), then 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 (IIS and Apache-based Servers Only)

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

```
# /itg/* must be mapped to one of the
# workers/itg/*=load_balancer

# The status page can be accessed at
# http://<web_server_host>:<web_server_port>/jkmanager
# To enable 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>`. When the Web server processes a URL that matches `<URL_PATTERN>`, worker `<WORKER_NAME>` is used to serve this request. The worker name (`<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 Mercury IT Governance Center–supported Web servers:

- Sun ONE Web Server
- Microsoft IIS
- Apache Web server

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

Configuring the Sun ONE Web Server

To configure the Sun ONE Web Server to run as the external Web server for the Mercury IT Governance Server:

1. Connect to the Sun ONE System administration server and create a new server named "ITG."

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

2. Stop the Mercury IT Governance Server.

   For information about how to stop the Mercury IT Governance Server, see *Starting and Stopping the Mercury IT Governance Server* on page 68.

3. Put the `workers.properties` file you that configured (see *Configuring the Workers Properties File* on page 112) 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="<ITG_Home>/
workers.properties" log_level="error" log_file="<path_to_
log_files>/itg_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="/itg/*"name="itg-servlet"
   ```

7. Place the following text after the end of the "Object" section (that is, after *`</Object>`*):

   ```
   <Object name="itg-servlet">
   Service fn="jk_service" worker="load_balancer"
   </Object>
   ```

   The `"itg-servlet"` strings must match.

Note

Note that `worker` attribute specifies the name of the JK worker used to serve requests with URLs that match the `path` attribute, which is `/itg/*` 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. Create a new (or select an existing) Web site under your IIS to integrate with the Mercury IT Governance Server.

   c. Create a new (or select an existing) directory in your file system in which to store the integration-related files.

      In this procedure, this directory is `<ISAPI_REDIRECTOR_HOME>`.

   d. Copy the `workers.properties` file, the `uriworkermap.properties` file, and `<ITG_Home>/integration/webserverplugins/iis/windows/x86-32/isapi_redirector.dll` file to the `<ISAPI_REDIRECTOR_HOME>` directory you created (or selected) in step c.

   e. Right-click the Web site you created (or selected) in step b.

   f. Select **New > Virtual Directory.**

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

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

   i. Click **Next.**

   j. On the Web Site Content Directory page, under **Directory,** type the full path of the `<ISAPI_REDIRECTOR_HOME>` directory that contains the `isapi_redirect.dll` file (the directory you created or selected in step c).

   k. Click **Next.**

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

m. Click **Finish.**

An example of this directory is `c:\inetpub\scripts`. Depending on the IIS root directory configuration, the drive and directory may vary. This directory must have run permission.

2. Configure a `workers.properties` file and a `uriworkermap.properties` file, as described in *Configuring the Workers Properties File* on page 112 and *Configuring the uriworkermap.properties File (IIS and Apache-based Servers Only)* on page 115.

3. To configure IIS to load `isapi_redirector.dll` as a filter:

   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 the value `/jakarta/isapi_redirector.dll`.

      iii. Add a string value with the name `worker_file` a value which is the full path to the workers.properties file, i.e. `<ISAPI_REDIRECTOR_HOME>\workers.properties` (for example, `c:\inetpub\scripts\workers.properties`).

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

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

      v. Add a string value with the name `log_file` and the directory path where you want the log file to reside. (Include the log file name, for example, `c:\ITG\isapi.log`.)

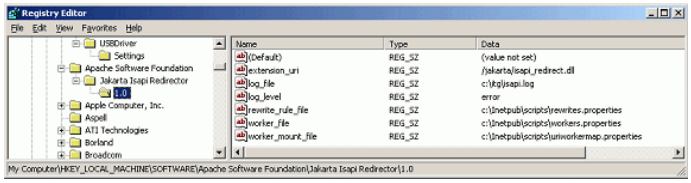
      vi. Add a string value with the name `worker_mount_file` and a value that is the full path to your `uriworkermap.properties` file, which is the *<ISAPI_REDIRECTOR_HOME>* directory (for example, `c:\inetpub\scripts\uriworkermap.properties`).

      vii. Create an empty file named `rewrites.properties`, and save it to the *<ISAPI_REDIRECTOR_HOME>* directory. Add a string value with the name `rewrite_rule_file` and assign it a value that is the full path to new `rewrites.properties` file, which is located in the *<ISAPI_REDIRECTOR_HOME>* directory (for example, `c:\inetpub\scripts\rewrites.properties`).

Note

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:



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

Note

Perform the following steps at the Web sites level.

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

   The Properties dialog box opens.

d. Click the **ISAPI Filter** tab.

e. Click **Add.**

   The Filter Properties window opens.

f. In the **Executable** field, enter the full path to the `isapi_redirector.dll` file (`<`*`ISAPI_REDIRECTOR_HOME`*`>\isapi_redirector.dll`).

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

h. Click **OK.**

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

   e. Click **Add.**

f. Type the path to the `isapi_redirector.dll` file (`<ISAPI_REDIRECTOR_HOME>\isapi_redirector.dll`).

g. Click **OK.**

5. Restart the IIS service.

Note

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

6. Start the Mercury IT Governance 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 the Apache-Based Web Server (Apache 2.0, IBM HTTP Server, or HP Web 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 2.0.

Compiling a Binary of JK

Configuring Apache on UNIX requires a dynamically linkable JK module binary named `mod_jk.so`. In most cases, the `<ITG_Home>/integration/webserver` 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.

If a precompiled binary is not available, then complete the following steps. Otherwise, proceed to *Configuring Apache 2.0* on page 122.

To compile a binary of JK:

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

   http://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 step 4 on page 122.

4. Run the `make` command to build the Apache module that forwards requests from the Apache Web server to the Mercury IT Governance Server using the AJP13 protocol.

Note

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

Configuring Apache 2.0

To configure the Apache 2.0 module:

1. Copy the Apache 2.0 module `mod_jk.so` from `<ITG_HOME>/integration/webserverplugins/apache/<os>/<platform>` to the Apache module directory (typically, `<APACHE_HOME>/modules` or `<APACHE_HOME>/libexec`).

2. Copy the `workers.properties` and `uriworkermap.properties` files from `<ITG_HOME>/integration/webserverplugins/configuration` to the Apache configuration directory (typically, `<APACHE_HOME>/conf`).

   Make sure that the name of the worker mapped to `/itg/*` pattern in the `uriworkermap.properties` file matches the name of the worker defined in `workers.properties` file. This worker must also be listed in the `worker.list` directive in the `workers.properties` file.

3. Configure a `workers.properties` file and a `uriworkermap.properties` file.

   For detailed instructions, see *Configuring the Workers Properties File* on page 112 and *Configuring the uriworkermap.properties File (IIS and Apache-based Servers Only)* on page 115.

4. Go to the Apache `conf` directory (typically, `<APACHE_HOME>/conf`), and open the `httpd.conf` file in a text editor such as Notepad.

5. Add the following lines of text to the `httpd.conf` file:

```
LoadModule jk_module <RELATIVE_MODULES_PATH>/mod_jk.so
JkWorkersFile <RELATIVE_CONF_PATH>/workers.properties
JkMountFile    <RELATIVE_CONF_PATH>/uriworkermap.properties
JkLogFile     <RELATIVE_LOGS_PATH>/jk.log
JKLogLevel    ERROR
```

   `<RELATIVE_MODULES_PATH>` is the path to the modules directory relative to `<APACHE_HOME>` (typically modules or libexec). `<RELATIVE_CONF_PATH>` is the path to the configuration directory relative to `<APACHE_HOME>` (typically conf). `<RELATIVE_LOGS_PATH>` is the path to the configuration directory relative to `<APACHE_HOME>` (typically logs).

   A typical `httpd.conf` modification looks as follows:

```
LoadModule jk_module modules/mod_jk.so
JkWorkersFile  conf/workers.properties
JkMountFile    conf/uriworkermap.properties
JkLogFile     logs/jk.log
JKLogLevel    ERROR
```

Note

The `httpd.conf` file is complex and highly configurable. Correct placement of these lines in the file depends on your Web server setup. You can add the lines in any section of the file that is related to the domain of the Web server that you are integrating with Mercury IT Governance Center.

For example, if you set up Apache to run several virtual servers, add these lines of text to the section of the file that controls the settings for the virtual server.

6. To implement your changes, restart the Apache server.

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 the `httpd.conf` file and exit the text editor.

Integrating an External Web Server with a Mercury IT Governance Server

To integrate the external Web server with the Mercury IT Governance Server, perform the following tasks:

1. Stop the Mercury IT Governance Server.

   For information about how to do this, see *Starting and Stopping the Mercury IT Governance Server* on page 68.

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 `<`*`ITG_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.

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 `BASE_URL`, see Appendix A, *Server Configuration Parameters,* on page 237. For more information about `kUpdateHtml.sh`, see *kUpdateHtml.sh* on page 299.

Verifying the Integration

To verify the integration between the external Web server and the Mercury IT Governance Server:

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

Note

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

For information about how to start the Mercury IT Governance Server, see *Starting and Stopping the Mercury IT Governance Server* on page 68. For information about supported browsers, see the document *System Requirements and Compatibility Matrix.*

Configuring a Server Cluster

This section provides the following information about server clustering in the Mercury IT Governance 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 Mercury IT Governance Server cluster, review the information provided in Chapter 2, *System Overview,* on page 19, particularly *Server Cluster Configurations* on page 26. The concepts described in this section are key to understanding configuring server clusters.

KINTANA_SERVER_NAME and the <*ITG_Home*>/server directory

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

Server nodes are the individual Mercury IT Governance Servers that comprise the server cluster. Each node, or server, in a cluster requires a separate directory in the `<ITG_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 `<ITG_Home>/server` must have a corresponding `KINTANA_SERVER_NAME` defined in `server.conf`, all with the same assigned value.

Note

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 Mercury IT Governance 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 6-1 on page 127 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 237.

Table 6-1. 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 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
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

Table 6-1. 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 .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

Note

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 IT Governance Server for integration with an external Web server in single-server mode.
2. Stop the Mercury IT Governance Server.

   For information about how to stop the Mercury IT Governance Server, see *Starting and Stopping the Mercury IT Governance Server* on page 68.
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 112.

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 6-1* on page 127):

- 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 Mercury IT Governance Server.

   For information about how to stop the Mercury IT Governance Server, see *Starting and Stopping the Mercury IT Governance Server* on page 68.

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 112.

   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 IT Governance 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
```

4. Create the new `<ITG_Home>/server` directory.

   Make a copy of the first server directory (the entire directory) at the same level as the first one.

   Example:

```
<ITG_Home>
+ server
      + node1
      + node2
```

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
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 `<ITG_Home>/bin`, run `kUpdateHtml.sh`.

7. If you have additional nodes in your cluster, repeat step 1 through step 6.

8. If the Mercury IT Governance Server is running in a Windows environment, start it using the Windows service called "`Mercury ITG 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 `<ITG_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 135.

External Web Server, Multiple Machines

In a server cluster, an `<ITG_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 Mercury IT Governance 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 110.

2. Stop the Mercury IT Governance Server.

   For information about how to stop the Mercury IT Governance Server, see *Starting and Stopping the Mercury IT Governance Server* on page 68.

3. Stop the external Web server.

4. Make sure that the common directories that the servers use (`<ITG_Home>/logs`, `<ITG_Home>/reports`, `<ITG_Home>/attachments`, and `<ITG_Home>/transfers`) are shared.

Note

> 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 IT Governance 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:

```
@node
# 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

# ORACLE_HOME of machine2
com.kintana.core.server.ORACLE_HOME=/opt/oracle

# <ITG_Home> for this node
com.kintana.core.server.BASE_PATH=/home/ITG

# 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
```

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 `<ITG_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 `<ITG_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:

   ```
   <ITG_Home>
      server/
         node1
   ```

   The directories on machine2 could be:

   ```
   <ITG_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 Mercury IT Governance 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 `<ITG_Home>/bin`, run `kConfig.sh`.

       The configuration wizard starts up.

    b. Select **Configure Windows Services.**

    c. Follow the prompts to create the service.

Note

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 Mercury IT Governance Server cluster configuration. A hardware load balancer is similar to an HTTP reverse-proxy server and forwards HTTP requests.

All Mercury IT Governance 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.

Note

Sticky sessions are required for hardware load balancing in the Mercury IT Governance Center environment.

Starting and Stopping Servers in a Cluster

If you stop any server in a Mercury IT Governance Server cluster, the Mercury IT Governance Server cluster continues to operate as long as at least one server in the cluster is running. If a server stops, the Mercury IT Governance Web server module detects that the server is unavailable and stops sending it HTTP requests. When the server becomes available again, the Mercury IT Governance 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 Mercury IT Governance Server* on page 68.

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 "`Mercury ITG <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
```

Note

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, check to 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 `<ITG_Home>/server/<server name>/log` for errors.

Example:

```
> cd <ITG_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, check to 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.

Warning

If several cluster environments (for example, for Development, Test, and Production) are on the same network segment, you must change the `MULTICAST_IP / MULTICAST_PORT` parameters in the `server.conf` file, and change the corresponding setting in `cache.conf` file. Otherwise, the cluster environments will conflict.

If clusters other than those related to Mercury IT Governance Center are set up, and these are using the same multicast ip/port, the environment may also conflict.

For information about server reports and how to run them, see *Running Server Reports from the Admin Tools Window* on page 144 and *Running Server Reports from the Command Line* on page 148.

Chapter

Maintaining the System

In This Chapter:

Overview of Administration Tools and System Maintenance

Two kinds of administration tools and facilities are available to Mercury IT Governance 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 Workbench

  These tools include:

  - Admin Tools let you submit and view server reports
  - SQL Runner lets you submit SQL statements against the Mercury IT Governance 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 Mercury IT Governance Center instances

Administration Tools in the Standard Interface

The Mercury IT Governance Center standard interface includes tools that you can use to:

- View running reports

Note

For information about viewing running reports, see the *Reports Guide and Reference.*

- View running executions
- View interrupted executions

You access these tools in the standard interface through the **Administration** menu.

Viewing Running Executions

To view running executions:

1. Log on to Mercury IT Governance 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 Mercury IT Governance 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.**

   The **Failed Executions** section lists the details of the selected interrupted execution.

Server Tools In the Workbench

You can use the server tools in the Workbench to:

- View the technical status of the Mercury IT Governance 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 7-1 lists the names and descriptions of the three access grants that give users various levels of access to the Server Tools window.

Table 7-1. Server tools access grants

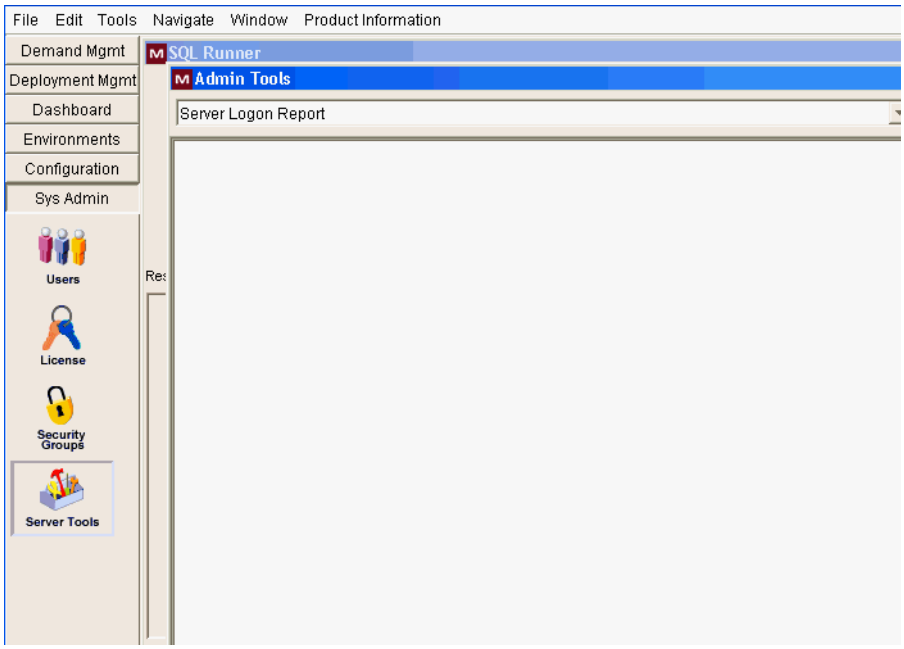
Access Grant	Description
Sys Admin: View Server Tools	Lets the user view the Admin Tools and SQL Runner windows in read-only mode.
Sys Admin: Server Tools: Execute Admin Tools	Lets the user: ■ Run server reports in the Admin Tools window ■ View the SQL Runner window in read-only mode
Sys Admin: Server Tools: Execute SQL Runner	Lets the user: ■ Run SQL queries in the SQL Runner window ■ View the Admin Tools window in read-only mode

For more information about security groups and access grants, see the document *Security Model Guide and Reference.*

Accessing and Using the Workbench Server Tools

To access the server tools in the Workbench:

- On the Workbench shortcut bar, select **Sys Admin > Server Tools.**



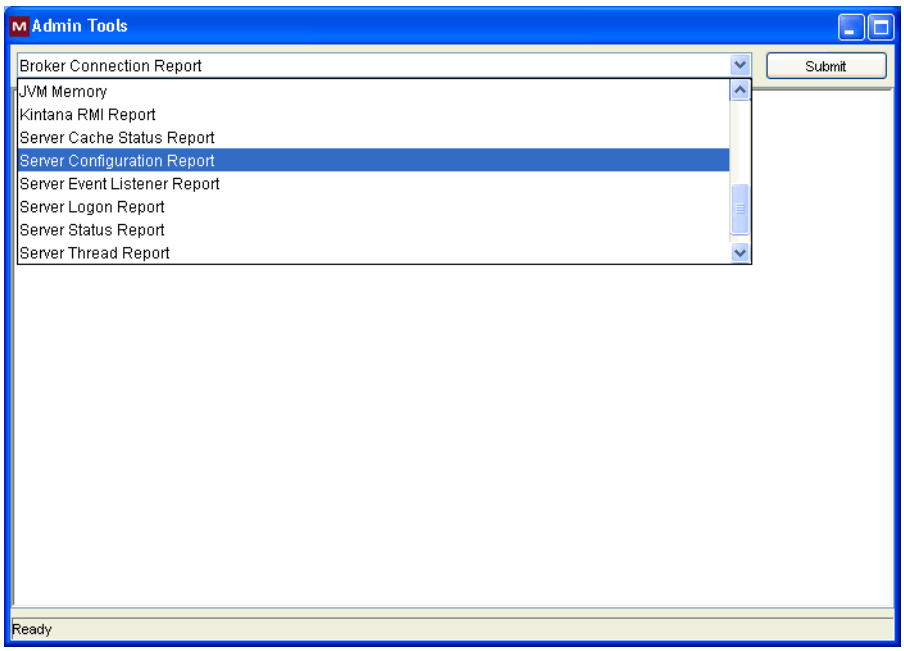
The Admin Tools window and the SQL Runner open.

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 7-2* on page 146 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.

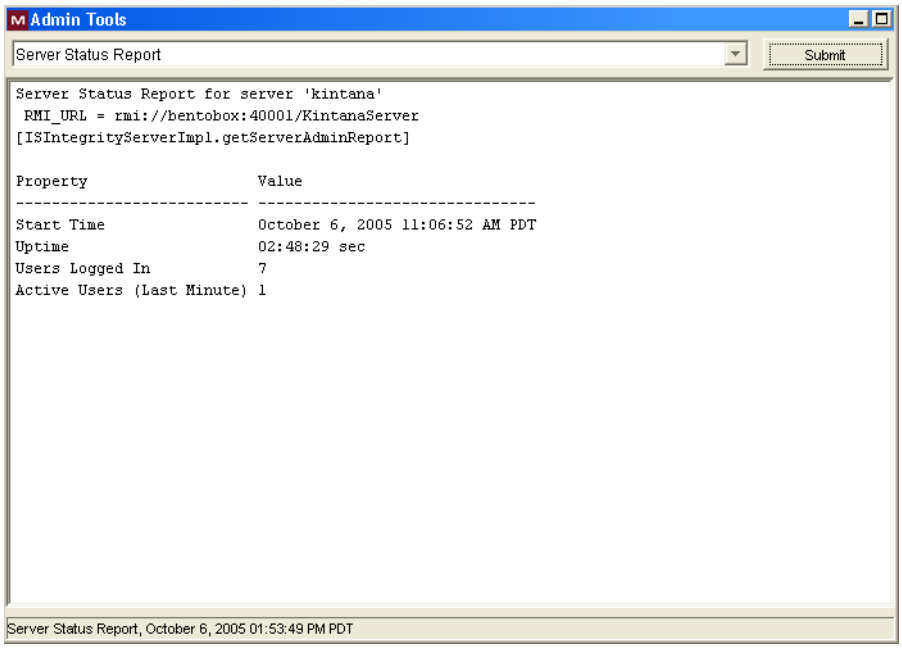


Table 7-2. Server reports (page 1 of 3)

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. For performance reasons, the Mercury IT Governance 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. If the Mercury IT Governance Server cannot allocate more connections, threads that need to access the database might need to wait for a connection. This report also shows: ■ Number of threads waiting for connections ■ Average duration threads had to wait for connections ■ Percentage of threads that had to wait for connections ■ Total number of connection requests, and if JDBC logging is enabled ■ 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 cacheable entity in Mercury IT Governance Center, including: ■ Hits, misses, and hit rate ■ Number of cache flushes (broken down by the categories "old," "idle," "reclaimed," and "max cache size reached") ■ Average load time ■ Cached object count and maximum idle time
Client Font	All supported fonts for the Mercury IT Governance Center installation.
Client Property	Details about the environment of the client computer currently running the Workbench.
Client Time Zone	All time zones recognized by the client.

Table 7-2. 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 Report	
JVM Memory	Free and total memory in the Mercury IT Governance Server JVM.
Kintana RMI Report	All RMI connection threads.
Server Cache Status	Shows the following cache information: ■ Cached entities ■ Number of units that can be cached ■ Number of free units ■ The number of hits and misses, and the miss rate ■ Number of entities swapped ■ Amount of memory taken up by the cache **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 Mercury IT Governance Server can send to the client.
Server Logon	Information about all users logged on to the Mercury IT Governance Server(s) and logon information such as IP address and idle time. This information is used to determine Mercury IT Governance Server load. If server clustering is used, this report provides a picture of load distribution.

Table 7-2. Server reports (page 3 of 3)

Report Name	Description
Server Status	Status information about Mercury IT Governance Server(s): ■ Whether the server is available and its start time ■ Length of time the server has been available ■ Number of users logged on to the server ■ Number of users active during the last minute
Server Thread	Information about running threads within a Mercury IT Governance Server(s). 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 Mercury IT Governance 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 Mercury IT Governance Server using the `kRunServerAdminReport.sh` script, which is located in the `<ITG_Home>/bin` directory. For more information about the `kRunServerAdminReport.sh` script, see *kRunServerAdminReport.sh* on page 296.

Running SQL Statements in the SQL Runner Window

You can use the SQL Runner window to run database queries directly against the Mercury IT Governance Center database schema using the 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.

Warning

Make sure 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 7-3 lists the controls in the SQL Runner window, along with a description of each.

Table 7-3. Controls in the SQL Runner window (page 1 of 2)

Control Name	Control Type	Description
SQL Statement	Text box	Use this box to type an SQL query for running and testing purposes. **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. 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. 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.

Table 7-3. Controls in the SQL Runner window (page 2 of 2)

Control Name	Control Type	Description
Ping Server	Button	Tests the connection speed between the client and the Mercury IT Governance Server.
Ping DB	Button	Tests the connection speed between the client and the database (via the Mercury IT Governance Server).
Open As Text	Button	Opens results in a text window. You can cut and paste information from this window.

Setting Debugging and Tracing Parameters

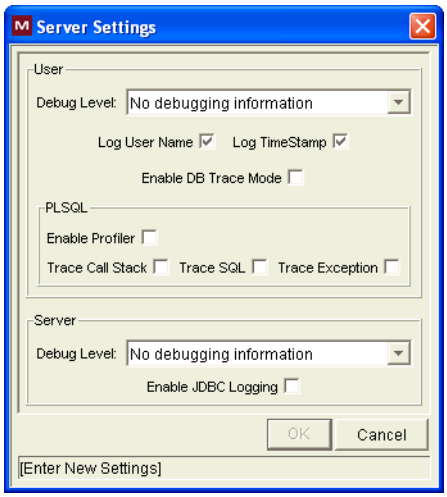
You use the Server Settings dialog box to set debugging and tracing parameters at both the user and server levels.

To open the Server Settings dialog box from the Workbench:

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

   The Workbench opens.
3. From the shortcut bar, select **Edit > Server Settings.**

   The Server Settings dialog box opens.



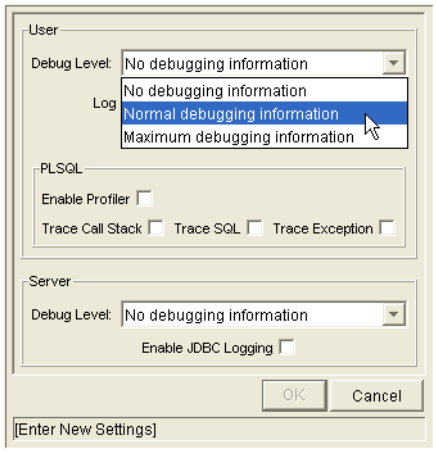
User Settings

This section provides information about the debug level and PL/SQL settings.

Debug Level Setting

To override the default debug level set for your Mercury IT Governance Center sessions:

- In the **Debug Level** list, select a different value.



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 when troubleshooting any problems you encounter in Mercury IT Governance 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.

Warning

Make sure that you do not to leave the server running in debug mode for too long. Lots 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 Mercury IT Governance 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 Mercury IT Governance Center applications use by calling the Oracle-supplied PL/SQL package `DBMS_PROFILER`.

Note

You must set up the PL/SQL package. For an example of how to do this, see *Setting Up the Oracle Profiler* on page 153.

The profiling information is logged in a JDBC log file in the Mercury IT Governance Center `log` directory. Enabling the profiler can help you to identify performance bottlenecks.

Note

Because running the `DBMS_PROFILER` package might slow system performance and reduce storage space, Mercury 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 Mercury IT Governance Center applications) use.

The output of the profiling information is saved to a JDBC log file in the Mercury IT Governance Center `log` directory.

Note

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 Mercury IT Governance Server, and not just your user session:

1. Under **Server,** in the **Debug Level** list, select one of the following:

   Note

   The following settings correspond to the settings for the DEFAULT_SERVER_LOGGING_LEVEL server configuration parameter. The value names, however, are different.

   - **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 Mercury IT Governance 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 237.

2. To have the Mercury IT Governance Server(s) maintain a Java Database Connectivity (JDBC) log file, select the **Enable JDBC Logging** checkbox.

Getting Information from Log Files

The Mercury IT Governance 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 `<ITG_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 Mercury IT Governance 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 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 Mercury Support, increase the debug level.

Unless instructed otherwise by 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 Workbench interface. You can also retrieve current server settings by accessing the Server Tools window and running the Server Configuration report.

Note

Unless instructed by 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:

Note

Do not enable HTTP logging if you use an external Web server.

1. Stop the IT Governance 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/itg/web/knta/
global/images/date_time.gif HTTP/1.1"200 155 "http://
localhost:8080/itg/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 `<ITG_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:

- `<ITG_Home>/logs/PKG_Package_ID`
- `<ITG_Home>/logs/REQ_Request_ID`
- `<ITG_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 `<ITG_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 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 `<ITG_Home>/logs/temp` directory are stored for temporary purposes. Unless requested otherwise by Mercury Support, you can delete these log files on a regular basis.

Periodically Stopping and Restarting the Server

The Mercury IT Governance Server generally requires very little maintenance. To help make sure your system operates smoothly, Mercury recommends that the server be stopped and restarted once a month.

For information about starting and stopping the server, see *Starting and Stopping the Mercury IT Governance Server* on page 68.

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 Mercury IT Governance Center.

Changing the Database Schema Passwords

If you must change the Mercury IT Governance 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 `<ITG_Home>/bin/kEnvUpdatePassword.sh` to automatically change all occurrences of a specific password for a particular host and user name.

  Note: This functionality is also available from the **Environments** section of the 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 Mercury IT Governance Center database schema passwords:

1. Check to make sure that all users are logged off the system.
2. Stop the Mercury IT Governance Server.

   For information about how to stop Mercury IT Governance Servers, see *Stopping the Mercury IT Governance Server* on page 186.
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.

Note

> 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 Mercury IT Governance Server.

   For information about restarting Mercury IT Governance Servers, see *Starting and Stopping the Mercury IT Governance Server* on page 68.

Maintaining Temporary Tables

The Mercury IT Governance Server uses several tables for temporary storage during processing (for example, during package migration) for:

- Logon attempts
- Debug messages
- Commands and parameters

Mercury IT Governance Server uses a set of services to monitor and clean up these temporary tables. Check to make sure the cleanup parameters (described in *Cleanup Parameters* on page 176 and in Appendix A, *Server Configuration Parameters,* on page 237) 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 Mercury IT Governance Server over the previous 14 days. This information includes:

- `USER_ID` of users who attempted to log on
- Success or failure status of each logon attempt
- Any messages generated during the logon attempt

The `KNTA_LOGON_ATTEMPTS` table is maintained only for auditing purposes. The Mercury IT Governance Server does not require the data to operate correctly. 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 Mercury PL/SQL database packages generate. After you analyze this data, you can safely purge it. The Mercury IT Governance Server purges this data automatically at the frequency determined by the `HOURS_TO_KEEP_MESSAGE_ROWS` server configuration parameter setting.

Backing Up Mercury IT Governance Center Instances

Backing up a Mercury IT Governance Center instance involves backing up both the file system and the database schema. Mercury stores all Mercury IT Governance Center configuration and transaction data in its associated database schema.

Because this information is so important, Mercury 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 Mercury IT Governance Server. For information about how to export a database schema, see your Oracle database documentation.

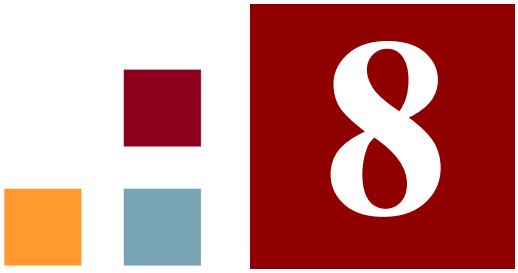
Mercury recommends that you back up the `<ITG_Home>/logs` directory daily. This directory contains transactional history files for each migrated package or request.

Note

Before you make critical changes to Mercury IT Governance Center, perform a full backup of the database schema and complete `<ITG_Home>` directory.

It is not necessary to back up registry settings.

Chapter



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 83 and Appendix A, *Server Configuration Parameters,* on page 237 contains information on the initial settings that Mercury recommends for the Oracle database and Mercury IT Governance Server. If Mercury IT Governance Center performance slows after these settings are in place, use the methodologies outlined in the flowcharts shown in *Figure 8-1* on page 165, *Figure* on page 166, *Figure 8-3* on page 167, and *Figure 8-4* on page 167 to isolate performance problems and determine how to fix them.

Figure 8-1. Identifying and addressing system performance problems

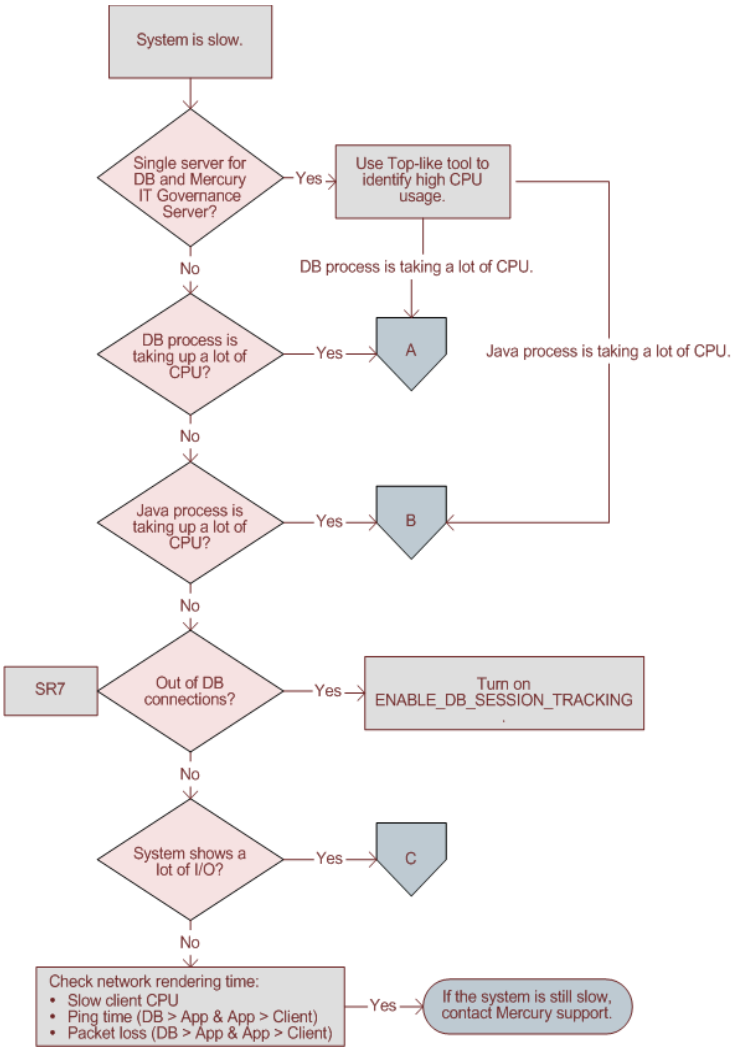


Figure 8-2. Identifying and addressing database performance problems (A)

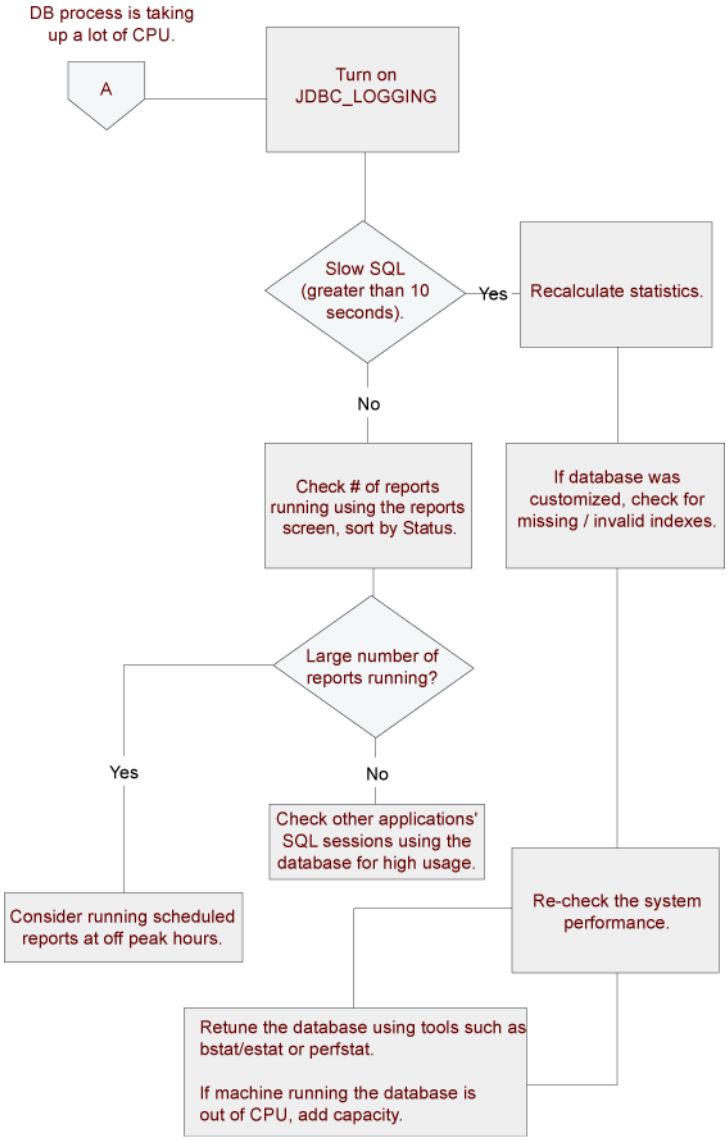


Figure 8-3. Identifying and addressing Java process performance problems (B)

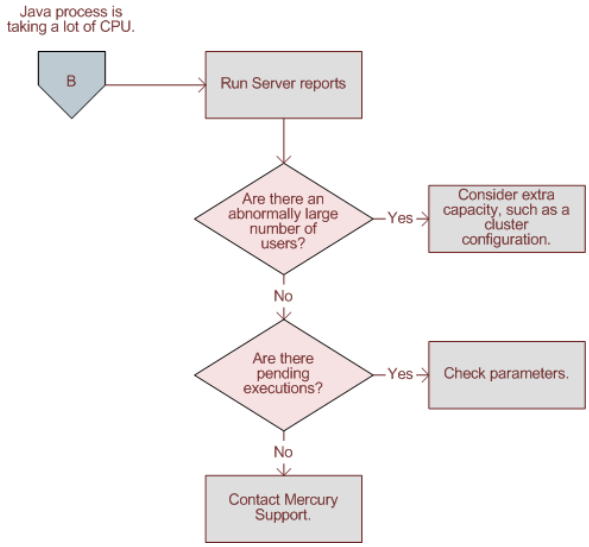
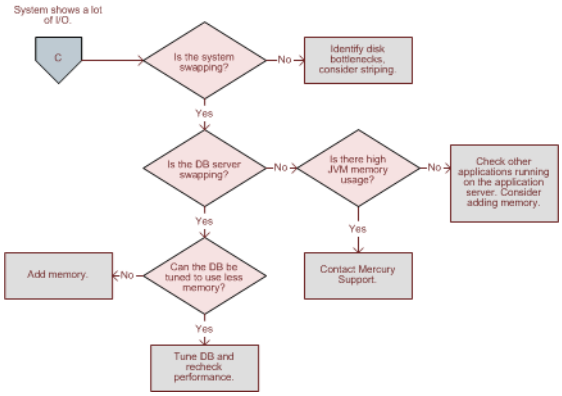


Figure 8-4. Identifying and addressing I/O performance problems (C)



Collecting Database Schema Statistics

This section provides information about collecting statistics about the Oracle database schema.

Setting the Database to Gather Statistics

Mercury IT Governance Server requires the gathering of database statistics. For information about the Oracle database parameter that enables statistics gathering, see *OPTIMIZER_MODE* on page 87. These statistics provide information on the number of rows in tables, data distribution, and frequency of values.

Collecting Additional Statistics by Setting Server Parameters

Collect additional 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 Mercury IT Governance Center service to collect this kind of data periodically about the Mercury IT Governance Center database schema. You can use the following parameters to collect database statistics on Mercury IT Governance 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 Mercury IT Governance Server parameters, see *Server Configuration Parameters* on page 237.

Using Scripts to Collect Additional Statistics

If statistics gathered using the Mercury IT Governance Center service are insufficient, you can use the following to gather additional statistics:

- The `dbms_stats` package that Oracle provides as part of the database
- The `kBuildStats.sh` script, which is located in the `<ITG_Home>/bin` directory

Running dbms_stats

Run the `dbms_stats` package, as follows:

```
begin
dbms_stats.gather_schema_stats (ownname => <ITG_User>,
cascade => TRUE,
method_opt => 'FOR ALL COLUMNS SIZE AUTO'
);
end;
/
```

You would typically run this package as the system user. To run it as a Mercury IT Governance Center user, grant the privilege to run the `dbms_stats` package by running the following SQL statement as the system user from an SQL*Plus session:

```
grant execute on dbms_stats to <ITG_User>;
```

Running kBuildStats.sh

You can also run the `kBuildStats.sh` script to gather statistics. This script, which is located in the `<ITG_Home>/bin` directory, runs the same commands as the `dbms_stats` package.

Sampling a Percentage of Data

For large Mercury IT Governance Center installations, running the `kBuildStats.sh` script can take a long time. In the case of such large installations, you can sample a percentage of data in each object instead of data from the entire Mercury IT Governance 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 => <ITG_User>,
cascade => TRUE,
method_opt => 'FOR ALL COLUMNS SIZE AUTO',
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.

Scheduled Reports Do Not Run on Schedule

Problem

Although the Mercury IT Governance 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 237.

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 237.

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 167.

Improving System Performance

This section provides information on how you can improve system performance.

Tuning Java Virtual Machine (JVM) Performance

Because the Mercury IT Governance 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 Mercury IT Governance 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.

Note

After you first start the Mercury IT Governance Server following an installation or upgrade, the server occupies approximately 600 MB in memory. As you use the product, the cache fills up and the JSPs are loaded into memory. The result is that, over time, the system gradually uses more memory. This is normal, and memory usage levels out over time. Memory usage typically increases to a maximum of 800 MB.

Tuning Server Cluster Performance

High transaction volumes and a large number of concurrent users on a Mercury IT Governance Server can degrade server response time. If the Mercury IT Governance 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 Mercury IT Governance Server cluster. In this system configuration, multiple Mercury IT Governance 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 126.

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 Mercury IT Governance Center database into the following segments:

- Mercury IT Governance Center tables
- Mercury IT Governance Center indexes
- Redo logs
- Rollback tablespaces
- Temporary tablespaces
- System tablespace
- Tablespace for management and related utilities

Mercury recommends that Mercury IT Governance 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 8-1*.

Table 8-1. Database disk recommendations

Disk	Recommendations for Data Placement
Disk 1	Mercury IT Governance Center tables
Disk 2	Mercury IT Governance Center indexes
Disk 3	Redo logs
Disk 4	■ Rollback tablespaces ■ Temporary tablespaces ■ System tablespace ■ Tablespace for management and related utilities

For Mercury IT Governance Center database instances that have higher transaction volumes (more than 10,000 requests per month), Mercury 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 Mercury IT Governance 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

Mercury IT Governance 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 Mercury IT Governance 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.

Adjusting Server Configuration Parameters

This section provides information about Mercury IT Governance 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 Mercury IT Governance Server parameters, see *Server Configuration Parameters* on page 237. 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 Mercury IT Governance 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 7-2* on page 146. For more information about the cleanup parameters, see *Server Configuration Parameters* on page 237.

Debug Parameters

Debug parameters control the debug and log output from the Mercury IT Governance 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 Mercury IT Governance Server. Users who have the required privileges can configure these parameters by selecting **Edit > Debug Settings** from the 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 237 and *Logging Parameters* on page 283.

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.

Note

Mercury recommends that you consult Mercury Support before enabling 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 283.

Timeout Parameters

Timeout parameters determine how long the Mercury IT Governance 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 Workbench sessions send a message to inform the Mercury IT Governance Server that the client is active.
- `DB_LOGIN_TIMEOUT` determines the duration (in seconds) for the Mercury IT Governance 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 Mercury IT Governance 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 Mercury IT Governance 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` 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 Mercury IT Governance 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 Mercury IT Governance Server checks for pending scheduled tasks, and starts the tasks if worker threads are available.
- `WF_SCHEDULED_TASK_PRIORITY` determined 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 Mercury IT Governance Server maintains. After the Mercury IT Governance 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 Mercury IT Governance 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 Mercury IT Governance 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 283

Chapter

10

Migrating Entities

In This Chapter:

About Entity Migration

Entity migrators are Mercury Deployment Management object types. Each migrator is designed to migrate a specific kind of Mercury IT Governance Center entity and all of its dependent objects from one Mercury IT Governance Center instance to another.

You can use Mercury Deployment Management to manage configuration changes to Mercury IT Governance Center. Mercury Deployment Management comes with an out-of-the-box set of object types, or *entity migrators,* that you can use to move Mercury IT Governance Center configuration entities (workflows, request types, and so on) between Mercury IT Governance Center instances. If you maintain scratch instances for developing and testing Mercury IT Governance 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 Mercury IT Governance Center implementation. You can build a workflow for every migrator object type, or create a single generic workflow for all migrator object types.

Note

You can only migrate entities between Mercury IT Governance Center instances of the same release.

You can migrate the following Mercury IT Governance Center entities:

- Special commands
- Object types
- Portlet definitions
- Dashboard modules
- Dashboard data sources
- Project types
- Workplan 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 workplan 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. Workplan 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 Mercury IT Governance 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 Mercury IT Governance 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 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 Mercury IT Governance 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.

Note

If you want to perform version control on changes to Mercury IT Governance Center configuration entities as they are migrated, you can version the Zip file that is extracted from the source instance.

Mercury recommends that you not unzip this file manually, except for debugging purposes.

5. The Production server unpacks the migrated Zip 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 Mercury IT Governance Center to a Production instance.

Note

To create, submit, and process migrations, you must have the required licenses and access grants. For more information, see the document *Security Model Guide and Reference*.

Before you perform the following steps, check to 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:

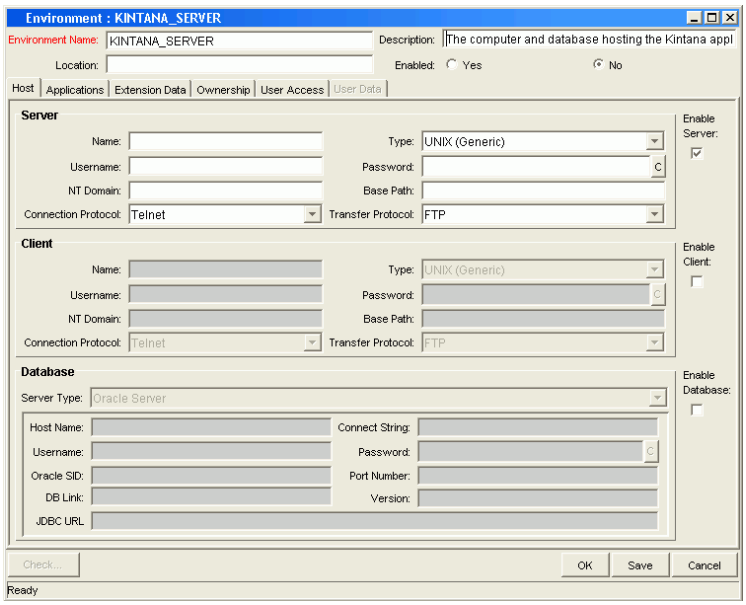
1. If the environment definition for the Mercury IT Governance Server is not configured, then you must first create the `KINTANA_SERVER` environment, as follows:

Note

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.

The Environment: KINTANA_SERVER window opens.



b. Define and enable the server information.

Note

Because this environment definition represents the Mercury IT Governance 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.

c. Define and enable the source environment.

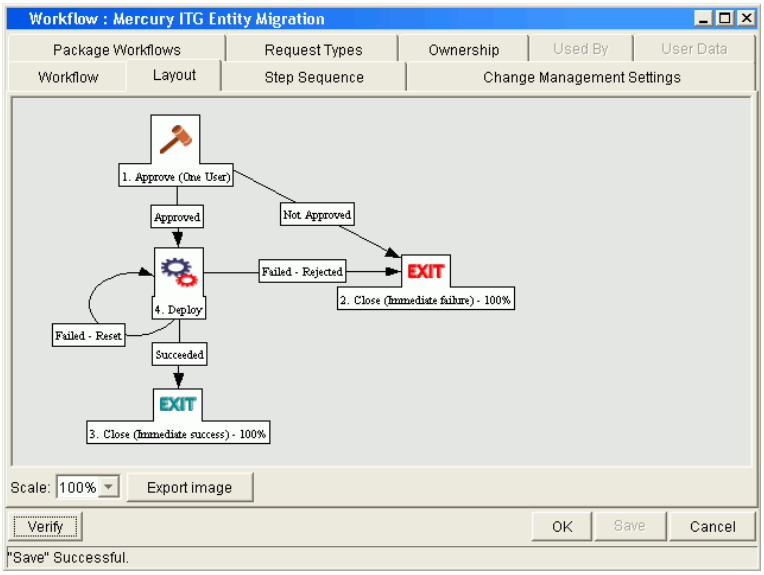
Note

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 document *Mercury 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 document *Security Model Guide and Reference*.

4. In the Package:<*Package Name*> window, in the **Workflow** field, enter the workflow you created.

5. Click **New Line.**

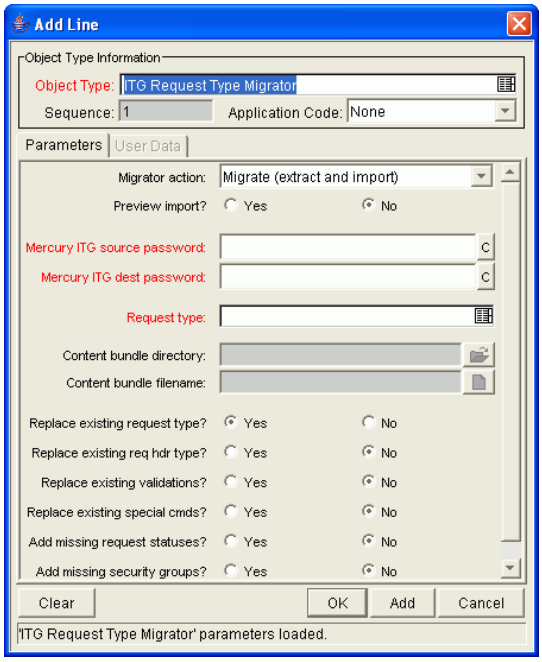
   The Add Line dialog box opens.

6. In the **Object Type** field, type `ITG Request Type Migrator.`

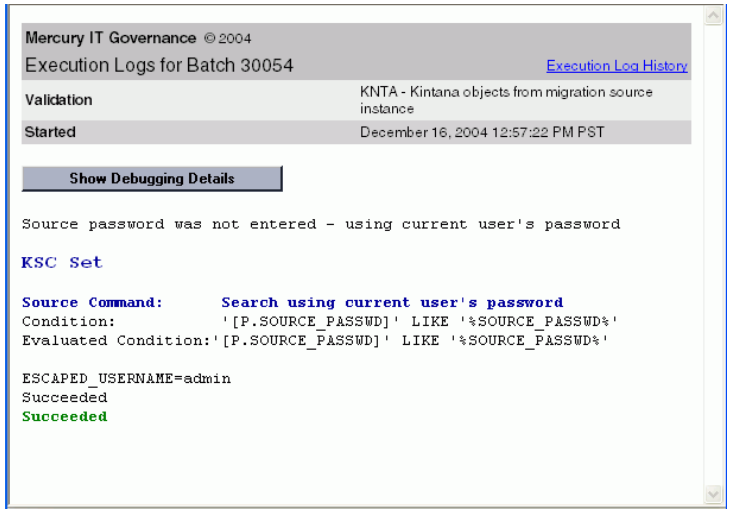
7. Enter the following required information:

   - In the **Mercury ITG source password** field, type the password for your Mercury IT Governance Center account in the source instance.
   - **Mercury ITG dest password** field, type the password for your Mercury IT Governance Center account in the destination instance.

- In the **Request type** field, type the name of the request type that you want to migrate.



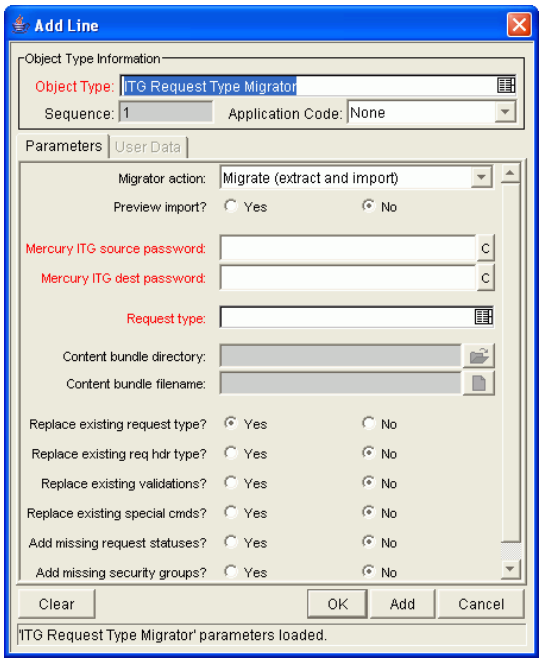
8. Click **OK.**
9. Submit the workflow.
10. Process the workflow.
11. Check the execution log to verify that the migration was successful.



Defining Entity Migrators

Each object type for the Mercury IT Governance Center entity migrators has a set of parameters similar to those described in this section (and as illustrated in the previous example). The Request Type Migrator shown in *Figure 10-1* is an example.

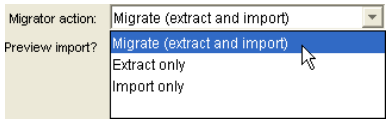
Figure 10-1. Add Line dialog box for the Request Type Migrator



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 10-2* shows the **Migrator action** list.

Figure 10-2. Migrator action list



In the **Migrator action** list, you can select one of the following actions:

- **Migrate (extract and import)**
- **Extract only**
- **Import only**

Table 10-1 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.

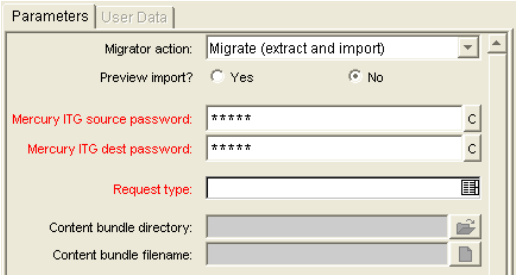
Table 10-1. Migrator action list dependencies

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 10-3*, the parameters are the entity name (in this case, the request type), content bundle directory, and content bundle filename.

Figure 10-3. Basic parameters



Content Bundle Controls

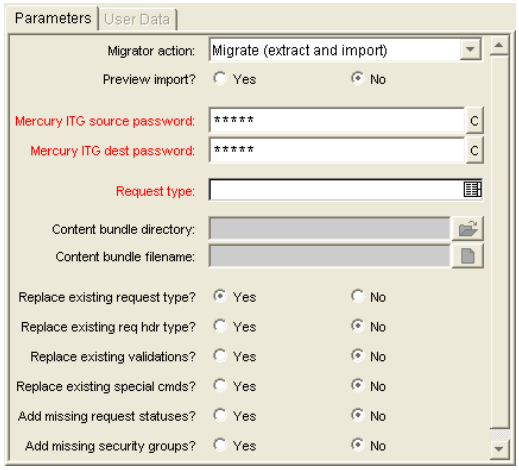
The behavior of controls related to the content bundle depends on the migrator action you select, as follows:

- 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 Mercury IT Governance Server under the `<ITG_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 Mercury IT Governance Server.

Import Flags

Use the import flags listed in the lower portion of the **Parameters** tab (shown in *Figure 10-4*) to control migrator behavior.

Figure 10-4. Import flags



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 Mercury IT Governance 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 Mercury IT Governance 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 Mercury IT Governance 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 Mercury IT Governance Center special commands (including parent and child special commands) that exist in the target Mercury IT Governance 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 Mercury IT Governance Center instance, you can decide whether or not to create them. The default value is **No.**

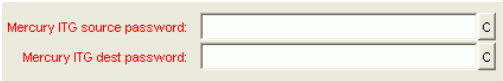
- **Add missing security groups?** If the entity to be migrated references security groups that are not included in the target instance, you can add those security groups. The default value is **No.**

  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 **Mercury ITG source password** and **Mercury ITG dest password** fields (*Figure 10-5*) are enabled.

Figure 10-5. Password fields



Source Password Field

When the migrator contacts the source server, it uses the credentials of the current Mercury IT Governance Center user to authorize the entity extraction. This user must be part of a security group that contains the access grant "System Admin: Migrate Kintana Objects." Confirm the user password for the source server in the **Mercury ITG source password** field.

Destination Password Field

When the migrator contacts the destination server, it uses the credentials of the current Mercury IT Governance Center user to authorize the entity import. This user must be part of a security group that has the "Sys Admin: Migrate Mercury ITG Objects" access grant. Confirm the user password for the destination server in the **Mercury ITG dest password** field.

Internationalization List

Typically, in an environment in which you are managing configuration across multiple Mercury IT Governance Servers, all of the Mercury IT Governance Center databases involved have the same localization settings. However, if you must migrate configuration entities between Mercury IT Governance 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. On the shortcut bar in the Workbench, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench window opens.

2. Click **List.**

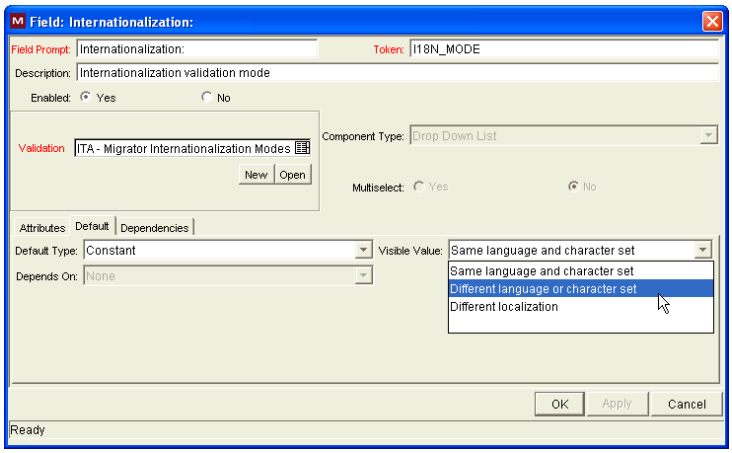
3. In the **Object Name** column on the **Results** tab, double-click **ITG Request Type Migrator.**

   The Object Type: ITG Request Type Migrator window opens.

4. In the **Prompt** column on the **Fields** tab, double-click **Internationalization.**

   The Field: Internationalization window opens.

5. Click the **Default** tab.



6. In the **Visible Value** list, select one of the following:

   - **Same language and character set.** This is the default option for migrating entities between Mercury IT Governance 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 Mercury IT Governance 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.

7. Click **OK.**

Environment Considerations

When migrating entities, Mercury Deployment Management logs on to remote machines in the same way another user would (that is, using FTP, SCP, SSH, or Telnet). Mercury Deployment Management can log to a remote server using any existing operating system user name and password.

Mercury recommends that you generate a new user (for example, Mercury IT Governance Center) on every machine to which Mercury Deployment Management has access. A user you create for this purpose must have full access to the `<ITG_Home>` directory on the Mercury IT Governance 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 Mercury IT Governance Center supports, see the *System Requirements and Compatibility Matrix* document and the *Mercury 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 document *Mercury Deployment Management Configuration Guide*.

Setting the SERVER_ENV_NAME Parameter

The Mercury IT Governance Center migrators depend on the `SERVER_ENV_NAME` server configuration parameter. This parameter specifies the name of an environment definition in the Mercury IT Governance Center system that describes the host server running that Mercury IT Governance Center instance.

When you installed Mercury IT Governance 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. Mercury IT Governance Center often refers to this server parameter to find the environment definition that contains information about the computer[s] that host the Mercury IT Governance 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 Mercury IT Governance Center users have ownership and control over different Mercury IT Governance 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 Mercury IT Governance 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.

Note

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 Mercury IT Governance 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

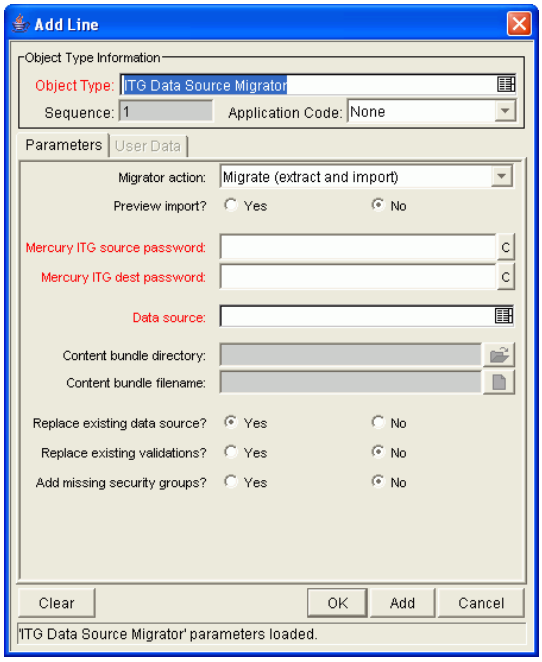
This section provides descriptions of Mercury IT Governance Center entity migrators:

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 Mercury IT Governance Center instances. (Data sources provide data displayed in Dashboard portlets.)

Figure 10-6 shows the parameters for the Dashboard Data Source migrator as they are displayed during package line creation.

Figure 10-6. Data Source Migrator

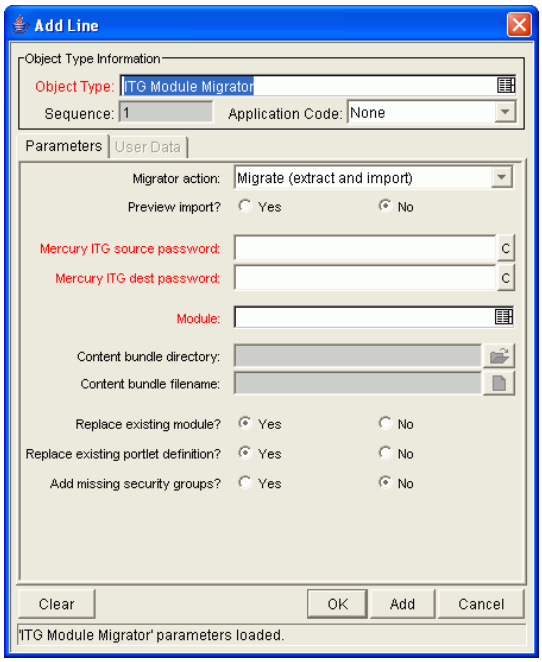


For information about the fields in this migrator, see *Defining Entity Migrators* on page 207. For information about how to create a portlet data source, see the document *Configuring the Standard Interface*.

Module Migrator

In the Mercury IT Governance Center standard interface, a module is the set of pages that an administrator sets up for users to view and navigate in the Dashboard. You can use the Module Migrator to move Mercury IT Governance modules from one Mercury IT Governance Center environment to another.

Figure 10-7. Module Migrator



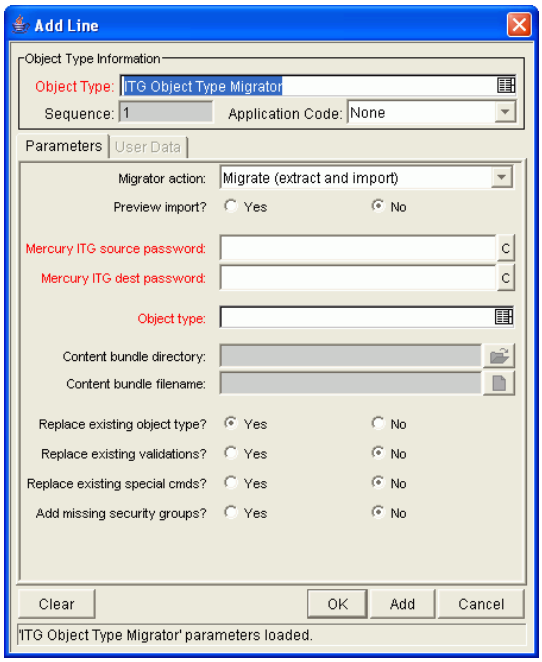
For information about the fields in this migrator, see *Defining Entity Migrators* on page 207. For information about how to create modules, see the document *Configuring the Standard Interface*.

Object Type Migrator

The Object Type Migrator (*Figure 10-8* on page 218) contains the additional option **Replace existing special cmds?** If the validation to be migrated references Mercury IT Governance Center special commands (including parent and child special commands) that exist in the target Mercury IT Governance 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 10-8. Object Type Migrator



For information about most of the controls in this migrator window, see *Defining Entity Migrators* on page 207.

Configuration Considerations

The ITG 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

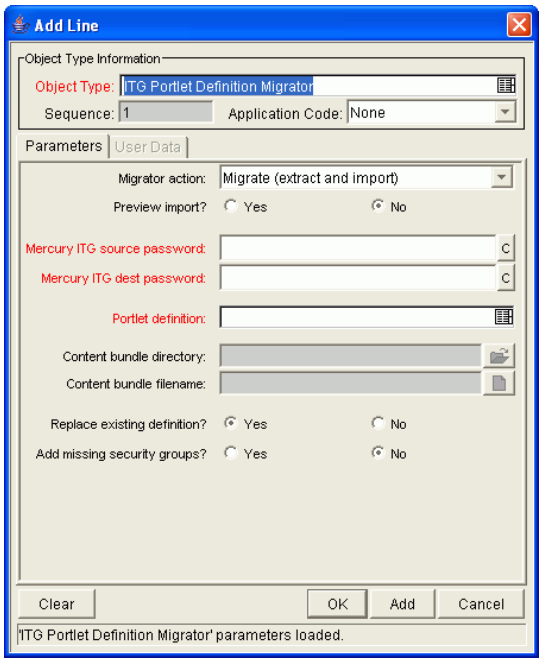
Note

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 10-9*) 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 Mercury IT Governance Center, the migrated changes are applied to all users who have added the same portlet to their Dashboard.

Figure 10-9. Portlet Definition Migrator

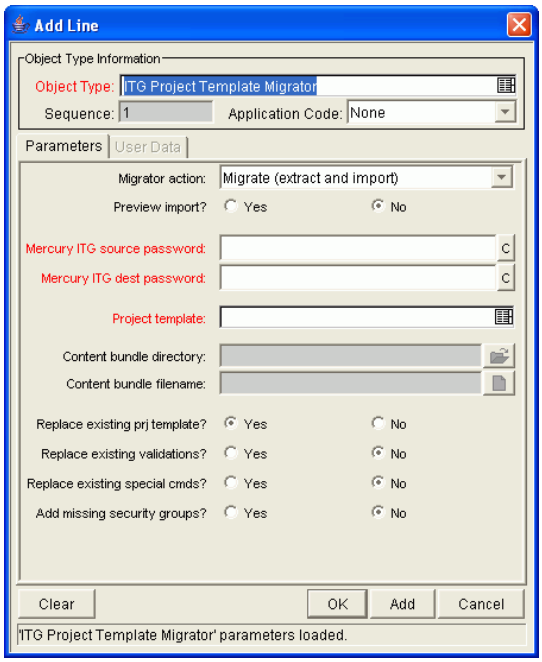


For information about the fields in this migrator, see *Defining Entity Migrators* on page 207.

Project Template Migrator

The Project Template Migrator (*Figure 10-10* on page 220) contains the additional option, **Replace Existing special cmds?** If the validation to be migrated references Mercury IT Governance Center special commands (including parent and children special commands) that already exist in the target Mercury IT Governance Center instance, you can decide whether or not to replace them. The default value is **No.** Regardless of their values, special commands missing from the destination instance are created automatically.

Figure 10-10. Project Template Migrator



For information about most of the fields in this migrator, see *Defining Entity Migrators* on page 207.

Configuration Considerations

The Project Template 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 already referenced elsewhere
- Security groups referenced by resource lists
- Notifications referenced by project tasks
- Notification intervals referenced by notifications
- Security groups referenced by notifications
- Ownership group information for the project template
- Project team tab information

Project templates can reference users and security groups. The Project Template Migrator transfers these references, but does not create a missing user or security group. If the referenced user or security group does not exist in the destination instance, the reference is discarded in transit. A message to that effect is placed in the execution log for the migration.

A project template can also contain references to project templates used to create the current template. The Project Template Migrator transfers these references, but does not create a missing nested project template.

Note

To ensure that you preserve these references, first migrate any project templates that are nested inside other project templates. Otherwise, if the referenced nested project template does not exist in the destination instance, the reference is discarded in transit.

Project Type Migrator

You can define project types in a development or testing instance of Mercury Project Management, and then use the Project Type Migrator to migrate them to production after testing.

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 workplan 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.

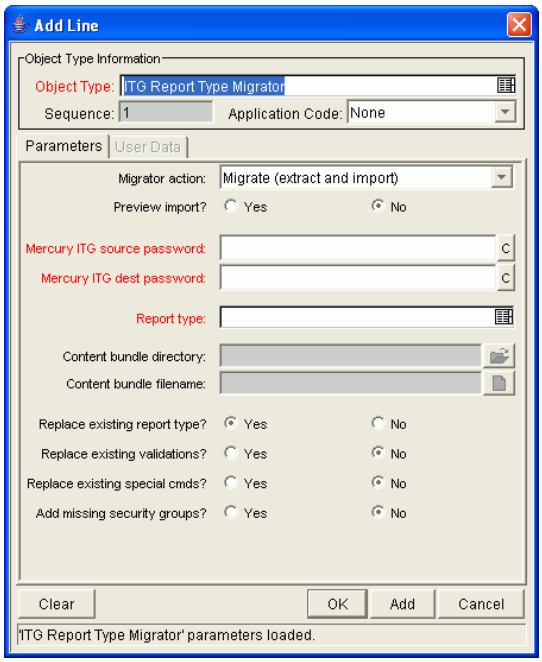
Note

The Project Type Migrator does not transport secondary objects as dependencies.

Report Type Migrator

The Report Type Migrator (*Figure 10-11*) contains the additional option **Replace Existing special cmds?** If the validation to be migrated references Mercury IT Governance Center special commands (including parent and child special commands) that already exist in the target Mercury IT Governance Center instance, you can choose to replace them (or not). (The default value is **No.**) Regardless of their values, Mercury IT Governance Center automatically recreates special commands that are missing from the destination instance.

Figure 10-11. Report Type Migrator



For information about most of the fields in this migrator, see *Defining Entity Migrators* on page 207.

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

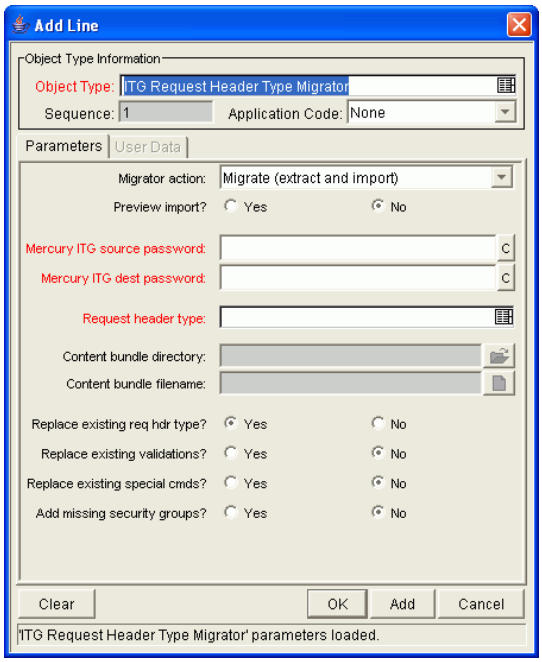
Note

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 10-12*) contains the additional option **Replace Existing special cmds?** If the validation to be migrated references Mercury IT Governance Center special commands that already exist in the target Mercury IT Governance 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, Mercury IT Governance Center automatically recreates special commands that are missing from the destination instance.

Figure 10-12. Request Header Type Migrator



For information about most of the fields in this migrator, see *Defining Entity Migrators* on page 207.

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

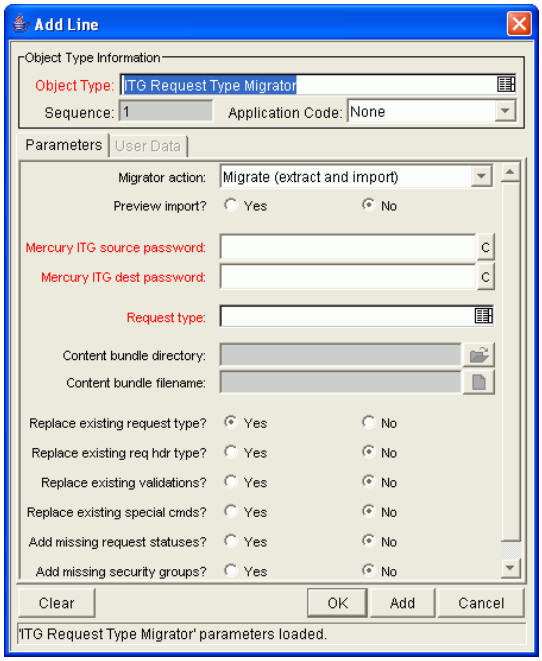
- Special commands referenced by other special commands
- Ownership group information for the request header type

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 10-13*) has additional import behavior options from which to choose.

Figure 10-13. Request Type Migrator



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 Mercury IT Governance 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 Mercury IT Governance Center special commands that already exist in the target Mercury IT Governance 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, Mercury IT Governance Center automatically recreates 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 Mercury IT Governance 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.

Note

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.

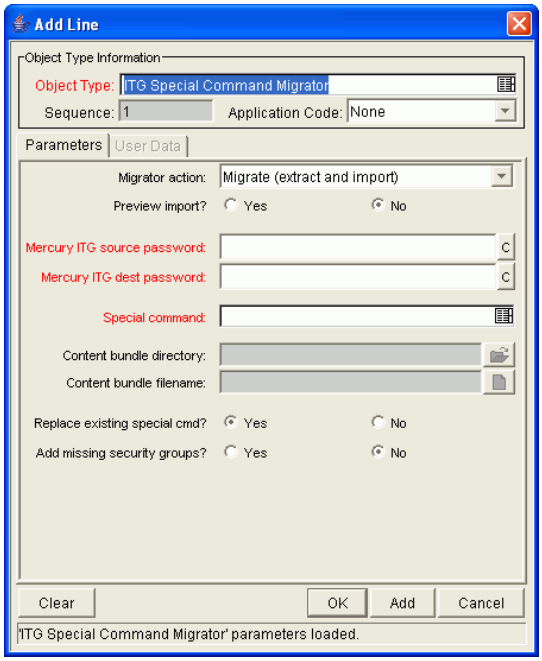
Circular references between request types and workflows could make it necessary to migrate either a request type or workflow twice:

- 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

The Special Command Migrator (*Figure 10-14*) migrates a Mercury IT Governance Center special command from one Mercury IT Governance Center environment to another.

Figure 10-14. Special Command Migrator

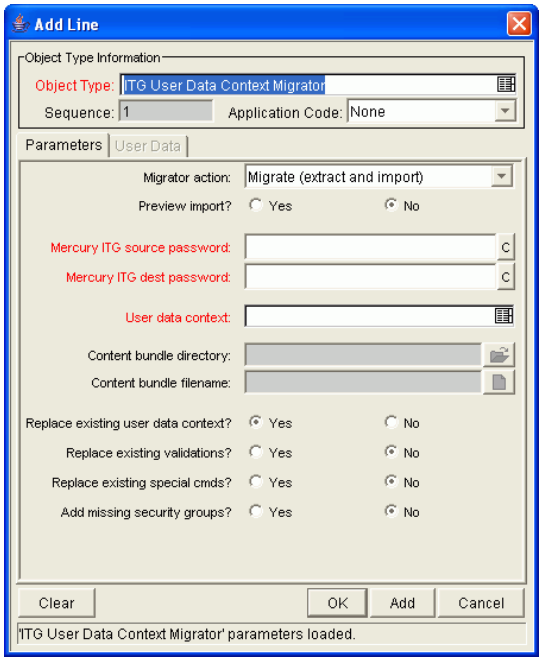


For information about the fields in this migrator, see *Defining Entity Migrators* on page 207.

User Data Context Migrator

The User Data Context Migrator (*Figure 10-15*) contains the additional option **Replace Existing special cmds?** If the validation to be migrated references Mercury IT Governance Center special commands that already exist in the target Mercury IT Governance 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, Mercury IT Governance Center automatically recreates special commands that are missing from the destination instance.

Figure 10-15. User Data Context Migrator

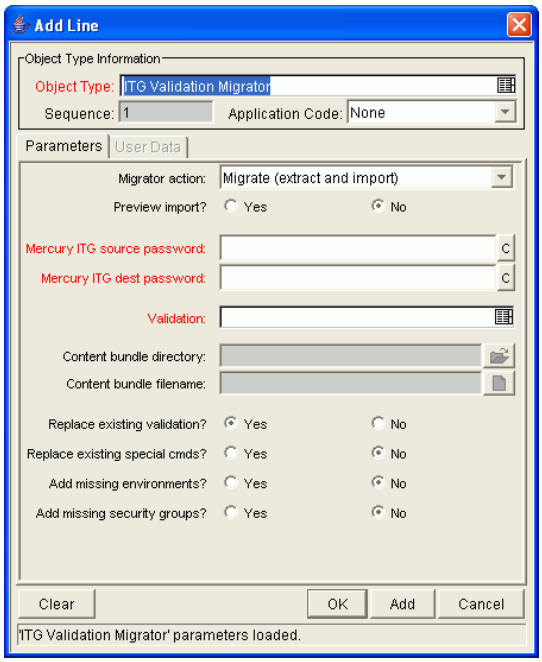


For information about most of the fields in the User Data Context Migrator, see *Defining Entity Migrators* on page 207.

Validation Migrator

The Validation Migrator is shown in *Figure 10-16*.

Figure 10-16. Validation Migrator



This migrator contains the following two additional import behavior options:

- **Replace existing special cmds?** If the validation to be migrated references Mercury IT Governance Center special commands that already exist in the target Mercury IT Governance 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, Mercury IT Governance Center automatically recreates 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 Mercury IT Governance 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 207.

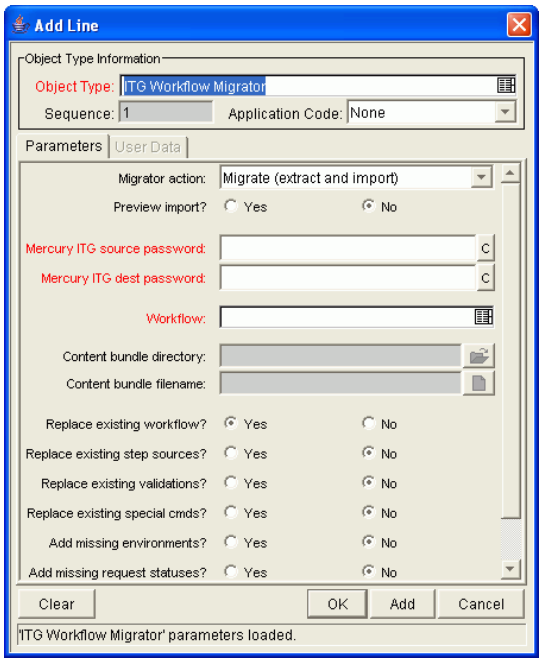
Configuration Considerations

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 10-17*.

Figure 10-17. Workflow Migrator



This migrator provides the following additional import behavior options:

- **Replace existing special cmds?** If the workflow to be migrated references Mercury IT Governance Center special commands that already exist in the target Mercury IT Governance 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 Mercury IT Governance 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 Mercury IT Governance 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 Mercury IT Governance 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 207.

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.

Mercury 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.

Workplan Template Migrator

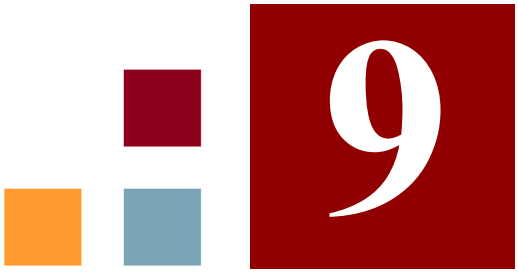
You can define workplan templates in a development or testing instance of Mercury Project Manager, and then migrate them to production after testing is completed.

The Workplan Template Migrator migrates the following:

- Header information such as workplan template name and list of owners (users)
- Workplan (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 Workplan Template Migrator does not transport any secondary objects (for example, validations) as dependencies.

Chapter



Migrating Instances

In This Chapter:

Overview of Instance Migration

Each Mercury IT Governance Center instance consists of a file system and an Oracle database, which can exist on Windows or UNIX machines. You can migrate Mercury IT Governance Center using one of the following methods:

- Copy an entire Mercury IT Governance 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 Mercury IT Governance 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 Mercury IT Governance Server. Migrating only the database schema does not require a new license key.

Enterprise environments typically have multiple Mercury IT Governance 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 Mercury IT Governance 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 Mercury 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 Mercury IT Governance 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 a Document Management Module (Optional)

If your source machine contains the Mercury document management module, see the *Document Management Guide and Reference* for information about how to migrate the document management module.

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 Mercury IT Governance Server, as described in the following sections.

Obtaining a New License Key

Mercury IT Governance Center is licensed based on the computer that hosts the Mercury IT Governance Server. If you plan to migrate the Mercury IT Governance 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:
   - Mercury IT Governance Center version number
   - Machine IP address
   - Operating system (Windows or UNIX)
   - Server purpose (development, test, or production)
2. Go to the Mercury Support site (support.mercury.com).

3. In the right panel of the Mercury 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 **IT Governance,** click **Get License.**

5. Enter the required information, and then click **Submit.**

Stopping the Mercury IT Governance Server

To ensure that you do not lose transactions, reports, or logs, stop the Mercury IT Governance Server before you migrate any part of a Mercury IT Governance Center instance. For information about how to stop the server, see *Starting and Stopping the Mercury IT Governance Server* on page 68.

Migrating the Mercury IT Governance Server

Before you migrate the Mercury IT Governance 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 Mercury IT Governance 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 185.

2. Stop the Mercury IT Governance Server.

   For information on how to stop the server, see *Starting and Stopping the Mercury IT Governance Server* on page 68.

3. Migrate the Mercury IT Governance Center file system:

   a. Make a Zip file of the entire `<ITG_Home>` directory.

   b. Copy the Zip file to the target machine, and then unzip the file.

4. Migrate the Mercury IT Governance Center database schema.

   For information about how to migrate the database schema, see *Migrating the Database Schemas* on page 191.

5. Reconfigure the Mercury IT Governance Server in the target location, as follows:

   a. Run the `kConfig.sh` script, which is located in the `<ITG_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 Mercury IT Governance Server.

        - `RMI_URL` specifies the port on which the Mercury IT Governance 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 Mercury IT Governance Server machine.

        - `BASE_PATH` specifies the full path to the directory where the Mercury IT Governance 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 Mercury IT Governance 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 Mercury IT Governance Server instance. If multiple Mercury IT Governance 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 `<ITG_Home>/bin` directory.

6. Create a Windows service for the new Mercury IT Governance Center instance, as follows:

   a. Navigate to the `<ITG_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 Mercury IT Governance Server.

For information about how to start the server, see *Starting and Stopping the Mercury IT Governance Server* on page 68.

Migrating to a UNIX Machine

To migrate the Mercury IT Governance Server to a UNIX machine:

1. Obtain a new license key, as described in *Obtaining a New License Key* on page 185.

2. Stop the Mercury IT Governance Server.

   For information about how to stop the Mercury IT Governance Server, see *Starting and Stopping the Mercury IT Governance Server* on page 68.

3. Migrate the Mercury IT Governance Center file system as follows:

   a. On the machine where the Mercury IT Governance Server is running, navigate to the parent of the `<ITG_Home>` directory.

b. Using an archiving utility (such as Tar or Zip), create an archive file of the entire `<ITG_Home>` directory.

   For example, if the `<ITG_Home>` directory is named "`ITG`," run the following TAR command:

   ```
   $ tar cf mitg70.tar ITG
   ```

c. Using FTP in binary mode, copy the archive file to the target machine. Put the archive file in the parent of the new `<ITG_Home>` directory.

d. Extract the archive file as follows:

   ```
   $ tar xf mitg70.tar
   ```

   This creates the new Mercury IT Governance Server directory structure. A directory named `ITG` is created automatically.

4. Migrate the Mercury IT Governance Center database schema.

   For information about how to migrate the database schema, see *Migrating the Database Schemas* on page 191.

5. Reconfigure the Mercury IT Governance Server in the target location as follows:

   a. Run the `kConfig.sh` script, which is located in the `<ITG_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 Mercury IT Governance Server.

        - `RMI_URL` specifies the port on which the Mercury IT Governance 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 Mercury IT Governance Server machine.
  - `BASE_PATH` specifies the full path to the directory where the Mercury IT Governance 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 Mercury IT Governance 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 Mercury IT Governance Server instance. If multiple Mercury IT Governance 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 `<ITG_Home>/bin` directory.

6. Place the new `license.conf` file into `<ITG_Home>/conf`.
7. Start the new Mercury IT Governance Server.

   For information on how to start the server, see *Starting and Stopping the Mercury IT Governance Server* on page 68.

Migrating the Database Schemas

This section provides the procedures used to migrate the Mercury IT Governance 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. Check to 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 Mercury Deployment Management Extension for Oracle E-Business Suite, you must consider the location of your Primary Object Migrator Host when migrating the Mercury IT Governance Center database schema, because Mercury Object Migrator might reside in the same database, or even the same schema, as Mercury IT Governance Center.

Migrating the schema does not require migrating the Mercury Object Migrator instance because the integration method in Mercury IT Governance Center can be refreshed to use the existing Mercury Object Migrator installation. If Object Migrator shares a database with Mercury IT Governance Center, and you intend to migrate it as well as Mercury IT Governance Center, the destination database must support Object Migrator. (For more information, see *Mercury Object Migrator Guide.*)

Unless Mercury IT Governance Center and Mercury Object Migrator share the same schema, the migration of Object Migrator is completely separate from the migration of Mercury IT Governance Center, and should be completed before you migrate the Mercury IT Governance Center database. Contact Mercury Support (support.mercury.com) for instructions on how to perform this migration.

If Mercury IT Governance Center and Mercury Object Migrator share the same schema and you want to migrate both, you must coordinate the migration activities. Contact Mercury Support for instructions.

Regardless of the configuration, refresh the integration definition after you migrate the Mercury IT Governance Center schemas.

To migrate the database schemas:

Note

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 196 and *Import Command Variables* on page 197.

1. Stop the Mercury IT Governance Server.

   For information about how to stop the Mercury IT Governance Server, see *Starting and Stopping the Mercury IT Governance Server* on page 68.

2. Export the Mercury IT Governance 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=<ITG_Username> LOG=c:/export_
   knta_700.log
   ```

3. Export the RML schema.

4. Create the new Mercury IT Governance Center database schema, as follows:

   a. Run the `CreateKintanaUser.sql` script (located in the `<ITG_Home>/install_700/mitg700/system` directory) from SQL*PLUS as the SYSTEM user.

      Example:

      ```
      SQL> @CreateKintanaUser.sql ITG_User ITG_Password Data_
      Tablespace Index_Tablespace Temp_Tablespace Clob_
      Tablespace
      ```

   b. Run the `GrantSysPrivs.sql` script (located in the `mitg700/sys` directory) from SQL*PLUS as the SYSTEM user.

   For more information, see *Preliminary Database Tasks* on page 305.

5. Create the new Mercury IT Governance Center RML database schema.

   To create a new, empty RML database schema in the target database, run the `CreateRMLUser.sql` script (located in the `mitg700/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 Mercury IT Governance 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_ITG_Username>
FROMUSER=<ITG_Username> LOG=c:/import_knta_700.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 Mercury IT Governance Center RML database schema:

Note

You can find the following scripts in the `<ITG_Home>/install_700/rml directory.`

- To set up the permissions between the two:

```
sqlplus <itg_user>/<itg_password>@SID
@RMLSetupInITGSchema.sql <rml_user>
```

- To create synonyms to IT Governance objects in the RML schema:

```
sqlplus <rml_user>/<rml_password>@SID
@RMLSetupInRMLSchema.sql <itg_user>
```

10. Configure the database schema to allow appropriate access to rebuild optimizer statistics.

Note

If Mercury IT Governance Center and Mercury Object Migrator share the same database schema, the Mercury IT Governance Center database schema is referred to as the Mercury IT Governance Center account, and the Mercury Object Migrator schema is referred to as the Mercury Object Migrator account.

To provide the necessary grants and permissions to the Mercury IT Governance Center user, run the `GrantSysPrivs.sql` script, as follows:

As the SYS user:

```
SQL> @GrantSysPrivs.sql <itg_user>
```

11. If any database links are defined in the Mercury IT Governance Center database schema, recreate the database links in the new Mercury IT Governance Center database schema as follows:

    - Run the `Recreate_db_links.sql` script from SQL*PLUS connected as the new Mercury IT Governance Center database schema account.

      The `Recreate_db_links.sql` script is located in the `<ITG_Home>/install_700/` directory. Running this script generates a file named `Recreate_customer_links.sql`.

    - Run the newly created `Recreate_customer_link.sql` script from SQL*PLUS connected as the new Mercury IT Governance Center database schema account.

12. If the Extension for Oracle E-Business Suite is in use and Mercury Object Migrator resides in the same schema as Mercury IT Governance Center, complete the Mercury Object Migrator migration.

    For assistance, contact Mercury Support.

13. If you are using the Extension for Oracle E-Business Suite, refresh the Primary Object Migrator Host definition.

14. Recompile invalid objects.

    To validate any invalid Mercury IT Governance Center database objects generated during link regeneration, run the `RecompileInvalid.sql` script, which is located in the `<ITG_Home>/install_700` directory. Run this script from SQL*PLUS connected as the new Mercury IT Governance Center database schema account.

15. Reconfigure the Mercury IT Governance Server to connect to the new database schema as follows:

    a. Start the configuration utility by running the `kConfig.sh` script located in the `<ITG_Home>/bin` directory.

    b. Update the server configuration parameters, which are described in *Server Configuration Parameters* on page 237.

Note

> If you edit the `server.conf` files manually, be sure to run the `kUpdateHTML.sh` script after you complete the edit.

16. Start the Mercury IT Governance Server (see *Starting and Stopping the Mercury IT Governance Server* on page 68).

Troubleshooting Instance Migrations

This section describes common problems that might occur when migrating Mercury IT Governance Center instances.

Mercury IT Governance Server Does Not Start

If you cannot start the Mercury IT Governance Server, check the `serverLog.txt` file (located in the `<ITG_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 `<ITG_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.

Note

> 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 Mercury IT Governance 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 Mercury IT Governance Server installation directory is specified correctly.

If the Mercury IT Governance Server has recently been upgraded and the URL has changed, check to make sure that any saved links to the previous Mercury IT Governance 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=<ITG_Username> LOG=c:/export_knta_
700.log
```

Note

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`)
ITG_Username	Name of the Mercury IT Governance 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_ITG_Username>
FROMUSER=<ITG_Username> LOG=c:/import_knta_700.log
```

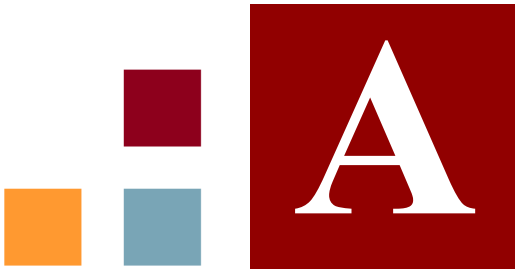
Note

The `imp` command might have a different name on Windows.

Table 9-2. Import command variables

Variable	Definition
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_ITG_Username	Name of the new Mercury IT Governance Center database schema.
ITG_Username	Name of the database schema that was previously exported.

Appendix



Server Configuration Parameters

In This Appendix:

Overview of Configuration Parameters

This appendix lists and describes the Mercury IT Governance Server configuration parameters located in three files in the `<ITG_Home>` directory:

- `server.conf`
- `logging.conf`
- `LdapAttribute.conf`

For more information about the Mercury IT Governance Server directory structure, see Appendix B, *Server Directory Structure and Server Tools,* on page 289.

Determining the Correct Parameter Settings

For most Mercury IT Governance Center installations, the default parameter values are correct. Considerations detailed in the parameter descriptions can help you determine under what circumstances you might change the parameter settings.

Required Parameters

The **Required** column shows whether the server parameter is required to set up a Mercury IT Governance Server. A value of `TRUE` in this column indicates that the parameter is required. A value of `FALSE` indicates that the parameter is optional. A condition 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 when 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 value 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 Mercury IT Governance Servers in a server cluster configuration, see *Configuring a Server Cluster* on page 126.

For information about how to specify your own parameters, see *Defining Custom and Special Parameters* on page 73.

Directory Path Names

Use forward slashes (/) when entering directory paths in the `server.conf` file, regardless of the operating system being used. Mercury IT Governance Center automatically uses the appropriate path separators when communicating with Microsoft Windows. Mercury 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 175.

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.

Note

> Mercury 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 Mercury IT Governance Server.
2. Run the `kConfig.sh` script.

Note

> 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.

3. Restart the Mercury IT Governance Server.

Note

> To see a list of the `server.conf` parameter values on an active Mercury IT Governance 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 144 and *Running Server Reports from the Command Line* on page 148.

Table A-1 provides descriptions of all 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` Valid values: `TRUE, FALSE`
APPLET_KEY_CLEANUP_INTERVAL	The frequency with which the *ENABLE_APPLET_KEY_CLEANUP* thread runs. *See also DAYS_TO_KEEP_APPLET_KEYS* on page 243.	Default: 21600 (seconds)
*ATTACHMENT_DIRNAME	Absolute pathname of the directory where attached documents are to be stored. This directory must: ■ Give read/write access to Web browsers ■ Be outside the directory tree if the system includes an external Web server In a server cluster, all servers must be able to access and share the specified directory.	Example: `c:\itg\eon\attachments`
AUTHENTICATE_REPORTS	If set to `TRUE`, access to all reports requires user authentication. (A user must provide a Mercury IT Governance Center user login ID).	Default: `TRUE` Valid values: `TRUE, FALSE`
*AUTHENTICATION_MODE	User authentication method. To specify multiple modes, use a comma-delimited list of valid values.	Default: ITG Valid values: `ITG, LDAP, NTLM, SITEMINDER`
AUTO_COMPLETE_SHORT_TYPE_MAX_ROWS	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 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. 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:\ITG\eon\logs`
*BASE_PATH	Full path to the directory where the Mercury IT Governance Server is installed.	The default value depends on the operating system platform. Example: `C:\ITG\eon\`
*BASE_URL	Web location (top directory name) of the Mercury IT Governance Server.	Example: `http://www.mydomain.com:8080`
BUDGET_IN_WHOLE_DOLLARS	Determines whether budget values are expressed in whole dollars.	Default: `FALSE` Valid values: `TRUE, FALSE`
CLIENT_TIMEOUT Category: Timeout	The value of this parameter determines the frequency (in minutes) with which the Workbench interface sessions sends a message to the Mercury IT Governance Server that indicates the client is still active. 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 Workbench.	Default: `FALSE` Valid values: `TRUE, FALSE`

Table A-1. Server configuration parameters (page 3 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
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 243.	Default: 16200
**CONC_LOG_ TRANSFER_ PROTOCOL Required if *ORACLE_ APPS_ENABLED* = TRUE	Transfer protocol used to transfer concurrent request logs and patching README files.	Default: FTP Valid values: FTP, SCP
**CONC_REQUEST_ PASSWORD Required if *ORACLE_ APPS_ENABLED* = TRUE	Encrypted password of the concurrent request user.	Encrypted example: `fnd`
**CONC_REQUEST_ USER Required if *ORACLE_ APPS_ENABLED* = TRUE	Valid user on the Oracle system that can be used to retrieve concurrent request output files. Set the retrieval method (FTP or SCP). *See **CONC_LOG_ TRANSFER_PROTOCOL* on page 242.	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` Valid values: `TRUE, FALSE`
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)

Table A-1. Server configuration parameters (page 4 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
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) 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* (see page 264) that the Dashboard module can use for database connections.	Default: 25 Valid values: Integer between 0 and 100
DASHBOARD_PAGE_ AUTO_REFRESH_ DISABLED	To disable the Dashboard auto-refresh feature (from the Personalize Page view), add this parameter to the `server.conf` file, and set it to `TRUE`.	Default: `FALSE` Valid values: `TRUE, FALSE`
DATE_NOTIFICATION_ INTERVAL	Interval at which the Mercury IT Governance 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 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 Category: Cleanup	Number of days to keep records of all logon attempts.	Default: 14
**DB_CONNECTION_ STRING (Required if RAC is used)	Oracle RAC (Real Application Clusters) service name.	Example: `K92RAC`
DB_LOGIN_TIMEOUT Category: Timeout	The amount of time that the Mercury IT Governance 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 (milliseconds)
*DB_PASSWORD	Password for the database schema that contains the Mercury IT Governance Center tables.	Example: `#!#<`*password*`>#!#`
*DB_USERNAME	Name of the database schema that contains the Mercury IT Governance Center tables.	Example: `knta`
DEBUG_MESSAGE_ CLEANUP_INTERVAL	Use this parameter to specify the run frequency for the *ENABLE_ DEBUG_MESSAGE_CLEANUP* thread (see page 245).	Default: 21600
DEFAULT_ COMMAND_TIMEOUT Category: Timeout	Determines the number of seconds the Mercury IT Governance 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). In new installations, this defaults to the slow connection setting. Mercury recommends that the system administrator review this setting after installation. 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_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`, the search results are returned unordered.	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. **Note:** Mercury recommends that you leave the default value unless the Mercury IT Governance 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. **Note:** Mercury recommends that you not change this value. It does not affect performance.	Default: 5000 (milliseconds)
DISTRIBUTION_LOG_ DIRECTORY	**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.	Default: Same as the default value for the *BASE_LOG_ DIRECTORY* parameter
DOCUMENT_ CLEANUP_SERVICE_ DELAY	Interval (in minutes) of document cleanup if Document Management is set up. Occasionally, if a user is attaching a document, but then cancels the operation, the document is attached anyway. This service cleans up such documents.	Default: 1440 (minutes)
EMAIL_ NOTIFICATION_ CHECK_INTERVAL Category: Scheduler/ services/thread	Determines the frequency (in seconds) with which the Mercury IT Governance Server checks for pending email notifications.	Default: 20
EMAIL_ NOTIFICATION_ SENDER	Email address of the default sender of email notifications. This sender receives any error messages associated with email notifications.	Example: `sender@itg.com`

Table A-1. Server configuration parameters (page 8 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
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 Workbench.) This parameter is associated with the frequency parameter *APPLET_KEY_CLEANUP_INTERVAL*.	Default: `TRUE` Valid values: `TRUE, FALSE`
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. This parameter is associated with the *COMMANDS_CLEANUP_INTERVAL* frequency parameter and the *DAYS_TO_KEEP_COMMANDS_ROWS* parameter.	Default: `TRUE` Valid values: `TRUE, FALSE`
ENABLE_CONCURRENT_REQUEST_UPDATES	This parameter is related to requests in 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` 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` 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_COST_ ROLLUP_SERVICE	Mercury IT Governance 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` 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` Valid values: `TRUE, FALSE`
ENABLE_ DASHBOARD_ LOADING_MESSAGE	If set to `TRUE`, the Dashboard displays a message as it loads a page.	Default: `FALSE` Valid values: `TRUE, FALSE`
ENABLE_DB_ SESSION_TRACKING Category: Low-level debug	If set to `TRUE`, enables a stack trace to be reported in the ITG DB Server Reports, which you can use to track the exact line of code used to request a database connection.	Default: `FALSE` 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` Valid values: `TRUE, FALSE`
ENABLE_ DIRECTORY_ CLEANUP	Determines whether the Directory Cleanup Service is enabled.	Default: `TRUE` 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_DOCUMENT_CLEANUP_SERVICE	Enables a server thread that periodically checks for documents that are no longer attached to a Mercury IT Governance Center entity, and removes those it finds from the Mercury IT Governance 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` Valid values: `TRUE, FALSE`
ENABLE_EXCEPTION_ENGINE 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. Set the exception engine interval with ***EXCEPTION_ENGINE_WAKE_UP_TIME* on page 255.	Default: `TRUE` Valid values: `TRUE, FALSE`
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 Mercury Portfolio Management.	Default: `TRUE` Valid values: `TRUE, FALSE`

Table A-1. Server configuration parameters (page 11 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
ENABLE_FLS_ PENDING_DENORM	Managing field-level security is very computationally expensive, so whenever the security settings at the field level are updated, the Mercury IT Governance Server performs a number of calculations that allow live security checks in performance. The server performs these calculations asynchronously, by a separate server thread. 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: ■ *FLS_PENDING_DENORM_WAKE_ UP_TIME* ■ *FLS_PENDING_DENORM_DAY_ OF_WEEK* ■ *FLS_PENDING_DENORM_WEEK_ INTERVAL*	`Default: TRUE` Valid values: `TRUE, FALSE`
ENABLE_FX_RATE_ UPDATE_SERVICE	Recalculates cost after financial exchange (FX) rates change.	Default: `TRUE` Valid values: `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` 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_JDBC_ LOGGING 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. This parameter is useful in debugging system performance problems. You can set this parameter in the Workbench interface without stopping the system (**Edit > Settings**).	Default: `FALSE` 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. **Remember my logon** sets a cookie on the local machine that lets a user log on to Mercury IT Governance Center later, without entering logon information. You can also view reports via 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. To disable this function, change the parameter value to `FALSE`.	Default: `TRUE` Valid values: `TRUE, FALSE`

Table A-1. Server configuration parameters (page 13 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` Valid values: `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` Valid values: `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` Valid values: `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` Valid values: `TRUE, FALSE`
ENABLE_PROGRAM_ SUMMARY_ CONDITION_ENGINE	If set to `TRUE,` enables the automatic update of program health indicators.	Default: `FALSE` 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_PROJECT_LAUNCH_FROM_ACTION_MENU	If set to `TRUE`, allows users with the required permission to open the Workbench as a stand-alone application.	Default: `TRUE` Valid values: `TRUE, FALSE`
ENABLE_QUALITY_CENTER_METRICS_SYNC	If set to `TRUE`, enables a service that synchronizes Mercury IT Governance Center with Mercury Quality Center™.	Valid values: `TRUE, FALSE`
ENABLE_QUICKLIST_UPDATE	Controls the visibility of the **Update** button on the Quick List.	Default: `TRUE` Valid values: `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: `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` Valid values: `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` Valid values: `TRUE, FALSE`

Table A-1. Server configuration parameters (page 15 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
ENABLE_SQL_TRACE Category: High-level debug	Determines whether performance statistics for all SQL statements run are placed into a trace file. The SQL trace facility generates the following statistics for each SQL statement: ■ Parse, run, and fetch counts ■ CPU and elapsed times ■ Physical reads and logical reads ■ Number of rows processed ■ Misses on the library cache ■ User name under which each parse occurred ■ Each commit and rollback This parameter corresponds to the **Enable DB Trace Mode** checkbox in the Server Settings dialog box.	Default: `FALSE` Valid values: `TRUE, FALSE`
ENABLE_STATISTICS_CALCULATION Category: Database statistics	Whether to automatically collect statistics for the cost-based optimizer. By default, statistics are rebuilt every Sunday at 1 a.m.	Default: `TRUE` Valid values: `TRUE, FALSE`
ENABLE_TIME_SHEET_NOTIFICATIONS_SERVICE	If set to `TRUE`, enables notification on time sheets.	Default: `FALSE` 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. **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` Valid values: `TRUE, FALSE`

Table A-1. Server configuration parameters (page 16 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
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. **Note:** If enabled on a busy system, Web access logging generates many log files.	Valid values: `TRUE, FALSE`
ENABLE_WEB_ SERVICES	To use the Mercury IT Governance Web services interface, set this to `TRUE`.	Valid values: `TRUE, FALSE`
**EXCEPTION_ ENGINE_WAKE_UP_ TIME Required if *ENABLE_ EXCEPTION_ENGINE* = TRUE Category: Scheduler/ services/thread	Time at which the exception engine process runs.	Default: 1 (that is, 1:00 a.m.) Valid values: 1 through 24
EXTERNAL_WEB_ PORT	If you are using an external Web server to serve IT Governance Center clients, you must configure this parameter as an available port that can communicate with the IT Governance Server. This port receives AJP (Apache JServe Protocol) requests from the external Web server. AJP is the standard protocol used for communication between a Web server and an application server. **Note:** If you are using an external Web server, you must still configure the standard Mercury IT Governance **HTTP_PORT*. This port is used internally by Mercury IT Governance Center reports, there is no need to make it accessible to the network.	Valid value: Any available port number

Table A-1. Server configuration parameters (page 17 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
FAIL_EXECUTIONS_ ON_STARTUP	If the Mercury IT Governance 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` Valid values: `TRUE, FALSE`
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` 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 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 Valid values: Number between 1 and 4, inclusive

Table A-1. Server configuration parameters (page 18 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
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.	
GRAPHICAL_WF_ENABLE	If set to `TRUE`, makes links to view Graphical Workflow available on submitted requests.	Default: `TRUE` Valid values: `TRUE, FALSE`
GZIP_ENCODING_ENABLED	Determines whether HTTP responses are compressed before they are sent to Mercury IT Governance Center HTML clients. If set to `TRUE`, then textual HTTP responses are compressed using GZIP compression (if the requesting browser supports GZIP). By default, this is set to `TRUE` to improve the responsiveness of the IT Governance Center standard (HTML) interface, because less overall data is carried across the Internet between the client and the Mercury IT Governance Server. If all Mercury IT Governance Center clients have fast network access to the Mercury IT Governance Server, then consider setting this parameter to `FALSE` to reduce the overhead of compressing and decompressing responses.	Default: `TRUE` 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

Table A-1. Server configuration parameters (page 19 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
*HOURS_TO_KEEP_DEBUG_MESSAGE_ROWS Category: Cleanup	The number of hours that rows in the `KNTA_DEBUG_MESSAGES` table are to be kept. For high-volume Mercury IT Governance Center installations, a large number of rows may be generated in this table. For such installations, decrease this value accordingly. *See also* *ENABLE_DEBUG_MESSAGE_CLEANUP* on page 248.	Default: 48
*HTTP_PORT	Port to use to communicate with the built-in HTTP server. If Mercury IT Governance Center is in stand-alone mode (that is, it is not integrated with an external Web server), then Mercury IT Governance Center clients must have access to the `HTTP_PORT`. If Mercury IT Governance 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 Mercury IT Governance 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 Mercury IT Governance Center clients (and thus, the port need not be exposed outside of the Mercury IT Governance Server).	Default: 8080 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). If unspecified, same as *I18N_SECTION_DIRECTION*.	Valid values: `ltr, rtl` (left to right, right to left)

Table A-1. Server configuration parameters (page 20 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
I18N_ENCODING	Character encoding to be used on all HTML pages in the Mercury IT Governance Center standard interface.	Default: ISO-8859-15
I18N_LAYOUT DIRECTION	Default layout direction of HTML pages in the Mercury IT Governance Center standard interface.	Default: ltr Valid values: `ltr, rtl` (left to right, right to left)
I18N_REPORT_HTML_ CHARSET	HTML character set to use in Mercury IT Governance Center reports. Must map to the character set specified in *I18N_REPORTS_ ENCODING*.	Default: ISO-8859-15 Valid values (Windows): windows-hebrew
I18N_REPORTS_ ENCODING	Character encoding to use to generate reports in Mercury IT Governance Center. 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). 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 Mercury IT Governance Center installation. The language code must match the Mercury IT Governance Center installation language.	Default: en_US Example: `de_DE`
INTERFACE_ CLEANUP_INTERVAL	The value of this parameter determines the frequency with which the *ENABLE_INTERFACE_ CLEANUP* thread runs.	Default: 11700

Table A-1. Server configuration parameters (page 21 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
JAVA_CLASSES_LOC	Specifies the location of the JRE classes.	Example: `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: `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 Dashboard.	Default: `FALSE` Valid values: `TRUE, FALSE`
*JDBC_URL **Note:** For Oracle RAC (Real Application Clusters), this parameter must contain the host and port information for all databases to which the Mercury IT Governance Server will connect.	Locator for the database containing the Mercury IT Governance Center database schema. Must be specified correctly for Mercury IT Governance Server to communicate with the database. Format: `jdbc:oracle.thin:@<hostname>:<port>:<SID>` Where: *<hostname>* is the DNS name or IP address of the system running the database. *<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. *<SID>* is the database system ID.	Example: `jdbc:oracle:thin:@DBhost.domain.com:1521:SID`

Table A-1. Server configuration parameters (page 22 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
JSP_RECOMPILE_ENABLED	Determines whether changes to JSP files are picked up on a running server, thereby quickly making them visible. 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. If you expect JSP pages to be updated regularly, set to `TRUE`. The Mercury IT Governance Server detects JSP changes without restarting.	Default: `FALSE` on production systems, `TRUE` on development systems Valid values: `TRUE, FALSE`
**KINTANA_LDAP_ID Required if *AUTHENTICATION_MODE = LDAP	Mercury IT Governance Center account on the LDAP server. Used by the Mercury IT Governance Server to bind to the LDAP server.	Example: `uid=admin, ou=dev`
**KINTANA_LDAP_PASSWORD Required if *AUTHENTICATION_MODE = LDAP	Mercury IT Governance Center password on the LDAP server. The Mercury IT Governance Server configuration utility automatically encrypts this password. To manually edit this value, surround the encrypted password with `#!#` delimiters.	Default: `#!##!#` Example: `#!#<`*`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. **Note:** Mercury recommends that you not reset this parameter.	Example: `kintanaHome.html`
KINTANA_SERVER_DIRECTORY	Specifies the server directory location. You define this value if you are using a multiple-server (clustered) setup.	Default: `/server/kintana/`

Table A-1. Server configuration parameters (page 23 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
KINTANA_SERVER_LIST	The server sets the (read-only) value of this parameter at runtime.	Example: `aeon!rmi://ice:27099/KintanaServer`
*KINTANA_SERVER_NAME	Name of the Mercury IT Governance Server instance. If multiple Mercury IT Governance 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.	Default: kintana
*KINTANA_SESSION_TIMEOUT	The time to elapse before the Mercury IT Governance Server terminates a user session (in the Workbench or standard interface) because of inactivity. A value of 0 denotes no timeout.	Default: 120 (minutes) Valid values: 10 through 720
**LDAP_GROUP_RECURSION_LIMIT Required if **AUTHENTICATION_MODE* = LDAP	Number of levels of subgroups to traverse when importing users from groups.	Default: 15
**LDAP_SSL_PORT Required if **AUTHENTICATION_MODE* = LDAP	SSL port number on the LDAP server. If not specified, all transactions are carried over the port specified by the ***LDAP_URL* parameter.	Default: 636

Table A-1. Server configuration parameters (page 24 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values

**LDAP_URL
Required if *AUTHENTICATION_MODE = LDAP
	Comma-delimited list of LDAP URLs, which the Mercury IT Governance Server queries in the order specified.
	If no port number is specified, the default port number 389 is used.
	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 URLs specified for `LDAP_URL_FULL` do not have a DN value, the value set for `LDAP_BASE_DN` is used.
		Example:
		`ldap://ldap.theurl.com:389`
		Example:
		`ldap://10.100.102.199: 389`

LDAP_URL_FULL	IT Governance 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. **NOTES:** ■ 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. ■ 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: `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 **Note:** Setting this parameter resolves the following potential problems: ■ 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. ■ 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. ■ 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 Workbench. 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. If you set this up correctly: ■ Client A running inside the firewall connects to the internal DNS server and the machine name resolves to an IP address. ■ Client B running outside the firewall connects to an external DNS server and the machine name resolves to a different IP address. Both clients can then connect, each to a different IP address.	Example: `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 (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 Deployment Management to integrate with a mainframe system, then you must enable this “watch dog” thread. When 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` 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 (minutes)
MAX_DB_ CONNECTION_IDLE_ TIME Category: Database connection	Amount of time that an unused database connection stays open before it is closed and removed from the pool.	Default: 60 (minutes)

Table A-1. Server configuration parameters (page 27 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
MAX_DB_ CONNECTION_LIFE_ TIME Category: Database connection	Amount of time 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. Therefore, do not keep database sessions open for the life of the Mercury IT Governance Server.	Default: 1440 (minutes)
MAX_DB_ CONNECTIONS Category: Database connection	The number of connections the Mercury IT Governance 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. After this number is reached, user sessions queue for the next available database connection. The Mercury IT Governance Server rarely requires more than 25 database connections.	Default: 60
*MAX_EXECUTION_ MANAGERS 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. If your organization processes a high volume of packages, you may require more execution managers.	Default: 15
MAX_ITG_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

Table A-1. Server configuration parameters (page 28 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
*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
*MAX_RELEASE_ EXECUTION_ MANAGERS Category: Scheduler/ services/thread	Number of command executions that can run in a release distribution simultaneously. Organizations processing a high volume of packages may require a larger number of release execution managers.	Default: 15 Valid values: Number greater than 1
MAX_STATEMENT_ CACHE_SIZE	Maximum number of prepared statements cached per database connection. Part of the database connection pool settings.	Default: 50 Valid values: Integer greater than 0
*MAX_WORKER_ THREADS 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. If the Mercury IT Governance 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
MULTICAST_ CLUSTER_NAME	Unique name of a Mercury IT Governance Server cluster. Do not configure two clusters with the same name running on the same subnet.	Example: `http://wwwserver.mydomain.com/itg`
MULTICAST_DEBUG	Whether or not incoming and outgoing multicast messages are to be logged to the Mercury IT Governance Server log.	Default: `FALSE` Valid values: `TRUE, FALSE`
MULTICAST_IP	Multicast IP address.	Default: 225.39.39.244 Valid values: 224.0.0.0 through 239.255.255.255
MULTICAST_LEASE_ MILLIS	Interval at which the Mercury IT Governance Server sends out heartbeats.	Default: 20000 (milliseconds)
MULTICAST_PORT	Multicast IP port.	Default: 9000
NOTIFICATIONS_ CLEANUP_PERIOD Category: Cleanup	Interval to clean up previously-sent notifications.	Default: 7 (days)
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. **Note:** This parameter also affects generation of the Efficient Frontier curve.	Default: 100 (iterations)

Table A-1. Server configuration parameters (page 30 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
OPTIMIZER_ NUMBER_OF_ TIMESHIFTS	Maximum number of periods the optimizer can shift start dates forward. This does not affect manually-shifted Portfolio Management entities; only on the optimizer. 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 out after that date.	Default: 6 (months)
ORACLE_APPS_ ENABLED	Determines whether Mercury IT Governance Center is to be integrated with Oracle applications. You must set this parameter to `TRUE` for installations running Mercury Object Migrator or Mercury GL Migrator.	Default: `FALSE` 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: `10.1.0.3.0`
*ORACLE_HOME	Full path to the Oracle home directory on the Mercury IT Governance Server. 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 Mercury IT Governance Center database schema.	Example: `d:/orant`
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

Table A-1. Server configuration parameters (page 31 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
*PASSWORD_ EXPIRATION_DAYS	Default expiration period of passwords for new users. A value of 0 indicates no expiration.	Default: 0 (days) Valid values: 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. The value 0 indicates no restriction.	Default: 0 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 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` Valid values: `TRUE, FALSE`
PENDING_EV_ UPDATES_CLEANUP_ INTERVAL	Specifies the interval at which to run pending earned value updates.	Default: 3600 (seconds)
PGA_AGGREGATE_ TARGET	Determines the maximum physical memory Oracle can use for working areas for all processes together. *See also* *WORKAREA_SIZE_ POLICY* on page 282.	Maximum number of MB that can be dedicated to working Oracle processes.

Table A-1. Server configuration parameters (page 32 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
PORTLET_EXEC_ TIMEOUT Category: Timeout	The amount of time (in seconds) after which portlets time out. 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)
REMOTE_ADMIN_ REQUIRE_AUTH	Determines whether user authentication is required for remote administration. If set to `TRUE`, users running `kStop.sh` to shut down the Mercury IT Governance Server are required to supply a valid Mercury IT Governance Center user name and password. If set to `FALSE`, any user with access to `kStop.sh` can shut down the server.	Default: `TRUE` 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 Mercury IT Governance Server directory structure), use this parameter to specify an alternate directory here. Make sure that the Mercury IT Governance Server has access to the directory so that the report output HTML files can be written here.	Example: `D:/<ITG_ Home>/700/ aeon/ reports/`

Table A-1. Server configuration parameters (page 33 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
REPORT_LOG_DIR	Directory in which the Mercury IT Governance Center report logs are stored. **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 Example: `D:/<ITG_Home>/700/aeon/logs/reports/`
REPORTING_STATUS_REFRESH_RATE Category: Scheduler/services/thread	The frequency with which report status is refreshed and displayed to the user.	Default: 5 (seconds)
REQUEST_LOG_DIR	Specifies the location for Request execution log outputs. **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 Example: `D:/ITG/700/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 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. **Note:** Mercury strongly recommends that you not change the value of this parameter.	Default: 3600 (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`

Table A-1. Server configuration parameters (page 34 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
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` 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` Valid values: `0` through `100`
RESTRICT_BYPASS_ EXECUTION_TO_ MANAGERS	Determines whether bypass execution of workflow steps in packages is restricted to managers. If set to `TRUE`, only users with an access grant of Package Manager or Request Manager access can bypass executions. If set to `FALSE`, all users eligible to act on executions can bypass them.	Default: `FALSE` 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` Valid values: `TRUE, FALSE`
RM_DEFAULT_ EFFORT_TYPE	Setting used to determine the default effort type (hours or full-time equivalents) used to display staffing profiles and resource pool information.	Default: `fte` (full-time equivalents) Valid values: `fte`, `hours`
RM_DEFAULT_ PERIOD_TYPE	Setting used to determine the default period type used to display staffing profiles and resource pool information.	Default: `month` Valid values: `quarter, month, week, year`

Table A-1. Server configuration parameters (page 35 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
*RMI_URL	Port on which the Mercury IT Governance Server listens to initiate RMI client/server communication. Must be a unique port, distinct from the Web server, SQL*Net, and the HTTP or HTTPS ports. Format: `rmi://<hostname>:<port>/KintanaServer`	Default:1099 Valid values: Port numbers higher than 1024 Example: `rmi://gold.itg.com:1099/ITGServer`
RMI_VALIDATE_SERVER_CERTIFICATE	This parameter is used if Mercury IT Governance Server is running in secure RMI mode. If set to `TRUE`, the client 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` 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. Must be the same as the database schema name used during installation.	Valid values: Any user name format that Oracle supports
*SCHEDULER_INTERVAL 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. Set to the maximum expected time for file transfer.	Default: 10000 (milliseconds)

Table A-1. Server configuration parameters (page 36 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
SEARCH_TIMEOUT Category: Timeout	The number of seconds after which searches time out. 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 (seconds)
SECURE_RMI	If set to `TRUE`, RMI network traffic between Workbench clients and the Mercury IT Governance Server is encrypted.	Default: `FALSE` Valid values: `TRUE, FALSE`
SERVER_ENV_NAME	Name of the Mercury IT Governance Center environment containing information about the Mercury IT Governance Server machine (for example, host name, user name, and password). Must be set before Mercury IT Governance entity migrators or commands involving secure copy can run.	Default: `KINTANA_SERVER`
SERVER_MODE	Specifies the server mode to use in case you want exclusive access to a running server.	Default: `NORMAL` Valid values: `Normal, Restricted, Disabled`
*SERVER_NAME	DNS name or IP address of the machine hosting the Mercury IT Governance Server.	Default: kintana Valid values: [any valid machine name]
SERVER_TYPE_CODE	Operating system on which the Mercury IT Governance Server is installed.	Valid values: `UNIX, WINDOWS`
SHOW_BASE_URL_ON_NOTIFICATIONS	Determines whether the URL for the Mercury IT Governance Center logon window is displayed at the top of each email notification.	Default: `TRUE` Valid values: `TRUE, FALSE`

Table A-1. Server configuration parameters (page 37 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
**SMTP_SERVER 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 No, not required, but you might need to specify if you have problems running PL/SQL-based Mercury IT Governance Center reports.	The Oracle SQL*Plus version installed on the machine that hosts the Mercury IT Governance Server. You must set this for some Mercury IT Governance Center reports that run from command-line SQL*Plus calls. If you encounter problems running PL/SQL-based reports in Mercury IT Governance Center, set this parameter.	Example: `com.kintana.core.server.SQLPLUS_VERSION=8.1.7.4.0`
**STATS_CALC_DAY_OF_WEEK Required if *ENABLE_STATISTICS_CALCULATION* = TRUE Category: Database statistics	Day of the week on which to calculate Oracle database statistics.	Default: 1 (designates Sunday) Valid values: 1 through 7
**STATS_CALC_WAKE_UP_TIME Required if *ENABLE_STATISTICS_CALCULATION* = TRUE Category: Database statistics	Hour of the day (using 24-hour clock) at which statistics are to be calculated.	Default: 1 (designates 1 a.m. or 01:00) Valid values: 0 (midnight) through 23 (11 p.m. or 23:00)

Table A-1. Server configuration parameters (page 38 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
**STATS_CALC_WEEK_INTERVAL Required if *ENABLE_STATISTICS_CALCULATION* = TRUE Category: Database statistics	Frequency (in weeks) with which statistics are calculated.	Default: 1 (designates weekly calculation) Valid values: 1 through 52 Example: 2 (designates every other week)
SYNC_EXEC_INIT_WAIT_TIME	Duration after which the intermediate Request Working page opens.	Default: 4 (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 Time Management or the My Tasks portlet.	Default: (minutes) Valid values:
THREAD_POOL_MAX_THREADS Category: Scheduler/services/thread	Maximum number of packages to run simultaneously within a release distribution. If a large number of packages in a distribution are processing, increasing this value can improve performance.	Default: 10
THREAD_POOL_MIN_THREADS Category: Scheduler/services/thread	Minimum number of packages to be run simultaneously within a release distribution. *See also THREAD_POOL_MAX_THREADS* on page 277.	Default: 5

Table A-1. Server configuration parameters (page 39 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
**TIME_ZONE Required if the Mercury IT Governance Server and the Oracle database are in different time zones	Use this parameter to set the time zone that is displayed to users to a time zone other than the time zone of the Mercury IT Governance Server and the associated Oracle database server. The `TIME_ZONE` parameter determines the displayed time zone on an instance-wide basis. That is, all times displayed to the user are in this time zone. If the IT Governance Server and the machine that hosts the Oracle database are in the same time zone, leave the parameter unspecified. If they are in different time zones, set this to the time zone of the host Oracle database. You can use the following formats to specify the `TIME_ZONE` value: ■ `GMT+/–hh:mm` This format indicates the hours and minutes before or after Coordinated Universal Time (UTC, which was formerly Greenwich Mean Time, or GMT). The valid range for hh:mm is from GMT–12:00 to GMT+14:00. ■ `Continent/City` Example: `America/New_York` To see examples of valid time zones, query the database table v$timezone names. If you do not a specify a value for the TIME ZONE parameter, the value defaults to the time zone in which the server is running. For a list of fully-qualified names, see the Client Timezone report described in *Table 7-2* on page 146.	Valid values: Any three-digit standard time zone designation such as PST, MST, CST, EST, and GMT. Do not use daylight savings modified time zones such as EDT or PDT.

Table A-1. Server configuration parameters (page 40 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
TMG_DATE_ NOTIFICATION_ INTERVAL	Parameter to set the running interval of the Time Sheet Notification Service in minutes. Defaults to 120. Must be greater than 0.	Default: 120 (minutes) Valid values: integers > 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
TRANSFER_PATH	This specifies the default temporary directory that Mercury IT Governance 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 Mercury Deployment Management. In a server cluster, all servers must be able to access and share the specified directory.	Example: `D:/<ITG_ Home>/700/ ionia/ transfers/`
TURN_ON_ CONCURRENT_ REQUEST_WATCH_ DOG	If you are using Deployment Management to integrate with Oracle applications (via Object Migrator), then you must enable this “watch dog” thread. When 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` Valid values: `TRUE, FALSE`

Table A-1. Server configuration parameters (page 41 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
TURN_ON_ NOTIFICATIONS 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` Valid values: `TRUE, FALSE`
TURN_ON_ SCHEDULER Category: Scheduler/ services/thread	Turns on the scheduler. Usage: To improve performance, turn off the scheduler in non-production instances.	Default: `TRUE` Valid values: `TRUE, FALSE`
TURN_ON_WF_ TIMEOUT_REAPER 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. Use the ***WF_TIMEOUT_ REAPER_INTERVAL* parameter to set the frequency with which the service checks for information.	Default: `TRUE` Valid values: `TRUE, FALSE`
TZ_IS_TIME_ZONE_ DEFAULTED		Default:
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

Table A-1. Server configuration parameters (page 42 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
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 Example: `D:/<ITG_Home>/700/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 Category: Scheduler/services/thread	Time interval at which the Mercury IT Governance Server checks for pending scheduled tasks, and starts the tasks if worker threads are available.	Default: 60 (seconds)
WF_SCHEDULED_TASK_PRIORITY Category: Scheduler/services/thread	Determines the priority of scheduled tasks. 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
**WF_TIMEOUT_REAPER_INTERVAL Required if *TURN_ON_WF_TIMEOUT_REAPER* = TRUE 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. Example: If you set a timeout value of 86400 (seconds), which is 24 hours, on Monday at 10 AM, then all active workflow steps would time out immediately at 10 AM on Tuesday.	Default: 900 (in seconds)

Table A-1. Server configuration parameters (page 43 of 43)

Parameter (*Required, **Required If)	Description, Usage	Default and Valid Values
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. **Note:** Mercury 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`. 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 Mercury IT Governance Workbench interface. Use this parameter to configure IT Governance Center to use a specific version (other than the default version) of the Java plug-in to open the Workbench.	Example: `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` Valid values: `TRUE, FALSE`

Logging Parameters

Table A-2 lists the Mercury IT Governance Server configuration parameters located in the `logging.conf` file, and provides a description of each. The `logging.conf` file is located in the `<ITG_Home>/conf` directory.

Note

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)	Definition, Description, Usage	Default, Valid Values, Example
CATCH_SYSTEM_ERR	Used to determine whether to redirect `System.err` to the server log.	Default: `TRUE` Valid values: `TRUE, FALSE`
CATCH_SYSTEM_OUT	Used to determine whether to redirect `System.out` to the server log.	Default: `TRUE` Valid values: `TRUE, FALSE`
DEFAULT_SERVER_LOGGING_LEVEL	Default debug level of the Mercury IT Governance Server. Controls the verbosity of logs generated by the Mercury IT Governance Server. The values, which can also be set dynamically at runtime in the Workbench Server Settings window, map as follows: ■ `ERROR` maps to None in the Server Settings window ■ `INFO` maps to Normal ■ `DEBUG` maps to Max For more information about the Server Settings window, see *Setting Debugging and Tracing Parameters* on page 150.	Valid values: ■ `NONE` - No information, (including errors) is logged ■ `ERROR` - Only errors are logged ■ `INFO` - Errors and additional information is logged ■ `DEBUG` - Includes verbose debugging messages ■ `ALL` - Displays all log messages generated

Table A-2. Logging parameters (page 2 of 3)

Parameter (*Required)	Definition, Description, Usage	Default, Valid Values, Example
DEFAULT_USER_DEBUG_LEVEL Category: High-level debug	Specifies the default debug level of a user's client session. 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. The values, which can also be set in the Workbench Server Settings window dynamically at runtime, map as follows: ■ ERROR maps to None in the Server Settings window ■ INFO maps to Normal ■ DEBUG maps to Max For more information about the Server Settings window, see *Setting Debugging and Tracing Parameters* on page 150.	Valid values: ■ `NONE` - No information, (including errors) is logged ■ `ERROR` - Only errors are logged ■ `INFO` - Errors and additional information is logged ■ `DEBUG` - Includes verbose debugging messages ■ `ALL` - Displays all log messages generated
ENABLE_CONSOLE_LOGGING	Enables logging by the Mercury IT Governance Server to the console.	Valid values: `TRUE, FALSE`
ENABLE_WEB_ACCESS_LOGGING	Whether or not to log information sent to the internal Mercury IT Governance Web server (Tomcat).	Valid values: `TRUE, FALSE`
FILE_RECHECK_INTERVAL	Time interval (in seconds) at which the `logging.conf` file is checked for changes. The file keeps being checked as long as the Mercury IT Governance Server is running.	Default: 30
FILE_UPLOAD_MAX_BYTES	Maximum size of an attached file in Mercury IT Governance Center.	Valid values: Up to 50MB
LOG_LAYOUT	Layout format of the log files.	Default: `TEXT` Valid values: `TEXT, XML`

Table A-2. Logging parameters (page 3 of 3)

Parameter (*Required)	Definition, 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 Mercury IT Governance 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 Mercury IT Governance Server renames it (to `serverLog_<timestamp>.txt`), and starts a new serverLog.txt file.	Default: 250
SERVER_DEBUG_LEVEL Category: High-level debug	Debug level of the Mercury IT Governance Server. Controls the verbosity of logs generated by independent server processes (for example, EmailNotificationAgent). 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 Mercury IT Governance Server configuration parameters in the `LdapAttribute.conf` file. This file is located in the `<ITG_Home>/conf` directory.

Use the `LdapAttribute.conf` file to map the attributes of the LDAP server with the attributes used by the Mercury IT Governance Server. The default mapping uses the standard LDAP attributes. All values are case-sensitive. Do not add spaces between tokens.

Note

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.

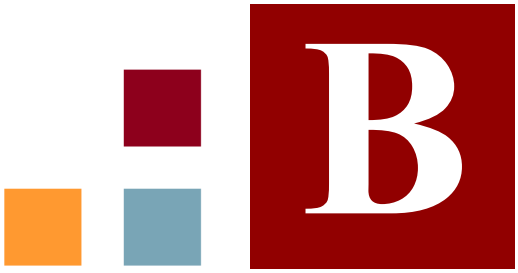
Table A-3. LDAP Attribute parameters (page 1 of 2)

Parameter (*Required)	Definition, Description, Usage	Default, Valid Values, Example
KNTA_USERS_INT	Target table for the import. Can be mapped to any LDAP attribute. Always map both `VISIBLE_USER_DATA` and `USER_DATA`. To disable default mapping, either comment out or delete the mapping line. Mappings: ■ USERNAME = sAMAccountName ■ FIRST_NAME = givenname ■ LAST_NAME = sn ■ EMAIL_ADDRESS = mail ■ PHONE_NUMBER = telephonenumber ■ DEPARTMENT_MEANING = departmentNumber ■ LOCATION_MEANING = locality ■ MANAGER_USERNAME = manager ■ USER_DATA1 = mail ■ VISIBLE_USER_DATA1 = mail	Format: ColumnName = LDAPAttribute

Table A-3. LDAP Attribute parameters (page 2 of 2)

Parameter (*Required)	Definition, Description, Usage	Default, Valid Values, Example
LDAP_TIME_ FORMAT	Attribute that keeps track of the time format used by the LDAP server.	Format for Active Directory servers: `yyyyMMddHHmmss'.0Z'` Format for Netscape LDAP servers: `yyyyMMddHHmmss'Z'`
LDAP_USER_ OBJECTCLASS	Objectclass attribute for a user on the LDAP server.	Default: person

Appendix



Server Directory Structure and Server Tools

In This Appendix:

Overview of Directory Structure

This appendix addresses the `mitg700` and `<ITG_Home>` directories and the scripts and tools they contain. The `mitg700` directory (the installation directory) contains two subdirectories that relate to the Oracle database schemas: `mitg700/sys` and `mitg700/system`.

The `<ITG_Home>` directory (the install directory for Mercury IT Governance 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 Mercury IT Governance Server or system.

mitg700/system Directory

The `mitg700/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 (page 1 of 2)

Variable	Description
ITG_User	User name of the new database schema.
ITG_Password	Password of the new database schema.
Data_Tablespace	Tablespace used to store Mercury IT Governance Center tables.

Table B-1. CreateKintanaUser.sql variables (page 2 of 2)

Variable	Description
Index_Tablespace	Tablespace used to store Mercury IT Governance 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 Mercury IT Governance Center database tables.
Rml_temp_tablepace	Temporary tablespace.

<*ITG_Home*>/bin Directory

The `bin` subdirectory of `<ITG_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 Mercury IT Governance Center database schema. This information can be very important in improving the overall performance of Mercury IT Governance Center. For information about how to use this script, see *Using Scripts to Collect Additional Statistics* on page 169.

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`), Mercury 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 68.

kDeploy.sh

The `kDeploy.sh` script is a command-line tool used to install Mercury Deployment Management Extensions, Mercury IT Governance Center Best Practices, and Mercury IT Governance 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 Mercury IT Governance Center version on top of which you can install the service pack.
- `<id>` is the unique identifier for service pack.
- `.#'` is the revision number for the deployment (optional)
- `.md` represents Mercury deployment.

For example, to install a product service pack SP1:

1. Extract the deployment JAR file.

   Note

   This file must be in the `<ITG_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 on page 294 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. For example, the command to install a Mercury IT Governance Center service pack (SP) could be: **`sh kDeploy.sh -i SP14`**

-l	Lists the deployments that are installed in an instance.
	For example:
	`sh kDeploy.sh -l`
	results in:
	`JAVA_HOME = /u1/java/j2sdk1_3_1_07`
	`java version "1.3.1_07"`
	`Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.1_07-b02)`
	`Java HotSpot(TM) Client VM (build 1.3.1_07-b02, mixed mode)`

-D	Searches for bundles in a given directory. For example, to search for a file in the `DIR` directory, run the following: `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 Mercury IT Governance Center database schema password in the command. Automates command execution but may be a security risk.
-u	Includes the Mercury IT Governance Center user name in the command.
-p	Includes the password for the Mercury IT Governance 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 Mercury IT Governance 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 Mercury 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 Mercury IT Governance Center standard (HTML) interface, the server must compile the page. To eliminate this initial performance drag, run the `kJSPCompiler.sh` script to precompile all of the HTML interface pages before users request them.

This gives first-time users faster access to the standard Mercury IT Governance Center interface.

kKeygen.sh

The `kKeygen.sh` script generates new security keys.

kMigratorExtract.sh

The script `kMigratorExtract.sh` is used in Mercury IT Governance Center entity migration.

kMigratorImport.sh

Use the `kMigratorImport.sh` script to migrate Mercury IT Governance 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 Mercury IT Governance Server. This utility provides a summary of current activity on the system and the number of database connections made.

Note

You can also access this functionality through the Workbench. To access and run these diagnostic reports from the 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 Mercury IT Governance Server as a background process. For more details about starting the server, see *Starting and Stopping the Mercury IT Governance Server* on page 68.

kStatus.sh

Run the `kStatus.sh` script to check the state of the Mercury IT Governance 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 Mercury IT Governance 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.

Note

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 Mercury IT Governance 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 Mercury IT Governance Server* on page 68.

kSupport.sh

Use the `kSupport.sh` script to gather information useful to 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 Mercury 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 Mercury IT Governance 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 Mercury IT Governance Workbench. When you run the script, it prompts you for your Mercury IT Governance Center user name and password, and for the message text.

setServerMode.sh

The setServerMode.sh script, located in the `<ITG_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 allows users with Administrator access grant to log on. The server cannot run scheduled executions, notifications, or the concurrent request manager while in this mode.

  Before you can install a Mercury 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 Mercury IT Governance 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 68.

<*ITG_Home*>/docs Directory

The `docs` subdirectory contains all documentation files for Mercury IT Governance Center (to view them, you need Adobe Reader). When you install Mercury Deployment Management Extensions, Accelerators, and Migrators (involves different installation procedures than Mercury IT Governance Center), the installation script also installs the corresponding Extension documentation into the `docs` directory.

You can also access product documentation:

- From **Product Information > Documentation** in either the Mercury IT Governance Center standard interface or the Workbench interface
- The Mercury IT Governance Download Center

<*ITG_Home*>/integration Subdirectory

The `integration` subdirectory contains information or examples for various common integrations between the Mercury IT Governance Server and external systems. For example, the `<ITG_Home>/integration/webserver` directory contains information about each external Web server that you can integrate with the Mercury IT Governance 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.

<*ITG_Home*>/logs Directory

The server directory structure has two log directories. The `<ITG_Home>/logs` directory contains the `reports` subdirectory, which contains a log file for each Mercury IT Governance Server report that is run, and directories named `PKG_number` and `REQ_number`. These subdirectories contain execution logs for Deployment Management packages and Request Management requests. The `<number>` variable in the directory name corresponds to the ID of the package or request being run.

The other log directory, `<ITG_Home>/server/kintana/log` contains all Mercury IT Governance 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 Workbench as Administrator and reset the server debug level to `Maximum debugging information` (select **Edit > Debug Settings**).

<*ITG_Home*>/reports Directory

The `reports` subdirectory contains the HTML files for all reports that Mercury IT Governance Center clients have run.

<*ITG_Home*>/server Directory

The `<ITG_Home>/server` directory contains the deployed Mercury IT Governance Server. Typically, administrators are not required to make any changes in this directory. Server configurations are handled through the provided admin scripts in the `<ITG_Home>/bin` directory.

<*ITG_Home*>/sql Directory

The `sql` subdirectory contains source code for the built-in Mercury IT Governance Center reports and core PL/SQL packages. This is provided for convenience and for customization needs.

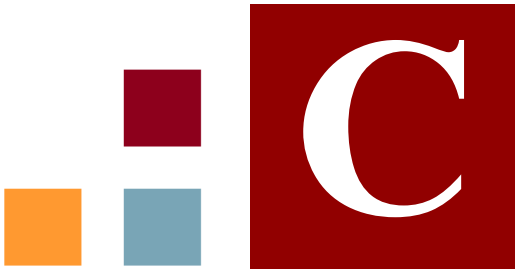
<*ITG_Home*>/transfers Directory

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 208.

Other Directories

Other directories contain reference files, as indicated by their names. You are not likely to require access to these directories.

Appendix



Preinstallation Checklists

In This Appendix:

Preliminary Tasks

Before you can install Mercury IT Governance 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 Mercury IT Governance 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 Mercury Product Support Organization (PSO) representative. The checklist will help your PSO representative make the necessary preparations before installation and speed up the installation process. If you have questions or concerns, contact the PSO representative or log a service request on our support site at support.mercury.com.

Note

The tables in the following sections describe some system requirements. For a complete list of requirements, see the *System Requirements and Compatibility Matrix* document.

Preliminary Database Tasks

Table C-1 lists the Oracle database-related tasks to perform before you install Mercury IT Governance Center.

Table C-1. Preinstall checklist for database tasks

	Database Task	Information	Date / Time
__	Identify the name and IP address of the database server.		
__	Install an Oracle database to house Mercury IT Governance Center solutions.	The database server can reside on the same machine as the Mercury IT Governance Server, or on a different machine.	
__	Create the two required database schemas, and then set up access grants for them. For information about how to run the script to create the database schemas, see *Creating the Database Schemas* on page 50. For information about how to run the script that sets up the required access grants for the schemas, see *When to Set Up Grants to the Database Schema* on page 37. **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: ■ `GRANT SELECT ON v_$parameter to <Mercury_ITG_Schema>` ■ `GRANT SELECT ON v_$mystat to <Mercury_ITG_Schema>` ■ `GRANT SELECT ON v_$process` to `<Mercury_ITG_Schema>` ■ `GRANT SELECT ON v_$session to <Mercury_ITG_Schema>` ■ `GRANT EXECUTE ON dbms_stats to <Mercury_ITG_Schema>` To set up these grants before (or during) installation, run the `GrantSysPrivs.sql` script (located in the `mitg700/sys` directory).	

Preliminary Application Server Tasks

Table C-2 lists the tasks to perform on every machine you plan to use as an Mercury IT Governance Center application server.

Table C-2. Preinstall checklist for application server tasks (page 1 of 2)

	Application Server Task	Information	Date / Time
—	Identify the operating system (UNIX or Windows) running on each machine on which you plan to install the Mercury IT Governance 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 document *System Requirements and Compatibility Matrix.*		
—	Create a system (mitg) user for Mercury IT Governance Center installation and future system maintenance activities on this server. Create an email account for this system user.	Specify a user name that is consistent with your corporate naming standards.	
—	Mercury IT Governance Center requires that you set `JAVA_HOME` in the system environment of the user account to be used to start the Mercury IT Governance 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 47 of this document.	

Table C-2. Preinstall checklist for application server tasks (page 2 of 2)

	Application Server Task	Information	Date / 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. For the exact version and operating system requirements, see the document *System Requirements and Compatibility Matrix.* For information about how to install the SDK, see *Installing the Software Developer Kit (SDK)* on page 49 of this document.	
__	Each Mercury IT Governance Server requires the Oracle client library. After you install the Oracle database and client libraries, check to make sure that you can connect to that instance from the command line by running: `sqlplus <username>/<password>@<SID>`	Mercury IT Governance 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 Mercury IT Governance Center application servers.

Table C-3. Preinstall checklist for Windows servers that interact with Mercury IT Governance Servers

	Task	Information	Date / Time
__	Check to make sure that the mitg user has Administrator-level access to the machine.		
__	Check to make sure that the regional setting on the server is English (United States).		
__	Check to make sure that FTP is installed and enabled, and that Bourne shell (bash) is installed.	Product support for Windows: Van Dyke (VShell Server) OpenSSH Cygwin provides a complete UNIX-like environment. For information how to download and install Cygwin UNIX Emulator, go to cygwin.com. **Note:** Mercury recommends Van Dyke vShell.	
__	If you plan to use Mercury IT Governance Center to perform deployments to other Windows machines in your environment, make sure that each Windows server with which Mercury IT Governance Center is to interact has the following: ■ UNIX Bourne shell emulator ■ 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 Mercury IT Governance Center.

Table C-4. Preinstall checklist for network tasks

	Network Task	Information	Date / Time
—	If you plan to use Mercury IT Governance Center to perform deployments to other Windows machines in your environment, make sure that each Windows server that is to interact with Mercury IT Governance Center has the following: ■ UNIX Bourne shell emulator ■ FTP, SSH, SSH2, or Telnet server	Mercury supports the following UNIX Bourne shell emulators: ■ Van Dyke (VShell Server) ■ OpenSSH (included in Cygwin) ■ Telnet server from Microsoft Windows Services for UNIX (SFU) (a supported remote command processor) **Note:** Mercury recommends Van Dyke vShell. Cygwin provides a complete UNIX-like environment. For information how to download and install Cygwin UNIX Emulator, go to cygwin.com.	
—	Although Mercury IT Governance Server comes with its own HTTP server, you may want to use an industry-standard external Web server to serve Mercury IT Governance Center clients. If you plan to use HTTPS or a server cluster configuration, you must install and configure an external Web server.	Mercury IT Governance Center supports the following external Web servers: ■ Microsoft IIS ■ Microsoft Windows Server Sun Java System Web Server ■ Apache HTTP Server ■ IBM HTTP Server (IHS) For information on supported versions, see *System Requirements and Compatibility Matrix.* Mercury IT Governance Server cluster uses an external Web server to load balance Web traffic across multiple application servers. For the most current configuration information, see the document *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 Mercury IT Governance Center Dashboard and Workbench.

Table C-5. Preinstall checklist for client machine tasks

	Client Task	Information	Date / Time
__	Check to make sure that the client machine has a supported Web browser installed.	For information on which Web browsers are supported, see the document *System Requirements and Compatibility Matrix.*	
__	Check to make sure that the client machine has sufficient RAM for the part of Mercury IT Governance Center it is to access on Workbench.	Client machines that access the Mercury IT Governance Workbench must have at least 256 MB of RAM. Client machines that are not intended to access the Workbench must have at least 128 MB of RAM. For client machines that access Mercury Project Management, Mercury recommends 512 MB of RAM.	
__	Check to make sure that the client machine has sufficient disk space.	100 MB free disk space is required	
__	Check to 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

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

N

O

P

R

S

T

U

V

W