

Mercury IT Governance Center™
**System Administration
Guide and Reference**

Version: 6.0



MERCURY™

This manual, and the accompanying software and other documentation, is protected by U.S. and international copyright laws, and may be used only in accordance with the accompanying license agreement. Features of the software, and of other products and services of Mercury Interactive Corporation, may be covered by one or more of the following patents: United States: 5,511,185; 5,657,438; 5,701,139; 5,870,559; 5,958,008; 5,974,572; 6,137,782; 6,138,157; 6,144,962; 6,205,122; 6,237,006; 6,341,310; 6,360,332, 6,449,739; 6,470,383; 6,477,483; 6,549,944; 6,560,564; 6,564,342; 6,587,969; 6,631,408; 6,631,411; 6,633,912; 6,694,288; 6,738,813; 6,738,933; 6,754,701; 6,792,460 and 6,810,494. Australia: 763468 and 762554. Other patents pending. All rights reserved.

Mercury, Mercury Interactive, the Mercury logo, the Mercury Interactive logo, LoadRunner, WinRunner, SiteScope and TestDirector are trademarks of Mercury Interactive Corporation and may be registered in certain jurisdictions. The absence of a trademark from this list does not constitute a waiver of Mercury's intellectual property rights concerning that trademark.

All other company, brand and product names may be trademarks or registered trademarks of their respective holders. Mercury disclaims any responsibility for specifying which marks are owned by which companies or which organizations.

Mercury
379 North Whisman Road
Mountain View, CA 94043
Tel: (650) 603-5200
Toll Free: (800) TEST-911
Customer Support: (877) TEST-HLP
Fax: (650) 603-5300

© 1997–2005 Mercury Interactive Corporation. All rights reserved.

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

Table of Contents

List of Figures	xi
List of Tables	xiii
Chapter 1: Introduction	15
About This Document.....	16
Who Should Read This Document	17
Prerequisite Documents	17
Related Documents.....	18
Chapter 2: System Overview	21
Architecture Overview	22
Client Tier.....	23
Application Server Tier	24
Database Tier	25
System Configurations	26
Single-server Configurations.....	26
Single-Server/Single-Machine Configuration	26
Single-Server/Multiple-Machine Configuration	28
Single-Server/External Web Server Configuration	29
Server Cluster Configurations.....	31
Server Cluster/External Web Server Configuration	32
Server Cluster Hardware Load Balancer Configuration	35
Chapter 3: Installation/Upgrade Overview	37
Key Questions.....	38
Installing for the First Time?	38

Upgrading?.....	39
Installing the Optional Document Management Module?	40
Installing/Upgrading Mercury Object Migrator or GL Migrator?	41
Installing/Upgrading a Mercury Change Management Extension?	41
All System Requirements Met?.....	41
License Keys Obtained?.....	41
(Windows Only) UNIX Emulator and Telnet Server Installed?.....	42
Installing or Upgrading Mercury Best Practices?	42
Key Decisions	43
Compile the JSP Files?.....	43
(Installation Only) Configure the Server During or After Installation?.....	43
(Installation Only) Give Grants to Database Schema Before or During Installation?	43
(Object Migrator, Upgrade) Run on a Single Database Schema?.....	44
(Installation Only) Create the Database Schemas Automatically?	44
(UNIX Only) Run in Graphic (Swing) or Console Mode?	44
Chapter 4: Installing Mercury IT Governance Center	47
Preparing for Installation.....	48
Collecting the Information Required.....	48
Downloading the Installation Files.....	51
Unzipping the Installation Files	52
Verifying that the JAVA_HOME Parameter is Set.....	52
Creating a Mercury IT Governance Center User.....	53
Installing the Software Developer Kit (SDK).....	53
(Optional) Creating the Database Schemas.....	55
Running the Installation Script	57
Installing on Windows	57
Installing on UNIX.....	59
(Windows Only) Configuring the FTP Server	61
Verifying the Installation	62
What to Do Next	62
Chapter 5: Upgrading to Release 6.0	63
Preparing for Upgrade	64
Backing Up the Existing Application.....	66
Backing Up the File System.....	66
Exporting the Database Schema	68
Collecting the Information Required.....	69
Downloading the Upgrade Files	70
Unzipping the Upgrade Files	70
Verifying That the JAVA_HOME Parameter is Set.....	71
Running the Upgrade Script	72
Upgrading on Windows.....	74

Upgrading on UNIX	74
(Windows Only) Configuring the FTP Server	75
Verifying the Upgrade	75
(Optional) Post-Upgrade Activities.....	76
(Microsoft Project Users Upgrading from Release 5.0) Updating Project Data.....	76
Updating Users with Resource Information	76
What to Do Next	77
Chapter 6: Configuring the System	79
Starting and Stopping the Mercury IT Governance Server	80
Setting Server Modes	80
Starting the Server	81
Stopping the Server.....	82
Configuring or Reconfiguring the Server	83
Standard Configuration Procedure.....	83
Defining Custom and Special Parameters.....	85
(Optional) Enabling Secure RMI.....	87
(Optional) Generating Password Security	88
Verifying Client Access to the Server	89
Configuring or Reconfiguring the Database.....	90
Database Parameters	90
DB_BLOCK_SIZE.....	90
DB_CACHE_SIZE.....	91
GLOBAL_NAMES	91
LOG_BUFFER.....	92
(RAC Only) MAX_COMMIT_PROPAGATION_DELAY	92
OPEN_CURSORS	92
OPEN_LINKS	93
OPTIMIZER_MODE	93
PGA_AGGREGATE_TARGET.....	93
PROCESSES	94
(Oracle 10G or Later) SGA_TARGET.....	95
SHARED_POOL_RESERVED_SIZE	95
SHARED_POOL_SIZE	95
TIMED_STATISTICS	96
WORKAREA_SIZE_POLICY	96
Oracle Database Configuration Examples.....	97
Oracle 9i Example.....	97
Oracle 10G Example	100
Granting Select Privileges to v_ \$session	103
(Optional) Generating Database Links.....	103
Configuring the Mercury IT Governance Workbench	105
Configuring the Java Plug-in	105

Setting the Correct Version of the Java Plug-In	105
Launching the Workbench	106
Troubleshooting Default JVM Problems	106
What to Do Next	107
Chapter 7: Optional and Future Installations and Configurations	109
Installing Mercury Best Practices	110
Installing/Upgrading Mercury Change Management Extensions	111
Installing Product Patches	111
Configuring the Workbench as a Java Application	112
Copying the jar Files	112
Creating the Batch File	113
Creating kintana.bat for Windows	113
Creating kintana.sh for UNIX	114
Setting the Default Web Browser	115
Integrating with an LDAP Server	116
Configuring an External Web Server	118
Process Overview	118
Choosing an External Web Port	118
Configuring a Workers Property File	119
(Sun Web Servers Only) Configuring a workers.properties File	119
(IIS or Apache Only) Configuring a workers2.properties File	121
Configuring the External Web Server	124
Configuring the Sun Java System or Sun ONE Web Server	124
Configuring the Microsoft IIS 5.0 Web Server	127
Configuring the Microsoft IIS 6.0 Web Server	130
Configuring the Apache Web Server	132
Integrating the External Web Server with the Mercury IT Governance Server	135
Setting the server.conf Parameters	135
Validating the Integration	136
Configuring a Server Cluster	137
Server Clustering Overview	137
Server Clustering Configuration Procedures	140
External Web Server, Single Machine	140
External Web Server, Multiple Machines	142
Hardware Load Balancer, Multiple Machines	145
Starting and Stopping Servers in a Cluster	146
Validating the Cluster Configuration	147
Chapter 8: Maintaining the System	149
Overview of Administration Tools and System Maintenance	150
Administration Tools in the Standard Interface	151
Accessing the Administration Tools	151

Viewing and Cancelling Running Reports.....	151
Viewing Running Executions.....	152
Viewing Interrupted Executions	152
Administration Tools in the Workbench Interface	154
The Server Tools Windows	154
Accessing the Server Tools Windows.....	154
Access Grants Required to Use Server Tools.....	155
Using Admin Tools.....	156
Running Server Reports Using Admin Tools.....	156
Running Server Reports Using kRunServerAdminReport.sh.....	159
Using SQL Runner	160
The SQL Runner Window	160
Running SQL Statements	161
Using the Server Settings Window	162
User Settings	163
Server Settings.....	164
Getting Information from Log Files.....	165
Server Log Files	165
Report Log Files.....	167
Execution Log Files.....	167
Execution Debug Log Files	168
Temporary Log Files.....	168
Periodically Stopping and Restarting the Server.....	169
Maintaining the Database	169
Changing the Database Schema Passwords	169
Maintaining Temporary Tables.....	170
The KNTA_LOGON_ATTEMPTS Table	171
The KNTA_DEBUG_MESSAGES Table.....	171
Cleanup Parameters.....	171
Backing Up Mercury IT Governance Center Instances	171
Chapter 9: Improving System Performance.....	173
Identifying Performance Problems.....	174
Isolating Performance Problems.....	174
Collecting Statistics About the Database Schema	179
Setting the Database to Gather Statistics	179
Collecting Additional Statistics by Setting Server Parameters.....	179
Collecting Additional Statistics Using Scripts	180
Troubleshooting Performance Problems.....	181
Scheduled Reports Do Not Run at the Scheduled Time	181
Packages Do Not Execute.....	182
Nightly Reports on Sunday Do Not Finish On Time, System Slows on Monday	182
Improving System Performance	183
Tuning JVM (Java Virtual Machine) Performance.....	183

Running in Interpreted Mode	183
Debugging Problems	183
Tuning Server Cluster Performance	184
Improving Input/Output Throughput	185
Improving Advanced Searches	186
Adjusting Server Configuration Parameters	186
Cleanup Parameters	187
Debug Parameters	187
Timeout Parameters	189
Scheduler/Services/Thread Parameters	189
Database Connection Parameters	190
Logging Parameters	190
Chapter 10: Migrating Instances	191
Overview of Instance Migration	192
Copying an Existing Instance to Create a Second One	193
Running the Installation Script Twice to Create Two Instances	194
(Optional) Migrating a Document Management Module	194
Preparing for Migration	195
Obtaining a New License Key	195
Stopping the Mercury IT Governance Server	195
Migrating the Mercury IT Governance Server	196
Migrating to a Windows Machine	196
Migrating to a UNIX Machine	198
Post-Migration Activities	200
Migrating the Database Schema	201
Troubleshooting Instance Migrations	206
The Mercury IT Governance Server Does Not Start	206
The Server Starts Running, but Applications Cannot be Accessed	206
Exp Command Variables	207
Imp Command Variables	208
Chapter 11: Migrating Entities	209
Overview of Entity Migration	210
Migration Process	211
Example Migration: Extracting an Object Type	212
Defining Entity Migrators	215
Migrator Action Field	215
Basic Parameters	216
Import Flags	217
Password Fields	218
Internationalization Field	219
Environment Considerations	220

Environment Connection Protocols	220
Environment Transfer Protocols.....	220
Setting the SERVER_ENV_NAME Parameter	223
Security Considerations.....	224
Migration and Ownership.....	224
Migrations and Entity Restrictions	225
Entity Migrators	226
Data Source Migrator	226
Module Migrator	227
Object Type Migrator	228
Overview Page Section Migrator	230
Portlet Migrator	231
Project Template Migrator.....	232
Report Type Migrator.....	234
Request Header Type Migrator	236
Request Type Migrator.....	238
Special Command Migrator.....	241
User Data Context Migrator	242
Validation Migrator	243
Workflow Migrator	245
Appendix A: Server Configuration Parameters	251
Overview of Configuration Parameters.....	252
Determining Appropriate Parameter Settings	252
Required Parameters	252
Directory Path Names	253
Categories of Performance-Related Parameters.....	253
server.conf Parameters	254
logging.conf Parameters.....	272
LdapAttribute.conf Parameters.....	275
Appendix B: Server Directory Structure and Server Tools	279
Directory Structure Overview	280
mitg600/system Directory.....	281
CreateKintanaUser.sql.....	281
CreateRMLUser.sql	281
ITG_Home/bin Directory	282
kBuildStats.sh.....	282
kCancelStop.sh	282
kConvertToLog4j.sh.....	282
kConfig.sh.....	283
kDeploy.sh.....	283
kEncrypt.sh.....	284

kGenPeriods.sh	285
kGenTimeMgmtPeriods.sh	285
kJSPCompiler.sh	285
kKeygen.sh	285
kMigratorExtract.sh	285
kMigratorImport.sh	285
kRunServerAdminReport.sh	286
kStart.sh	286
kStatus.sh.....	286
kStop.sh.....	286
kSupport.sh	287
kUpdateHtml.sh.....	287
kWall.sh	288
setServerMode.sh	288
ITG_Home/docs Directory	289
ITG_Home/integration Directory	289
ITG_Home/logs Directory.....	290
ITG_Home/reports Directory	290
ITG_Home/server Directory	291
ITG_Home/sql Directory	291
ITG_Home/transfers Directory.....	291
Other Directories	291
Index	293

List of Figures

Figure 2-1	Mercury IT Governance Center architecture.....	22
Figure 2-2	Single-server/single-machine configuration.....	27
Figure 2-3	Single-server/multiple-machine configuration.....	28
Figure 2-4	Single-server/external Web server configuration.....	29
Figure 2-5	Server cluster/external Web server configuration.....	32
Figure 2-6	Server cluster/hardware load balancer configuration	35
Figure 8-1	Standard Interface: Administration menu.....	151
Figure 8-2	View Running Executions page.....	152
Figure 8-3	View Interrupted Executions page.....	152
Figure 8-4	Server Tools window.....	154
Figure 8-5	Admin Tools window.....	156
Figure 8-6	Sample Server Status report.....	157
Figure 8-7	SQL Runner window	160
Figure 8-8	Sample text results for Ping DB.....	162
Figure 8-9	Server Settings window.....	162
Figure 9-1	Identifying and addressing system performance problems.....	175
Figure 9-2	Identifying and addressing database performance problems (A)	176
Figure 9-3	Identifying and addressing Java process performance problems (B).....	177
Figure 9-4	Identifying and addressing I/O performance problems (C)	178
Figure 10-1	Moving from a single instance to multiple instances	194
Figure 10-2	Migrating data between DEV and PROD	194
Figure 11-1	KINTANA_SERVER environment	213
Figure 11-2	Change Management workflow	213
Figure 11-3	Package line definition	214

List of Figures

Figure 11-4 Execution log	214
Figure 11-5 Migrator action field.....	215
Figure 11-6 Basic parameters.....	216
Figure 11-7 Import flags.....	217
Figure 11-8 Password fields.....	218
Figure 11-9 FTP (server to server).....	221
Figure 11-10 FTP (active).....	222
Figure 11-11 FTP (passive).....	222
Figure 11-12 Data Source migrator.....	226
Figure 11-13 Module migrator.....	227
Figure 11-14 Object Type migrator	228
Figure 11-15 Overview Page Section migrator.....	230
Figure 11-16 Portlet migrator	231
Figure 11-17 Project Template migrator	232
Figure 11-18 Report Type migrator	234
Figure 11-19 Request Header Type migrator.....	236
Figure 11-20 Request Type migrator	238
Figure 11-21 Special Command migrator	241
Figure 11-22 User Data Context migrator.....	242
Figure 11-23 Validation migrator	243
Figure 11-24 Workflow migrator	245

List of Tables

Table 4-1	Required installation information	49
Table 4-2	UNIX installation modes.....	59
Table 5-1	Required upgrade information	69
Table 6-1	Special configuration parameters	86
Table 6-2	Example parameters for Oracle 9i	97
Table 6-3	Example parameters for Oracle 10G.....	100
Table 6-4	Server parameters related to the Java plug-in.....	106
Table 7-1	Server parameters affected by clustering.....	138
Table 8-1	Server Tools access grants.....	155
Table 8-2	Server reports.....	157
Table 8-3	SQL Runner window fields and buttons.....	160
Table 9-1	Database disk recommendations	185
Table 10-1	Exp command variables.....	207
Table 10-2	Imp command variables.....	208
Table 11-1	Migrator action field dependencies.....	215
Table A-1	server.conf parameters	255
Table A-2	logging.conf parameters.....	272
Table A-3	LdapAttribute.conf parameters.....	275
Table B-1	CreateKintanaUser.sql variables.....	281
Table B-2	CreateRMLUser.sql variables	281
Table B-3	Selected command-line parameters for kDeploy.sh.....	284

Chapter 1 Introduction

In This Chapter:

- *About This Document*
 - *Who Should Read This Document*
 - *Prerequisite Documents*
 - *Related Documents*
-

About This Document

This document provides information about installing, upgrading, configuring, and maintaining the Mercury IT Governance Center™ system, including:

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

Key topics in this document include the following:

- In this chapter:
 - *Who Should Read This Document* on page 17
 - *Prerequisite Documents* on page 17
 - *Related Documents* on page 18
- Chapter 2, *System Overview*, on page 21
- Chapter 3, *Installation/Upgrade Overview*, on page 37
- Chapter 4, *Installing Mercury IT Governance Center*, on page 47
- Chapter 5, *Upgrading to Release 6.0*, on page 63
- Chapter 6, *Configuring the System*, on page 79
- Chapter 7, *Optional and Future Installations and Configurations*, on page 109
- Chapter 8, *Maintaining the System*, on page 149
- Chapter 9, *Improving System Performance*, on page 173
- Chapter 10, *Migrating Instances*, on page 191
- Chapter 11, *Migrating Entities*, on page 209
- Appendix A: *Server Configuration Parameters* on page 251
- Appendix B: *Server Directory Structure and Server Tools* on page 279

Who Should Read This Document

This document is for the following users of Mercury IT Governance Center:

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

This document assumes that you are moderately knowledgeable about enterprise application development and highly skilled in enterprise system and database administration.

For More Information

For information about audience types, see the *Guide to Documentation*.

Prerequisite Documents

Documents that provide general (prerequisite) information about Mercury IT Governance Center include:

- *Getting Started*
- *Configuring the Standard Interface*
- *Key Concepts*

Documents that provide critical prerequisite installation and upgrade information to system and database administrators:

- *System Requirements and Compatibility Matrix*

Before you install or upgrade to release 6.0 of Mercury IT Governance Center, you should make sure that your operating environment meets the minimum system requirements. This document provides that information, including:

- System requirements and product compatibility information for organizations installing or upgrading to release 6.0 of Mercury IT Governance Center

- A list of products supported in prior releases of Mercury IT Governance Center that have been desupported in release 6.0
- *What's New in Release 6.0*

This document is particularly important for you to read if you are installing or upgrading Mercury IT Governance Center, because it provides information about:

- New features in release 6.0 of Mercury IT Governance Center
- Potential impacts for customers upgrading from release 5.0 or 5.5 to 6.0
- *Release Notes*

This document contains product issues and anomalies, along with any last-minute information not in the regular documentation set.

Always check the latest version of the *Release Notes*, which are on the Mercury IT Governance Download Center, before installing, re-installing, upgrading, or adding a product patch to release 6.0 of Mercury IT Governance Center.

For More Information

For information about these documents and how to access them, see the *Guide to Documentation*.

Related Documents

Key related documents, which you might need to refer to when you are configuring or maintaining 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*

For More Information

For information about these documents and how to access them, see the *Guide to Documentation*.

Chapter 2 System Overview

In This Chapter:

- *Architecture Overview*
 - *Client Tier*
 - *Application Server Tier*
 - *Database Tier*
 - *System Configurations*
 - *Single-server Configurations*
 - *Server Cluster Configurations*
-

Architecture Overview

As shown in *Figure 2-1*, Mercury IT Governance Center employs a three-tier architecture consisting of:

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

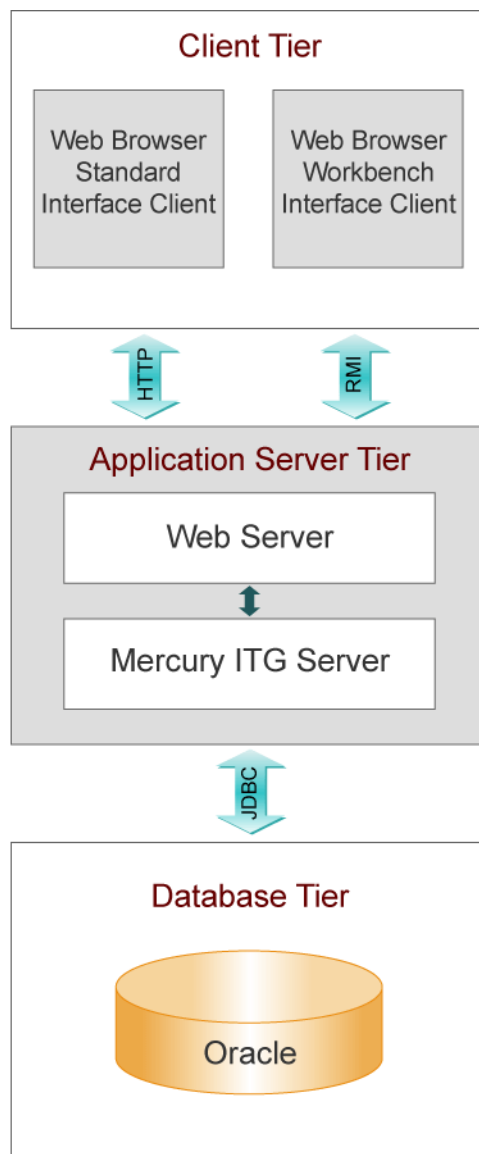


Figure 2-1. Mercury IT Governance Center architecture

Client Tier

The client tier consists of:

- The Mercury IT Governance Center standard interface, which is rendered using JSP (Java Server Pages) and is accessed using a Web browser.
- The Mercury IT Governance Workbench interface, executed using a Java applet installed on the client machine, and launched using the Sun Java plug-in to a Web browser

Communication between the client and application server tiers is accomplished as follows:

- For the standard interface, HTTP or HTTPS, with no code required on end users' machines

The client accesses information from the database through the J2EE application server using a shared database session pool.

- For the Workbench interface, RMI (Remote Method Invocation) or SRMI (Secure Remote Method Invocation), which has been optimized by Mercury for use in Mercury IT Governance Center

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

For More Information

For more information about the standard and Workbench interfaces, see the *Key Concepts* document and the *Getting Started* guide.

Application Server Tier

The application server:

- Runs on the Microsoft Windows, Sun Solaris, HP-UX, IBM AIX, and Red Hat Linux 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 like code deployment to remote systems, dynamic routing, and email notifications
- Can run on one or more machines as a cluster to improve performance and scale hardware as usage increases
- Can optionally integrate with external Web servers like 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 system failure

The protocol used for communication between Mercury IT Governance Server and Mercury IT Governance Web server is AJP13—a protocol similar to HTTP that has been optimized for performance.

Communication between the application server and database tiers is achieved using JDBC (Java Database Connectivity).

For More Information

For more information about configuring an external Web server, see [Configuring an External Web Server on page 118](#).

Database Tier

The database tier consists of an Oracle database containing the tables, procedures, PL/SQL packages, and other components used by the Mercury IT Governance Center products. 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.

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 pool to eliminate creation of 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. Mercury IT Governance Servers can be deployed in either a single-server configuration or a server cluster configuration.

The following sections discuss each of these configurations in detail.

Single-server Configurations

Single-server configurations (containing one Mercury IT Governance Server and one Oracle database) represent the norm for Mercury IT Governance Center configurations. The single Mercury IT Governance Server handles the entire user load and also functions as the Web server. The Mercury IT Governance Server machine also houses the file system for the program code, reports, execution logs, and attachments files. All other data is stored in the Oracle database.

Single-server configurations can be configured in the following ways, which are discussed in the sections following:

- 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 consists of one machine hosting both the Mercury IT Governance Server and the Oracle database.

Figure 2-2 shows a logical diagram of the single-server/single-machine configuration.

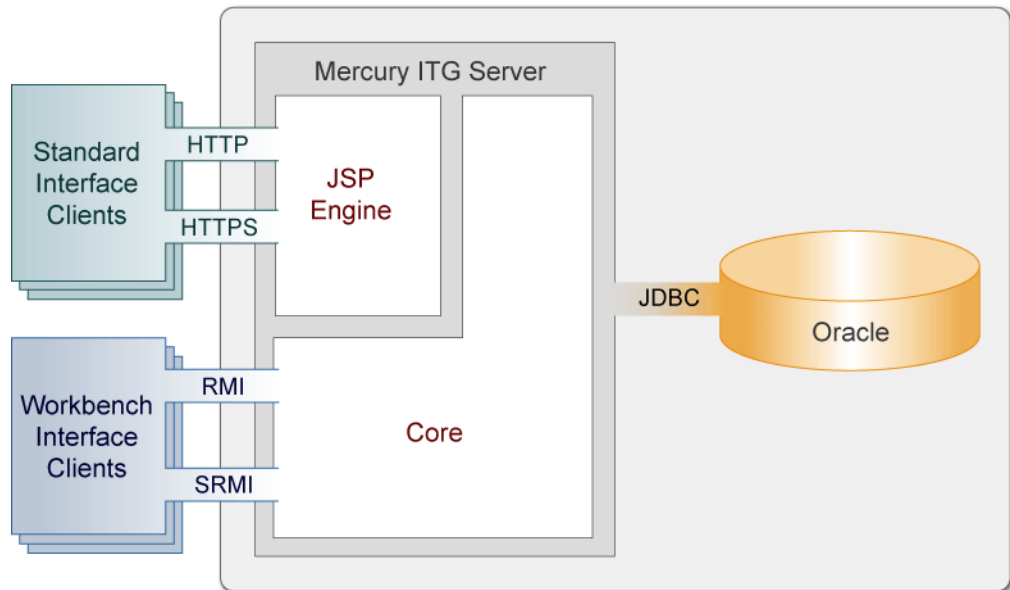


Figure 2-2. Single-server/single-machine configuration

As shown in [Figure 2-2](#):

- 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 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.

Organizations typically use this configuration when they require a dedicated machine for all Mercury IT Governance Center services and database operations. User load, transaction capacity, and system performance depends on the machine's available resources. Load balancing and server failover features are not supported by this configuration.

For More Information

For information about setting up a single-server/single-machine configuration, see [Chapter 4, *Installing Mercury IT Governance Center*, on page 47](#) or the [Chapter 5, *Upgrading to Release 6.0*, on page 63](#).

Single-Server/Multiple-Machine Configuration

In the single-server/multiple-machine configuration, the Mercury IT Governance Server and the Oracle database reside on separate machines. This configuration offers additional performance capacity and modularizes the activities of maintenance of the application server and database tiers. The separate machines do not have to be running the same operating systems, thereby allowing greater flexibility.

Figure 2-3 shows a logical diagram of the single-server/multiple-machine configuration.

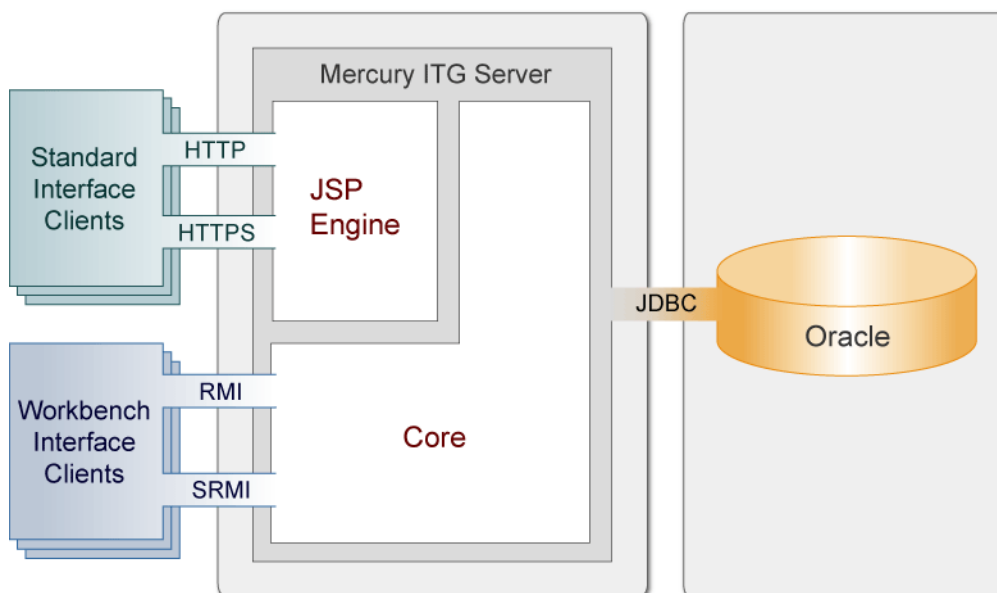


Figure 2-3. Single-server/multiple-machine configuration

As shown in *Figure 2-3*:

- 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 reside on separate machines and communicate with each other using JDBC.

Organizations typically use this configuration when they require a separate machine for database operations. User load, transaction capacity, and system performance depend on the Mercury IT Governance Server machine's

available resources. Load balancing and server failover features are not supported by this configuration.

For More Information

For information about setting up a single-server/multiple-machine configuration, see [Chapter 4, *Installing Mercury IT Governance Center*](#), on page 47 or [Chapter 5, *Upgrading to Release 6.0*](#), on page 63.

Single-Server/External Web Server Configuration

In the single-server/external Web server configuration, Web traffic comes into the Web server and is then passed to Mercury IT Governance Center. Communication between the external Web server and the Mercury IT Governance Server is achieved using AJP13, a proprietary protocol that can be more efficient than HTTP or HTTPS when communicating with Mercury IT Governance Servers using an external Web server.

[Figure 2-4](#) shows a logical diagram of the single-server/external Web server configuration.

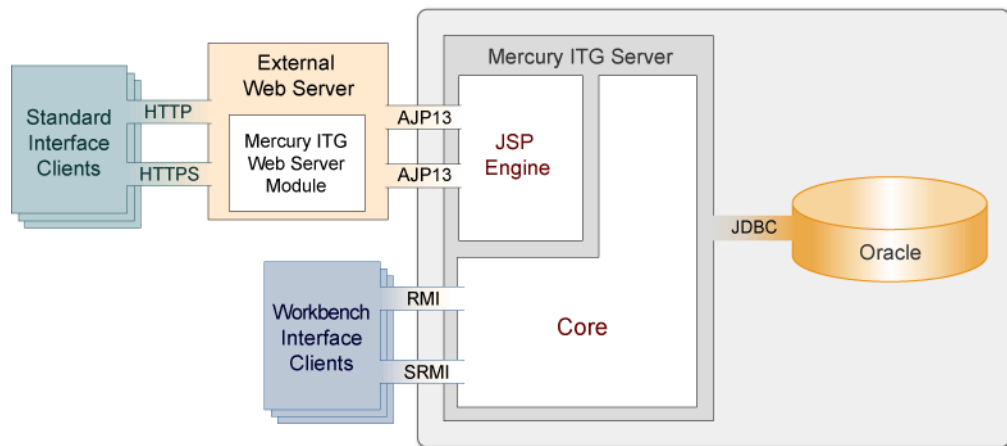


Figure 2-4. Single-server/external Web server configuration

As shown in [Figure 2-4](#):

- Standard interface clients communicate with an external Web server using HTTP, or HTTPS for secure communication. Communication between the external Web server and Mercury IT Governance Servers is achieved using AJP13.

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

This configuration is suitable for organizations that:

- Already utilize a standard Web server within their network infrastructure
- Want to 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 their standard 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. Mercury does not recommend that the Mercury IT Governance Server run as root; therefore, integration with an external Web server is recommended in this case.

As with other single-server configurations, user load, transaction capacity, and system performance depend on the Mercury IT Governance Server machine's available resources. Load balancing and server failover features are not supported by this configuration.



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 More Information

For information about setting up a single-server/external Web server configuration, see [Chapter 4, *Installing Mercury IT Governance Center*, on page 47](#) or [Chapter 5, *Upgrading to Release 6.0*, on page 63](#), and [Chapter 7, *Optional and Future Installations and Configurations*, on page 109](#).

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

Server Cluster Configurations

Server cluster configurations improve performance on systems that handle high transaction volumes or large numbers of concurrent users. In addition to meeting the needs of larger user loads and providing greater scalability, server cluster configurations support load balancing and server failover features to help make sure that mission-critical systems can provide constant and optimal accessibility 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 balance user connections across multiple Mercury IT Governance Servers. If a Mercury IT Governance Server becomes unavailable, the activities running on that server are automatically transferred to an available Mercury IT Governance Server in the cluster. This server failover feature helps to make sure that Mercury IT Governance Center system services, for example, email notifications and scheduled executions, remain operational.

Server cluster configurations contain two or more Mercury IT Governance Servers and an Oracle database. One of the Mercury IT Governance Servers (the first one installed) is called the primary server. The other server (assuming a two-server setup) is called 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.

Server cluster configurations can be implemented with a single machine or multiple machines. To run multiple Mercury IT Governance Servers on a single machine, its memory capacity and CPU usage must meet the memory and CPU requirements of the 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.

As discussed in the following sections, server clusters can be configured in one of the following ways:

- Server cluster/external Web server
- Server cluster/hardware load balancer

Server Cluster/External Web Server Configuration

Server cluster/external Web server configurations utilize an external Web server to distribute 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 want 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 organizations that already utilize a standard Web server within their network infrastructure.

User load, transaction capacity, and system performance can usually be improved using this configuration. The degree of improvement depends on the number of Mercury IT Governance Servers in the cluster and their available resources. Load balancing and server failover features are supported by this configuration.

Figure 2-5 shows a logical diagram of the server cluster/external Web server configuration.

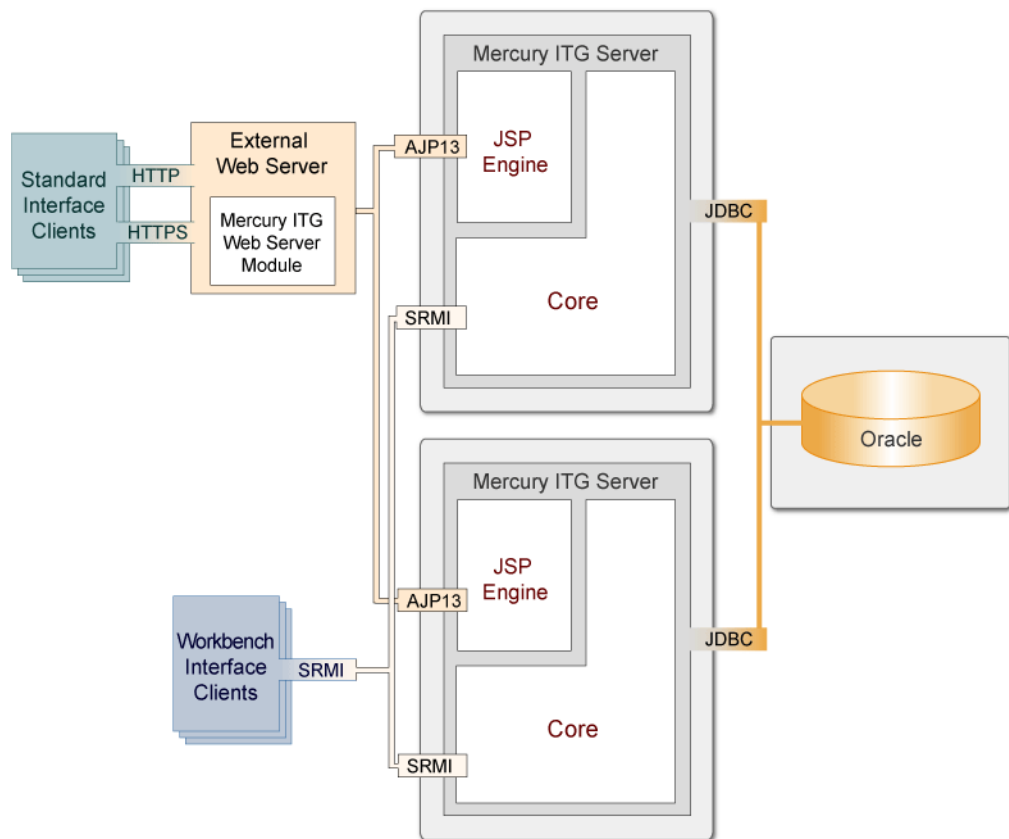


Figure 2-5. Server cluster/external Web server configuration

As shown in *Figure 2-5*:

- 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. All users can see is 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 protocol used between the Mercury IT Governance Web server module and the Mercury IT Governance Servers is AJP13.
- The Mercury IT Governance Servers also accept RMI or SRMI connections from Workbench interface users who run applets in browsers to directly connect to the Mercury IT Governance Server using this protocol.
- The Mercury IT Governance Server communicates with the Oracle database using JDBC.

Software Load Balancing

The Mercury IT Governance Center Web server module can be used 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 makes 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

Single sign-on integration with products like Netegrity SiteMinder can be achieved through integration with an external Web server.

Using SSL Accelerators

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

Communication between the external Web server and Mercury IT Governance Servers is achieved using AJP13, a proprietary protocol that can be more efficient than HTTP when communicating with Mercury IT Governance Servers using an external Web server.

For More Information

For information about setting up a server cluster/external Web server configuration, see [Chapter 7, *Optional and Future Installations and Configurations*](#), on page 109.

Server Cluster Hardware Load Balancer Configuration

The server cluster/hardware load balancer configuration is similar to the server cluster/external Web server configuration, except that, 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 client connections to be distributed evenly among available Mercury IT Governance Servers based on server load and availability.

Figure 2-6 shows a logical diagram of the server cluster/load balancer configuration.

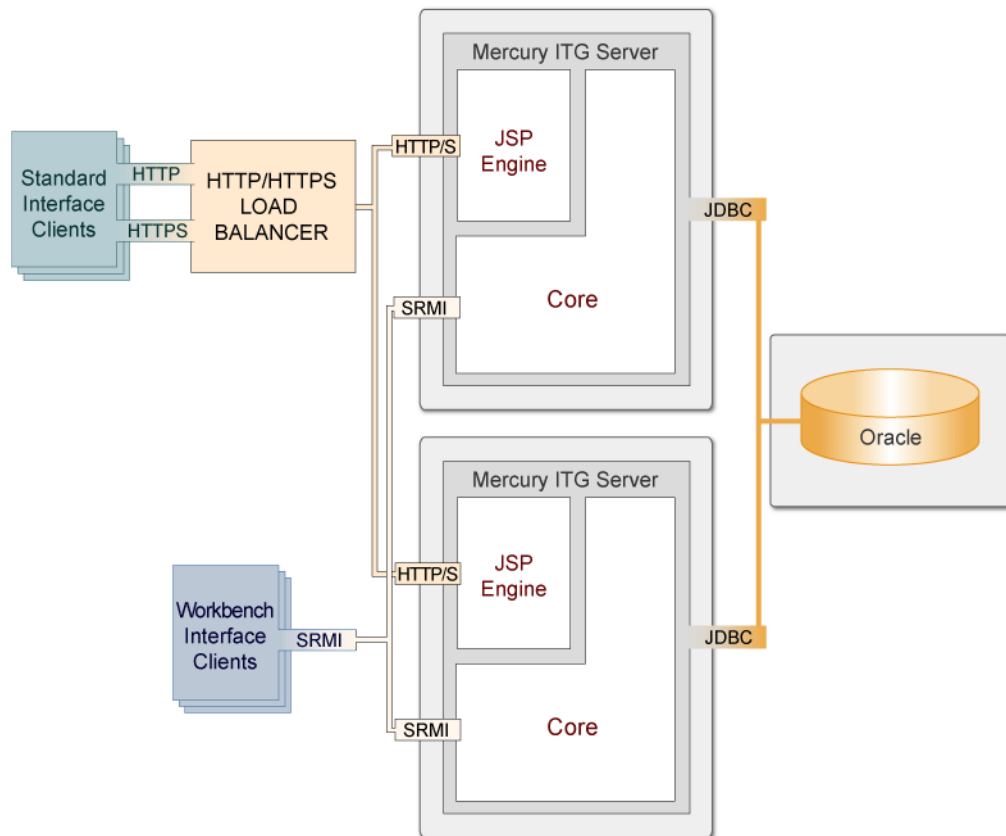


Figure 2-6. Server cluster/hardware load balancer configuration

As shown in [Figure 2-6](#):

- 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. In this configuration, the hardware load balancer behaves like a reverse proxy server and Mercury IT Governance Servers listen for HTTP or HTTPS requests that it distributes.



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 typically 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.

User load, transaction capacity, and system performance are improved using this configuration. The degree of improvement depends on the number of Mercury IT Governance Servers in the cluster and their available resources. Load balancing and server failover features are supported in this configuration.

For More Information

For information about setting up a server cluster/hardware load balancer configuration, see [Chapter 7, *Optional and Future Installations and Configurations*](#), on page 109.

Installation/Upgrade Overview

In This Chapter:

- *Key Questions*
 - *Installing for the First Time?*
 - *Upgrading?*
 - *Installing the Optional Document Management Module?*
 - *Installing/Upgrading Mercury Object Migrator or GL Migrator?*
 - *Installing/Upgrading a Mercury Change Management Extension?*
 - *All System Requirements Met?*
 - *License Keys Obtained?*
 - *(Windows Only) UNIX Emulator and Telnet Server Installed?*
 - *Installing or Upgrading Mercury Best Practices?*
 - *Key Decisions*
 - *Compile the JSP Files?*
 - *(Installation Only) Configure the Server During or After Installation?*
 - *(Installation Only) Give Grants to Database Schema Before or During Installation?*
 - *(Object Migrator, Upgrade) Run on a Single Database Schema?*
 - *(Installation Only) Create the Database Schemas Automatically?*
 - *(UNIX Only) Run in Graphic (Swing) or Console Mode?*
-

Key Questions

Answering the key questions in the following sections will help get you started in installing or upgrading to Mercury IT Governance Center.

Installing for the First Time?

If you are installing Mercury IT Governance Center for the first time (rather than upgrading from a prior product release), follow these steps:

1. Read the rest of this chapter.
2. Read the *System Requirements and Compatibility Matrix* document, which is described in [Prerequisite Documents on page 17](#).
3. Read the *Release Notes*, which are also described in [Prerequisite Documents on page 17](#).
4. If you are also installing Mercury Object Migrator™, Mercury GL Migrator, or one of the Mercury Change Management Extensions, see the appropriate document or documents.

For information about the documentation for those products, see the *Guide to Documentation*.

5. Go to [Chapter 4, Installing Mercury IT Governance Center, on page 47](#) for the installation procedure you need to follow.

That chapter explains how to:

- Prepare for installation
 - Run the installation procedure
 - Verify the installation
6. Configure the Mercury IT Governance Server and system environment, which is discussed in [Chapter 6, Configuring the System, on page 79](#).
 7. Install or configure any optional products you have purchased to work with Mercury IT Governance Center. These optional installations are discussed in [Chapter 7, Optional and Future Installations and Configurations, on page 109](#).

Upgrading?

If you are upgrading from release 5.0 or 5.5 of Mercury IT Governance Center, follow these steps:

1. Read the rest of this chapter.
2. Read the *System Requirements and Compatibility Matrix* document, which is described in [Prerequisite Documents on page 17](#).
3. Read the *What's New in Release 6.0* document.

This document, which is described in [Prerequisite Documents on page 17](#), contains descriptions of new features in release 6.0 as well as potential impacts customers may encounter in upgrading from a prior release.

4. Read the *Release Notes*, which are described in [Prerequisite Documents on page 17](#).
5. If you are also upgrading Mercury Object Migrator, Mercury GL Migrator, or one of the Mercury Change Management Extensions, see the appropriate document or documents.

For information about the documentation for those products, see the *Guide to Documentation*.

6. Go to [Chapter 5, Upgrading to Release 6.0, on page 63](#) for the upgrade procedure you need to follow.

That chapter explains how to:

- Prepare for upgrade
 - Run the upgrade procedure
 - Verify the upgrade
7. Configure the Mercury IT Governance Server and system environment, which is discussed in [Chapter 6, Configuring the System, on page 79](#).
 8. Install or configure any optional products you have purchased to work with Mercury IT Governance Center. These optional installations are discussed in [Chapter 7, Optional and Future Installations and Configurations, on page 109](#).

Supported Upgrades

If the upgrade process detects an existing release 5.0 system, it upgrades in stages—first to release 5.5 and then to release 6.0. You can choose to follow this process yourself (by first upgrading your system to release 5.5 using the instructions in the 5.5 documentation, and then upgrading to release 6.0 using the instructions in the release 6.0 documentation), but this is not required.



The upgrade script actually follows that general process: It first upgrades your system to release 5.5 and then to release 6.0 (with no pause in between).

If you are upgrading from a Kintana release prior to 5.0, you must first upgrade to release 5.0 of Mercury IT Governance Center before doing a second upgrade to release 6.0. This is true for both the Mercury IT Governance Center applications as well as the Mercury Change Management Extensions and Migrators.

For information about upgrading to release 5.0, see the appropriate Mercury documentation for release 5.0.

Installing the Optional Document Management Module?

Mercury provides you with both the Mercury-configured Documentum code (which is based on EMC/Documentum Content Server 5.25, SP1) and the Documentum documentation you need to install the Mercury IT Governance Center documentation management functionality available in release 6.0.

Installing the document management functionality is a separate procedure from installing Mercury IT Governance Center.

For More Information

For more information, see the *Mercury Document Management Guide and Reference*.

Installing/Upgrading Mercury Object Migrator or GL Migrator?

If you are running Mercury IT Governance Center in the Oracle environment, and have purchased release 6.0 of Mercury Object Migrator or Mercury GL Migrator, you need to consult not only the installation or upgrade instructions in this document, but also the special instructions in the Mercury Object Migrator or Mercury GL Migrator documentation.

For More Information

For more information about the Mercury Object Migrator and Mercury GL Migrator documentation, see the *Guide to Documentation*.

Installing/Upgrading a Mercury Change Management Extension?

If you have purchased release 6.0 of any of the Mercury Change Management Extensions, you need to consult not only the installation or upgrade instructions in this document, but also the special instructions in the Mercury Change Management Extensions documentation.

If you are installing or upgrading one or more Mercury Change Management Extensions, you must do it after you have installed (or upgraded) and configured Mercury IT Governance Center, and before you use Mercury IT Governance Center for processing.

For More Information

For more information about the Mercury Change Management Extensions documentation, see the *Guide to Documentation*.

All System Requirements Met?

Before you begin your installation or upgrade, be sure your system environment meets all the requirements in the *System Requirements and Compatibility Matrix* document, which is available from the Mercury IT Governance Download Center.

License Keys Obtained?

Have you already purchased the Mercury products you intend to install at this time (you can purchase and install more products at a later time), and have you already obtained the license file?

Whether you are installing for the first time or upgrading from Mercury IT Governance Center 5.0 or 5.5, you will need a 6.0-compliant license file. Mercury IT Governance Center's license keys are delivered in a file (`license.conf`) that is copied into the `ITG_Home/conf` directory.

When customers purchase Mercury Change Management Extensions or Migrators, they are given a username and password that allows them to download product code and documentation from the Mercury IT Governance Download Center.

(Windows Only) UNIX Emulator and Telnet Server Installed?

To run Mercury IT Governance Center on Microsoft Windows, you need to have a UNIX emulator (for example, cygwin) and a telnet server (for example, MSFT Telnet).

For More Information

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

Installing or Upgrading Mercury Best Practices?

Mercury Best Practices consists of content in the form of zip files that are imported into the Mercury IT Governance Center database.

If the product license your organization has purchased includes Mercury Best Practices, you can install it with the `kDeploy.sh` script after you have installed or upgraded Mercury IT Governance Center.

For More Information

For more information about installing Mercury Best Practices, see [Installing Mercury Best Practices on page 110](#).

Key Decisions

This section describes a number of decisions you need to make before you begin to install or upgrade to release 6.0 of Mercury IT Governance Center products.

Compile the JSP Files?

One of the prompts in the installation or upgrade procedure asks whether or not the JSP files should be compiled. Compiling the JSP files helps to make sure system performance is optimal. If the files are not compiled, the Mercury IT Governance Server needs to compile the JSP files the first time a page is accessed.

Although compiling JSP files will add about 15 minutes to the installation process, Mercury strongly recommends that you compile your JSP files.

(Installation Only) Configure the Server During or After Installation?

The Mercury IT Governance Server must be configured before it can be started. If you decide to do this during installation, the procedure prompts you to input the values of various server parameters. If you choose not to configure during installation, relevant information gathered during the procedure is inserted into the server configuration file, and you can complete the task as a post-installation step.

It may be necessary to configure the server later if required information (for example, valid port numbers) is not available at the time of installation.

(Installation Only) Give Grants to Database Schema Before or During Installation?

During a step in the installation procedure, statistics are rebuilt for the Oracle optimizer. This is done to improve the performance of Mercury IT Governance Center.

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 `mitg600/sys` directory) performs these required grants.

You can execute this script now (as the SYS user, or SYSTEM on Oracle 9i) or during the installation procedure.



The installation cannot complete until the privileges have been granted and the statistics have been rebuilt.

(Object Migrator, Upgrade) Run on a Single Database Schema?

If you have Mercury Object Migrator installed, are running Object Migrator and Mercury IT Governance Center on the same schema, and are upgrading to release 6.0, Mercury recommends that you perform a cold backup before beginning to upgrade. If you encounter problems during upgrade, reverting to the cold backup will preserve your Object Migrator installation.

(Installation Only) Create the Database Schemas Automatically?

The Mercury IT Governance Server requires two database schemas to store application data. These schemas can be created automatically during the installation procedure, or a database administrator can create them ahead of time.

To create the schemas prior to the installation, follow the instructions in [\(Optional\) Creating the Database Schemas on page 55](#). If the schemas already exist at the time of installation, the procedure populates the schemas with the entities and data required to run the Mercury IT Governance Server.

(UNIX Only) Run in Graphic (Swing) or Console Mode?

On Windows platforms, Mercury IT Governance Server installation or upgrade procedures are run exclusively in graphic (also called *swing*) mode.

On UNIX platforms, Mercury IT Governance Server upgrades can be run in either graphic or console mode. When determining the mode you want to use, consider the following.

- Graphic mode is more user friendly.

It allows you to go back and change parameters before starting the installation.

- In some cases, console upgrades may be the only option.

For example, if you choose to upgrade graphically and are accessing the target machine remotely, you may need additional software or configuration. If you are accessing a UNIX system from a Windows system you will need software that allows the UNIX application to redirect the display to Windows.

Installing Mercury IT Governance Center

In This Chapter:

- *Preparing for Installation*
 - *Collecting the Information Required*
 - *Downloading the Installation Files*
 - *Unzipping the Installation Files*
 - *Verifying that the JAVA_HOME Parameter is Set*
 - *Creating a Mercury IT Governance Center User*
 - *Installing the Software Developer Kit (SDK)*
 - *(Optional) Creating the Database Schemas*
 - *Running the Installation Script*
 - *Installing on Windows*
 - *Installing on UNIX*
 - *(Windows Only) Configuring the FTP Server*
 - *Verifying the Installation*
 - *What to Do Next*
-

Preparing for Installation

Before beginning the Mercury IT Governance Center installation procedure, you need to:

1. Check the document titled *System Requirements and Compatibility Matrix* to be sure your system meets the minimum requirements.

For more information about this document, see the *Guide to Documentation*.

2. Collect the information you will need during the installation procedure.
3. Download the installation files (which are called `mitg-600-install.zip`) from the Mercury IT Governance Download Center.
4. Unzip the installation files in a temporary directory.
5. Verify that the `JAVA_HOME` parameter is set.
6. Create an IT Governance Center user.
7. Install the SDK.
8. (Optional) Create the database schemas.

These tasks are described in the following sections.



Note

The variable `ITG_Home`, which is used repeatedly throughout this document, refers to the root directory where Mercury IT Governance Center is installed. The name of this directory and its location is up to you.

You should *not* unzip the installation files in your `ITG_Home` directory—instead, choose a temporary directory in another location.

Collecting the Information Required

The Mercury IT Governance Center installation procedure prompts for several parameters used in creating and configuring the Mercury IT Governance Server. Each piece of information entered is validated before the installation continues.

[Table 4-1](#) lists the information that is required to complete the installation process.

Table 4-1. *Required installation information*

Prompt	Description
Install Location	Directory in which the Mercury IT Governance Server will be installed and configured. If the directory does not exist it will be created. The directory path cannot contain spaces.
License Configuration File	File containing valid Mercury IT Governance Center license keys. The Mercury IT Governance Server is activated by license keys, provided in a <code>license.conf</code> file, which you must obtain before installation. If you do not have a valid <code>license.conf</code> file, contact Mercury Support.
JAVA_HOME	(Windows only) Directory in which Java is installed.
ORACLE_HOME	Directory in which the Oracle client tools are installed. The directory path string cannot contain spaces.
SQL*PLUS	Location of the SQL*Plus utility. SQL*Plus is not needed for the installation, but it is required by the Mercury IT Governance Server.
System Password	(If you are creating database users during the installation) Your system password.

Table 4-1. Required installation information [continued]

Prompt	Description
Database Access Information	<p>In addition to installing the Mercury IT Governance Center file system, the installation optionally creates and populates the database schemas needed to store application data. To access the database, the installation prompts for a user name and password, and the valid components of a JDBC URL.</p> <p>If you choose to have the installation create the database schemas, you need to enter the system username and password. If you choose to create the database schemas before installation, you need to enter the Mercury IT Governance Center database schema username and password.</p> <p>The JDBC URL is used by the Mercury IT Governance Server to connect to the Oracle database — it is of the form: <code>jdbc:oracle:thin:@hostname:port:SID</code></p> <p>where:</p> <ul style="list-style-type: none"> • <i>hostname</i> is the DNS name or IP address of the computer running the database • <i>port</i> is the port used by SQL*Net to connect to this database. Its value is generally 1521, but the actual value can be obtained by looking at the corresponding entry in <code>tnsnames.ora</code>. • <i>SID</i> is the SID of the database. This is usually identical to the database connect string. If it is different, an extra parameter is required. <p>For Oracle Real Application Clusters (RAC), the <code>JDBC_URL</code> parameter must contain the host and port information for all databases to which the Mercury IT Governance Server will connect. An example to enable the Mercury IT Governance Server to communicate with databases <code>Jaguar1</code> and <code>Jaguar2</code> appears below: <code>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))</code></p>
Mercury ITG Schema	(If you have chosen to create the schema during the installation) User name and password of the Mercury IT Governance Center database schema.
Reporting Meta Layer Schema	User name and password of the Mercury IT Governance Center RML schema.
Tablespaces	Table, index, CLOB, and temporary tablespaces of the Oracle database that are to be used in the creation of schemas and database objects.

Table 4-1. Required installation information [continued]

Prompt	Description
Windows Service Name	(Windows only) Name of the service for the Mercury IT Governance Server. The installation process prefaces the service name with “Mercury ITG” to identify the service. The service name is also used to create the Start Menu item.
Holiday Schedule	Holiday schedule on which the Mercury IT Governance Center regional calendar will be based. If you choose None , a new regional calendar with no holidays is set as the system default regional calendar, which you must name in the System Calendar prompt.
System Calendar	(Only if Holiday Schedule = None) 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 the installation procedure, you cannot change it.
Region Name	Name the region for the installation, which is defined by a combination of calendar and currency. If you have only one region, name it Enterprise or your company name.
Configure Server	If you answer Yes , initiates the wizard that prompts you for values for the required (also called <i>standard</i>) set of server configuration parameters. You can choose to configure the server now or later. Table A-1 on page 255 lists the server configuration parameters. It identifies the required parameters with an asterisk.

Downloading the Installation Files

Mercury IT Governance Center installation files, as well as Mercury Change Management Extensions and Migrators installation files, are distributed from the Mercury IT Governance Download Center. To access the files, you must have a username and password to the Web site download area, which is provided to you by Mercury at the time you purchase software.

Download the Mercury IT Governance Center installation file (`mitg-600-install.zip`).

If you are also installing one or more Mercury Change Management Extensions or Migrators, see the appropriate Mercury product documentation for specific instructions about downloading and installing. This documentation is described in the *Guide to Documentation*.

Unzipping the Installation Files

Prior to running the installation driver script, you must unzip the installation files for the Mercury IT Governance Center software.

Unzip the files into a temporary directory. You can do this with a graphical application like WinZip, or using a similar command-line tools like Unzip. You can also extract bundles with `jar xvf <>`.

The unzip procedure creates a new subdirectory named `mitg600/`.

Verifying that the JAVA_HOME Parameter is Set

Mercury IT Governance Center requires that `JAVA_HOME` be set in the system environment of the user account that will be used to start the Mercury IT Governance Server.

Determining the Path in DOS

To determine the `JAVA_HOME` path in DOS:

```
echo %JAVA_HOME%
```

Determining the Path in UNIX

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

```
echo $JAVA_HOME
```

Setting the Parameter in Windows or DOS

To set the value of `JAVA_HOME` in Windows:

1. Select **Start > Settings > Control Panel**.
2. Open the System Properties window.
3. Click the **Environments** tab.
4. Set the `JAVA_HOME` variable and value.

To set the value of `JAVA_HOME` in DOS:

```
set JAVA_HOME="JVM_Install_Directory"
```

Setting the Variable in UNIX

To set the value of `JAVA_HOME` in UNIX:

```
JAVA_HOME="JVM_Install_Directory"  
export JAVA_HOME
```

Creating a Mercury IT Governance Center User

You need to create a system user for Mercury IT Governance Center installation and future system maintenance activities.

Always log on to the server machine as this user when performing 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 if a special Mercury IT Governance Center user is not used.

Creating the User in Windows

The user in Windows should be configured to be a member of the Administrators and Domain Users groups, at a minimum. This user should have full access to the installation directory for Mercury IT Governance Center and all of its subdirectories. The Administrators screen group must have at least read access to these directories.

Creating the User in UNIX

In UNIX, Mercury IT Governance Center does not require root access to be installed. Do not install the server as the root user.

Installing the Software Developer Kit (SDK)



Note

You must install the complete SDK. The JRE alone is not supported.

Since the Mercury IT Governance Server is Java-based, the machine that hosts the Mercury IT Governance Server must also host a Java Virtual Machine (JVM), which is part of a Software Developer Kit (SDK). SDKs native to the operating systems supported by Mercury IT Governance Center are available from either Sun Microsystems or from your operating system vendor.

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

```
http://java.sun.com
```

2. Install the SDK according to the vendor's instructions.

Some vendors provide custom installation packages that can be automatically installed with a command like `pkgadd`. Other vendors provide a `tar` file that needs to be extracted.

The directory where the SDK is installed will be referred to in this document as *SDK_Install_Dir*.

Note

The name of the directory path cannot contain spaces.

Note

Many operating systems require that OS-specific patches be applied before the SDK is installed. Be sure to follow all instructions provided by the vendor.

3. After installing the SDK, verify that the user that Mercury IT Governance Center will be run under has the Java executable in its path. The easiest method to verify this is to log on and run the command:

```
java -version
```

This should return the version of Java. If you receive an error message, modify the path environment variable as appropriate.

4. Be sure that the `JAVA_HOME` environment variable has been set correctly.

To check this, run the following command:

```
echo %JAVA_HOME%
```

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

To set the value of JAVA_HOME in DOS:

```
set JAVA_HOME="SDK_Install_Dir"
```

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

```
JAVA_HOME="JVM_Install_Dir"  
export JAVA_HOME
```

(Optional) Creating the Database Schemas

To create the database schemas:

1. Generate at least one rollback segment for each of your tablespaces. For Oracle 9i or above, use 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 distinct database schemas to store application data. A database administrator can create these schemas prior to the installation. Creating database schemas require privileges that a database administrator might not want to grant to a Mercury IT Governance Center administrator, so be sure this has been done prior to installation, or that the right person is available during installation.

To create the database schemas and the permissions between them:

1. Unpack the Mercury IT Governance Center installation bundle as outlined in [Running the Installation Script on page 57](#).

A directory named `mitg600` will be created. The `mitg600/sys` and `mitg600/system` directories contain the scripts you need to create the database schemas.

2. Run the script `CreateKintanaUser.sql` (located in `mitg600/system`) against the database into which Mercury IT Governance Center will be installed.

The script prompts for a username and password, and the tablespaces that should be used by the Mercury IT Governance Center database schema.

```
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 `mitg600/system`).

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

```
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 (located in `mitg600/sys`).

This script grants the privileges required by the Mercury IT Governance Server.

If you created the schemas prior to installation, specify **Please use existing schemas** when prompted during the installation procedure. Supply the same values as those used in this procedure (that is, the values of `mercury_itg_username` and `rml_username`).

Running the Installation Script

The following steps are required to install the database objects and data used by the server. The steps in this section can be performed 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 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

To install the Mercury IT Governance Server on Windows:

1. Extract all files from `mitg-600-install.zip` to the file system.

All the files and scripts necessary to install Mercury IT Governance Center are extracted. The installation procedure prompts for the location where the software should be installed—the location is immaterial.

The `mitg600` directory resulting from the extraction contains:

- The `install.exe` executable
 - Several `jar` files
 - A `system` directory
 - A `sys` directory
2. Locate the executable file `install.exe` that was extracted, and double-click it.
 3. Enter the information prompted by the installation program (see [Collecting the Information Required on page 48](#)).

Once all information has been entered, the installation program installs the Mercury IT Governance Center files and configures the database. Status bars indicate the status of the installation. A summary of the installation displays any problems that were encountered.

Once the installation has completed successfully, Mercury IT Governance Center is installed as a Windows service. You can view the properties for this service through the Services Control Panel item.

4. To complete the service setup, select the Mercury IT Governance Center service in the Services Control Panel and click **Start**.

Mercury recommends that you set the startup type to **Automatic** so that the Mercury IT Governance Server restarts automatically when the computer is rebooted. Also, if a custom Mercury IT Governance Center user is generated as recommended, set the Log On As parameter to this username.

5. To save the settings, click **Save**.

A **Start** menu item corresponding to the Windows service name entered during the installation is also created. The menu provides links to Mercury IT Governance Center documentation and an uninstall program.

If you chose not to configure the Mercury IT Governance Server during installation, see [Configuring or Reconfiguring the Server on page 83](#).



Note

Do not map the *ITG_Home* directory to be accessible from an external Web server. This introduces a potential security risk. Mercury recommends using the Mercury-supplied Web server, unless you have the special requirements described in [Single-Server/External Web Server Configuration on page 29](#).

Installing on UNIX

To install the Mercury IT Governance Server on UNIX:

1. Extract the files into an empty directory from the download bundle by running:

```
jar xvf mitg-600-install.zip
```

or

```
unzip mitg-600-install.zip
```

All the files and scripts necessary to install Mercury IT Governance Center are extracted. The installation procedure prompts for the location where the software should be installed—the location is immaterial.

The `mitg600` directory resulting from the extraction contains:

- The `install.sh` shell script
 - Several `jar` files
 - A `system` directory
 - A `sys` directory
2. Start the installation by running the installation script (as the `SYS` user) and specifying the installation mode:

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

Table 4-2. UNIX installation modes

Mode	Meaning
<code>-swing</code>	GUI mode. A wizard guides the user through the installation.
<code>-console</code>	Command line mode. The installation script runs within the terminal session.

The script performs the following actions:

- Prompts for information required for installing the server (see [Collecting the Information Required on page 48](#)).
- Generates all database tables in the tablespace specified.

- 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 for system performance reasons.

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.

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



The installation cannot complete until the privileges have been granted and the statistics have been rebuilt.

(Windows Only) Configuring the FTP Server

Mercury IT Governance Center uses FTP to perform file migrations from one machine to another. 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 Windows computers usually require some additional FTP server configuration to function with Mercury IT Governance Center.

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

Configure the FTP server according to directions from your third-party supplier.

For the File and Directory Chooser components to work properly, you must set the FTP server directory listing style to UNIX and not MS-DOS.

To set the directory listing style:

1. In Windows, open the Internet Service Manager.
2. In the left-hand panel, open the Internet Information server under Console Root.
3. Select the applicable machine name.
4. Right-click the Default FTP site that appears in the right-hand panel and select **Properties**.

The Default FTP Site window opens.

5. Click on the **Home Directory** tab.
6. Under Directory Listing Style, select **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, Mercury IT Governance Center should be able to do the same.

Verifying the Installation

To verify the installation, perform the following tasks:

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

If you run into any unexpected problems, contact Mercury Support.

What to Do Next

Once you are confident that Mercury IT Governance Center has installed successfully, delete all subdirectories of the `install_600` directory except the `logs` subdirectory.

Go on to [Chapter 6, *Configuring the System*, on page 79](#).

Chapter 5 Upgrading to Release 6.0

In This Chapter:

- *Preparing for Upgrade*
 - *Backing Up the Existing Application*
 - *Collecting the Information Required*
 - *Downloading the Upgrade Files*
 - *Unzipping the Upgrade Files*
 - *Verifying That the JAVA_HOME Parameter is Set*
 - *Running the Upgrade Script*
 - *Upgrading on Windows*
 - *Upgrading on UNIX*
 - *(Windows Only) Configuring the FTP Server*
 - *Verifying the Upgrade*
 - *(Optional) Post-Upgrade Activities*
 - *(Microsoft Project Users Upgrading from Release 5.0) Updating Project Data*
 - *Updating Users with Resource Information*
 - *What to Do Next*
-

Preparing for Upgrade

Before beginning the Mercury IT Governance Center upgrade procedure, you need to:

1. Check the document titled *System Requirements and Compatibility Matrix* to be sure your system meets the minimum requirements.

For more information about this document, see the *Guide to Documentation*.

2. Shut down your current Mercury IT Governance Center system.
3. Back up your existing Mercury IT Governance Center (both the file system and the database).
4. Separately back up any custom graphics you want to preserve from your release 5.0 or 5.5 system.

Graphics of this type left in the system will be overridden during the upgrade process.

5. Make your own copy of the `tune.conf`, `cache.conf`, `siteminder.conf`, and `LdapAttribute.conf` files, as you like.

These files are automatically backed up as part of the upgrade, and overwritten. If you have made mods to these, you can revert each `.conf` file to `.conf.pre6.0.0`.

You might want to have your own backed-up copies of these files, as well.

6. Separately back up any Special Ops solutions you want to re-apply to release 6.0.
7. Collect the information you will need during the upgrade process.
8. Download the upgrade files (which are called `mitg-600-upgrade.zip`) from the Mercury IT Governance Download Center.
9. Unzip the upgrade files.
10. Verify that the `JAVA_HOME` parameter is set.

These tasks are described in the following sections.



Note

The variable `ITG_Home` is used repeatedly throughout this document. This variable refers to the root directory where Mercury IT Governance Center is installed. The name of this directory and its location is up to you.

You should *not* unzip the upgrade files in your `ITG_Home` directory — instead, choose a temporary directory in another location.

Upgrade Procedure Summary

During the upgrade to release 6.0:

- In the file system, files are removed, replaced, and added.
- In the database:
 - Reference entities are replaced.
 - Non-reference entities are changed or upgraded.



The upgrade to release 6.0 takes longer than the release 5.0 and 5.5 upgrade procedures.

Backing Up the Existing Application

Always back up the file system and database schema before upgrading Mercury IT Governance Center.

Backing Up the File System

There are many ways to back up the Mercury IT Governance Server file system. A common approach is described below. Note that however you create the backup, the initial step to stop the server is required.



The following procedures calls for you to stop the Mercury IT Governance Server, and it will not be restarted until the upgrade procedure is complete. The upgrade will not run if the server is running. If the server is not stopped, there is a possibility that users may still be accessing the system; any new data stored between the time of backup and the time the upgrade completes may be unrecoverable if any unexpected problems occur.

Backing Up on Windows

To back up on Windows:

1. Open the Services control panel.

Stop the Mercury IT Governance Center service by selecting it (the default name is Mercury ITG *id*) and click the **Stop** button.

2. Open a DOS window and navigate to the parent directory of *ITG_Home*. Create a zip archive by issuing the following command:

```
zip backup_filename.zip -r ITG_Home
```

The archive named *backup_filename.zip* is put in the parent directory of *ITG_Home*. Keep this archive at least until you have verified that the upgrade completed successfully.

If it becomes necessary to later restore the archive, move *backup_filename.zip* into the parent directory of *ITG_Home* and issue the command:

```
unzip backup_filename.zip
```



Note

The Windows instructions described here will vary depending on the version of the zip utility on your system. Mercury recommends using WinZip.

Backing Up on UNIX

To back up on UNIX:

1. Stop the server by running *kStop.sh*, which is located in the *ITG_Home/bin* directory:

```
sh kStop.sh -now
```

2. In the *ITG_Home* directory, create a tar archive of the complete IT Governance directory tree by issuing the following command (making sure you have enough disk space):

```
tar cvf ../backup_filename.tar .
```

The archive named *backup_filename.tar* is put in the parent directory of *ITG_Home*. Keep this archive at least until you have verified that the upgrade completed successfully.

If it becomes necessary to later restore the archive, move *backup_filename.tar* into *ITG_Home* and unarchive it with the command:

```
tar xvf backup_filename.tar
```

Exporting the Database Schema

Before you begin the upgrade procedure, you need to back up all schemas and tablespaces used by Mercury IT Governance Center. Described here is a common way to back up the system.

From the command line on an Oracle server that has direct or SQL*Net access to the database on which the Mercury IT Governance Center database schema resides, issue the following command (all on one line):

```
exp USERID=system/system_password@connect_string COMPRESS=N  
INDEXES=Y FILE=mitg_date.dmp OWNER=knta
```

In this example, *mitg* is the name of the IT Governance Center database schema, and the file that will contain the export is called *mitg_date.dmp*, where *date* is the current date.

Keep the export file at least until you have verified that the upgrade has completed successfully.

If you need to restore the archive later, you must first drop the database schema, reinitialize it, and then import from the export file. To do this, execute the `CreateKintanaUser.sql` script (located in the installation bundle, and extracted to `mitg600/system`) and the `GrantSysPrivs.sql` script (located in the installation bundle, and extracted to `mitg600/sys`).

Collecting the Information Required

The Mercury IT Governance Center upgrade prompts for the information listed and described in [Table 5-1](#).

Table 5-1. Required upgrade information

Prompt	Description
Mercury ITG Schema	Password of the Mercury IT Governance Center database schema.
System Password	(If required) System password.
Rollback Segment	Rollback segment required to upgrade the system. Mercury recommends using a tablespace with at least 100 megabytes of space available with a single rollback segment.
License Configuration File	File containing valid Mercury IT Governance Center license keys. The Mercury IT Governance Server is activated by license keys, provided in a <code>license.conf</code> file, which you must obtain before installation. If you do not have a valid <code>license.conf</code> file, contact Mercury Support.
JAVA_HOME	(Windows only) Directory in which Java is installed.
Name Display Format	(Upgrade from release 5.5 only)
Holiday Schedule	Holiday schedule on which your IT Governance Center default regional calendar will be based. Holidays in the holiday schedule you select will be preset for up to ten years as non-working days in your current (5.0 or 5.5) base calendar, which will become the system default regional calendar. All other non-working days currently configured in the base calendar are also preserved. If you select None , no holidays will be added to the current base calendar (system default regional calendar). Warning: Whatever you choose (or the system default) will be applied to existing data during the upgrade procedure. You can change these values later, if you like.
System Calendar	Name of the IT Governance Center default regional calendar. The current (5.0 or 5.5) base calendar will become the system default regional calendar, which is used by default whenever a regional calendar is not specified or available. The system default regional calendar is also used as the default calendar for request work item fields. Warning: Whatever you choose (or the system default) will be applied to existing data during the upgrade procedure. You can change these values later, if you like.

Table 5-1. Required upgrade information [continued]

Prompt	Description
Region Name	<p>Name of the region for the installation, which is defined by a combination of calendar and currency.</p> <p>This region name will be added to all existing entities to preserve their current settings.</p> <p>If you have only one region, name it Enterprise or your company name.</p>
Currency Code	<p>Three-letter code for the default currency, in which company financial information is reported.</p> <p>Existing currency values will be assumed to be in this currency.</p> <p>The system default is US dollars (USD).</p> <p>Warning: Once you choose your currency during the installation procedure, you cannot change it.</p>

Downloading the Upgrade Files

Mercury IT Governance Center upgrade files, as well as Mercury Change Management Extensions and Migrators files, are distributed from the Mercury IT Governance Download Center. To access the files, you must have a username and password to the Web site download area, which is provided to you by Mercury at the time of software purchase.

Download the Mercury IT Governance Center upgrade file (`mitg-600-upgrade.zip`) and put it in the `ITG_Home` directory.

If you are also upgrading one or more Mercury Change Management Extensions or Migrators, see the appropriate product documentation for specific instructions about downloading and installing. The documentation is described in the *Guide to Documentation*.

Unzipping the Upgrade Files

Prior to running the upgrade driver script, you must unzip the upgrade files for the Mercury IT Governance Center software.

To unzip the upgrade files:

1. Make sure that the Mercury IT Governance Center installation files (`mitg-600-upgrade.zip`) is located in the `ITG_Home` directory.
2. Unzip the files into the `ITG_Home` directory.

This can be done with a graphical application like WinZip, or from a DOS window.

The unzip procedure creates a new subdirectory named *ITG_Home/upgrade_600*.



Note

Once extracted, the file `mitg-600-upgrade.zip` is no longer needed.

Verifying That the `JAVA_HOME` Parameter is Set

Mercury IT Governance Center requires that `JAVA_HOME` be set in the system environment of the user account that will be used to start the Mercury IT Governance Server.

Determining the Path in DOS

To determine the `JAVA_HOME` path in DOS:

```
echo %JAVA_HOME%
```

Determining the Path in UNIX

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

```
echo $JAVA_HOME
```

Setting the Variable in Windows or DOS

To set the value of `JAVA_HOME` in Windows:

1. Select **Start > Settings > Control Panel**.
2. Open the System Properties window.
3. Click the **Environments** tab.
4. Set the `JAVA_HOME` variable and value:

```
set JAVA_HOME="JVM_Install_Directory"
```

Setting the Variable in UNIX

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

```
JAVA_HOME="JVM_Install_Directory"  
export JAVA_HOME
```

Running the Upgrade Script



Mercury strongly recommends that you upgrade a test instance before upgrading your production instance.

The executable (Windows) or upgrade driver script (UNIX) starts the file system and database schema upgrade process, which could take several hours. You will be prompted for the passwords for the Mercury IT Governance Center user and possibly a user with system-level database accounts.

The following sections contain the procedures for running the upgrade script on Windows and UNIX.



If you do not have a public grant to `v_$session`, you need to do the following. For Mercury IT Governance Center to be able to keep track of the open database sessions it is using, be sure that a public grant exists on the `v_$session` dynamic performance table. To do this, connect as `SYS` to the database containing the Mercury IT Governance Center database schema and issue the following SQL statement:

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

Server Directory Cleanup

During the upgrade procedure your server directory is cleaned up. Contents of the directory not part of the node directories is moved to a backup directory.

The upgrade procedure looks through the values in the `server.conf` file to determine which server nodes are running on the `ITG_Home` directory being upgraded, and it applies changes to only those directories. Any other files and directories found in the `server` directory are moved to `ITG_Home/BACKUP` in a

time-stamped directory. Once the server has been upgraded and is operational, it is safe for you to delete or archive the `BACKUP` directory.

The upgrade procedure prompts you to confirm these details before continuing. If you believe something is not correct, you can cancel the upgrade and reconcile changes. The upgrade procedure reads all values of `KINTANA_SERVER_NAME` in `server.conf` and upgrades only matching directory names. All files and directories in `ITG_Home/server` that do not have corresponding `KINTANA_SERVER_NAME` definitions are moved.

For more information about server nodes, see [ITG_Home/server Directory on page 291](#).

(Upgrades from release 5.0) Additional Server Cleanup Information

If you are upgrading from release 5.0 of Mercury IT Governance Center, you might need to do some reconfiguration based on changes introduced in release 5.5.

The upgrade procedure attempts to create the proper `ITG_Home/server` directories for you, as described in the prior section, but if it cannot determine which server directories to create, it upgrades the server to a directory named `TO_BE_CONFIGURED`.

As a special post-upgrade step, you need to properly configure your server.

For more information about server nodes, see [ITG_Home/server Directory on page 291](#).

Upgrading on Windows

To run the upgrade on a Windows machine:

1. Navigate to the `ITG_Home/upgrade_600` directory.
2. Double-click on `upgrade.exe`.

The upgrade (for both Windows and UNIX) prompts you for password information and then performs some system tests before continuing the upgrade. These tests check for the existence of any temporary tables left over from a previous upgrade. If any of these tests fail, follow the on-screen instructions to fix the problem. Once all temporary tables have been reconciled, you can restart the upgrade. If you encounter unexpected errors at that point, contact Mercury Support.

Upgrading on UNIX

To run the driver in console mode:

1. Navigate to the `ITG_Home/upgrade_600` directory:

```
cd ITG_Home/upgrade_600
```

2. Do one of the following:

- a. Run the script in console mode:

```
sh upgrade.sh -console
```

- b. Run the script in graphical mode:

```
sh upgrade.sh -swing
```



Note

If you are running the driver in graphical mode, you must run in an X Window session.

(Windows Only) Configuring the FTP Server

If you are configuring Mercury IT Governance Server after the upgrade of a prior product release, you should already have an FTP server set up. If so, be sure your currently installed FTP server is supported (supported FTP servers are listed in the *System Requirements and Compatibility Matrix* document).

If you need to set up an FTP server, see [\(Windows Only\) Configuring the FTP Server on page 61](#).

Verifying the Upgrade

To verify the installation, perform the following tasks:

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

If you run into any unexpected problems or have additional questions, contact Mercury Support.

(Optional) Post-Upgrade Activities

The activities described in the following sections can be done as a final step in the upgrade process to resolve outstanding data conversions or population issues.

(Microsoft Project Users Upgrading from Release 5.0) Updating Project Data

If your project integration style is specified as either **Microsoft Project controlled** or **Project Management controls actuals**, you need to use the Microsoft Project synchronization wizard to synchronize your Mercury Project Management™ project with your Microsoft Project file.

In release 5.0 and prior releases, Kintana Drive (now Mercury Change Management™) did not synchronize scheduled effort or actual effort values for projects where the project settings marked these two fields as disabled. Since the Mercury Resource Management™ visualizations (new in release 5.5 of Mercury IT Governance Center) rely on effort values, a release 6.0 upgrade from release 5.0 defaults effort values for these projects.

This default value is calculated with the assumption that the assignments in Microsoft Project were at 100% units (full-time allocation). This means that after the release 6.0 upgrade, the new Resource Management visualizations might show some resources as over-allocated, in cases where the Microsoft Project assignments were at less than 100% units.

The inaccuracies are corrected the first time you synchronize each of your Mercury Project Management projects with Microsoft Project.

Updating Users with Resource Information

Mercury IT Governance Center user accounts are tied to a resource account in Mercury IT Governance Center. You can populate the resource information associated with each Mercury IT Governance Center user using the User window (one user at a time) or by using Mercury IT Governance Center's User Open Interface. For instructions, see the *Open Interface Guide and Reference*.

What to Do Next

Once you are confident that Mercury IT Governance Center has upgraded successfully, delete all subdirectories of the `upgrade_600` directory except the `logs` subdirectory.

Go on to [Chapter 6, *Configuring the System*](#), on page 79.

Chapter 6 Configuring the System

In This Chapter:

- *Starting and Stopping the Mercury IT Governance Server*
 - *Setting Server Modes*
 - *Starting the Server*
 - *Stopping the Server*
 - *Configuring or Reconfiguring the Server*
 - *Standard Configuration Procedure*
 - *Defining Custom and Special Parameters*
 - *(Optional) Enabling Secure RMI*
 - *(Optional) Generating Password Security*
 - *Verifying Client Access to the Server*
 - *Configuring or Reconfiguring the Database*
 - *Database Parameters*
 - *Oracle Database Configuration Examples*
 - *Granting Select Privileges to v_\$session*
 - *(Oracle Object Migration Only) Generating Database Links*
 - *Configuring the Mercury IT Governance Workbench*
 - *Configuring the Java Plug-in*
 - *Troubleshooting Default JVM Problems*
 - *What to Do Next*
-

Starting and Stopping the Mercury IT Governance Server

The following sections explain 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 31](#) and [Configuring a Server Cluster on page 137](#).



Note

Unless otherwise indicated, references to “the server” in this document mean the Mercury IT Governance Server or Mercury IT Governance Application Server, not the server machine.

Setting Server Modes

Mercury IT Governance Center supports the following server modes:

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

The server must be in restricted mode to install or upgrade any of the Mercury Change Management Extensions.

- **Disabled mode.** In disabled mode the server is prevented from being started. This happens only when a Mercury IT Governance Center upgrade has exited prior to completing.

Setting the Server Mode with `setServerMode.sh`

The `setServerMode.sh` script, located in the `ITG_Home/bin` directory, manually sets the server mode in situations where you want to obtain exclusive access to a running server. For example, to set the server mode to restricted:

```
sh setServerMode.sh -Restricted
```

Setting the Server Mode with `kConfig.sh`

You can also set the server mode with `kConfig.sh`:

1. Run `sh kConfig.sh`.

2. Select **Set Server Mode**.
3. Select **Restricted Mode** from the list.
4. Select **Finish**.

For More Information

For more information about `setServerMode.sh`, see [setServerMode.sh](#) on page 288.

For more information about `kConfig.sh`, see [kConfig.sh](#) on page 283.

Starting the Server

Starting the Server on Windows

To start the server on Windows:

1. (If you are installing or upgrading one of the Mercury Change Management Extensions) Set the server to restricted mode (see [Setting Server Modes](#) on page 80).
2. Open the Microsoft Services control panel.
3. Select the Mercury IT Governance Center service (which starts with “Mercury ITG”).
4. Click **Start**.
5. (If you have installed or upgraded one of the Extensions) Set the server to normal mode (see [Setting Server Modes](#) on page 80).

Starting the Server on UNIX

To start the server on UNIX:

1. (If you are installing or upgrading one of the Mercury Change Management Extensions) Set the server to restricted mode (see [Setting Server Modes](#) on page 80).
2. CD to the `ITG_Home/bin` directory.
3. Run the `kStart.sh` script.

4. (If you have installed or upgraded one of the Extensions) Set the server to normal mode (see [Setting Server Modes on page 80](#)).

For More Information

For more information about `kStart.sh`, see [kStart.sh on page 286](#).

For information about starting servers in a cluster, see [Starting and Stopping Servers in a Cluster on page 146](#).

Stopping the Server

Stopping the Server on Windows

To stop the server on Windows:

1. Open the Microsoft Services control panel.
2. Select the Mercury IT Governance Center service (which starts with “Mercury ITG”).
3. Click **Stop**.

Stopping the Server on UNIX

To stop the server on UNIX:

1. Navigate to the `ITG_Home/bin` directory.
2. Run the `kStop.sh` script:

```
sh kStop.sh -now -user username
```

Be sure `username` is valid user name with Administrator access privileges.

For More Information

For more information about `kStop.sh`, see [kStop.sh on page 286](#).

For information about stopping servers in a cluster, see [Starting and Stopping Servers in a Cluster on page 146](#).

Configuring or Reconfiguring the Server

If you configured the Mercury IT Governance Server during installation, you probably don't need to reconfigure it until your environment or requirements change. If you didn't configure during installation, do it now.

Most of the configuration can be done using the standard configuration procedure, which is described in the next section, [Standard Configuration Procedure](#).

In some special cases, however, it may be necessary to add custom parameters. This procedure is described in [Defining Custom and Special Parameters on page 85](#).

The server configuration tool runs in both a console and a graphical mode. In the Windows environment, the tool requires an X Window session when running graphically.

Standard Configuration Procedure

This section outlines the standard configuration procedure and describes all of the required settings for a typical installation.

To configure the Mercury IT Governance Server:

1. From the `ITG_Home/bin` directory, execute `kConfig.sh` from a DOS or UNIX command line:

- To run in graphical mode:

```
sh kConfig.sh
```

- (UNIX only) To run in console mode:

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



Note

You need to run this utility in an X Window session, since it displays a graphical user interface.

2. The configuration wizard guides you through the configuration of the Mercury IT Governance Server.

Enter a value for every parameter that is required or appropriate for your system environment. To help you determine the correct value for a given parameter, see the tool-tip description that is visible when you move your cursor over the parameter name, or see [server.conf Parameters on page 254](#).

All confidential information (for example, passwords) remains hidden and is encrypted before it is stored.

Do not change values that are defaulted except in special circumstances—for example, if you are sure the default value does not meet your organization's needs.



Use forward slashes (/), not backslashes (\), for all file path separators, regardless of your operating system environment. Mercury IT Governance Center automatically uses the appropriate backslash path separators when communicating with Windows, but expects to read only forward slashes from the configuration file.

The last section in the server configuration wizard is Custom Parameters. This is where you can define parameters you may need to set for special needs:

- If you have no custom parameters to add, leave this section blank.
- If you need to add and define custom parameters, see [Defining Custom and Special Parameters on page 85](#).

3. After you have finished with the configuration, click **OK**.

This causes the wizard to:

- Write the configuration parameters to the `server.conf` file.
- Generate other files needed internally by the Mercury IT Governance Server (for example, `jboss-service.xml`).

4. Stop and restart the server.

For information about how to do that, see [Starting and Stopping the Mercury IT Governance Server on page 80](#).



Note

You can also modify these parameters directly in the `.conf` files, which are described in [Appendix A: Server Configuration Parameters on page 251](#).

If you decide to do that, you need to run the script `kUpdateHtml.sh` after you finish the modifications.

For More Information

For a list and description of all the standard server configuration parameters, see [Appendix A: Server Configuration Parameters on page 251](#).

Defining Custom and Special Parameters

Mercury IT Governance Center has two kinds of server parameters in addition to the Mercury-supplied standard parameters:

- Custom parameters are created and defined by customers.

Custom parameter names must be preceded by the prefix `com.kintana.core.server`. For example, if a custom parameter named `NEW_PARAMETER` is being added, the name in the `Key` field should be `com.kintana.core.server.NEW_PARAMETER`.

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

- Special parameters have been created, named, and defaulted by Mercury, but they can be used in special situations if you add them to the custom parameters folder.

The special parameters are documented in [Table 6-1](#).

Table 6-1. Special configuration parameters

Parameter	Description	Sample Value
com.kintana.core.server.DB_CONNECTION_STRING	<p>When the JDBC_URL parameter is specified, the SID of the database on which the Mercury IT Governance Center schema resides is requested. It is assumed that the connection string for this database is the same as the SID. However, this is not always the case.</p> <p>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.</p>	PROD
com.kintana.core.server.NON_DOMAIN_FTP_SERVICES	<p>Windows environment only: FTP servers on Windows typically require the entry of the Windows domain along with the username (in the form <i>Domain\Username</i>) when opening an FTP session. By default, Mercury IT Governance Center includes the domain name along with the username in an FTP session to a Windows computer.</p> <p>If you use an FTP server that does not require the domain name, you can use this parameter to override the default functionality.</p> <p>Contact Mercury Support for more information.</p>	WAR-FTPD
com.kintana.core.server.TEMP_DIR	<p>A Mercury IT Governance Center temporary directory. This defaults to a <code>temp</code> subdirectory of the <code>logs</code> directory.</p> <p>Include the full path when using this parameter.</p>	

(Optional) Enabling Secure RMI

To enable SRMI (RMI over SSL), you need to:

1. Create a keystore for SSL to use.

The standard way to do this is using Java's `keytool` application.

To get more information about the `keytool` application, see www.churchillobjects.com/c/11201e.html.

The "store password" that you use when running `keytool` is the password you need to set in `KEY_STORE_PASSWORD` (see [step 2](#)).

2. Define three parameters in the `server.conf` file:

- `RMI_URL`

Use the "rmis" prefix instead of "rmi."

- `KEY_STORE_FILE`

Set this parameter to point to the keystore file.

- `KEY_STORE_PASSWORD`

This parameter must contain the keystore password, and it 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 like this:

- `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`



Note

You can create a self-signed certificate, if you like.

(Optional) Generating Password Security

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. Encryption is done using the server's public key and only the server is able to 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 the installation of Mercury IT Governance Center, reside in `ITG_Home/security`. The key pair needs to be generated only once, unless you think that the security of the server has been breached. In that case, you can regenerate the key pair and re-encrypt 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 is needed for the server to communicate with the database)
- `DB_USERNAME` (the username for the Mercury IT Governance Center database schema)
- `DB_PASSWORD` (the password for the Mercury IT Governance Center database schema)

When the script completes, it puts two keys in the `ITG_Home/security` directory:

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

On Windows, the files remain readable by anyone. As the system administrator, you need to make sure that non-trusted users do not have read privilege to those files.

On UNIX, the files are read only for the user running the script. If this user is not the same user who started the server, the server will not be able to read the keys and will be unable to start.

For More Information

For more information about `kKengen.sh`, see [kKeygen.sh](#) on page 285.

Verifying Client Access to the Server

All Mercury IT Governance Center clients log on to Mercury IT Governance Center using the same URL. The URL for Mercury IT Governance Center is formed by taking the value of the `BASE_URL` `server.conf` parameter and appending `/itg/web/knta/global/Logon.jsp`:

```
http://wwwserver.mydomain.com:port/itg/web/knta/global/Logon.jsp
```

To verify client access to the Mercury IT Governance Server after installation or upgrade, log on to a client machine as administrator.

To log on to Mercury IT Governance Center:

1. On a client machine using one of the supported browsers, enter the URL for your Mercury IT Governance Center site.

The Mercury IT Governance Center logon screen appears.

2. Enter username `admin` and password `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 once you have generated accounts for all your users.

3. Click **Submit**.

The Mercury IT Governance Center standard interface opens.

For More Information

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

Configuring or Reconfiguring the Database

The settings described in this section are meant to be starting values only—you need to monitor the database and analyze performance data to fine-tune the settings for your system environment. Tuning an Oracle database should involve an Oracle database administrator.

The recommendations in this section assume that Mercury IT Governance Center is the only application using the database instance. You need to adjust recommended parameter values if other applications are sharing the database.

Database Parameters

This section defines and describes the key Oracle database parameters that can affect overall Mercury IT Governance Center system performance. Recommended parameter settings are also provided for the Mercury IT Governance Center environment.

For More Information

For more detailed information about the Oracle parameters described in the following sections, refer to the Oracle database documentation.

DB_BLOCK_SIZE

Size (in bytes) of Oracle database blocks.

DB_BLOCK_SIZE cannot be changed after the database has been created.

For older installations upgrading to a newer version of Mercury IT Governance Center that use a smaller block size, make a full export of the database, re-create the database with the new block size, and import the data back into the database.

Recommended Setting

Set this parameter to 8 kilobytes.

DB_CACHE_SIZE

Size (in bytes) 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 (kilobyte or megabyte). Smaller values are automatically rounded up to the granule size.

A value of zero is not allowed, because zero is the size of the default pool for the standard block size, which is the block size for the system tablespace.

Recommended Setting

Start with 300 megabytes.

GLOBAL_NAMES

Whether or not a database link is required to have the same name as the database to which it connects.

Recommended Setting

Set to FALSE (if multiple Mercury IT Governance Center test instances use the same database instance, you must set GLOBAL_NAMES to FALSE).

If GLOBAL_NAMES is set to TRUE, loopback database link creation fails. 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 One

```
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 Two

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

LOG_BUFFER

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 this parameter based on the number of concurrent users:

- For systems with fewer than 50 concurrent users, set this parameter to 512 kilobytes (512000 bytes).
- For systems with more than 50 concurrent users, set this parameter to 1 megabyte (1000000 bytes).

(RAC Only) MAX_COMMIT_PROPAGATION_DELAY

Time delay (in milliseconds) after a change committed on one instance is propagated to other instances on the RAC (Real Application Clusters) system.

Recommended Setting

Set to 0.

OPEN_CURSORS

Number of cursors one session can hold open at a given time.

Oracle uses cursors to handle updates, inserts, deletes, and result sets returned by queries.

Recommended Setting

Set to 1000 or above.

OPEN_LINKS

Number of open database links connections to other databases that can be active at a given time.

Recommended Setting

Set to 20.

OPTIMIZER_MODE

Default behavior for choosing an optimization approach for executing a query.

Recommended Setting

For Oracle database versions Oracle 9i, set to CHOOSE.

For Oracle database version Oracle 10G or higher, set to ALL_ROWS (the default Oracle setting).

Database statistics gathering is required.

For More Information

For information about collecting database statistics, see [Collecting Statistics About the Database Schema](#) on page 179.

PGA_AGGREGATE_TARGET

Aggregate 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 like sort, group-by, hash-join, bitmap merge, and bitmap create.

Use this parameter with WORKAREA_SIZE_POLICY set to AUTO.

PGA_AGGREGATE_TARGET replaces the traditional SORT_AREA_SIZE.

Recommended Setting

Set this value based on the total amount of memory available for the Oracle instance. This value can then be tuned and dynamically modified at the instance level.

Recommended initial value for the parameter `PGA_AGGREGATE_TARGET` is $PGA_AGGREGATE_TARGET = (total_mem * 80\%) * 40\%$.

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

PROCESSES

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

Although concurrent usage and usage nature are factors used to determine the number of connections used, it is rare for a Mercury IT Governance Server to use 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.

Recommended Setting

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

Example One

For single-server configurations, set to 45 (default).

Example Two

For a Mercury IT Governance Server cluster configuration running three servers, set to $(3 \times 25) + 20 = 95$.

(Oracle 10G or Later) SGA_TARGET

Maximum size of all SGA components combined in the instance.

If you specify SGA_TARGET, you do not need to provide individual values for SGA components like SHARED_POOL_SIZE, JAVA_POOL_SIZE, LARGE_POOL_SIZE, and DB_CACHE_SIZE.

SHARED_POOL_RESERVED_SIZE

This parameter helps to make sure that a portion of the shared pool (set 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 due to shared pool fragmentation.

For an object to be put in the reserved area, it must be larger than the SHARED_POOL_RESERVED_MIN_ALLOC value. Mercury recommends using the default value for the SHARED_POOL_RESERVED_MIN_ALLOC parameter.

Recommended Setting

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

SHARED_POOL_SIZE

Size (in bytes) of the shared pool.

The shared pool contains shared cursors and stored procedures. Larger values can improve performance in multi-user systems, but they use more memory. Smaller values use less memory, but they may degrade performance of multi-user systems.

Recommended Setting

Start with 300 megabytes.

TIMED_STATISTICS

Specifies whether or not statistics related to time are collected.

Setting this parameter helps to make sure that information about the database and timing information about internal activities is readily available. The overhead of enabling this function is minimal, and the data obtained can be extremely helpful.

Recommended Setting

Set to TRUE.

WORKAREA_SIZE_POLICY

This parameter controls the mode in which working areas are tuned. Its value can be AUTO or MANUAL.

If the value is AUTO, work areas used by memory-intense operators are sized automatically based on the PGA memory used by the system and the target PGA memory set in PGA_AGGREGATE_TARGET.

If the value is MANUAL, work areas are set manually and based on the value of the *_AREA_SIZE parameter.

Recommended Setting

Set to AUTO.

Oracle Database Configuration Examples

The following sections contain configuration examples for Oracle 9i and Oracle 10G.

Oracle 9i Example

Table 6-2 lists example parameters for Oracle 9i.

Table 6-2. 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")

Table 6-2. Example parameters for Oracle 9i (page 2 of 3)

Category/Parameter	Value
Instance Identification	
instance_name	ardent
Job Queues	
job_queue_processes	10
MTS	
dispatchers	“(PROTOCOL=TCP) (SERVICE=ardentXDB)”
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 (Oracle does not recommend sort_area_size, use pga_aggregate_target instead.)	
#pga_aggregate_target	25165824
pga_aggregate_target	1500M

Table 6-2. Example parameters for Oracle 9i (page 3 of 3)

Category/Parameter	Value
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	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 6-3 lists example parameters for Oracle 9i.

Table 6-3. 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
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")

Table 6-3. Example parameters for Oracle 10G (page 2 of 3)

Category/Parameter	Value
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
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

Table 6-3. Example parameters for Oracle 10G (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	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



Note

This grant is normally given during Mercury IT Governance Center installation or upgrade.

For Mercury IT Governance Center to be able to keep track of the open database sessions it is using, check to be sure that a public grant exists on the v_\$\$session dynamic performance table. To do this, connect as SYS to the database containing the Mercury IT Governance Center database schema and issue the following SQL statement:

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

(Oracle Object Migration Only) Generating Database Links

Mercury IT Governance Center can make use of database links to communicate with other databases. Usually a database link is created and associated with a particular environment in Mercury IT Governance Center and can then be used in situations like AutoCompleteSQL.

Some examples of situations where database links are used include:

- Custom object types that are designed to provide parameter value lists directly from a source or destination database during Mercury Change Management activities
- Some Mercury Change Management Extensions, including the Extension for Oracle E-Business Suite and the Extension for PeopleSoft, to facilitate Change Management activities

Database links can be defined as the need arises. For each database link you need (which probably would also include a link to the Mercury IT Governance Center database), issue a 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

For more information about database links and their usage, see:

- *Mercury Change Management Extension for Oracle E-Business Suite Guide*
- *Mercury Change Management Extension for PeopleSoft Enterprise Guide*
- *Mercury Object Migrator Guide*
- *Mercury GL Migrator Guide*
- *Oracle SQL Language Reference Manual*

Configuring the Mercury IT Governance Workbench

This section explains how to configure the Java plug-in and launch the Mercury IT Governance Workbench.

You should consider restricting use of the Workbench to users who have a need to do the kind of configuration and administration tasks accomplished through that interface.

For More Information

For more information about the Mercury IT Governance Workbench, see the *Getting Started* document.

Configuring the Java Plug-in

The Java plug-in is required to access the Mercury IT Governance Workbench interface.

Setting the Correct Version of the Java Plug-In



Under normal circumstances, this is not required—the plug-in is downloaded directly from the Sun Web site.

When users access the Mercury IT Governance Workbench, the system checks for the correct version of the Java plug-in on their computer. If it is not correct, users are guided through a procedure for installing the Java plug-in. This procedure needs to be performed only once.

Table 6-4 lists the default settings for the server configuration parameters related to the Java plug-in.

Table 6-4. Server parameters related to the Java plug-in

Server Parameter Name	Default
JAVA_PLUGIN_PATH_IE	http://java.sun.com/products/plugin/autodl/jinstall-minimum_supported_version-windows-i586.cab
JAVA_PLUGIN_PATH_NS	http://java.sun.com/j2se/minimum_supported_version/
JAVA_PLUGIN_VERSION	<i>Minimum_supported_version</i>

For More Information

For the minimum supported version of the Java plug-in 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 6-1*, see *server.conf Parameters* on page 254.

Launching the Workbench



If a pop-up blocker has been installed in the Web browser, the Workbench may not open. You can configure the blocker to allow pop-ups from Mercury IT Governance Center.

To launch the Workbench (from the Mercury IT Governance Center standard interface) select **Administration > Open Workbench**.

Troubleshooting Default JVM Problems

Workbench users might run into the following kinds of problems resulting from the Java plug-in's setting itself as the default JVM for their browser:

- The Workbench might throw a “class not found” exception error.
- Users might be running other applications requiring different versions of the Java plug-in.

To resolve these issues, do not mark any installed Java plug-ins as the default.

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

1. Open the Windows control panel.
2. Open the Java plug-in icon.
3. Click the **About** tab.

A window appears that identifies the Java plug-in being used by Mercury IT Governance Center, along with any other Java plug-ins you may have installed.

4. Click the **Browser** tab and deselect the default browser association.

After this change has been applied, other applications should be able to use the version of the Java plug-in they require, and the Workbench should still function properly.

What to Do Next

If you intend to do any of the installations or configurations described in [Chapter 7, *Optional and Future Installations and Configurations*, on page 109](#), (for example, if you are going to install or upgrade a Mercury Change Management Extension) do them now.

If you are finished your installation or upgrade tasks for now, begin to test your installed or upgraded release 6.0 system. As you do that, be sure you understand the maintenance procedures you need to do periodically on your system. Those procedures are described in [Chapter 8, *Maintaining the System*, on page 149](#).

Optional and Future Installations and Configurations

In This Chapter:

- *Installing Mercury Best Practices*
 - *Installing/Upgrading Mercury Change Management Extensions*
 - *Installing Product Patches*
 - *Configuring the Workbench as a Java Application*
 - *Copying the jar Files*
 - *Creating the Batch File*
 - *Setting the Default Web Browser*
 - *Integrating with an LDAP Server*
 - *Configuring an External Web Server*
 - *Process Overview*
 - *Choosing an External Web Port*
 - *Configuring a Workers Property File*
 - *Configuring the External Web Server*
 - *Integrating the External Web Server with the Mercury IT Governance Server*
 - *Configuring a Server Cluster*
 - *Server Clustering Overview*
 - *Server Clustering Configuration Procedures*
 - *Starting and Stopping Servers in a Cluster*
 - *Validating the Cluster Configuration*
-

Installing Mercury Best Practices

Mercury IT Governance Center Best Practices provides customers with experience-derived information and advice about configuring and using Mercury Portfolio Management™, Mercury Program Management™, and Mercury Project Management.

Mercury Best Practices installs various entities (for example, workflows and request types) on your system.

If the product license your organization has purchased includes Mercury Best Practices, you can install it with the following procedure.

Before you begin the installation procedure:

- Install and configure Mercury IT Governance Center.
- Be sure your `license.conf` file has the correct keys for Mercury Best Practices. (License information for individual solutions appear on separate lines in the file.)
- Create the user name (if required) that is required for the installation of Best Practices. (You can access Best Practices with the admin user but not the user name you have created for the database administrator.)

To install Mercury Best Practices:

1. Set the Mercury IT Governance Server to restricted mode.
2. Start the Mercury IT Governance Server.
3. Run the `kDeploy.sh` script:

```
sh kDeploy.sh -best-practices
```
4. Stop the server and restart it in normal mode.

For More Information

For information about setting the server to restricted mode, and starting and stopping the server, see [Starting and Stopping the Mercury IT Governance Server on page 80](#).

For more information about `kDeploy.sh`, see [kDeploy.sh on page 283](#).

Installing/Upgrading Mercury Change Management Extensions

For specific instructions for installing or upgrading a Mercury Change Management Extension, see the product documentation for the Extension you have purchased.

For More Information

For more information about the documentation for the Mercury Change Management Extensions, see the *Guide to Documentation*.

Installing Product Patches

Mercury occasionally delivers product update patches to licensed customers of Mercury IT Governance Center.

The script used to apply these product patches is `kDeploy.sh`, a command-line tool. Product patches are distributed as deployments, which are software bundles containing files and data, and are in the format:

```
mitg-itg_server_version-deployment_id.jar
```

The variable `itg_server_version` is the Mercury IT Governance Server version number and `deployment_id` (a variable-length name) is a unique identifier for the deployment.

For example, to install product patch PL9:

1. Issue the following command:

```
cp mitg-600-PL9.jar itg-home
```

2. Stop the Mercury IT Governance Server.

3. Issue the following command:

```
sh kDeploy.sh -i PL9
```

Follow the instructions you receive as the script runs.

4. Start the Mercury IT Governance Server.

For More Information

For information about starting and stopping the server, see [Starting and Stopping the Mercury IT Governance Server](#) on page 80.

For more information about `kDeploy.sh`, see [kDeploy.sh](#) on page 283.

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.



When running the Workbench interface as an application, be sure that client files are deployed properly. If the Mercury IT Governance Server is upgraded or patched, the client files might also need to be patched.

Copying the jar Files

To run the Workbench interface as an application, copy the following `jar` files from the `ITG_Home/html/client` and `ITG_Home/classes` directories into a single directory accessible by the client machine:

- `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 the necessary jar files have been copied to a single directory accessible by the client machine, 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 `kintana.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
jview /p /cp %CLASSPATH% com.kintana.core.gui.LogonApplet
your_company.domain.com:1200
```

Note

This example uses the Microsoft JVM (`jview`). To use the Sun SDK, replace the `jview` command line with the following command:

```
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 `jview` command to reflect the host name and RMI port of the primary server.

Note

If you are using the Sun SDK, edit the `java` command to reflect the hostname and RMI port of the primary server.

4. If you have any Mercury Change 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 `kintana.bat` file that you created.

Creating 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 `kintana.sh` with the following content:

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

CLASSPATH=.
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 ITG
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 host name and RMI port of the primary server.
4. If you have any Mercury Change 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 `kintana.sh` script that you created.

Setting the Default Web Browser

When running the Workbench interface as an application, users must set the default browser setting in their user profile.

To set the default browser setting:

1. In Workbench interface, open the **Edit** menu.
2. Select **User Profiles**.
3. Click the **General** tab.
4. Select the default Web browser.

If access to a URL is required, the default Web browser is used.

Integrating with an LDAP Server

The Mercury IT Governance Server can be integrated with any LDAP v3-compliant server—for example, Microsoft Windows Active Directory.

Integrating with an LDAP server might help minimize the cost of setup and maintenance associated with user account management. With the addition of 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.

In an LDAP environment, the Mercury IT Governance Server does authentication 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.conf` parameters. If passwords are not supplied in the `server.conf` file, the Mercury IT Governance Server does 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=username`. The `uid` attribute might 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.

For More Information

For more information about server parameters related to LDAP integration, see [LdapAttribute.conf Parameters](#) on page 275.

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`
2. Install the server's certificate in the JRE's database of trusted certificates.
3. Be sure the parameters in the `LdapAttribute.conf` file are set correctly.

For More Information

For more information about `server.conf` parameters, see [Table A-1 on page 255](#).

For more information about `LdapAttribute.conf` parameters, see [Table A-3 on page 275](#).

Configuring an External Web Server

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 [Single-Server/External Web Server Configuration on page 29](#) and [Server Cluster/External Web Server Configuration on page 32](#).

The following sections describe how to configure an external Web server to work with a Mercury IT Governance Center Server cluster.

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

Process Overview

As described in the following sections, you need to complete the following tasks to configure an external Web server:

1. Choose an external Web server (Sun Java System Web server, Sun ONE Web server, Microsoft IIS, or Apache)
2. Choose an EXTERNAL_WEB_PORT.
3. Configure a workers property file.
4. Configure the external Web server.
5. Integrate the external Web server with the Mercury IT Governance Server.
6. (Optional) Enable cookie logging on the external Web server.

Choosing an External Web Port

Choose the port through which the external Web server and the Mercury IT Governance Server will communicate. This port must be one not being currently used on the machine that is running Mercury IT Governance Center.

You will need to identify this port in the Mercury IT Governance Center `server.conf` file and your workers property file.

Configuring a Workers Property File

The workers property file stores information about the Mercury IT Governance Server. It contains information like the machine name, ports, and load balance. The external Web server will use this information to direct traffic to the Mercury IT Governance Center applications, as appropriate.

The following sections describe how to configure a workers property file for:

- Sun Java System Web Server or Sun ONE Web server (workers.properties)
- Microsoft IIS 5.0 and 6.0 (workers2.properties)
- Apache 1.3 and 2.0 (workers2.properties)

(Sun Web Servers Only) Configuring a workers.properties File

If your external Web server is Sun Java System Web server or Sun ONE Web server, you need to configure a workers.properties file.

A sample workers.properties file is shown below (this file is also located in the *ITG_Home/integration/webserver/conf* directory).

The definition as it exists below is for a single-server configuration. Information representing a clustered configuration is commented out. For more information about configuring a server cluster, see [Configuring a Server Cluster on page 137](#).

Sample File

```
# server2 (commented out in this file)
# represents the second server in a clustered configuration.

# Defines a load balancer to handle requests to the ITG
# server.
worker.list=load_balancer

# 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 be unique for each ITG instance, and added
# to the balanced_workers list below.
worker.server1.host=k1.acme.com
worker.server1.port=8009
worker.server1.type=ajp13
worker.server1.lbfactor=1

# 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

# Defines a load balancer.
worker.load_balancer.type=lb
worker.load_balancer.balanced_workers=server1
# worker.load_balancer.balanced_workers=server1,server2
# List all servers in the ITG cluster
# in the balanced_workers group.
```

Configuration Procedure

To configure a `workers.properties` file:

1. Using a text editor, open the sample `workers.properties` file located in the `ITG_Home/integration/webserver/conf` directory.
2. Set the `worker.list` parameter to `load_balancer`.
3. For the single server (or for each Mercury IT Governance Server in the server cluster), configure the following values:
 - a. Set the `worker.server#.host` parameter to the network address of the machine where the Mercury IT Governance Center instance is installed (you can use `localhost` if the Mercury IT Governance Center instance is located on the same computer as the Web server).
 - b. Set the `worker.server#.port` parameter to the `EXTERNAL_WEB_PORT` you have decided to use.
 - c. Set the `worker.server#.type` parameter to `ajp13`, which is the protocol used to connect to the remote server.
 - d. 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 server and 1 to the less-laden 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](#)).

(IIS or Apache Only) Configuring a workers2.properties File

The `workers2.properties` file is used for the JK2 module on IIS and Apache.



For IIS, the `workers2.properties` file can exist anywhere in the file system. For Apache, it must be in the `APACHE_HOME/conf` directory.

The version of `workers2.properties` in the `ITG_Home/integration/webserver/conf` directory should be used only for reference.

A sample `workers2.properties` file is shown below (this file is also located in the `ITG_Home/integration/webserver/conf` directory).

Sample File

```
# jk2.x configuration file. This file tells the external
# Web server how to connect to
# the Mercury IT Governance Servers.

# Temporary file created by the module. This file needs to be
# in a location that can be written to by the Web server.
[shm:]
file=${serverRoot}/logs/jk2.shm
size=1048576

# Defines the load balancer, which can be used
# for single or multi-server environments.
# In a multi-server environment, the module spreads load
# based on the load factor defined for each server below.
[lb:load_balancer]

# Defines a Mercury IT Governance Server instance.
# You can copy this block for each
# additional server in an ITG cluster. Be sure the port
# matches the port defined in the server.conf file as
# EXTERNAL_WEB_PORT. The "tomcatId" parameter must be set to
# a unique name that does not contain the period character to
# make sure Web sessions remain on the same server.
[channel.socket:localhost:8009]
port=8009
host=127.0.0.1
group=load_balancer
lb_factor=1
tomcatId=server1

# URLs matching http://host:port/itg/* will be forwarded to
# the load balancer.
[uri:/itg/*]
group=lb:load_balancer

# Optionally enables the JK status page. This provides
# some statistics and information on the JK plugin and load
# balancer.
```

```
[status:]  
  
# URLs http://host:port/jkstatus show JK status  
# information  
[uri:/jkstatus/*]  
group=status:
```

Configuration Procedure

To configure a `workers2.properties` file:

1. Using a text editor, open the sample `workers2.properties` file located in the `ITG_Home/integration/webserver/conf` directory.
2. Set the `file` parameter to the temporary shared memory file used by the JK2 module.

This file must be in a directory into which the Web server can write (Mercury recommends putting the file in the Web server's `log` directory).

It is typically unnecessary to change the `size` parameter.

3. Leave the `lb` parameter as shown:

```
[lb:load_balancer]
```

This parameter defines the load balancer that will be used in the cluster.

4. For the single server or each of the Mercury IT Governance Servers in the cluster, configure the following values:
 - a. Set the `channel.socket` parameter to the machine name and external Web port of the Mercury IT Governance Center instance as shown in the following example:

```
[channel.socket:k1.acme.com:8009]
```

- b. Set the `port` parameter to the external Web port you have decided to use.
- c. Set the `host` parameter to the network address of the network address of the machine where the Mercury IT Governance Center instance is installed (you can use `localhost` if the Mercury IT Governance Center instance is located on the same machine as the Web server).
- d. Set the `group` parameter to `load_balancer`, which defines the cluster to which this Mercury IT Governance Server belongs.

- e. Set the `lb_factor` parameter to the load balancing factor used to distribute load.

If all servers in a cluster can handle approximately the same amount of load, assign one to each server. If a server can handle twice as much load as another server, assign 2 to that server and 1 to the less-robust server.

- f. Set the `tomcatId` parameter to a unique name for that server, as shown in the following example:

```
tomcatId=server1
```

This parameter ties the user session to the appropriate server in the cluster.

The value must be unique for each server and must not contain a period (.). Mercury recommends setting this name to the same value as the `KINTANA_SERVER_NAME` parameter.

5. Define the URL that this Web server should forward to the Mercury IT Governance Servers—for example:

```
[uri:/itg/*]
```

At a minimum, the `/itg/*` URL must be forwarded, but you can forward all accesses to this Web server by changing the block:

```
[uri:/*]
```

6. Set the `group` parameter to `lb:load_balancer`.

This parameter defines the cluster that will service the URL set in [step 5](#).

7. To allow old URLs to operate properly with notifications, add the following code:

```
[uri:/kintana/*]  
group=lb:load_balancer
```

Configuring the External Web Server

The following sections describe how to set up external Web servers supported by Mercury IT Governance Center (for a list of supported versions, see the *System Requirements and Compatibility Matrix* document):

- Sun Java System Web Server
- Microsoft IIS
- Apache

Configuring the Sun Java System or Sun ONE Web Server

To configure the Sun Java System Web server or Sun ONE Web server:

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

You will be configuring this server to run as the external Web server for the Mercury IT Governance Server.

A directory called `https-ITG` is created that contains two files called `magnus.conf` and `obj.conf`.

2. Stop the Mercury IT Governance Server.

For information about how to do that, see [Starting and Stopping the Mercury IT Governance Server on page 80](#).

3. Do one of the following:

- (If your Sun Java System Web Server is installed on the same machine as your Mercury IT Governance Server) Put the `workers.properties` file you configured in [\(Sun Web Servers Only\) Configuring a workers.properties File on page 119](#) in the `ITG_Home/conf` directory.
- (If your Sun Java System Web Server is *not* installed on the same machine as your Mercury IT Governance Server) Put the `workers.properties` file you configured in [\(Sun Web Servers Only\) Configuring a workers.properties File on page 119](#) in the `Sun_Home/https-webserver_name/config` directory.

4. Do one of the following:

- (Windows only) Copy the `nsapi_redirector.dll` 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.

- (UNIX only) 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 `magnus.conf` (even though the text might wrap, each `init fn=` must be a full line):

```
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` right at the beginning of the “Object” section (that is, after `<Object name=default>`):

```
NameTrans fn="assign-name" from="/itg/*" name="itg-servlet"
```

7. Add the following after the end of the “Object” section (that is, after `</Object>`):

```
<Object name="itg-servlet">  
ObjectType fn=force-type type=text/html  
Service fn="jk_service" worker="load_balancer" path="/itg/*"  
</Object>
```

The `itg-servlet` strings should match.

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

To enable cookie logging:

1. Stop the Sun Java System Web Server.
2. In the `magnus.conf` file, find the line that initializes flex. This line begins:

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

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

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

The line should now look like this:

```
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 IIS 5.0 Web Server

To configure the Microsoft IIS Web server on Windows:

1. Create a virtual directory called `jakarta` pointing to the IIS scripts directory.

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

2. Copy `isapi_redirector2.dll` from `ITG_Home/integration/webserver/iis/win32` to the IIS scripts directory (usually `c:\inetpub\scripts`)

If `isapi_redirector2.dll` is being upgraded from a previous Mercury IT Governance Center instance, first stop the IIS server using `net stop w3svc`.

3. Be sure you have created a `workers2.properties` file, as described in ([IIS or Apache Only](#)) *Configuring a workers2.properties File* on page 121.

4. Configure IIS to load `isapi_redirector2.dll` as a filter:

- a. Define registry values for IIS with JK2.

- i. Add the following new registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software  
Foundation\Jakarta Isapi Redirector\2.0
```

- ii. Add a string value `serverRoot` and the `ITG_Home` directory as a value (for example, `serverRoot = c:\ITG`).

- iii. Add a string value `extensionUri` and a value `/jakarta/isapi_redirector2.dll`.

- iv. Add a string value `workersFile` and the *complete path* to the `workers2.properties` file as a value (for example, `c:\inetpub\scripts\workers2.properties`)

- v. Add a string value `logLevel` and a value `ERROR` (other options could include `DEBUG` and `INFO` for more verbose logging).

Logging information will be saved into the Windows event log.

- b. Open the Microsoft management console.



Perform these actions must be performed at the server level, *not* at the Web site level.

- c. Right-click the server name.

A pop-up menu opens.

- d. Select **Properties** from the menu.

The Properties window opens.

- e. From the Master Properties drop-down list, select **WWW Service**.

- f. Click **Edit**.

The Master Properties window opens.

- g. Click **Add**.

The Filter Properties window opens.

- h. In the Executable field, set the path to **isapi_redirector2.dll**.

In the Filter Name field, name the filter **jakarta**.

- i. In the `Web Service Extensions` folder of the configuration screen, set **All Unknown ISAPI Extensions** to **Enabled**.

- j. Click **OK**.

The Filter Properties window closes.

5. Using the control panel, restart IIS.

You can optionally restart IIS by running `net stop w3svc` and then `net start w3svc`.

6. Start the Mercury IT Governance Server(s).

(Optional) Enabling Cookie Logging on IIS 5.0

To enable cookie logging:

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 property sheet, select **Cookies**.
6. Click **Apply**.

Configuring the Microsoft IIS 6.0 Web Server

To configure the Microsoft IIS Web server on Windows:

1. Complete all the steps ([step 1](#) through [step 6](#)) in the procedure in [Configuring the Microsoft IIS 5.0 Web Server on page 127](#).
2. In addition, complete the following two steps:
 - a. Put IIS into IIS 5.0 isolation mode.

In the Windows management console:

- Go to **Web Site Properties > Service > Isolation Mode**.
- Select **Run WWW service in IIS 5.0 isolation mode**.

- b. Allow Tomcat's redirector DLL in Web service extensions.

In the Windows management console:

- Click **Web Services Extensions**.
- Select **Add a new Web service extension**.
- Type the extension name (for example, Jakarta-Tomcat).
- Select **Set extension status to Allowed**.
- Click **Add**.
- Type the path to `isapi_redirector2.dll`.
- Click **OK**.

(Optional) Enabling Cookie Logging on IIS 6.0

To enable cookie logging:

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 property sheet, select **Cookies**.
6. Click **Apply**.

Configuring the Apache Web Server

The following sections describe how to:

- Compile a binary of JK2 (which you need to only if a precompiled binary is not available in the `ITG_Home/integration/webserver` directory)
- Configure Apache 1.3
- Configure Apache 2.0

(Only If Required) Compiling a Binary of JK2

Configuring Apache on UNIX requires `mod_jk2.so`, a dynamically linkable JK2 module. In most cases, precompiled binaries of JK2 already exist in the `ITG_Home/integration/webserver` directory. Before attempting to compile the JK2 module, first check this directory and verify if the binaries already exist (binaries for several operating systems are available).

If a precompiled binary is not available, then complete the following procedure. Otherwise, proceed to the appropriate section for instructions on configuring your version of Apache (either [Configuring Apache 1.3 on page 133](#) or [Configuring Apache 2.0 on page 134](#)).

To compile a binary of JK2:

1. Unpack the following source code bundle:

```
ITG_Home/integration/webserver/src/jk2.0.2-src.tar.gz
```

2. Change to the following directory:

```
cd jarkarta-tomcat-connectors-jk-2.0.2-src/jk/native2
```

3. Run the following shell script:

```
sh buildconf.sh
```

4. Run the configuration script:

For Apache 1.3:

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

For Apache 2.0:

```
./configure --with-apxs2=/path_to_apache_bin/apxs
```

The configuration script generates the `make` files for the current machine environment. The `make` files are required for running the `make` command, as described in [step 5](#).

5. 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:

```
make
```

Configuring Apache 1.3

To configure the Apache 1.3 module:

1. Copy the Apache module to the Apache `libexec` directory:

- To copy a precompiled module:

```
cp ../ITG_Home/integration/webserver/path_to_JK2/mod_
jk2.so path_to_apache_libexec
```

- To copy a self-compiled module:

```
cp ../build/jk2/apache13/mod_jk2.so path_to_apache_
libexec
```

2. Navigate to the Apache `conf` directory. Edit the `httpd.conf` file:

```
LoadModule jk2_module libexec/mod_jk2.so
AddModule mod_jk2.so
```

3. Be sure you have created a `workers2.properties` file, as described in [\(IIS or Apache Only\) Configuring a workers2.properties File on page 121](#).

The file must be in `APACHE_DIR/conf`.

4. For the changes to take effect, restart the Apache server.

(Optional) Enabling Cookie Logging on Apache 1.3

To enable cookie logging:

1. Open the log file.
2. Add the following string after `%b`:

```
%{Cookie}i
```

The log format and custom log lines should now look like this:

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

Configuring Apache 2.0

To configure the Apache 2.0 module:

1. Copy the Apache module to the Apache module directory:

- To copy a precompiled module:

```
cp ../ITG_Home/integration/webserver/path_to_JK2/mod_
jk2.so path_to_apache_module
```

- To copy a self-compiled module:

```
cp ../build/jk2/apache2/mod_jk2.so path_to_apache_modules
```

2. Navigate to the Apache `conf` directory. Edit the `httpd.conf` file:

```
LoadModule jk2_module modules/mod_jk2.so
AddModule mod_jk2.so
```

3. Be sure you have created a `workers2.properties` file, as described in [\(IIS or Apache Only\) Configuring a `workers2.properties` File on page 121](#).

The file must be in `APACHE_DIR/conf`.

4. For the changes to take effect, restart the Apache server.

(Optional) Enabling Cookie Logging on Apache 1.3

To enable cookie logging:

1. Open the log file.
2. Add the following string after `%b`:

```
%{Cookie}i
```

The log format and custom log lines should now look like this:

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

Integrating the External Web Server with the Mercury IT Governance Server

To integrate the external Web server with the Mercury IT Governance Server, you need to complete the following steps, which are described in the following sections:

1. Stop the IT Governance Server, which is described in [Starting and Stopping the Mercury IT Governance Server](#) on page 80.
2. Set the `server.conf` parameters, which is described in the following section.
3. Validate the integration, which is described in the following section, [Validating the Integration](#).

Setting the server.conf Parameters

To set the `server.conf` parameters:

1. Back up `ITG_Home/server.conf`.
2. Add `com.kintana.core.server.EXTERNAL_WEB_PORT` and set it to the port number in the workers property file.
3. Change `BASE_URL` to the base URL of the external Web server.
4. Run the `kUpdateHtml.sh` script.

For More Information

For more information about `BASE_URL`, see [Appendix A: Server Configuration Parameters](#) on page 251.

For more information about `kUpdateHtml.sh`, see [kUpdateHtml.sh](#) on page 287.

Validating the Integration

To validate 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/web/knta/global/Home.jsp`.

The page should load correctly.



Note

You must use the complete path. `BASE_URL/itg` will not work.

For More Information

For information about starting the Mercury IT Governance Server, see [Starting and Stopping the Mercury IT Governance Server on page 80](#).

For information about supported browsers, see the *System Requirements and Compatibility Matrix* document.

Configuring a Server Cluster

This section contains the following information and tasks related to server clustering in the Mercury IT Governance Center environment:

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

Server Clustering Overview

Before you begin to set up a Mercury IT Governance Server cluster, you should review the information in [Chapter 2, System Overview, on page 21](#), particularly [Server Cluster Configurations on page 31](#).

In addition, the following clustering concepts are important to understand in configuring server clusters.

Server Nodes

Nodes are the individual Mercury IT Governance Servers that comprise the clustered server.

KINTANA_SERVER_NAME and the ITG_Home/server directory

A Mercury IT Governance Server is made up of the common code that is contained 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.

Each node in a cluster needs 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 definition of `KINTANA_SERVER_NAME` in `server.conf`, and the values must be the same.



Server directories cannot contain spaces, commas, or other non-alphanumeric characters, except hyphen (-) or underscore (_).

For example, `server1_1` is an acceptable name, but `server 1,1` is not.

@node Directives in server.conf

The `@node` directive in `server.conf` (that is, `@node` alone on a line) tells the Mercury IT Governance Server that the variables after `@node` are specific to only one node of the cluster.

You need to specify one `@node` directive for each server in your cluster.

Variables that appear above the first `@node` are common to all servers.



Note

A common practice in single-server environments is to append new `server.conf` parameters to the bottom of the file. Doing this in a clustered environment will apply the variable to only the last node defined in the file.

Be sure to 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 7-1 shows which `server.conf` variables you must define per node, based on the type of clustering you are using. For more information about these variables, see [server.conf Parameters on page 254](#).

Table 7-1. Server parameters affected by clustering

Parameter	External Web Server, Single Machine	External Web Server, Multiple Machines	Hardware Load Balancer, Multiple Machines
<code>com.kintana.core.server.KINTANA_SERVER_NAME</code>	X	X	
<code>com.kintana.core.server.BASE_PATH</code>		X	X
<code>com.kintana.core.server.ORACLE_HOME</code>		X	X
<code>com.kintana.core.server.BASE_URL</code>	X	X	X
<code>com.kintana.core.server.BASE_LOG_DIR</code>		X	
<code>com.kintana.core.server.SERVER_LOG_DIR</code>		X	

Table 7-1. Server parameters affected by clustering [continued]

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

Process Overview

Here are the general steps you need to follow to configure a server cluster (for more detailed instructions, see the following sections):

1. (External Web server only) Set up your IT Governance Server to be integrated with an external Web server in single-server mode.
2. Stop the Mercury IT Governance Server.
3. (External Web server only) Stop the external Web server.
4. (External Web server only) Configure the workers property file (see [Configuring a Workers Property File on page 119](#)) to include information for the multiple cluster nodes. Each node will need an external Web port defined (using the `EXTERNAL_WEB_PORT` configuration parameter).
5. Configure the server nodes on the file system.
6. Configure the server nodes in `server.conf`.

Server Clustering Configuration Procedures

This section contains three procedures, which match the configuration types in [Table 7-1 on page 138](#):

- 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 do that, see [Starting and Stopping the Mercury IT Governance Server on page 80](#).

2. Stop the external Web server.
3. Add the new node and relevant information to the workers property file.

Example for a `workers.properties` file (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
```

Example for a `workers2.properties` file (IIS and Apache):

```
# node1, already defined when integrating with
# the external Web server
[channel.socket:machine1:8009]
port=8009
```

```
host=machine1
group=load_balancer
lb_factor=1
tomcatId=server1

# node2, added as part of a cluster
[channel.socket:machine1:8010]
port=8010
host=machine1
group=load_balancer
lb_factor=1
tomcatId=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, here 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 property 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=21600
```

6. From `ITG_Home/bin`, run `kUpdateHtml.sh` to propagate the changes to all the servers in the cluster.
7. (If you have additional nodes in your cluster) Repeat [step 1](#) through [step 6](#).
8. (Windows only) Start the Mercury IT Governance Server using the Windows service called “Mercury ITG `server_name`”, where `server_name`

is the value of the `KINTANA_SERVER_NAME` for the node in the cluster.

You need to generate a new service for the new node:

- a. From `ITG_Home/bin`, run `kConfig.sh`.

This starts the configuration wizard.

- b. Select **Configure Windows Services**.

The wizard guides you through the steps to create the service.

9. Validate the cluster using the procedure in [Validating the Cluster Configuration on page 147](#).

External Web Server, Multiple Machines

In a multiple-machine cluster, an `ITG_Home` directory must exist 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 to be integrated with an external Web server.

For information about how to do that, see [Configuring an External Web Server on page 118](#).

2. Stop the Mercury IT Governance Server.

For information about how to do that, see [Starting and Stopping the Mercury IT Governance Server on page 80](#).

3. Stop the external Web server.

4. Make sure that the common directories used by each server (`ITG_Home/logs`, `ITG_Home/transfers`, and `ITG_Home/transfers`) are shared.



Permissions for the shared directories must be set so that they are readable and writable by users of each of the cluster machines.

5. Add the new node and relevant information to the workers property file.

Example for a `workers.properties` file (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
```

Example for a `workers2.properties` file (IIS and Apache):

```
# node1, already defined when integrating with
# the external Web server
[channel.socket:machine1:8009]
port=8009
host=machine1
group=load_balancer
lb_factor=1
tomcatId=server1

# node2, added as part of a cluster
[channel.socket:machine2:8010]
port=8010
host=machine2
group=load_balancer
lb_factor=1
tomcatId=server2
```

6. Configure `server.conf` to include the new node.

For a multiple-machine clustered environment, here is a typical `server.conf` excerpt:

```
@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 property 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. Once the first server is configured 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. Once the file is copied, cd to `ITG_Home/server` on the new machine and rename the `node1` directory to `node2`.

The server name must match the `KINTANA_SERVER_NAME` value.

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 propagate the `server.conf` changes.
12. (Windows only) Start the Mercury IT Governance Server using the Windows service. In a multiple-machine configuration, you need to generate the services on all machines.

You need to generate a new service for the new node:

- a. From `ITG_Home/bin`, run `kConfig.sh`.

This starts the configuration wizard.

b. Select **Configure Windows Services**.

The wizard guides you through the steps to create the service.



The keys in the security directory are needed to read encrypted values in `server.conf` and the database. The same keys must be present on all nodes of a multimachine cluster.

Hardware Load Balancer, Multiple Machines



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

A hardware load balancer can be used 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 need to listen for HTTP requests on a unique port. Each server in the cluster must have its `HTTP_PORT` parameter set to a value that is unique for each server and does not conflict with other external applications. This parameter must be defined in the `@node` section of the `server.conf` file for all servers in a cluster.

Starting and Stopping Servers in a Cluster

Any server in a Mercury IT Governance Server cluster can be stopped and the Mercury IT Governance Server cluster can continue to operate and be accessible, as long as at least one server in the cluster is still running. If a server becomes unavailable, the Mercury IT Governance Web server module detects that the server is unavailable and stops dispatching HTTP requests to it. When the server becomes available again, the Mercury IT Governance Web server module again detects the available server and begins to dispatch requests to this server again.

Starting and stopping the primary server is identical to the process used in a single-server configuration. This procedure is described in [Starting and Stopping the Mercury IT Governance Server on page 80](#).

To start a secondary server, use the `-name server-name` argument in the `kStart.sh` script. To stop a secondary server, use the `-name server-name=KINTANA_SERVER_NAME=server/server-name` argument in the `kStop.sh` script.

On Windows, there is one service per server (called “Mercury ITG *server-name*”).

Even when a script for starting or stopping all servers in a cluster is unavailable, you can write an appropriate script for a custom cluster. For example, the following script for the UNIX environment will start 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 will stop all three servers in a cluster configuration:

```
#!/bin/sh
./kStop.sh -name serv1
./kStop.sh -name serv2
./kStop.sh -name serv3
```



Note

When making changes 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 `server.conf` on all machines

Validating the Cluster Configuration

To validate the 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 be sure the values in the workers property file are correct. Since you have already validated the external Web server configuration, the problem is likely here.

2. Start one of the servers and try to connect to it.

If you cannot connect to it, the problems is likely in `server.conf`. Check it and correct any errors you find.

3. Start the remaining servers in the cluster.

4. Once the servers are started, use the `kStatus.sh` script to confirm that all nodes are running.

All the servers should be listed as running. If a node is not running, check the server log files in `ITG_Home/server/KINTANA_SERVER_NAME/log` for errors.

For 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)
```


Chapter 8 Maintaining the System

In This Chapter:

- *Overview of Administration Tools and System Maintenance*
 - *Administration Tools in the Standard Interface*
 - *Accessing the Administration Tools*
 - *Viewing and Cancelling Running Reports*
 - *Viewing Running Executions*
 - *Viewing Interrupted Executions*
 - *Administration Tools in the Workbench Interface*
 - *The Server Tools Windows*
 - *Using Admin Tools*
 - *Using SQL Runner*
 - *Using the Server Settings Window*
 - *Getting Information from Log Files*
 - *Server Log Files*
 - *Report Log Files*
 - *Execution Log Files*
 - *Execution Debug Log Files*
 - *Temporary Log Files*
 - *Periodically Stopping and Restarting the Server*
 - *Maintaining the Database*
 - *Changing the Database Schema Passwords*
 - *Maintaining Temporary Tables*
 - *Backing Up Mercury IT Governance Center Instances*
-

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 allow you to:

- View and cancel running reports
- View running executions
- View interrupted executions

- Administration tools accessible from the Workbench interface

These tools include:

- Admin Tools, which allows you to submit and view server reports
- SQL Runner, which allows you to submit SQL statements against the Mercury IT Governance Center database

These tools and facilities are discussed in the following sections.

This chapter also discusses:

- How to access and use log files
- Procedures for:
 - Periodically stopping and restarting the server
 - Maintaining the database
 - Backing up Mercury IT Governance Center instances

Administration Tools in the Standard Interface

The Mercury IT Governance Center standard interface includes tools to:

- View and cancel running reports
- View running executions
- View interrupted executions

Accessing the Administration Tools

You access the administration tools in the standard interface through the **Administration** menu, as shown in *Figure 8-1*.

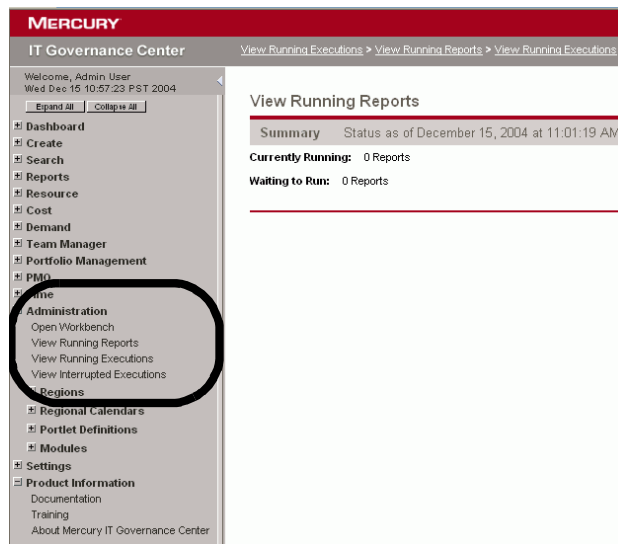


Figure 8-1. Standard Interface: Administration menu

Viewing and Cancelling Running Reports

To view and cancel running reports, select **Administration > View Running Reports** (see *Figure 8-1*).

For More Information

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

Viewing Running Executions

To view running executions:

1. Select **Administration > View Running Executions** (see [Figure 8-1](#)).

The View Running Executions page opens, as shown in [Figure 8-2](#).

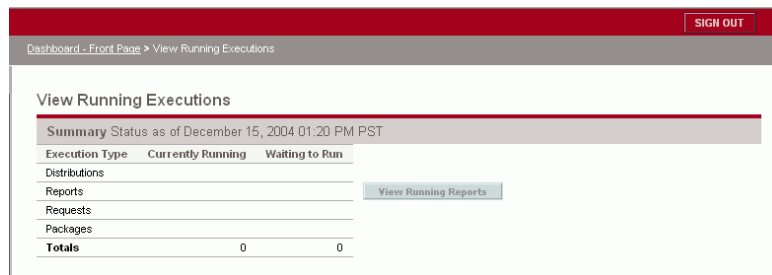


Figure 8-2. View Running Executions page

2. Any running distributions, reports, requests, or packages are displayed in the Summary section.
3. To view running reports, click the **View Running Reports** button.

Viewing Interrupted Executions

This section describes the procedure for viewing interrupted executions (including reports).

To view interrupted executions:

1. From the standard interface menu bar, select **Administration > View Interrupted Executions** (see [Figure 8-1](#))

The View Interrupted Executions page opens, as shown in [Figure 8-3](#).



Figure 8-3. View Interrupted Executions page

2. In the drop-down list below View Interrupted Executions for a Server Startup, select the date of the interrupted execution you want to see and click **View**.

3. Details of the selected interrupted execution are shown in the Failed Executions section.

Administration Tools in the Workbench Interface

From the Server Tools windows in the Mercury IT Governance Workbench interface, you can:

- View the technical status of the Mercury IT Governance Server by running server reports (Admin Tools window)
- Access the database directly and run SQL statements (SQL Runner window)

In addition, you can edit server settings in the Server Settings window.

The Server Tools Windows

Figure 8-4 shows the Server Tools windows.

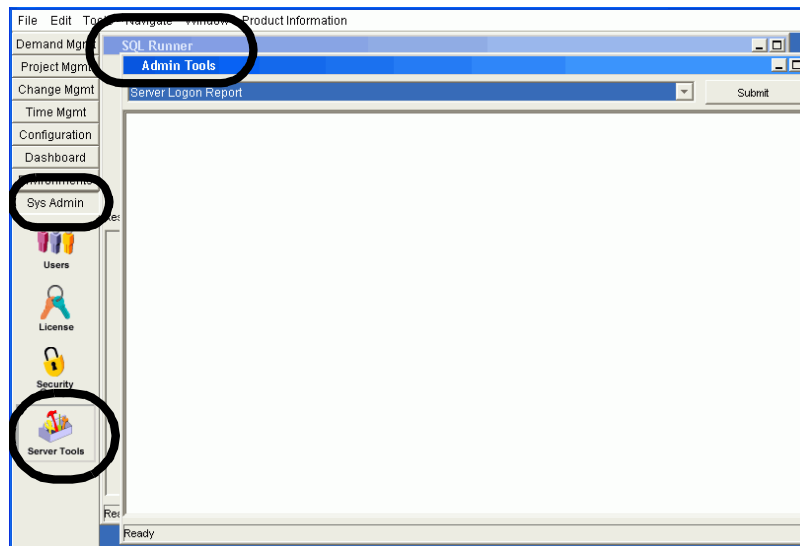


Figure 8-4. Server Tools window

Accessing the Server Tools Windows

You access the Server Tools windows in the Workbench interface by clicking **Sys Admin > Server Tools**, as shown in *Figure 8-4*.

Access Grants Required to Use Server Tools

Access to the Server Tools windows requires one of the three access grants listed and described in [Table 8-1](#).

Table 8-1. Server Tools access grants

Access Grant	Description
Sys Admin: View Server Tools	Allows the user to view the Admin Tools and SQL Runner windows in read-only mode.
Sys Admin: Server Tools: Execute Admin Tools	Allows the user to: <ul style="list-style-type: none">• Execute server reports in the Admin Tools window• View the SQL Runner window in read-only mode
Sys Admin: Server Tools: Execute SQL Runner	Allows the user to: <ul style="list-style-type: none">• Execute 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 *Security Model Guide and Reference*.

Using Admin Tools

Use the Admin Tools window to run server reports that contain information like server status and user details.

Figure 8-5 shows the Admin Tools window, with part of the list of available reports displayed. These reports are described in *Table 8-2*.

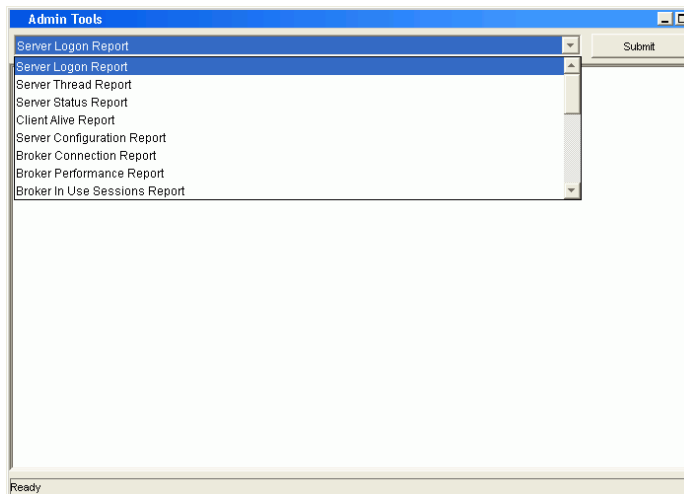


Figure 8-5. Admin Tools window

Running Server Reports Using Admin Tools

To run server reports using Admin Tools:

1. Select the report to run.
2. Click **Submit**.

The report is shown in the result box of the Admin Tools window.

A sample Server Status report is shown in *Figure 8-6*.

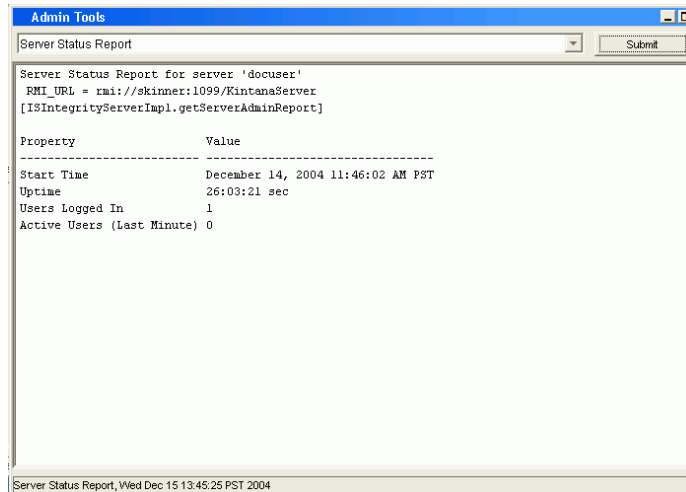


Figure 8-6. Sample Server Status report

Table 8-2. Server reports

Report	Description
Server Logon	Users logged on to the Mercury IT Governance Server(s) and their logon information (for example, IP address and idle time). This information is useful in determining Mercury IT Governance Server load. When server clustering is used, this report provides a picture of load distribution.
Server Thread	Information about running threads within a Mercury IT Governance Server. This information is useful in determining which services are running. When server clustering is used, this report also provides information on which server is running these services.
Server Status	Status information about Mercury IT Governance Server(s): <ul style="list-style-type: none"> • 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
Client Alive	Users with active sessions. For Workbench interface users, this report also shows when the client sent the last Keep Alive signal.
Server Configuration	All server parameters in effect for each of the active servers. Includes parameters not specifically set in the <code>server.conf</code> file.

Table 8-2. *Server reports [continued]*

Report	Description
Broker Connection	Information about database pool connections in use, organized by the connection ID.
Broker In Use Sessions	Information about database pool connections in use, organized by user. If the server parameter <code>DB_SESSION_TRACKING</code> is set to true, this report also shows stack traces of where the connection was allocated.
Broker Performance	<p>Statistics on how many database pool connections are held open and how many are in use.</p> <p>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.</p> <p>If the Mercury IT Governance Server cannot allocate more connections, threads that need to access the database might need to wait for a connection.</p> <p>This report also shows:</p> <ul style="list-style-type: none"> • 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)
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.
Kintana RMI Detail	All RMI connection threads.
Service Controller	Enabled services for the Mercury IT Governance Server(s), when services were last run, and when they are scheduled to run again.
Client Property	Client environment details.
Client Font	All supported fonts for the Mercury IT Governance Center installation.
Client Timezone	All time zones recognized by the client.

Table 8-2. Server reports [continued]

Report	Description
Server Cache Status	Shows the following cache information: <ul style="list-style-type: none"> • Cached entities • Number of units that can be cached • Number of units that are free • Miss rate, hits, and misses • Number of entities swapped • Amount of memory taken up by the cache
Server Event Listener	Events that the Mercury IT Governance Server can send to the client.
CacheManager Statistics	Cache manager statistics.
CacheManager Sizes	Cache manager size information.
JVM Memory	Free and total memory in JVM.

Running Server Reports Using `kRunServerAdminReport.sh`

You can also run server reports using the `kRunServerAdminReport.sh` script, which is located in the `ITG_Home/bin` directory.

For More Information

For more information about the `kRunServerAdminReport.sh` script, see [kRunServerAdminReport.sh](#) on page 286.

Using SQL Runner

You can use the SQL Runner window to run database queries using the Workbench interface instead of using an external program like SQL*Plus. The major benefit of using SQL Runner is being able to access the database directly without the database password. This could be important when you want to run performance tests without being exposed to sensitive data contained with the database. Additionally, developers and administrators can use the SQL Runner window to develop and test custom validations.

The SQL Runner Window

Figure 8-7 shows the SQL Runner window. The sample results shown in *Figure 8-7* are for **Ping DB**, one of the buttons in the SQL Runner window.

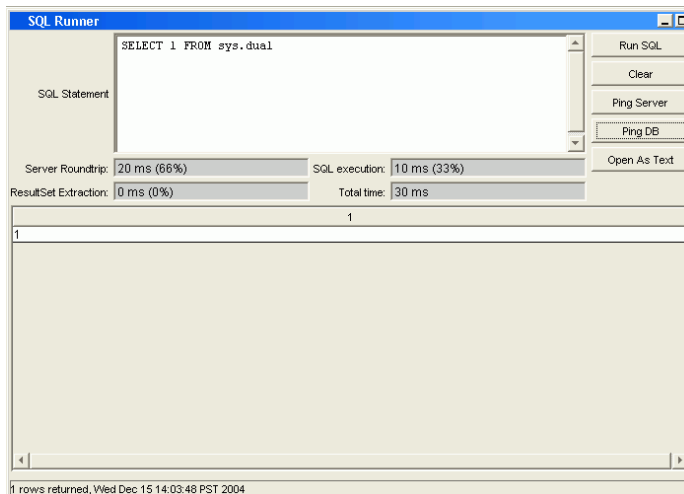


Figure 8-7. SQL Runner window

Table 8-3 lists the fields and buttons used in the SQL Runner window.

Table 8-3. SQL Runner window fields and buttons

Field/Button		Description
Name	Type	
SQL Statement	Text area	Allows the user to enter an SQL query for running and testing purposes.
Server Roundtrip	Results return (read-only)	Duration (in milliseconds) it takes to send the SQL statement out to the network and back. Used to show network latency and performance.

Table 8-3. SQL Runner window fields and buttons [continued]

Field/Button		Description
Name	Type	
SQL execution	Results return (read-only)	Duration (in milliseconds) it takes the database to execute the SQL statement. Used to tune validations or write complex statements that might help address performance concerns. Can also be used for database performance testing.
ResultSet Extraction	Results return (read-only)	Duration (in milliseconds) it takes to extract the data from the database.
Total time	Results return (read-only)	Duration (in milliseconds) it takes for SQL Runner to complete the operation.
Run SQL	Button	Runs the SQL statement entered in the SQL Statement field.
Clear	Button	Clears the window.
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 then cut and paste information from this window.

Running SQL Statements

To run an SQL statement using the SQL Runner window:

1. In the SQL Statement field, type an SQL statement.
2. To run the SQL statement, click **Run SQL**.

The execution results are shown in the result box in the SQL Runner window.

3. To see the results as text, click the **Open As Text** button.

The sample text results for **Ping DB** are shown in [Figure 8-8](#).

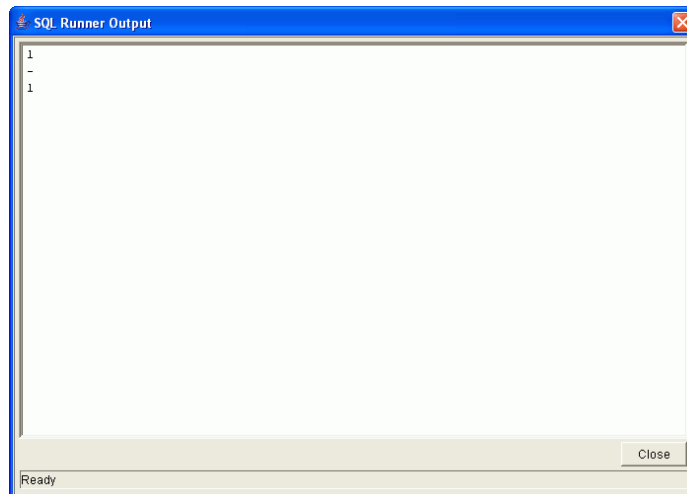


Figure 8-8. Sample text results for Ping DB

Using the Server Settings Window

The Server Settings window, shown in *Figure 8-9*, enables you to set debugging and tracing parameters at both the user and server levels.

You access the Server Settings window from the Workbench through the **Edit > Server Settings** menu.

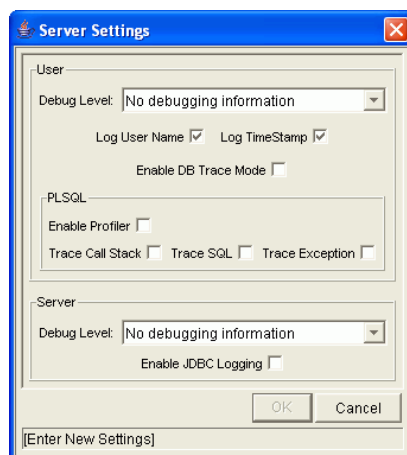


Figure 8-9. Server Settings window

User Settings

The following sections discuss the debug level and PL/SQL settings.

Debug Level Settings

The settings in this section are equivalent to the values for the `DEFAULT_USER_LOGGING_LEVEL` parameter in the `server.conf` file.

However, the values here are different. They map as follows:

- None maps to ERROR in `DEFAULT_USER_LOGGING_LEVEL`
- Normal maps to INFO
- Max maps to DEBUG

For More Information

For more information about the `DEFAULT_USER_LOGGING_LEVEL` parameter in `server.conf`, see [server.conf Parameters on page 254](#).

PL/SQL Settings

Enable Profiler

The Enable Profiler setting enables you to profile the run-time behavior of the PL/SQL code used in Mercury IT Governance Center applications by calling the Oracle-supplied PL/SQL package `DBMS_PROFILER`, which you need to set up.

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

Using this setting makes it easier to identify performance bottlenecks that can then be investigated more closely.



Note

Because running the `DBMS_PROFILER` package might have a negative impact on system performance and storage space, use it only in debugging situations.

The following example shows how you might 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

These settings enable you to use functionality from the Oracle DBMS_TRACE package to be used in PL/SQL programs used in Mercury IT Governance Center applications.

The output of the profiling information is logged in a JDBC log file in the IT Governance Center log directory.



Note

Because running the DBMS_TRACE package might have a negative impact on system performance and storage space, use it only in debugging situations.

Server Settings

The settings in this section are equivalent to the values for the DEFAULT_SERVER_LOGGING_LEVEL parameter in the `server.conf` file.

However, the values here are different. They map as follows:

- None maps to ERROR in DEFAULT_SERVER_LOGGING_LEVEL
- Normal maps to INFO
- Max maps to DEBUG

For More Information

For more information about the `DEFAULT_SERVER_LOGGING_LEVEL` parameter in `server.conf`, see [server.conf Parameters on page 254](#).

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 provides maintenance recommendations for each type of log file.

Server Log Files

Server log files are stored in the `ITG_Home/server/kintana/log` 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 that will 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 settings:

```
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 levels to **LOW** or **HIGH**.

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 > Server 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 **HIGH**. This can generate very large log files in the file system that could degrade system performance.

Enabling HTTP Logging



Note

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

To enable HTTP logging:

1. Stop the IT Governance Server.
2. Set the **ENABLE_WEB_ACCESS_LOG** `server.conf` parameter to **TRUE**.
3. Execute 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
```

For example:

```
127.0.0.1 - - [11/Dec/2004:19:08:16 +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 More Information

For information about starting and stopping the server, see [Starting and Stopping the Mercury IT Governance Server](#) on page 80.

Maintaining the Database

This section covers common topics related to maintaining the Oracle database that is part of Mercury IT Governance Center.

Changing the Database Schema Passwords

When you change the Mercury IT Governance Center database schema passwords, you need to change them both in the database and in the `server.conf` file. Before changing all the database schema passwords, consider the following:

- You need to check all environments:
 - Check both server and client passwords, along with database passwords.
 - Check passwords associated with application codes.
- Passwords might be hard-coded into commands in workflow steps, requests, and object types.
- Commands that use tokens for passwords (that is, [SOURCE_ENV.DB_PASSWORD]) should not need changing, as long as the password has been changed in the environment.

To change the Mercury IT Governance Center database schema passwords:

1. Be sure all users are logged off the system.
2. Stop the Mercury IT Governance Server.

For information about stopping Mercury IT Governance Servers, see [Stopping the Mercury IT Governance Server on page 195](#).

3. Change the passwords you want to change in the database.
4. To change the passwords in the `server.conf` file, run the `kConfig.sh` script to set the `DB_PASSWORD`, `CONC_REQUEST_PASSWORD`, and `RML_PASSWORD` server parameters.



When changing the passwords, do not edit the `server.conf` file directly. In order to properly encrypt password values, 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 80](#).

Maintaining Temporary Tables

The Mercury IT Governance Server uses a number of tables as temporary storage during processing (for example, during package migration) for:

- Logon attempts
- Debug messages
- Commands and parameters

A set of services employed by the Mercury IT Governance Server monitors and cleans up these temporary tables.

Be sure the cleanup parameters (described in [Cleanup Parameters on page 187](#) and also in [Appendix A: Server Configuration Parameters on page 251](#)) are set properly so that the temporary tables will not use too much database space.

The KNTA_LOGON_ATTEMPTS Table

This table contains the last 14 days of logon attempts to the Mercury IT Governance Server. Information stored in this table includes:

- USER_ID of users attempting to log on
- Success or failure of the each logon
- Any messages that were generated during logon

The KNTA_LOGON_ATTEMPTS table is maintained only for auditing purposes—the data is not required by the Mercury IT Governance Server.

The KNTA_DEBUG_MESSAGES Table

This table contains output information for a Mercury IT Governance Center debug session.

Cleanup Parameters

Use the HOURS_TO_KEEP_MESSAGE_ROWS server parameter to clean up the old records.

Use the DAYS_TO_KEEP_LOGON_ATTEMPT_ROWS server parameter to adjust the number of rows to keep.

Backing Up Mercury IT Governance Center Instances

A complete Mercury IT Governance Center instance backup procedure requires that you back up both the file system and the database schema.

With the exception of server log files, all the data generated and used by Mercury IT Governance Center is stored in the Oracle database. Because this information is so important, you should do a daily backup of the database schema. You can do the backup using the Oracle export command, or using the hot backup procedure (which does not require the Mercury IT Governance Server to be shut down). For information about the appropriate procedures to export database schema, see the Oracle database documentation.

In addition, you should also do daily backups of the *ITG_Home/logs* directory. This directory contains transactional history files for each migrated package or request.



Note

Before upgrading or making critical changes to Mercury IT Governance Center, do a full backup of the database schema and complete *ITG_Home* directory.

For More Information

For more information about backing up the file system and database schema, see [Backing Up the Existing Application on page 66](#).

Improving System Performance

In This Chapter:

- *Identifying Performance Problems*
 - *Isolating Performance Problems*
 - *Collecting Statistics About the Database Schema*
 - *Troubleshooting Performance Problems*
 - *Improving System Performance*
 - *Tuning JVM (Java Virtual Machine) Performance*
 - *Tuning Server Cluster Performance*
 - *Improving Input/Output Throughput*
 - *Improving Advanced Searches*
 - *Adjusting Server Configuration Parameters*
-

Identifying Performance Problems

The following sections discuss isolating performance problems, collecting statistics about the database schema, and troubleshooting performance problems.

Isolating Performance Problems

Configuring or Reconfiguring the Database on page 90 and *Appendix A: Server Configuration Parameters* on page 251 discuss the Mercury-recommended initial settings for the Oracle database and Mercury IT Governance Server. If Mercury IT Governance Center exhibits slower-than-desired performance after these settings are in place, use the methodologies outlined in the flowcharts shown in *Figure 9-1* on page 175, *Figure 9-2* on page 176, *Figure 9-3* on page 177, and *Figure 9-4* on page 178 to isolate performance problems and plan the actions required to fix them.

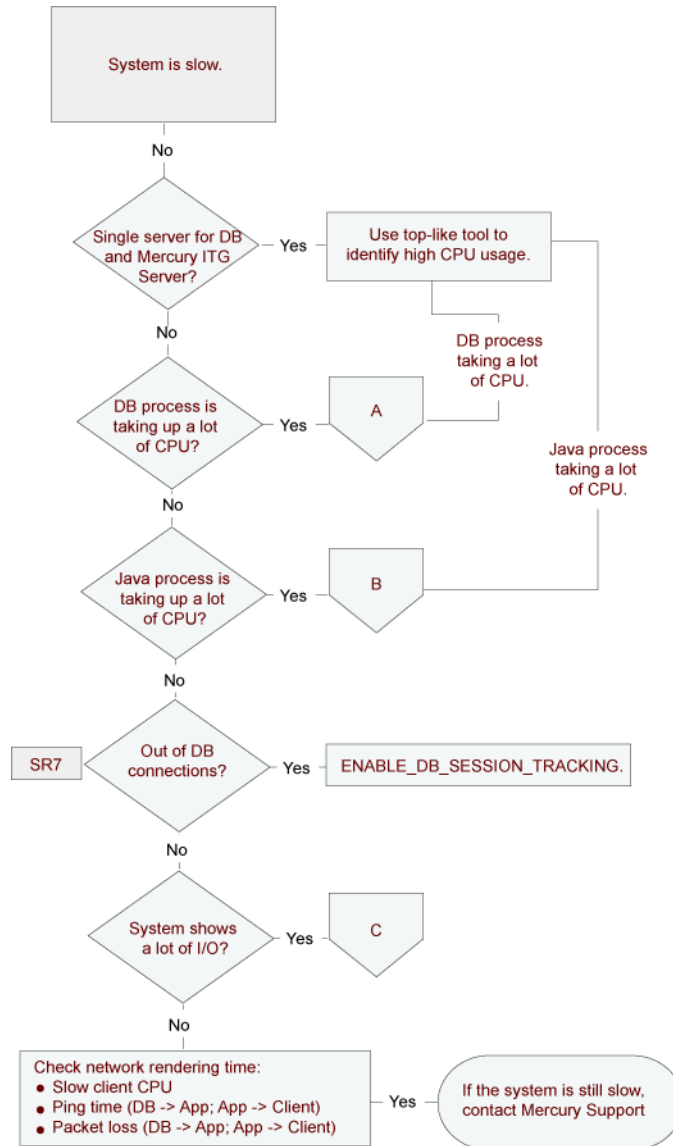


Figure 9-1. Identifying and addressing system performance problems

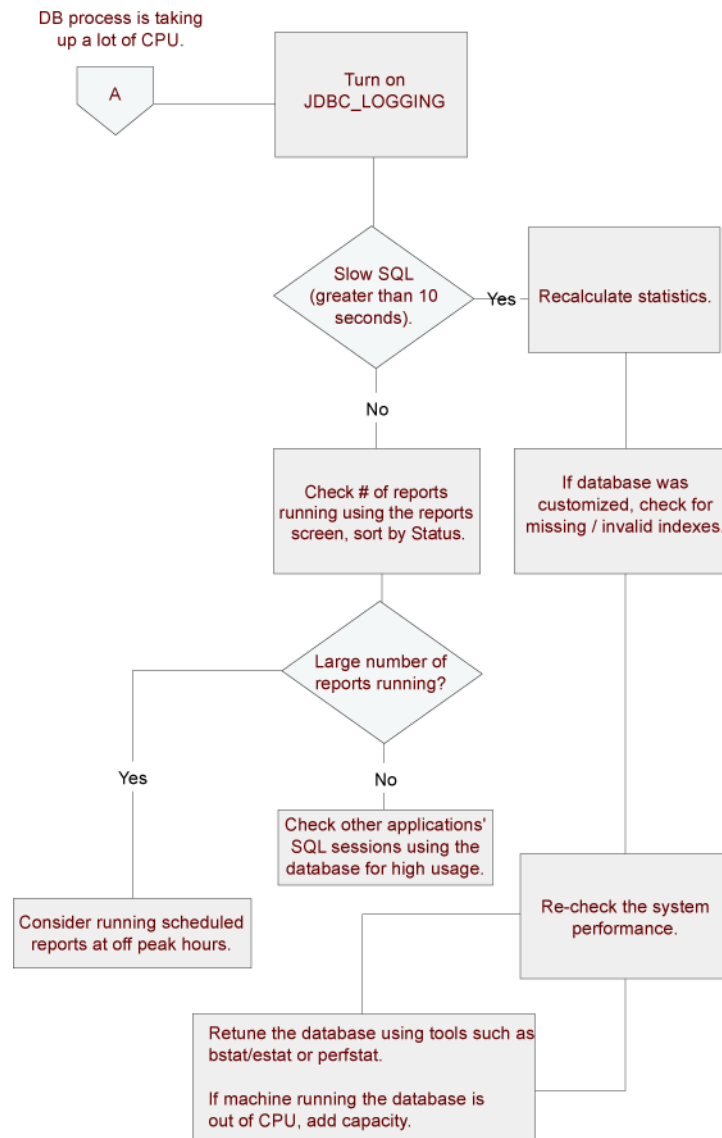


Figure 9-2. Identifying and addressing database performance problems (A)

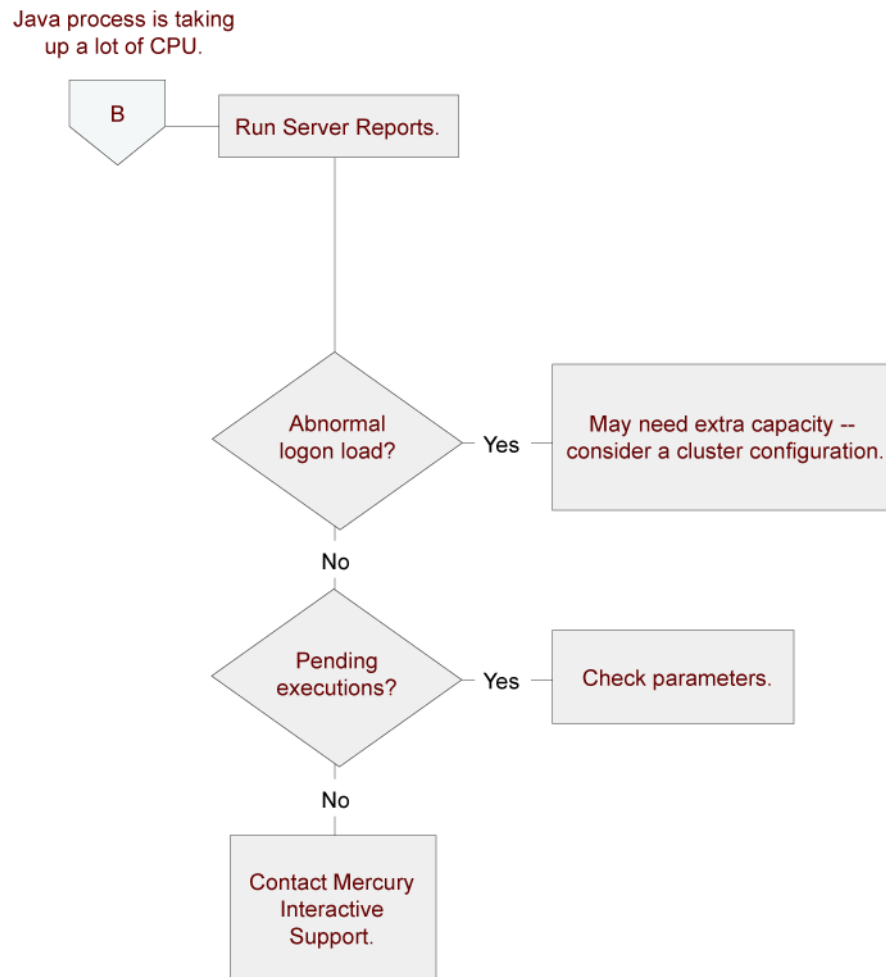


Figure 9-3. Identifying and addressing Java process performance problems (B)

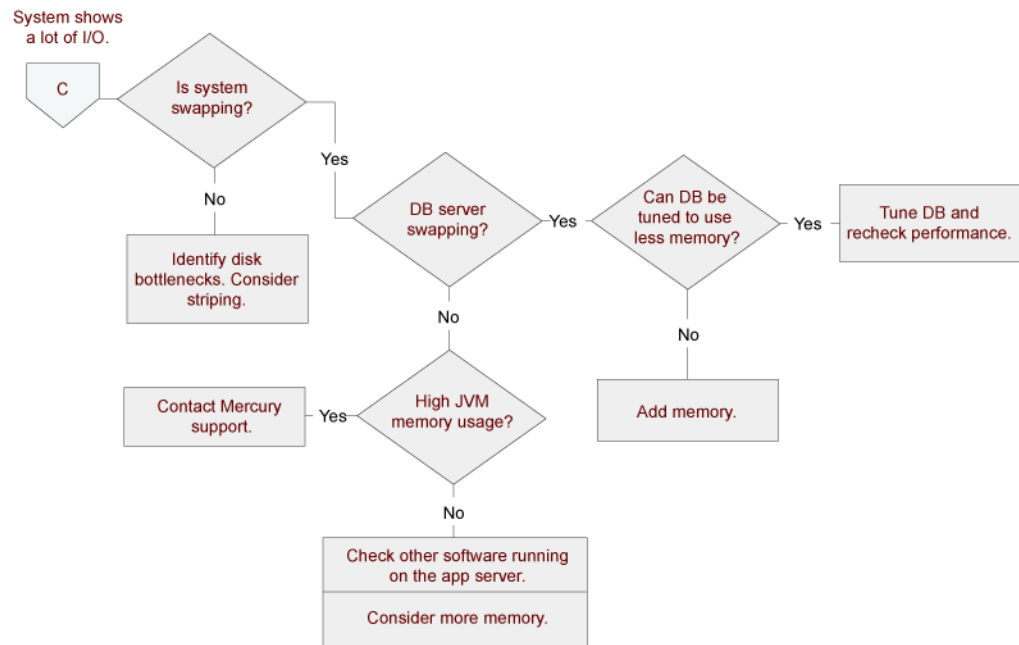


Figure 9-4. Identifying and addressing I/O performance problems (C)

Collecting Statistics About the Database Schema

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

Setting the Database to Gather Statistics

Database statistics gathering is required by Mercury IT Governance Server (see [OPTIMIZER_MODE on page 93](#) for information about the Oracle database parameter that enables statistics gathering). These statistics provide information on the numbers of rows in tables, data distribution, and occurrence frequency of values.

Collecting Additional Statistics by Setting Server Parameters

You should collect additional statistics when performing the following operations:

- 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 DMO or PMO
- Importing large projects (or a large number of projects) using Microsoft Project

You can set a Mercury IT Governance Center service to periodically collect this kind of data about the Mercury IT Governance Center database schema. The following Mercury IT Governance Server parameters are related to collecting database statistics:

- ENABLE_STATISTICS_CALCULATION
- STATS_CALC_INTERVAL
- STATS_CALC_WAKE_UP_TIME
- STATS_CALC_DAY_OF_WEEK
- STATS_CALC_WEEK_INTERVAL

For More Information

For a complete list of and descriptions for Mercury IT Governance Server parameters, see [Server Configuration Parameters](#) on page 251.

Collecting Additional Statistics Using Scripts

If statistics gathered using the Mercury IT Governance Center service are not sufficient, you can gather additional statistics in one of two ways:

- Using the `dbms_stats` package, which is provided by Oracle as part of the database
- Using the `kBuildStats.sh` script, which is located in the `ITG_Home/bin` directory

Running `dbms_stats`

To run this package:

```
begin
dbms_stats.gather_schema_stats (ownname => 'ITG_User',
cascade => TRUE,
method_opt => 'FOR ALL COLUMNS SIZE 1'
);
end;
/
```

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

```
"grant execute on dbms_stats to ITG_User;"
```

Running `kBuildStats.sh`

You can also gather statistics by running the `kBuildStats.sh` script located in the `ITG_Home/bin` directory. The `kBuildStats.sh` script runs the same commands as the `dbms_stats` package discussed in the following section.

Sampling a Percentage of Data

For large Mercury IT Governance Center installations, running this script may take a long time. In this case, you can choose to sample a percentage of data in each object instead of data from the entire Mercury IT Governance Center database schema.

Using a percentage to sample data may not be effective when data sets are small. Once the data set has grown, this method is almost as effective as calculating statistics for the entire database schema.

To calculate using a percentage, run the following script:

```
begin
dbms_stats.gather_schema_stats (ownname => 'ITG_User',
cascade => TRUE,
method_opt => 'FOR ALL COLUMNS SIZE 1',
estimate_percent => percentage_to_sample
);
end;
/
```

Troubleshooting Performance Problems

This section describes some common performance problems and how to correct them. As a general rule, if you are not using the default or recommended settings, reset your parameters first to those values before trying other solutions to performance problems.

Scheduled Reports Do Not Run at the Scheduled Time

Although the Mercury IT Governance Server has available capacity, the next scheduled tasks do not start.

This may be caused by a limitation specified in the MAX_WORKER_THREADS server parameter.

To run a larger number of scheduled reports simultaneously, set this parameter to a higher value.

For More Information

For more information about MAX_WORKER_THREADS, see [server.conf Parameters on page 254](#).

Packages Do Not Execute

If packages do not execute, it is possible that not enough execution managers are available to service the packages being processed by the system.

Try altering the `MAX_EXECUTION_MANAGERS` server configuration parameter.

For More Information

For more information, see [server.conf Parameters on page 254](#).

Nightly Reports on Sunday Do Not Finish On Time, System Slows on Monday

By default, database server statistics are collected on Sunday at 1:00 a.m. For large installations, the job may take a long time to run.

To avoid a problem on Monday morning, move the statistics collection time to a more appropriate time 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.

Determine CPU utilization. If the system is slow due to high peak load, you might require additional or faster hardware.

For More Information

For more information about gathering statistics, see [Collecting Statistics About the Database Schema on page 179](#).

Improving System Performance

The following sections describe possible ways to improve system performance.

Tuning JVM (Java Virtual Machine) Performance

Since 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

For performance reasons, the JVM uses a JIT (Just-In-Time) compiler. For debugging purposes, the JIT compiler can be disabled to run the JVM in interpreted mode. Exceptions encountered when running in interpreted mode contain line numbers that are helpful for debugging problems.

To run the JVM in interpreted mode, set a variable in the environment the server is started from 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 Problems

Several parameters in the Mercury IT Governance Server startup script (`kStart.sh`) can be used for debugging purposes. The `kStart.sh` JVM debugging parameters are `-ms550m` and `-mx550m`, which indicate that the JVM should start up with a heap size of 550 megabytes, and is limited to a maximum heap size of 550 megabytes.

Under normal circumstances these settings are sufficient. In sites with heavy usage, however, this value may need to be larger. Required memory depends on factors like cache sizes and number of Oracle connections.



Note

When the Mercury IT Governance Server is first started after installation or upgrade, the server occupies approximately 600 megabytes in memory. As the product is used, the cache fills up and the JSPs are loaded into memory. This results in a gradual increase in the memory used by the system. This effect is normal, and the memory usage levels out over time. Memory usage normally increases to a maximum of 800 megabytes.

Tuning Server Cluster Performance

High transaction volumes and high numbers of concurrent users on a Mercury IT Governance Server can cause a degradation of server response time. If the Mercury IT Governance Server is running on a multi-processor system, it may be possible that spare CPU is available, but JVM limitations can prevent the spare CPU capacity from being used.

In this case, you should 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 allows increased 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 More Information

For information about setting up cluster configurations, see [Configuring a Server Cluster](#) on page 137.

Improving Input/Output Throughput

An important database performance consideration is the distribution of input and output across multiple disks. 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.

The Mercury IT Governance Center database can be split into the following discrete pieces:

- 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 9-1](#).

Table 9-1. 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	<ul style="list-style-type: none"> • Rollback tablespaces • Temporary tablespaces • System tablespace • Tablespace for management and related utilities

For Mercury IT Governance Center database instances with even higher transaction volume (more than 10,000 requests per month), all the pieces may

be required to be on separate disks. The data and index tablespaces may also need to be striped across multiple disks to provide adequate disk throughput.

For Mercury IT Governance Center database instances with extremely high transaction volume (over 25,000 requests per month), move specific tables and indexes to separate tablespaces on separate disks. This will provide better control and further increase 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. Advanced searches allow users to locate requests based on information that is defined to be critical to their business processes.

As the number of requests logged in the system increases, advanced searches might experience slower performance. To improve performance on advanced searches:

- 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.

Be careful not to add too many indexes, since this can affect performance of inserts and updates to the database.

Adjusting Server Configuration Parameters

This section provides a categorized list of the Mercury IT Governance Server parameters related to system performance. In addition, it provides some usage considerations for the parameters. The 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. Other file names are noted.

For More Information

For a complete list of and descriptions for Mercury IT Governance Server parameters, see [Server Configuration Parameters on page 251](#). The following lists identify parameter categories.

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`
- `DAYS_TO_KEEP_LOGON_ATTEMPT_ROWS`
- `ENABLE_INTERFACE_CLEANUP`
- `HOURS_TO_KEEP_DEBUG_MESSAGE_ROWS`
- `NOTIFICATIONS_CLEANUP_PERIOD`

If periodic slowdowns occur, check these parameters in conjunction with the Service Controller report for correlation between the times when cleanup services run and the slowdowns occur. Change these parameters, if necessary, to avoid running cleanup services during peak periods.

For More Information

For information about the Service Controller report, see [Table 8-2 on page 157](#).

For more information about the cleanup parameters, see [server.conf Parameters on page 254](#).

Debug Parameters

Debug parameters control the debug and log output from the Mercury IT Governance Server. There are two subcategories of debug parameters:

- **High-level debug parameters:**

High-level debug parameters can be changed without causing system downtime on the Mercury IT Governance Server. Workbench interface

users with the appropriate privileges can configure these parameters by selecting **Edit > Settings** from the Workbench interface.

The high-level debug parameters (which are defined in the `server.conf` file unless otherwise indicated) are:

- `DEFAULT_USER_DEBUG_LEVEL` (`logging.conf` file)
- `ENABLE_JDBC_LOGGING`
- `ENABLE_SQL_TRACE`
- `SERVER_DEBUG_LEVEL` (`logging.conf` file)
- **Low-level debug parameters:**

Enable the low-level debug parameters only if you require debugging information for a specific area. Enabling these parameters may degrade system performance, because they consume additional CPU and generate large log files.



Mercury recommends that you consult Mercury Support before enabling these 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`

Timeout Parameters

Timeout parameters determine the duration that the Mercury IT Governance Server waits before timing out. Timeout values can be set for logon sessions, commands being run, and workflows.

The timeout parameters (which are all defined in the `server.conf` file) are:

- CLIENT_TIMEOUT
- DB_LOGIN_TIMEOUT
- DEFAULT_COMMAND_TIMEOUT
- PORTLET_EXEC_TIMEOUT
- SEARCH_TIMEOUT
- WEB_SESSION_TIMEOUT
- WORKBENCH_SESSION_TIMEOUT

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:

- AUTOCOMPLETE_STATUS_REFRESH_RATE
- EMAIL_NOTIFICATION_CHECK_INTERVAL
- ENABLE_EXCEPTION_ENGINE
- ENABLE_PENDING_PROJECT_CHANGE
- EXCEPTION_ENGINE_INTERVAL
- EXCEPTION_ENGINE_WAKE_UP_CHECK_FREQUENCY
- EXCEPTION_ENGINE_WAKE_UP_TIME
- MAX_EXECUTION_MANAGERS
- MAX_RELEASE_EXECUTION_MANAGERS
- MAX_WORKER_THREADS
- PENDING_PROJECT_CHANGE_INTERVAL
- REPORTING_STATUS_REFRESH_RATE

- SCHEDULER_INTERVAL
- THREAD_POOL_MAX_THREADS
- THREAD_POOL_MIN_THREADS
- TURN_ON_IF_TIMEOUT_REAPER
- TURN_ON_NOTIFICATIONS
- TURN_ON_SCHEDULER
- WF_SCHEDULED_TASK_INTERVAL
- WF_SCHEDULED_TASK_PRIORITY
- WF_TIMEOUT_REAPER_INTERVAL

Database Connection Parameters

Database connection parameters relate to the management of the database connection pool maintained by the Mercury IT Governance Server. When the Mercury IT Governance Server starts, one database connection is established. As usage increases, additional database connections spawn.

These parameters (which are all defined in the `server.conf` file) are:

- MAX_DB_CONNECTION_IDLE_TIME
- MAX_DB_CONNECTION_LIFE_TIME
- MAX_DB_CONNECTIONS
- MAX_STATEMENT_CACHE_SIZE

Logging Parameters

The logging parameters are in the `logging.conf` file.

For more information, see [logging.conf Parameters](#) on page 272.

Chapter 10 Migrating Instances

In This Chapter:

- *Overview of Instance Migration*
 - *Copying an Existing Instance to Create a Second One*
 - *Running the Installation Script Twice to Create Two Instances*
 - *(Optional) Migrating a Document Management Module*
 - *Preparing for Migration*
 - *Obtaining a New License Key*
 - *Stopping the Mercury IT Governance Server*
 - *Migrating the Mercury IT Governance Server*
 - *Migrating to a Windows Machine*
 - *Migrating to a UNIX Machine*
 - *Post-Migration Activities*
 - *Migrating the Database Schema*
 - *Troubleshooting Instance Migrations*
 - *The Mercury IT Governance Server Does Not Start*
 - *The Server Starts Running, but Applications Cannot be Accessed*
 - *Exp Command Variables*
 - *Imp Command Variables*
-

Overview of Instance Migration

This section provides instructions for migrating Mercury IT Governance Center instances from one environment to another.

A Mercury IT Governance Center instance consists of both the Mercury IT Governance Server and the Mercury IT Governance Center database schema. The Mercury IT Governance Server contains the file system that stores configuration information, log files, server reports, attachments, and license keys. The Mercury IT Governance Center database schema consists of two layers:

- A data layer containing data related to entities such as requests, packages, and projects
- A reporting metalayer containing views for reporting purposes

Migrating a Mercury IT Governance Center instance might include either the Mercury IT Governance Server or the Mercury IT Governance Center database schema, or both. You might migrate a Mercury IT Governance Center instance to:

- Clone the entire Mercury IT Governance Center instance by copying an existing instance and migrating the copy to another location

Cloning requires a new license key if the new location is a different machine.

- Migrate the Mercury IT Governance Server to a new machine but maintain the existing database schema

Migrating the server requires a new license key.

- Migrate the database schema but maintain the existing Mercury IT Governance Server

Migrating only the database schema does not require a new license key.

The following sections cover the procedures necessary to migrate a Mercury IT Governance Center instance.

Enterprise environments usually have several instances (for example, development, test, and production). The following sections discuss the simplest multiple instance configuration, consisting of two instances:

- DEV (development instance)
- PROD (production instance)

Each instance is located on a different machine. These basic migration principles can then be extended to support the number of instances used at your site.

There are two implementation scenarios for using multiple instances. The process for migrating these instances differs depending on the following scenarios:

- If you currently have an instance already in use, you can copy that instance to create a new one.
- If you have not yet installed any instance, you need to create multiple Mercury IT Governance Center instances by running the installation script multiple times.

These two scenarios are described in the following sections.

Copying an Existing Instance to Create a Second One

To implement multiple instances when a single production (PROD) instance is currently in use, clone the PROD instance. Each Mercury IT Governance Center instance consists of a file system and an Oracle database. These can exist on Windows or UNIX machines.

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 like workflows, object types, request types, validations, security groups, and environments.

3. Configure a package workflow to migrate the configuration data from DEV to PROD.

This process should be implemented in the PROD instance.

4. Migrate data from the DEV instance into the PROD instance.

Again, this activity is performed from the PROD instance. Therefore, it may be helpful to think of migrating the data as an import process.

This procedure is shown in [Figure 10-1](#).

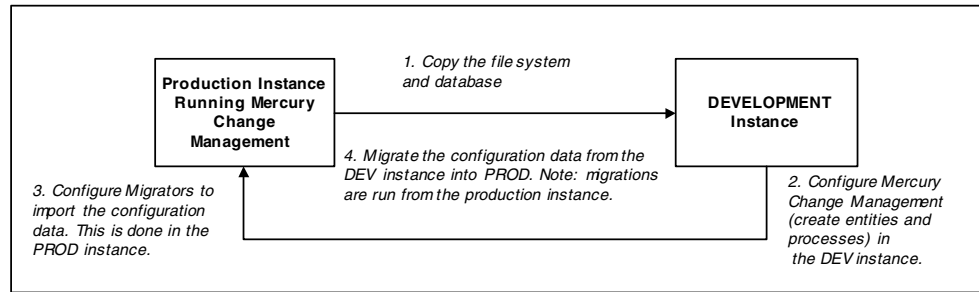


Figure 10-1. Moving from a single instance to multiple instances

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 existing data from one instance into another later on.

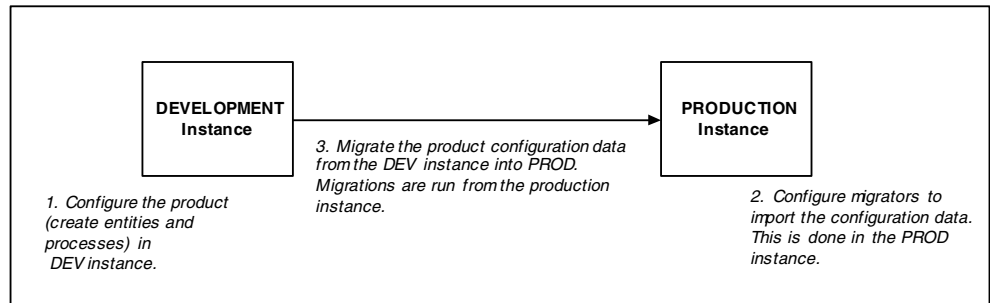


Figure 10-2. Migrating data between DEV and PROD

(Optional) Migrating a Document Management Module

If your source and target machines contain (and will contain) the Mercury document management module, see the *Document Management Guide and Reference* for instructions in how to migrate the document management module.

Preparing for Migration

Before you can begin to migrate, you need to obtain a new license key (if required) 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 running the Mercury IT Governance Server. If the Mercury IT Governance Server is migrated to a new machine, you must obtain a new license key for the target machine. (If you are migrating only the database schema, you do not need a new license key.)

To obtain a new license key, contact Mercury Support with the following information:

- Mercury IT Governance Center version
- Machine IP address
- Operating system (Windows or UNIX)
- Server purpose (development, test, or production)

Stopping the Mercury IT Governance Server

To help make sure that transactions, reports, and logs will not be lost, stop the Mercury IT Governance Server before migrating any part of the Mercury IT Governance Center instance.

For More Information

For instructions on how to stop the server, see [Stopping the Server on page 82](#).

Migrating the Mercury IT Governance Server

Before migrating the Mercury IT Governance Server, be sure that the target machine meets the requirements as documented in the *System Requirements and Compatibility Matrix* document.

Migrating to a Windows Machine

To migrate to a Windows machine:

1. Obtain a new license key, as described in [Obtaining a New License Key on page 195](#).
2. Stop the Mercury IT Governance Server (`kStop.sh`).
3. Migrate the Mercury IT Governance Center file system:
 - Make a `zip` file of the entire `ITG_Home` directory.
 - Copy this `zip` file to the target machine and unzip the file.
4. Migrate the Mercury IT Governance Center database schema.

For information about migrating the database schema, see [Migrating the Database Schema on page 201](#).

5. Reconfigure the Mercury IT Governance Server in the target location.
 - Run the `kConfig.sh` script located in the `ITG_Home/bin` directory.
 - The `kConfig.sh` script starts the server configuration utility. Values from the previous server configuration are displayed in the window for each server parameter.
 - Update the following parameters:
 - Any parameters that reference the DNS name or IP address of the old server
 - Server directories
 - ORACLE_HOME
 - BASE_PATH

- BASE_URL
 - RMI_URL
 - ATTACHMENT_DIRNAME
 - SERVER_NAME
6. Create a Windows service for the new Mercury IT Governance Center instance.
- Navigate to the `ITG_Home/bin` directory.
 - Run `kConfig.sh`:
 - Select `Configure Windows services`.
 - Select `Change service parameters and refresh the services`.
 - Enter a valid `JAVA_HOME`.
 - Select `Finish`.
7. Complete the post-migration activities, which are described in [Post-Migration Activities on page 200](#).
8. Start the new Mercury IT Governance Server:
- Using the Windows Control Panel, navigate to `Services`.
 - In the `Services` dialog box, select the Mercury IT Governance Center service name.
 - Click **Start**.

Migrating to a UNIX Machine

To migrate to a UNIX Machine:

1. Obtain a new license key, as described in [Obtaining a New License Key on page 195](#).
2. Stop the Mercury IT Governance Server.

For information about stopping the Mercury IT Governance Server, see [Starting and Stopping the Mercury IT Governance Server on page 80](#).

3. Migrate the Mercury IT Governance Center file system:

- On the machine where the Mercury IT Governance Server is running, navigate to the parent of the `ITG_Home` directory.
- 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 mitg60.tar ITG
```

- 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.
- Extract the archive file:

```
$ tar xf mitg60.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 migrating the database schema, see [Migrating the Database Schema on page 201](#).

5. Reconfigure the Mercury IT Governance Server in the target location.

- Run the `kConfig.sh` script located in the `ITG_Home/bin` directory.

The `kConfig.sh` script starts the server configuration utility. Values from the previous server configuration are displayed in the window for each server parameter.

- Update the following parameters:
 - Any parameters that reference the DNS name or IP address of the previous server machine
 - Server directories
 - ORACLE_HOME
 - BASE_PATH
 - BASE_URL
 - RMI_URL
 - ATTACHMENT_DIRNAME
 - SERVER_NAME
- 6. Complete post-migration activities, which are described in [Post-Migration Activities on page 200](#).
- 7. Put the new `license.conf` file in `ITG_Home/conf`.
- 8. Start the new Mercury IT Governance Server.
- 9. Run the `kStart.sh` script located in the `ITG_Home/bin` directory.

For more information about starting Mercury IT Governance Servers, see [Starting and Stopping the Mercury IT Governance Server on page 80](#).

Post-Migration Activities

Once the Mercury IT Governance Server has been migrated to the target machine, update the following server parameters to complete the migration process:

- ATTACHMENT_DIRNAME
- BASE_PATH
- BASE_URL
- ORACLE_HOME
- SERVER_TYPE_CODE
- RMI_URL
- SERVER_NAME

For more information about these server parameters, see [Appendix A: *Server Configuration Parameters* on page 251](#).

Be sure to:

- Stop the Mercury IT Governance Server prior to updating server parameters (`kStop.sh`).
- Use forward slashes when entering directory paths.
- When using a text editor to update the `server.conf` file, run the `kUpdateHtml.sh` script from the `ITG_Home/bin` directory before restarting the Mercury IT Governance Server.

Migrating the Database Schema

This section contains procedures for migrating the Mercury IT Governance Center database schema from one database to another.

If You Use the Extension for Oracle E-Business Suite

If you have Mercury Change 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, since 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. (See the *Mercury Object Migrator Guide* for more information.)

Except when 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 prior to migrating the Mercury IT Governance Center database. Contact Mercury Support for instructions on how to do this migration.

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

Regardless of the configuration, you should refresh the integration definition after migrating the Mercury IT Governance Center schema.

Migration Procedure

To migrate the database schema:



Exporting and importing the schema involves using the `exp` and `imp` commands, whose variables are described in [Exp Command Variables on page 207](#) and [Imp Command Variables on page 208](#).

1. Stop the Mercury IT Governance Server.

For information about how to do that, see [Starting and Stopping the Mercury IT Governance Server on page 80](#).

2. Export the Mercury IT Governance Center database schema into 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_
600.log
```

3. Create the new Mercury IT Governance Center database schema.

Run the following scripts (used to create the new database schema) from SQL*PLUS as the SYSTEM user (the file locations are indicated in parentheses):

- `CreateKintanaUser.sql` (installation zip file)
- `CreatorMLUser.sql` (installation zip file)
- `RMLSetupInITGSchema.sql` (*ITG_Home/install_600*)
- `RMLSetupInRMLSchema.sql` (*ITG_Home/install_600*)

Run the following script from SQL*PLUS as the SYS user (the file location is indicated in parentheses):

`GrantSysPrivs.sql` (installation or upgrade zip file)

Run the scripts from SQL*PLUS connected as the SYSTEM user, as shown in the following example:

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

Be sure you are creating and using these same tablespaces in the new databases.

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

In the target database, create a new empty RML database schema by running `CreateRMLUser.sql` from the downloaded Mercury IT Governance Center database scripts. Run the script from SQL*PLUS connected as the SYSTEM user, as shown in the following example:

```
SQL> @CreateRMLUser.sql Rml_User Rml_Password Rml_data_
tablespace Rml_temp_tablespace
```

5. Define any additional tablespaces required, and be sure they have adequate space. Mercury Change Management Extension for Oracle E-Business Suite is likely to use additional tablespaces.

6. Import the Mercury IT Governance Center database schema.

Import data from the export file that was previously created into the new empty Mercury IT Governance Center database schema by running the `imp` command, as shown 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_600.log
```

7. Create the RML-related packages in the RML schema:

```
sqlplus rml_user/rml_password@SID @rmlpackages
```

8. Grant privileges to the Mercury IT Governance Center RML database schema:

- 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
```

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



When Mercury IT Governance Center and Mercury Object Migrator are sharing 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.

You need to provide the necessary grants and permissions to the Mercury IT Governance Center user by running the `GrantSysPrivs.sql` script.

As the SYS user:

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

10. (If any database links are defined in the Mercury IT Governance Center database schema) Re-create the database links in the new Mercury IT Governance Center database schema.

To re-create the database links:

- 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_600/` 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.
11. (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. (Contact Mercury Support for assistance.)
 12. (If the Extension for Oracle E-Business Suite is in use) Refresh the Primary Object Migrator Host definition.
 13. Recompile invalid objects.

To validate any invalid Mercury IT Governance Center database objects generated when the links were regenerated, run the `RecompileInvalid.sql` script, which is located in the `ITG_Home/`

`install_600` directory. Run this script from SQL*PLUS connected as the new Mercury IT Governance Center database schema account.

14. Reconfigure the Mercury IT Governance Server to connect to the new database schema:

- Start the configuration utility by running the `kConfig.sh` script located in the `ITG_Home/bin` directory.
- Update the appropriate server configuration parameters, which are described in [server.conf Parameters on page 254](#).



If you edit the `server.conf` files manually, be sure to run `kUpdateHTML.sh` after you have completed the edit.

- Start the Mercury IT Governance Server, which is described in [Starting and Stopping the Mercury IT Governance Server on page 80](#).

Troubleshooting Instance Migrations

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

The Mercury IT Governance Server Does Not Start

Check the `serverLog.txt` file for error messages. The `serverLog.txt` file is located in the `ITG_Home/logs` directory.

Additionally, verify the parameters specified in the `server.conf` file.

The Server Starts Running, but Applications Cannot be Accessed

If the Web browser accessing the Mercury IT Governance Center URL generates a Not Found or Access Denied error, check the `server.conf` file and the external Web server (if one exists) to be sure that access to the directory where the Mercury IT Governance Server is installed is specified correctly.

If the Mercury IT Governance Server has recently been upgraded and the URL has changed, be sure that any saved links to the previous Mercury IT Governance Center URL (for example, existing requests) are updated to the new URL.

Exp Command Variables

Table 10-1 lists and describes the `exp` command variables.



Note

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

Table 10-1. Exp command variables

Variable	Description
<i>password</i>	Password of the SYSTEM user on the Oracle database.
<i>db</i>	Database connect string.
<i>Export_Filename</i>	Name of the file that will contain the export. The filename must use the <code>dmp</code> file extension (for example, <code>kntaExport.dmp</code>).
<i>ITG_Username</i>	Name of the Mercury IT Governance Center database schema being exported.

Imp Command Variables

Table 10-2 lists and describes the `imp` command variables.



Note

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

Table 10-2. Imp command variables

Variable	Definition
<i>Password</i>	Password for the SYSTEM user on the database.
<i>DB</i>	Database connect string.
<i>Export_Filename</i>	Name of the file that contains the export file. The filename must use the <code>dmp</code> file extension (for example, <code>kntaExport.dmp</code>).
<i>New_ITG_Username</i>	Name of the new Mercury IT Governance Center database schema.
<i>ITG_Username</i>	Name of the database schema that was previously exported.

Chapter 11 Migrating Entities

In This Chapter:

- *Overview of Entity Migration*
 - *Migration Process*
 - *Example Migration: Extracting an Object Type*
- *Defining Entity Migrators*
 - *Migrator Action Field*
 - *Basic Parameters*
 - *Import Flags*
 - *Password Fields*
 - *Internationalization Field*
- *Environment Considerations*
 - *Environment Connection Protocols*
 - *Environment Transfer Protocols*
 - *Setting the SERVER_ENV_NAME Parameter*
- *Security Considerations*
 - *Migration and Ownership*
 - *Migrations and Entity Restrictions*
- *Entity Migrators*
 - *Data Source Migrator*
 - *Module Migrator*
 - *Object Type Migrator*
 - *Overview Page Section Migrator*
 - *Portlet Migrator*
 - *Project Template Migrator*
 - *Report Type Migrator*
 - *Request Header Type Migrator*

- *Request Type Migrator*
 - *Special Command Migrator*
 - *User Data Context Migrator*
 - *Validation Migrator*
 - *Workflow Migrator*
-

Overview of Entity Migration

Migrators are Mercury Change Management object types. Each migrator is designed to migrate a specific Mercury IT Governance Center entity, as well as all of its dependent objects, from one Mercury IT Governance Center instance to another.

Migrating configurations using entity migrators and workflows allows you to 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 to use.



Migrations using entity migrators can be done only between Mercury IT Governance Center instances of the same release.

The following entities can be migrated:

- Commands (special commands)
- Object types
- Portlets
- Project templates
- Report types
- Request header types
- Request types
- User data contexts

- Validations
- Workflows

Migration Process

Following are the events that occur when a package line with an entity migrator object type is processed through a migration step. The example assumes that the package is created in the destination instance.

This example process includes events in the following order:

1. The destination Mercury IT Governance Server connects to the source server using telnet or SSH, and makes a request for the specified configuration data.
2. The source server pulls the requested configuration data from its database using a JDBC connection.
3. The source server creates a content bundle containing XML files.
4. The content bundle is transferred to the destination server using FTP.



Note

A content bundle can be imported into another Mercury IT Governance Center instance, or archived separately. When archived separately, a content bundle takes the form of a `zip` file containing a collection of XML files that can later be stored or imported into a Mercury IT Governance Center instance.

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

5. The destination server unpacks the content bundle.
6. The destination server writes the configuration data into its database using a JDBC connection.
7. The destination server generates an execution log.

Example Migration: Extracting an Object Type

Following is a sample procedure to extract an object type, and example screens to illustrate the procedure.



Note

The user creating, submitting, and processing the migrations must have the proper licenses and access grants. For more information, see the *Security Model Guide and Reference*.

Procedure

1. Create the KINTANA_SERVER environment (see [Figure 11-1](#)).
 - a. In the Environment Workbench, open the KINTANA_SERVER environment.
 - b. Define and enable the server environment.
2. Create a Change Management workflow (see [Figure 11-2](#)).
3. Create a package specifying the workflow you have just created.
4. Add a package line that names the object type as ITG Object Type Migrator (see [Figure 11-3](#)).
5. Submit the workflow and process it.
6. Review the execution log (see [Figure 11-4](#)) to be sure your migration was successful.

Key Screens

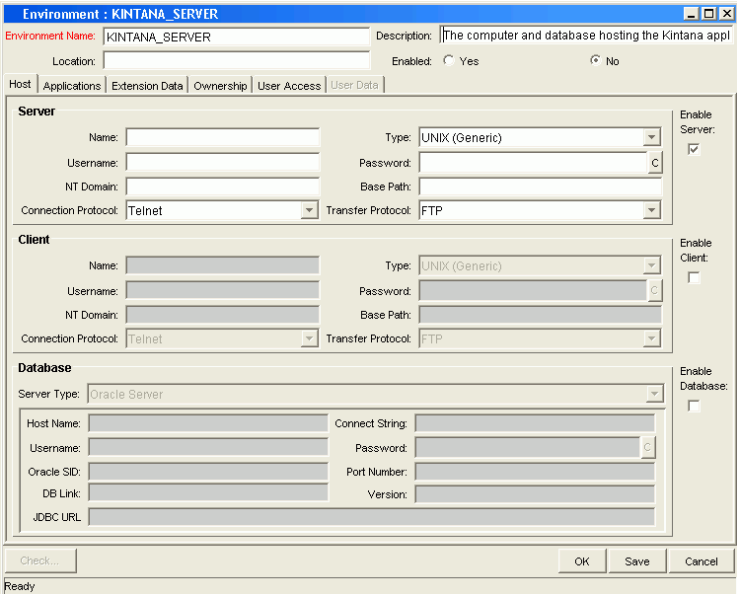


Figure 11-1. KINTANA_SERVER environment

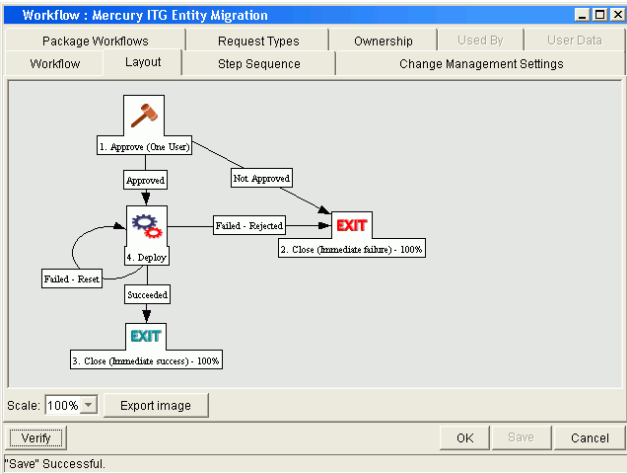


Figure 11-2. Change Management workflow

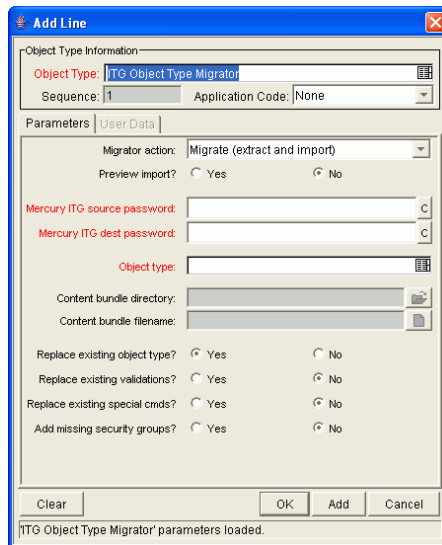


Figure 11-3. Package line definition



Figure 11-4. Execution log

Defining Entity Migrators

All migration package lines have similar settings, which are described in this section using the ITG Object Type Migrator shown in [Figure 11-3](#) as an example.

Migrator Action Field

The Migrator action field is shown in [Figure 11-5](#).

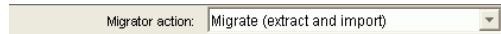


Figure 11-5. Migrator action field

The three choices for this field are:

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

[Table 11-1](#) summarizes the behavior of the fields that are dependent on the value of the Migrator action field.

Table 11-1. Migrator action field dependencies

Field and Field Set Names	Migrator Action Value		
	Migrate (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

The basic parameters are shown in *Figure 11-6*.

A screenshot of a web-based form with a light beige background. It contains three input fields. The first field is labeled 'Object type:' in red text and has a small grid icon to its right. The second field is labeled 'Content bundle directory:' and has a folder icon to its right. The third field is labeled 'Content bundle filename:' and has a document icon to its right. All fields are currently empty.

Figure 11-6. Basic parameters

Whether or not the basic parameters are enabled or required depends on the migrator action selected. In *Figure 11-6*, the parameters are entity name (in this case, Object type), Content bundle directory, and Content bundle filename.

Content Bundle Fields

These fields behave differently depending on the migrator action specified.

- Migrate (extract and import)—A content bundle is temporarily created, but you are not prompted to provide any information related to this temporary file.
- Extract only—You can specify the content bundle location and file name, or leave these fields blank and accept the default behavior. By default, the bundle will be created in the file system of the source Mercury IT Governance Center application 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.
- Import only—You must specify the content bundle location and file name. You can select the file by browsing the file system of the destination Mercury IT Governance Server.

Import Flags

The import flags are highlighted in [Figure 11-7](#).

The screenshot shows a configuration window for migrating entities. At the top, there is a dropdown menu for 'Migrate to (destination instance ID)'. Below it, the 'Preview import?' field has radio buttons for 'Yes' and 'No', with 'No' selected. There are two password fields: 'Mercury ITG source password:' and 'Mercury ITG dest password:'. Below these are fields for 'Object type:', 'Content bundle directory:', and 'Content bundle filename:'. At the bottom, there are four radio button groups for import flags: 'Replace existing object type?' (Yes selected), 'Replace existing validations?' (No selected), 'Replace existing special cmds?' (No selected), and 'Add missing security groups?' (No selected).

Figure 11-7. Import flags

The import flags vary depending on the object type.

Preview Import Field

If you set the Preview Import? field to **Yes**, the entity is not migrated. Instead, the migration is simulated and an execution log is generated.

Import Behavior Fields

These fields modify the specific import behavior for the entity to be migrated.

- **Replace existing (entity)**—If the entity to be migrated already exists in the target Mercury IT Governance Center instance, you can decide whether or not to replace it. The default value is **Yes**.

If the entity does not exist in the destination instance, it is created.

- **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 field's value, any validations that are missing from the destination instance is automatically created.

- 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 Fields

The password fields are shown in [Figure 11-8](#).



The image shows two text input fields stacked vertically. The top field is labeled 'Mercury ITG source password:' and the bottom field is labeled 'Mercury ITG dest password:'. Both labels are in red text. To the right of each input field is a small square button with the letter 'c' inside.

Figure 11-8. Password fields

The password fields are enabled if the Migrator action field is set to Migrate (extract and import).

Source Password Field

When the migrator contacts the source application server, the current user's name and password are used to log on to the target instance.

An application administrator performing a migration should already be linked to the proper security group containing the access grant allowing access to the source Mercury IT Governance Center instance, Sys Admin: Migrate Mercury ITG objects. However, if the password for the current user in the source instance is different from the password in the current instance, you can use this field to enter that override value.



If you modify the object, you need to enter the source and destination passwords for import and export.

Destination Password Field

When the migrator contacts the destination Mercury IT Governance Server, the current user's name and password is used to log on to the source instance.

An application administrator performing a migration should already be linked to the proper security group containing the access grant that allows access to

the destination Mercury IT Governance Center instance, Sys Admin: Migrate Mercury ITG Objects. However, if the password for the current user in the destination instance is different from the password in the current instance, you can use this field to enter that override value.



Note

If you modify the migrator object type, you need to enter the source and destination passwords for import and export.

Internationalization Field

This field might not be displayed on migrator object types, but is enabled and defaulted to **Same language and character set**. To change this value, edit the migrator object type.

This field takes three possible values:

- **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.
- **Different language or character set**—This option allows you to override character set or language incompatibilities within the same localization.
- **Different localization**—This option provides for the migration of content between instances belonging to different localizations (for example, English to German, or German to English).

Environment Considerations

When migrating entities, Mercury Change Management logs on to remote machines in the same way any other user would (that is, using FTP, SCP, SSH, or Telnet). Mercury Change Management can log on using any existing username and password.

Mercury recommends that you generate a new user (for example, Mercury IT Governance Center) on every machine that Mercury Change Management will access.

The user you create for this purpose must have full access to the *ITG_Home* directory on the Mercury IT Governance Server, as well as read and write permissions on other required directories.

Environment Connection Protocols

You must specify in the environment definition the communication protocols (for example, telnet) that will be used to connect to the server or client.

For More Information

For information about connection protocols supported by Mercury IT Governance Center, see the *System Requirements and Compatibility Matrix* document.

Environment Transfer Protocols

You must specify in the environment definition the transfer protocols that will be used to transfer files to the server or client. Mercury IT Governance Center supports the following transfer protocols:

- FTP
- FTP (active)
- FTP (passive)
- Secure Copy
- Secure Copy 2

Setting Up the FTP Protocols

The following capabilities must be enabled on the source and destination machines:

- FTP (server to server) (see [Figure 11-9](#)):
 - Either the source or the destination environment needs to allow outgoing connections to a third party
 - The FTP PORT command must be enabled on one of the environments
 - The FTP PASV command must be enabled on the other environment

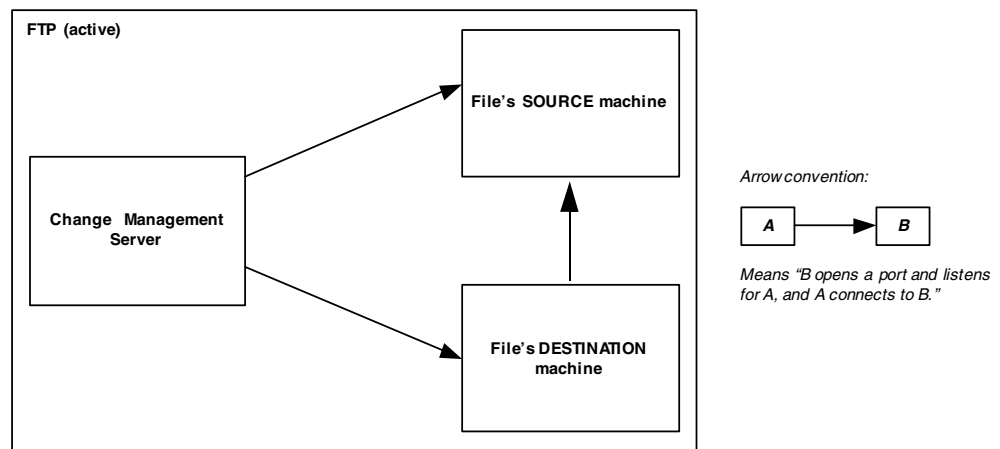


Figure 11-9. FTP (server to server)

- FTP (active) (see *Figure 11-10*):

The PORT command must be enabled on both the source and destination environments.

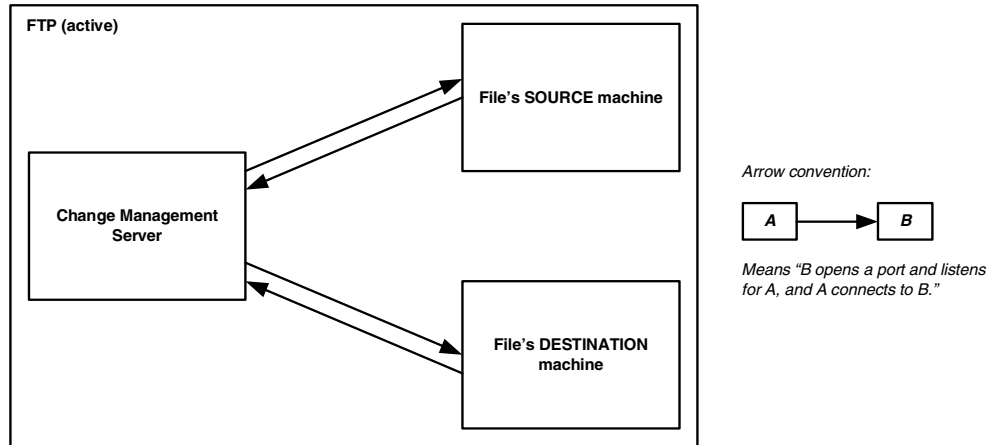


Figure 11-10. FTP (active)

- FTP (passive) (see *Figure 11-11*).

PASV must be enabled on both the source and destination environments. In this configuration, the Mercury IT Governance Server sends a command to the source or destination instructing that environment to open a port. The Mercury IT Governance Server then connects to that port.

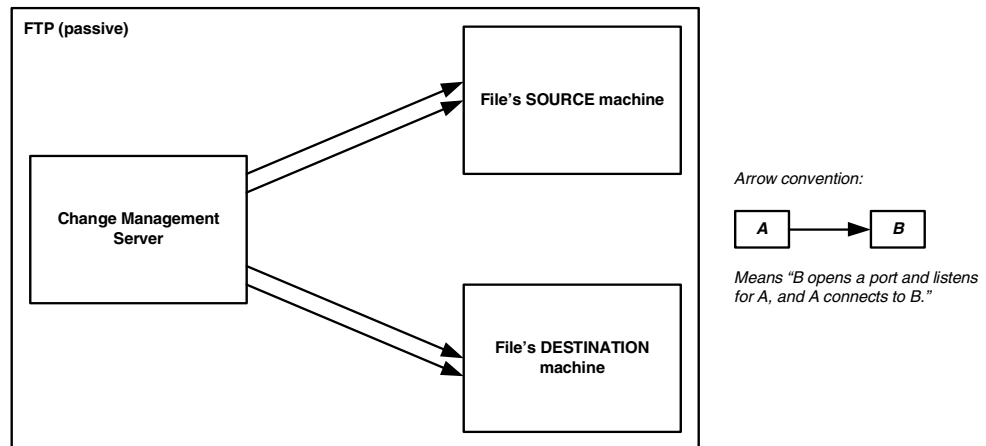


Figure 11-11. FTP (passive)

For More Information

For information about transfer protocols supported by Mercury IT Governance Center, see the *System Requirements and Compatibility Matrix* document.

Setting the SERVER_ENV_NAME Parameter

The Mercury IT Governance Center migrators depend on the SERVER_ENV_NAME server configuration parameter. You need to set this parameter to the name of an environment in the Mercury IT Governance Center system that describes the host server running that Mercury IT Governance Center instance. Since the destination instance should be the driving instance, SERVER_ENV_NAME should be set to KINTANA_SERVER.

On either platform, by default, the server environment configuration entry should appear as below. Be sure this parameter is properly configured.

```
SERVER_ENV_NAME=KINTANA_SERVER
```

When a Mercury IT Governance Center installation automatically generates the environment KINTANA_SERVER, it is also captured as the default SERVER_ENV_NAME parameter in the `server.conf` file.

**Note**

It is unlikely that you will need to change the default value of KINTANA_SERVER. In case you must modify it, you need to synchronize the `server.conf` parameter and the Mercury IT Governance Center environment name.

Security Considerations

The following sections provide information about security considerations relating 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 able to perform migrations can migrate it.
- If ownership is configured for the entity and the user migrating is not in the ownership group, the migration will fail.
- If ownership is configured for the entity and the user migrating is in the ownership group, the migration succeeds.
- If ownership is configured for the entity 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 import, but not export.

Migrations and Entity Restrictions

A report type might refer to security groups through entity restrictions. The Report Type migrator will transfer references to security groups, but will not create a security group.

If the referenced security group does not exist in the destination instance, the reference will be discarded in transit. A message to that effect will appear in the migration's execution log.

If the source instance contains security groups that do not exist in the destination instance at the time of migration, the entity restrictions for the migrated report type will not be accurate.

You should therefore manually verify report types that contain entity restrictions in the destination instance following migration.

Entity Migrators

This section lists and describes the entity migrators.

Data Source Migrator

Figure 11-12 shows the Data Source migrator.

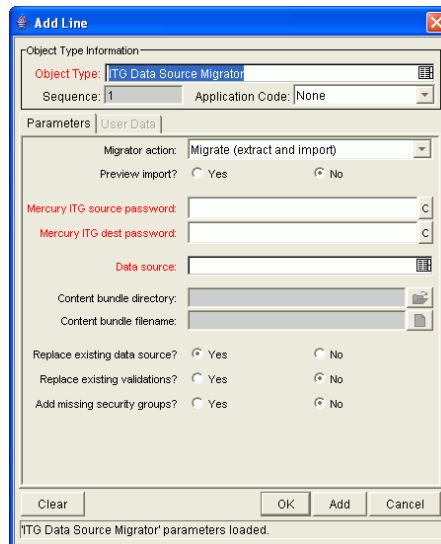


Figure 11-12. Data Source migrator

For information about the fields in this migrator, see [Defining Entity Migrators](#) on page 215.

Module Migrator

Figure 11-13 shows the Module migrator.

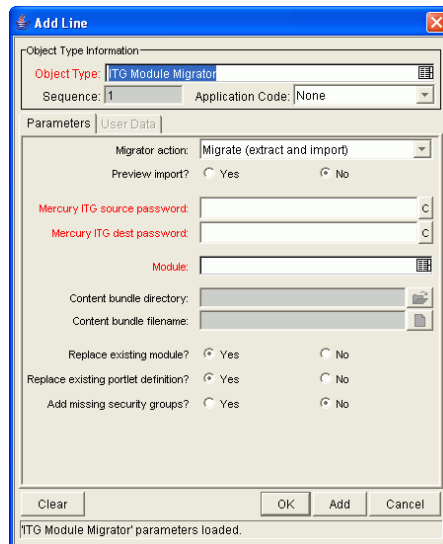
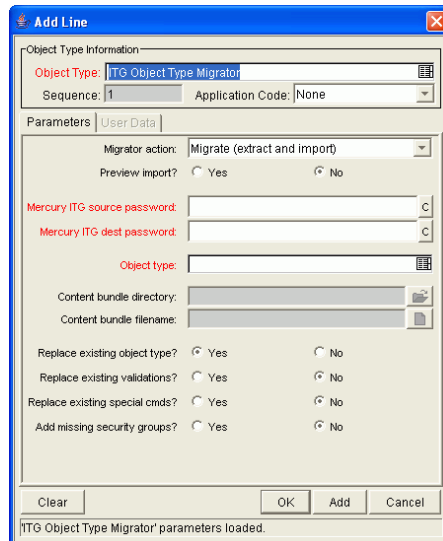


Figure 11-13. Module migrator

For information about the fields in this migrator, see [Defining Entity Migrators](#) on page 215.

Object Type Migrator

Figure 11-14 shows the Object Type migrator.



For information about most of the fields in this migrator, see [Defining Entity Migrations on page 215](#).

This migrator contains one additional field (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 the value, special commands missing from the destination instance are created automatically.

Configuration Considerations

The migrator also transfers the following information:

- Special commands referenced by command steps
- Validations referenced by fields
- Environments referenced by validations
- Special commands referenced by validations

- Special commands referenced by other special commands
- Ownership group information for the entity



The migrator will transfer references to environments from validations, but will not create an environment. If the referenced environment does not exist in the destination instance, the migration will fail. In this case, you need to create the missing environment manually in the destination instance.

Overview Page Section Migrator

Figure 11-15 shows the Overview Page Section migrator.

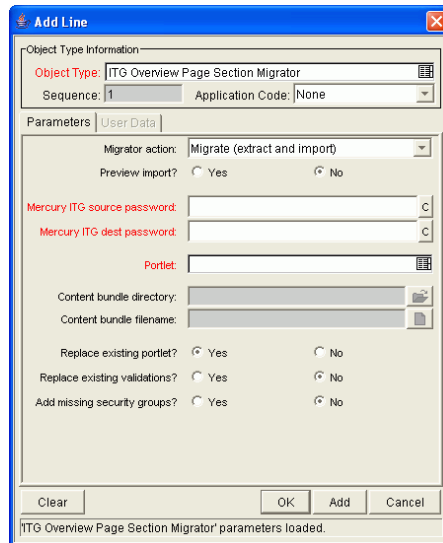


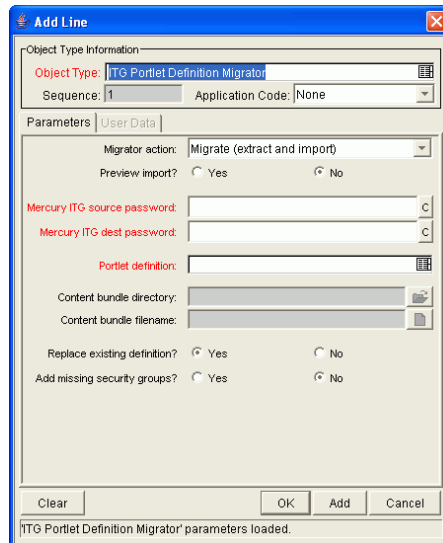
Figure 11-15. Overview Page Section migrator

For information about the fields in this migrator, see [Defining Entity Migrators on page 215](#).

Portlet Migrator

The Portlet migrator contains all standard entity migrator object type fields. When migrating a portlet to replace an existing enabled portlet in an active instance, the changes migrated will be propagated to all users who have added the same portlet to their Dashboard.

Figure 11-16 shows the Overview Page Section migrator.



The screenshot shows a dialog box titled "Add Line" with a close button in the top right corner. The dialog is divided into several sections:

- Object Type Information:** Contains a text field for "Object Type" with the value "ITG Portlet Definition Migrator", a "Sequence" field with the value "1", and a dropdown for "Application Code" set to "None".
- Parameters:** A tabbed section with "User Data" selected. It includes:
 - A dropdown for "Migrator action" set to "Migrate (extract and import)".
 - Radio buttons for "Preview import?" with "No" selected.
 - Text fields for "Mercury ITG source password:" and "Mercury ITG dest password:" with clear buttons.
 - A text field for "Portlet definition:" with a list icon.
 - Text fields for "Content bundle directory:" and "Content bundle filename:" with folder selection icons.
 - Radio buttons for "Replace existing definition?" with "Yes" selected.
 - Radio buttons for "Add missing security groups?" with "No" selected.
- Buttons:** "Clear", "OK", "Add", and "Cancel" are located at the bottom.
- Status Bar:** Displays the message "ITG Portlet Definition Migrator' parameters loaded."

Figure 11-16. Portlet migrator

For information about the fields in this migrator, see [Defining Entity Migrators](#) on page 215.

Project Template Migrator

Figure 11-17 shows the Project Template migrator.

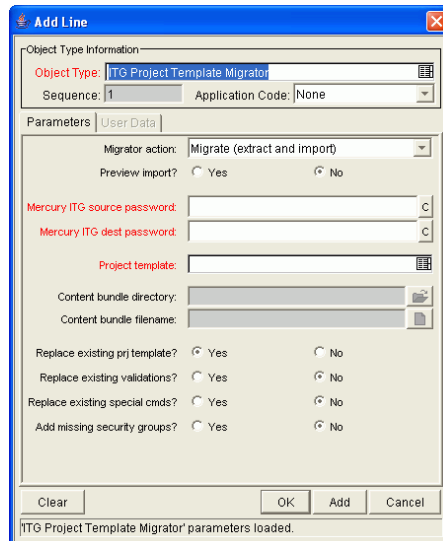


Figure 11-17. Project Template migrator

For information about most of the fields in this migrator, see [Defining Entity Migrators on page 215](#).

This migrator contains one additional field (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 the value, special commands missing from the destination instance are created automatically.

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 appears in the migration's execution log.

A project template can also contain references to other project templates that have been used to create the current template. The project template migrator transfers these references, but does not create a missing nested project template.



Note

To be sure these references are preserved, migrate first any project templates that have been 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.

Report Type Migrator

Figure 11-18 shows the Report Type migrator.

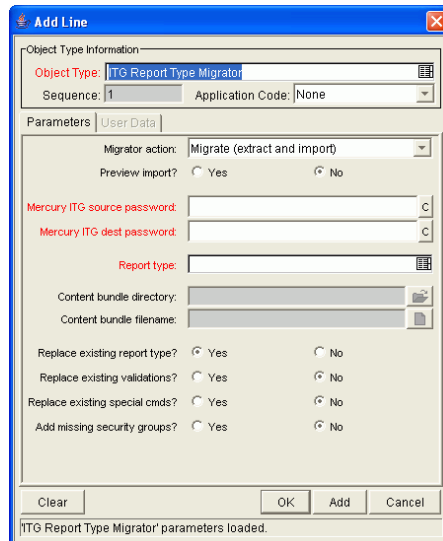


Figure 11-18. Report Type migrator

For information about most of the fields in this migrator, see [Defining Entity Migrations on page 215](#).

This migrator contains one additional field (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 the value, special commands missing from the destination instance are created automatically.

Configuration Considerations

The Report Type migrator also transfers the following information:

- Special commands referenced by command steps
- Validations referenced by fields
- Environments referenced by validations
- Special commands referenced by validations

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



The Report Type migrator transfers references to environments from validations, but does not create an environment. If the referenced environment does not exist in the destination instance, the migration fails. In this case, you need to create the missing environment manually in the destination instance.

Request Header Type Migrator

Figure 11-19 shows the Request Header Type migrator.

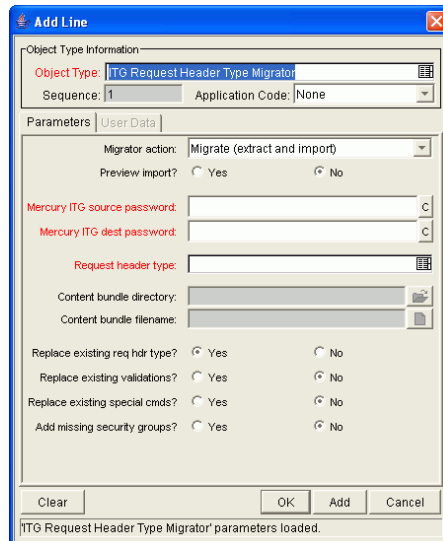


Figure 11-19. Request Header Type migrator

For information about most of the fields in this migrator, see [Defining Entity Migrations on page 215](#).

This migrator contains one additional field (Replace Existing special cmds?). If the validation to be migrated references Mercury IT Governance Center special req 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 the value, special commands missing from the destination instance are created automatically.

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 need to create the missing environment manually in the destination instance.

Request Type Migrator

Figure 11-20 shows the Request Type migrator.

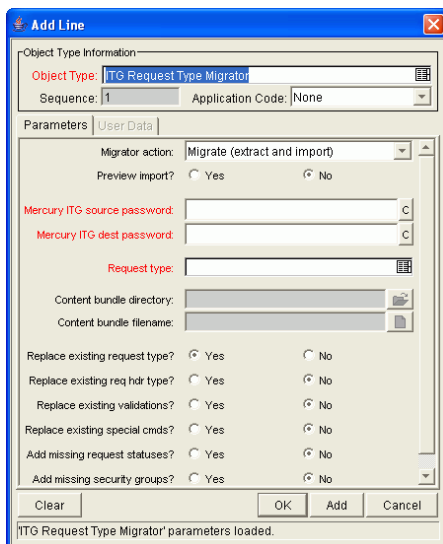


Figure 11-20. Request Type migrator

For information about most of the fields in this migrator, see [Defining Entity Migrators on page 215](#).

The Request Type migrator contains three additional import behavior fields:

- **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 the value, special commands missing from the destination instance are created automatically.

- **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**.

A message appears in the execution log for each referenced request status that is not created.



If this field 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 need to 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 (in 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 need to 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 need to 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. Since 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

Figure 11-21 shows the Special Command migrator.

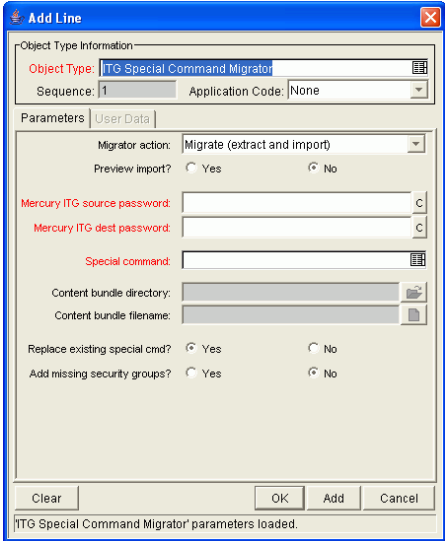


Figure 11-21. Special Command migrator

For information about the fields in this migrator, see *Defining Entity Migrators* on page 215.

User Data Context Migrator

Figure 11-22 shows the User Data Context migrator.

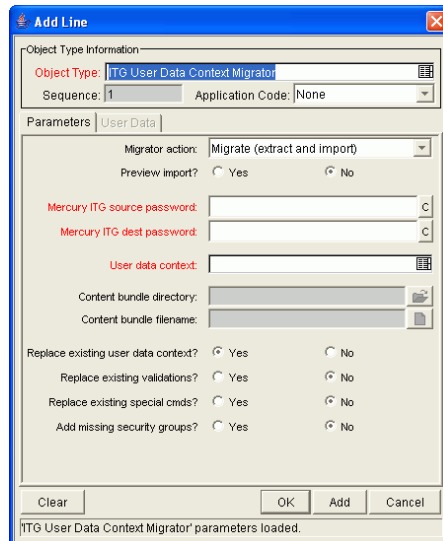


Figure 11-22. User Data Context migrator

For information about most of the fields in this migrator, see [Defining Entity Migrations on page 215](#).

This migrator contains one additional field (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 the value, special commands missing from the destination instance are created automatically.

Validation Migrator

Figure 11-23 shows the Validation migrator.

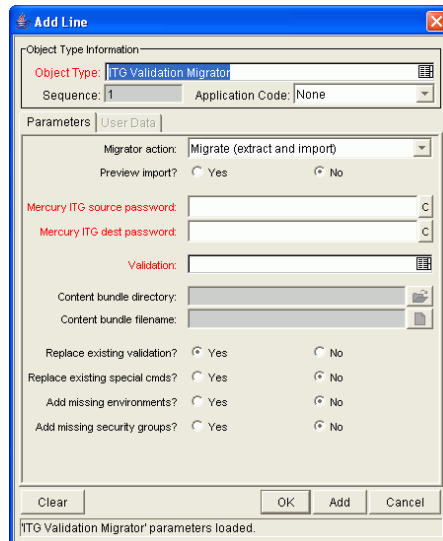


Figure 11-23. Validation migrator

For information about most of the fields in this migrator, see [Defining Entity Migrators on page 215](#).

This migrator contains two additional import behavior fields:

- **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 the value, special commands missing from the destination instance are created automatically.

- **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, you should confirm and complete environment data manually if any environments have been created by the migrator.

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

Figure 11-24 shows the Workflow migrator.

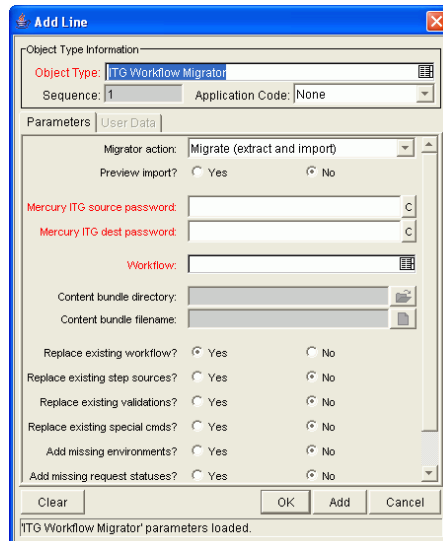


Figure 11-24. Workflow migrator

For information about most of the fields in this migrator, see [Defining Entity Migrations on page 215](#).

This migrator contains the following additional import behavior fields:

- **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 decide whether or not to replace them. This includes both special commands directly referenced by the workflow, and also special commands referenced by these special commands. Special commands in validations referenced by the workflow 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 already exist in the target Mercury IT Governance Center instance, you can decide whether or not to replace them. If the existing step sources are already in use by workflows in the destination instance, however, certain fields cannot be changed even if **Replace Existing Step Sources?** is set to **Yes**. These fields include **Workflow Scope**, **Validation**, and **Decision Type**.

- 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 decide whether or not to create them. 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, you should confirm and complete environment data manually if any environments have been created by the migrator.

- Add missing request statuses?—If the workflow 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**.

Configuration Considerations

The workflow migrator also transfers the following information:

- Subworkflows referenced by workflow steps
- Special commands referenced by command steps
- Workflow step sources referenced by workflow steps
- Validations referenced by parameters or workflow step sources
- Environments and environment groups referenced by workflow steps
- Environments referenced by environment groups referenced by workflow Steps
- Environments referenced by validations
- Special commands referenced by validations
- Special commands referenced by workflow step sources
- Special commands referenced by other special commands already referenced elsewhere
- Security groups referenced by workflow steps
- Request statuses referenced by workflow steps
- Notifications referenced by workflow steps

- Notification intervals referenced by notifications
- Security groups referenced by notifications
- 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, this notification interval is created. The workflow migrator does not replace an existing notification interval.

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 reference is discarded in transit. A message to that effect appears in the migration's execution log.

The workflow migrator transfers references to request types, but does not create a request type. If the referenced request type does not exist in the destination instance, the reference is discarded in transit. A message to that effect appears in the migration's execution log.

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

- A new request type referring to a new workflow is migrated. Since 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, since the workflow it refers to exists, the references are preserved.

Replacing an Existing Workflow

There are some restrictions in using the Workflow migrator to make changes to an existing process that is already in use (by requests or package lines). These restrictions help to make sure that these existing requests or package lines are not damaged by the migration.

Specifically, the workflow migration does succeed unless the migrator logic can find a workflow step in the incoming process that corresponds to each step in the previous process. 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 for matching workflow steps to each other. If the step source is not identical, then two workflow steps cannot be considered to match.

- When both the incoming and previous workflows assign a unique name to each workflow step, these workflow step names are used in combination with the step source to assess matching.
- When a workflow step name is repeated within either process, the step sequence is used instead, in combination with the step source, to assess matching.

The Workflow migrator is not able to 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.

Due to this matching restriction, each open request is on the same process step following the migration as it was prior to the migration. The migration might have changed the name of this step, but it has not transitioned request workflows.

It is important to note, however, 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 allows open requests to move forward.

The migration’s execution log contains a table listing old and new workflow steps.

Mercury recommends using the **Preview import** mode first when replacing an existing workflow, and inspecting this table of matched workflow steps before running such a 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 already 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 change of name.

2. Migrate the new version of the workflow into the production instance, under the original name.

Since the production instance no longer contains a workflow by this name, the migration treats this as a new workflow.

3. Following 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 the original name.



Server Configuration Parameters

In This Chapter:

- *Overview of Configuration Parameters*
 - *Determining Appropriate Parameter Settings*
 - *Required Parameters*
 - *Directory Path Names*
 - *Categories of Performance-Related Parameters*
 - *server.conf Parameters*
 - *logging.conf Parameters*
 - *LdapAttribute.conf Parameters*
-

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

For more information about the Mercury IT Governance Server directory structure, see [Appendix B: Server Directory Structure and Server Tools](#) on page 279.

Determining Appropriate Parameter Settings

For the majority of Mercury IT Governance Center installations, the default settings for these parameters is appropriate.

Considerations are provided in the parameter descriptions that will help you determine under what circumstances you might change the parameter settings.

Required Parameters

The Required column shows whether the server parameter is a required parameter for setting up a Mercury IT Governance Server. A value of TRUE in this column indicates that the parameter is required. A value of FALSE in this column 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 137.

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. In particular, do not use backslashes (\) when entering directory paths in the `server.conf` file, since Mercury IT Governance Center does not recognize backslashes.

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 186.

server.conf Parameters

Table A-1 lists the key Mercury IT Governance Server configuration parameters located in the `server.conf` file in the `ITG_Home` directory.

The `server.conf` file contains the values of all of the server parameters when the server configuration utility was last run. The `server.conf` file is automatically regenerated by the configuration utility.

You can also edit the `server.conf` file directly. If you do that, follow these steps:

1. Stop the server (`kStop.sh`).
2. Edit `server.conf`.
3. Run the `kUpdateHtml.sh` script to propagate the changes.
4. Restart the server (`kStart.sh`).



Note

A short form of the parameter name is used in *Table A-1*—prepended to each name is the string `com.kintana.core.server`.



Note

To get a list of the `server.conf` parameters on your active Mercury IT Governance Server, along with their current values, run the Server Configuration report. For information about how to do that, see [Running Server Reports Using Admin Tools on page 156](#).

Table A-1. *server.conf* parameters (page 1 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
ALLOW_SAVE_REQUEST_DRAFT	Allows requests to be saved without automatically submitting them in the standard interface.	Default: FALSE Valid: TRUE, FALSE
*ATTACHMENT_DIRNAME	Absolute pathname of the directory where attached documents will be stored. This directory: <ul style="list-style-type: none"> • Must give read/write access to Web browsers • Should be outside the directory tree when using an external Web server 	Example: c:/itg/attachments
*AUTHENTICATION_MODE	User authentication method. Specify multiple modes by using a comma-delimited list of valid values.	Default: ITG Valid: ITG, LDAP, NTLM, SITEMINDER Example: ITG, LDAP
AUTOCOMPLETE_STATUS_REFRESH_RATE Category: Scheduler/services/thread	Interval (in seconds) at which the command status is refreshed to provide a list of values in an auto complete.	Default: 5
*BASE_PATH	Full path to the directory where the Mercury IT Governance Server is installed.	The default depends on the operating system platform
*BASE_URL	Web location (top directory name) of the Mercury IT Governance Server.	http://www.mydomain.com:8080
CLIENT_TIMEOUT Category: Timeout	Interval (in minutes) at which the Workbench interface sessions sends a message to inform the Mercury IT Governance Server that the client is still active. Under normal operation, do not change this value.	Default: 5
**CONC_LOG_TRANSFER_PROTOCOL (Required if ORACLE_APPS_ENABLED = TRUE)	Transfer protocol to use when transferring concurrent request logs and patching README files.	Default: FTP Valid: FTP, SCP

Table A-1. server.conf parameters (page 2 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
**CONC_REQUEST_PASSWORD (Required if ORACLE_APPS_ENABLED = TRUE)	Encrypted password of the concurrent request user.	Example: fnd (encrypted)
**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) with CONC_LOG_TRANSFER_PROTOCOL.	Example: applmgr
DATE_NOTIFICATION_INTERVAL	Interval (in minutes) at which the Mercury IT Governance Server is to check whether date-based notifications are waiting to be sent, and to send them.	Default: 60
DAYS_TO_KEEP_APPLET_KEYS	Duration (in days) to keep records of applet keys.	Default: 1
DAYS_TO_KEEP_COMMAND_ROWS	Duration (in days) to keep records of all commands.	Default: 1
DAYS_TO_KEEP_INTERFACE_ROWS Category: Open Interface	Duration (in days) to keep records of all interfaces.	Default: 5
DAYS_TO_KEEP_LOGON_ATTEMPT_ROWS Category: Cleanup	Duration (in 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	Duration (in seconds) for the Mercury IT Governance Server to keep attempting to log on to the database before reporting that the database is not available.	Default: 30
*DB_PASSWORD	Password of the database schema containing the Mercury IT Governance tables.	Example: #!#password#!#
*DB_USERNAME	Name of the database schema containing the Mercury IT Governance tables.	Example: knta

Table A-1. *server.conf* parameters (page 3 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
DEFAULT_COMMAND_TIMEOUT Category: Timeout	Duration (in seconds) for the Mercury IT Governance Server to keep attempting to run commands before timing out.	Default: 90
EMAIL_NOTIFICATION_CHECK_INTERVAL Category: Scheduler/services/thread	Interval (in seconds) at which the Mercury IT Governance Server is to check if notifications are waiting to be sent out.	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
ENABLE_DB_SESSION_TRACKING Category: Low-level debug	Whether to enable the resource and cost update service.	Default: FALSE Valid: TRUE, FALSE
ENABLE_EXCEPTION_ENGINE Category: Scheduler/services/thread	Whether to enable the exception engine, which runs a process to determine whether active projects are running on time or not. Set the exception engine interval with EXCEPTION_ENGINE_INTERVAL.	Default: TRUE Valid: TRUE, FALSE
ENABLE_JDBC_LOGGING Category: High-level debug	Whether to enable JDBC logging, which records SQL run against the database, the time required to execute the SQL, and the time to fetch the results. This information is recorded in jdbc.System_Name.log in the server log directory. This parameter can be useful in debugging system performance problems. You can set this parameter in the Workbench interface without stopping the system (Edit > Settings).	Default: FALSE Valid: TRUE, FALSE

Table A-1. server.conf parameters (page 4 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
ENABLE_PENDING_PROJECT_CHANGE Category: Scheduler/services/thread	Enables a service to process a state change. The target state change results in a state change in another object (for example, a request waiting for a task that just completed). The target state change cannot be performed because the target entity is locked. The parameter that determines the interval this service is run is PENDING_PROJECT_STATE_CHANGE.	Default: TRUE Valid: TRUE, FALSE
ENABLE_SQL_TRACE Category: High-level debug	Corresponds to the Enable DB Trace Mode checkbox in the Server Settings screen. Records information used to debug the Oracle database.	Default: FALSE Valid: 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: TRUE, FALSE
**EXCEPTION_ENGINE_INTERVAL Required if ENABLE_EXCEPTION_ENGINE = TRUE)	Interval (in seconds) when the exception engine process runs.	Default: 4200
**EXCEPTION_ENGINE_WAKE_UP_CHECK_FREQUENCY (Required if ENABLE_EXCEPTION_ENGINE = TRUE) Category: Scheduler/services/thread	Interval (in seconds) before a task is verified for exceptions.	Default: 1500
**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: 1 through 24

Table A-1. server.conf parameters (page 5 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
*HOURS_TO_KEEP_DEBUG_MESSAGE_ROWS Category: Cleanup	Duration (in 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. In this case, decrease this value accordingly.	Default: 48
*HTTP_PORT	Port to be used to communicate with the built-in HTTP server.	Default: 8080 Valid: Unique port greater than 1024 and distinct from the Web server, SQL*Net, and RMI ports.
I18N_CARAT_DIRECTION	Caret position on input fields (for example, text fields). If unspecified, same as I18N_SECTION_DIRECTION.	Valid: ltr, rtl
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: ltr, rtl
I18N_REPORTS_ENCODING	Character encoding to be used to generate reports in Mercury IT Governance Center. Recommended for Windows systems: IW8MSWIN1255	Valid: Any encoding algorithm that Oracle can interpret.
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 (Windows): windows-hebrew
I18N_SECTION_DIRECTION	Layout direction of custom sections (for example, request detail sections). If unspecified, same as I18N_LAYOUT_DIRECTION.	Valid: ltr, rtl

Table A-1. *server.conf* parameters (page 6 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
*INSTALLATION_LOCALE	<p>Language and country code of the Mercury IT Governance Center installation.</p> <p>Language code must match the Mercury IT Governance Center installation language.</p>	<p>Default: en_US</p> <p>Example: de_DE</p>
<p>*JDBC_URL</p> <p>Note:</p> <p>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.</p> <p>Example (connection to Jaguar1 and Jaguar2):</p> <pre>jdbc:oracle:thin:@(DESCRIPTION= N= (AADDRESS_LIST=(ADDRESS= (PROTOCOL=TCP) (HOST=jaguar1) (PORT=1521)) (ADDRESS= (PROTOCOL=TCP) (HOST=jaguar2) (PORT=1521))) (CONNECT_ DATA- (SERVICE_NAME=J920)))</pre>	<p>Locator for the database containing the Mercury IT Governance Center database schema.</p> <p>Must be specified correctly for Mercury IT Governance Server to communicate with the database.</p> <p>Format:</p> <pre>jdbc:oracle:thin:@hostname:port:SID</pre> <p>where:</p> <ul style="list-style-type: none"> - <i>hostname</i> is the DNS name or IP address of the system running the database. - <i>port</i> 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. - <i>SID</i> is the database system ID. 	<p>Example:</p> <pre>jdbc:oracle:thin: @DBhost.domain.com: 1521:SID</pre>
JSP_RECOMPILE_ENABLED	<p>Whether changes to JSP files are to be picked up on a running server, thereby making them immediately visible.</p> <p>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.</p> <p>If JSP pages are expected to be updated regularly, set to TRUE.</p>	<p>Default: FALSE</p> <p>Valid: TRUE, FALSE</p>
<p>**KINTANA_LDAP_ID</p> <p>(Required if AUTHENTICATION_MODE = LDAP)</p>	<p>Mercury IT Governance Center account on the LDAP server.</p> <p>Used by the Mercury IT Governance Server to bind to the LDAP server.</p>	<p>Examples: uid=admin, ou=dev</p>

Table A-1. *server.conf* parameters (page 7 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
**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_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	Duration (in minutes) before the Mercury IT Governance Server terminates a user session due to inactivity. A value of 0 denotes no timeout.	Default: 120 Valid: 10 through 720
**LDAP_BASE_DN (Required if AUTHENTICATION_MODE = LDAP)	Base in the LDAP server from which the search will start. If not specified, the LDAP server is queried to determine the base.	
**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
**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.	Example: ldap:// ldap.theurl.com: 389

Table A-1. server.conf parameters (page 8 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
<p>LOCAL_IP</p> <p>Note:</p> <p>Setting this parameter as indicated resolves the following potential problems:</p> <ul style="list-style-type: none"> ● 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 machine name of the machine running the firewall, clients inside the firewall can connect, but clients outside cannot, because they cannot resolve the host name. ● 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 address translation is not accomplished between the different IP address to the IP address on the machine running the firewall. 	<p>Name of the machine running the firewall.</p> <p>Before you set this parameter, register the external IP address on the external DNS server.</p> <p>If this is set up properly, the following is true:</p> <ul style="list-style-type: none"> ● 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. <p>Both clients can then connect, each to a different IP address.</p>	
<p>*LOGON_TRIES_INTERVAL</p>	<p>Time interval (in minutes) during which logon attempts are monitored.</p>	<p>Default: 1</p>
<p>MAX_DB_CONNECTION_IDLE_TIME</p> <p>Category: Database connection</p>	<p>Duration (in minutes) that an unused database connection is held open before it is closed and removed from the pool.</p>	<p>Default: 60</p>
<p>MAX_DB_CONNECTION_LIFE_TIME</p> <p>Category: Database connection</p>	<p>Duration (in minutes) that a database session is held open before it is closed and removed from the pool.</p> <p>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.</p>	<p>Default: 1440</p>

Table A-1. server.conf parameters (page 9 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
MAX_DB_CONNECTIONS Category: Database connection	Number of database connections to hold open. In a server cluster configuration, the number of database connections for each Mercury IT Governance Server. Once this number has been reached, user sessions queue for the next available database connection. It is rare for the Mercury IT Governance Server to use more than 25 database connections?	Default: 60
*MAX_EXECUTION_MANAGERS Category: Scheduler/services/thread	Number of command executions that can run simultaneously. If one or more package lines are selected for execution, and execution manager is used to run the package lines serially. Organizations processing a high volume of packages may require a larger number of execution managers.	Default: 15
*MAX_LOGON_TRIES	Maximum number of logon attempts in the time interval specified by LOGON_TRIES_INTERVAL.	Default: 0
*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: Number greater than 1
MAX_STATEMENT_CACHE_SIZE	Maximum number of prepared statements that are cached per database connection. Part of the DB connection pool settings.	Default: 50 Valid: Number greater than 0
*MAX_WORKER_THREADS Category: Scheduler/services/thread	The number of threads can run simultaneously to process scheduled tasks (for example, reports or request commands). If the Mercury IT Governance Server is heavily loaded, lower this value to reduce the server's workload. If there are a lot of pending tasks and additional capability is available on the server, raise this value to improve performance.	Default: 10

Table A-1. *server.conf* parameters (page 10 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
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: <code>http://wwwserver.mydomain.com/itg</code>
MULTICAST_DEBUG	Whether or not incoming and outgoing multicast messages are to be logged to the Mercury IT Governance Server log.	Default: FALSE Valid: TRUE, FALSE
MULTICAST_IP	Multicast IP address.	Default: 225.39.39.244 Valid: 224.0.0.0 through 239.255.255.255
MULTICAST_LEASE_MILLIS	Interval (in milliseconds) at which the Mercury IT Governance Server sends out heartbeats.	Default: 20000 (every 20 seconds)
MULTICAST_PORT	Multicast IP port.	Default: 9000
NOTIFICATIONS_CLEANUP_PERIOD Category: Cleanup	Interval (in days) to clean up previously sent notifications.	Default: 7
ORACLE_APPS_ENABLED	Whether or not Mercury IT Governance Center is to be integrated with Oracle Applications. This parameter must be TRUE in installations running Mercury Object Migrator, Mercury GL Migrator, or Mercury Patch Migrator.	Default: FALSE Valid: TRUE, FALSE
*ORACLE_HOME	Full path to the Oracle_Home directory on the Mercury IT Governance Server machine. The Oracle_Home/network/admin directory must contain the proper TNS names (or a file containing the names: <code>tnsnames.ora</code>) required to connect to the Mercury IT Governance database schema.	Example: <code>d:/orant</code>
*PASSWORD_EXPIRATION_DAYS	Default expiration period (in days) of passwords for new users. A value of 0 indicates no expiration.	Default: 0 Valid: 0 through 366

Table A-1. server.conf parameters (page 11 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
*PASSWORD_REUSE_RESTRICTION_DAYS	Duration (in days) to restrict reuse of an old password after a new password is set. A value of 0 indicates no restriction.	Default: 0 Valid: 0 through 2192
PENDING_COST_EV_UPDATE_SERVICE_DELAY	Duration (in seconds) to wait after completion of the Pending Cost EV Update service before restarting the service.	Default: 30 Valid: Number greater than 0
PENDING_COST_EV_UPDATE_SERVICE_ENABLED	Enables a service that asynchronously propagates external updates to the Pending Cost EV Updates service when updates could not be made immediately.	Default: FALSE Valid: TRUE, FALSE
**PENDING_PROJECT_CHANGE_INTERVAL (Required if ENABLE_PENDING_PROJECT_CHANGE = TRUE) Category: Scheduler/services/thread	Interval (in seconds) at which the service is to be run.	Default: 300
PORTLET_EXEC_TIMEOUT Category: Timeout	Time duration (in seconds) after which portlets time out. 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
REMOTE_ADMIN_REQUIRE_AUTH	Whether or not to require user authentication 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 username and password. If set to FALSE, any user with access to kStop.sh can shut down the server.	Default: FALSE Valid: TRUE, FALSE
REPORTING_STATUS_REFRESH_RATE Category: Scheduler/services/thread	Time interval (in seconds) at which report status is refreshed and displayed to the user.	Default: 5

Table A-1. *server.conf* parameters (page 12 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
RESOURCE_CACHE_SIZE	Size of caches of internal string resources.	Default: 20 Valid: Number greater than 0
RESTRICT_BYPASS_EXECUTION_TO_MANAGERS	Whether or not bypass execution of workflow steps in packages is restricted to managers. If set to TRUE, only users with an access grant of Package Manager Access can bypass executions. If set to FALSE, all users eligible to act on executions can bypass them.	Default: FALSE Valid: TRUE, FALSE
*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: <i>rmi://hostname:port/KintanaServer</i>	Default: <i>port</i> is 1099 Valid: <i>port</i> must be greater than 1024 Example: <i>rmi://gold.itg.com:1099/ITGServer</i>
*RML_PASSWORD	Password of the Oracle schema name specified in RML_USERNAME.	Valid: [encrypted password]
*RML_USERNAME	Oracle schema name for the meta layer schema. Must be the same as the database schema name used during during installation.	Valid: [whatever username format Oracle supports]
*SCHEDULER_INTERVAL Category: Scheduler/services/thread	Time interval (in seconds) at which the scheduler wakes up to check for services to be run.	Default: 60
SCPCLIENT_TIMEOUT	Time duration (in milliseconds) 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

Table A-1. *server.conf* parameters (page 13 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
SEARCH_TIMEOUT Category: Timeout	Time duration (in 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 searches completed.	Default: 60
SECURE_RMI		Default: FALSE Valid: TRUE, FALSE
SERVER_ENV_NAME	Name of the Mercury IT Governance Center environment containing information about the Mercury IT Governance Server machine (for example, hostname, username, and password). Must be set before Mercury IT Governance entity migrators or commands involving secure copy can run.	Default: KINTANA_SERVER
SERVER_MODE	Server mode.	Default: NORMAL
*SERVER_NAME	DNS name of IP address of the machine hosting the Mercury IT Governance Server.	Default: kintana Valid: [any valid machine name]
SERVER_TYPE_CODE	Operating system on which the Mercury IT Governance Server is installed.	Valid: UNIX, WINDOWS
SHOW_BASE_URL_ON_NOTIFICATION	Whether or not the URL for the Mercury IT Governance Center logon window is displayed at the top of each email notification.	Default: TRUE Valid: TRUE, FALSE
**SMTP_SERVER (Required if notifications are used)	Hostname of the SMTP-compliant mail server that acts as the gateway for email notifications.	Example: mailserver.mydomain.com
*SQLPLUS	Name of the command-line SQL*Plus executable, which must be in the <i>Oracle_Home/bin</i> directory.	Default: sqlplus

Table A-1. server.conf parameters (page 14 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
**STATS_CALC_DAY_OF_WEEK (Required if ENABLE_STATISTICS_CALCULATION = TRUE) Category: Database statistics	Day of the week on which Oracle database statistics are to be calculated.	Default: 1 (which designates Sunday) Valid: Numbers from 1 through 7 (which designates Saturday)
**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 (which designates 1 a.m. or 1:00) Valid: 0 (which designates midnight) through 23 (which designates 11 p.m. or 23:00)
**STATS_CALC_WEEK_INTERVAL (Required if ENABLE_STATISTICS_CALCULATION = TRUE) Category: Database statistics	Time interval (in weeks) at which statistics are calculated.	Default: 1 (which designated every week) Valid: Numbers from 1 through 52 Example: 2 (which designates every other week)
SYNC_EXEC_INIT_WAIT_TIME	Time duration (in seconds) after which the intermediate Request Working page opens.	Default: 4
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 seconds) at which to poll for completion of a request after the intermediate Request Working page has opened.	Default: 15
THREAD_POOL_MAX_THREADS Category: Scheduler/services/thread	Maximum number of packages to be executed simultaneously within a release distribution. When a large number of packages in a distribution are processing, increasing this value may improve performance.	Default: 10

Table A-1. *server.conf* parameters (page 15 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
THREAD_POOL_MIN_THREADS Category: Scheduler/services/thread	Minimum number of packages to be executed simultaneously within a release distribution. See also THREAD_POOL_MAX_THREADS.	Default: 5
**TIME_ZONE (Required if the Mercury IT Governance Server and the Oracle database are in different time zones)	Time zone of the Oracle database. Leave the parameter blank if the IT Governance Server and the host machine of the Oracle database are in the same time zone. If they are in different time zones, set this to the time zone of the host Oracle database. Use a valid 3-digit standard time zone (for example, PST, MST, and GMT). Do not use daylight savings-modified time zones (for example, EDT or PDT). For assistance in setting these modified times, contact Mercury Support. You can also use a fully qualified time zone name (you're not restricted to three digits) like "America/Los_Angeles" or "Australia/LHI". For a list of fully qualified names, refer to the Client Timezone report, described in Table 8-2 on page 157 .	See the description.
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: 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: TRUE, FALSE
TURN_ON_WF_TIMEOUT_REAPER Category: Scheduler/services/thread	Turns on the timeout reaper, which scans all active workflow steps to verify if they have timed out according to the settings for the step.	Default: TRUE Valid: TRUE, FALSE
USER_PASSWORD_MAX_LENGTH	Maximum length (in characters) of user passwords.	Default: 16

Table A-1. server.conf parameters (page 16 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
USER_PASSWORD_MIN_DIGITS	Minimum length (in characters) of user passwords.	Default: 0
USER_PASSWORD_MIN_LENGTH	Minimum length of a user password.	Default: 4
USER_PASSWORD_MIN_SPECIAL	Minimum number of non-alphabetic (special) characters in user passwords.	Default: 0
VISUALIZATION_EXEC_TIMEOUT	Time duration (in seconds) that costing and resource management visualizations can run before timing out.	Default: 180
WF_SCHEDULED_TASK_INTERVAL Category: Scheduler/services/thread	Time interval (in seconds) at which the Mercury IT Governance Server checks for pending scheduled tasks, and starts the tasks are if worker threads are available.	Default: 60
WF_SCHEDULED_TASK_PRIORITY Category: Scheduler/services/thread	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.	Default: 10
**WF_TIMEOUT_REAPER_INTERVAL (Required if TURN_ON_WF_TIMEOUT_REAPER = TRUE) Category: Scheduler/services/thread	Time interval (in seconds) at which the service checks for information.	Default: 900
WORK_ITEM_BREAKDOWN_SERVICE_DELAY	Time duration (in seconds) to wait after the work item breakdown service has completed a session before restarting the service.	Default: 30 Valid: Number greater than 0
WORK_ITEM_BREAKDOWN_SERVICE_ENABLED	Enables the work item breakdown service. This service asynchronously decomposes the scheduled and actual effort of work item assignments into daily units. These daily units provide the building blocks for resource management visualizations.	Default: TRUE Valid: TRUE, FALSE

Table A-1. *server.conf* parameters (page 17 of 17)

Parameter (*Required, **Required If), Category	Definition, Description, Usage	Default, Valid Values, Examples
WORK_ITEM_UPDATE_SERVICE_DELAY	Time duration (in seconds) to wait after the work item update service has completed a session before restarting the service.	Default: 120 Valid: Number greater than 0
WORK_ITEM_UPDATE_SERVICE_ENABLED	Enables the work item update service. This service asynchronously propagates external updates to work items when updates can not be made immediately.	Default: TRUE Valid: TRUE, FALSE

logging.conf Parameters

Table A-2 lists and describes the Mercury IT Governance Server configuration parameters located in the `logging.conf` file in the `ITG_Home/conf` directory.

Table A-2. `logging.conf` parameters (page 1 of 3)

Parameter (*Required)	Definition, Description, Usage	Default, Valid Values, Example
CATCH_SYSTEM_ERR	Whether or not to redirect <code>System.err</code> to the server log	Default: TRUE Valid values: TRUE, FALSE
CATCH_SYSTEM_OUT	Whether or not to redirect <code>System.out</code> to the server log.	Default: TRUE Valid values: TRUE, FALSE
DEFAULT_SERVER_LOGGING_LEVEL	<p>Default debug level of the Mercury IT Governance Server.</p> <p>Controls the verbosity of logs generated by the Mercury IT Governance Server.</p> <p>The values, which can also be set dynamically at runtime in the Workbench Server Settings window, map as follows:</p> <ul style="list-style-type: none"> • ERROR maps to None in the Server Settings window • INFO maps to Normal • DEBUG maps to Max <p>For more information about the Server Settings window, see Using the Server Settings Window on page 162.</p>	<p>Valid values:</p> <ul style="list-style-type: none"> • NONE - no information, including no 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 being produced

Table A-2. logging.conf parameters (page 2 of 3)

Parameter (*Required)	Definition, Description, Usage	Default, Valid Values, Example
DEFAULT_USER_DEBUG_LEVEL Category: High-level debug	<p>Specifies the default debug level of a user's client session.</p> <p>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.</p> <p>The values, which can also be set in the Workbench Server Settings window dynamically at runtime, map as follows:</p> <ul style="list-style-type: none"> • ERROR maps to None in the Server Settings window • INFO maps to Normal • DEBUG maps to Max <p>For more information about the Server Settings window, see Using the Server Settings Window on page 162.</p>	<p>Valid values:</p> <ul style="list-style-type: none"> • NONE - no information, including no 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 being produced
ENABLE_CONSOLE_LOGGING	Enables logging by the Mercury IT Governance Server to the console.	Valid: TRUE, FALSE
ENABLE_WEB_ACCESS_LOGGING	Whether or not to log information sent to the internal Mercury IT Governance Center Web server (Jetty).	Valid values: TRUE, FALSE
FILE_RECHECK_INTERVAL	<p>Time interval (in seconds) at which the logging.conf file is checked for changes.</p> <p>The file keeps being checked as long as the Mercury IT Governance Server is running.</p>	Default: 30
LOG_LAYOUT	Layout format of the log files.	Default: TEXT Valid values: TEXT, XML
MAX_BACKUP_INDEX	Limits the number of backup logs kept in the system.	Default 20

Table A-2. logging.conf parameters (page 3 of 3)

Parameter (*Required)	Definition, Description, Usage	Default, Valid Values, Example
ROTATE_LOG_SIZE	Size (in kilobytes) at which Mercury IT Governance Server logs are closed and a new log opened.	Default: 250
SERVER_DEBUG_LEVEL Category: High-level debug	<p>Debug level of the Mercury IT Governance Server.</p> <p>Controls the verbosity of logs generated by independent server processes (for example, EmailNotificationAgent).</p> <p>Corresponds to the Debug Level drop down list in the Server section of the Server Settings screen.</p>	Valid values: NONE, LOW, HIGH

LdapAttribute.conf Parameters

Table A-3 lists and describes the Mercury IT Governance Server configuration parameters located in the `LdapAttribute.conf` file in the `ITG_Home/conf` directory.

Table A-3. *LdapAttribute.conf* parameters (page 1 of 2)

Parameter (*Required)	Definition, Description, Usage	Default, Valid Values, Example
KNTA_USERS_INT	<p>Target table for the import. Could be mapped to any LDAP attribute.</p> <p>Always map both <code>VISIBLE_USER_DATA</code> and <code>USER_DATA</code>.</p> <p>To disable a default mapping, either comment out or delete the mapping line.</p> <p>Mappings:</p> <ul style="list-style-type: none"> • <code>USERNAME = sAMAccountName</code> • <code>FIRST_NAME = givenname</code> • <code>LAST_NAME = sn</code> • <code>EMAIL_ADDRESS = mail</code> • <code>PHONE_NUMBER = telephonenumber</code> • <code>DEPARTMENT_MEANING = departmentNumber</code> • <code>LOCATION_MEANING = locality</code> • <code>MANAGER_USERNAME = manager</code> • <code>USER_DATA1 = mail</code> • <code>VISIBLE_USER_DATA1 = mail</code> 	<p>Format: <code>ColumnName = LDAPAttribute</code></p>
LDAP_DYNAMIC_GROUP_MEMBERS	<p>Dynamic group members on the LDAP server.</p> <p>This parameter and the <code>LDAP_STATIC_GROUP_MEMBERS</code> parameter are used to determine the attribute that indicates the members of the group.</p> <p>In Active Directory, these would map to the same attribute.</p>	
LDAP_GROUP_NAME	Group name on the LDAP server.	
LDAP_GROUP_OBJECTCLASS	Objectclass attribute for a group on the LDAP server.	

Table A-3. *LdapAttribute.conf* parameters (page 2 of 2)

Parameter (*Required)	Definition, Description, Usage	Default, Valid Values, Example
LDAP_LOGON_ID	Logon ID on the LDAP server. This parameter needs to be different from the LDAP_USER_ID if the LOGON_METHOD = LOGON_ID in the <code>server.conf</code> file.	
LDAP_MODIFY_TIMESTAMP	Attribute that keeps track of the last modified time for an object on the LDAP server.	
LDAP_OBJECTCLASS	Objectclass attribute on the LDAP server. Usually maps to "objectclass."	
LDAP_ORG_UNIT_NAME	Organizational unit on the LDAP server.	
LDAP_STATIC_GROUP_MEMBERS	Static group members on the LDAP server. This parameter and the LDAP_DYNAMIC_GROUP_MEMBERS parameter are used to determine the attribute that indicates the members of the group. In Active Directory, these would map to the same attribute.	
LDAP_TIME_FORMAT	Attribute that keeps track of the time format used by the LDAP server.	Format (Active Directory): yyyyMMddHHmmss '0Z'
LDAP_USER_ID	User ID on the LDAP server. This parameter is used to resolve the users on the LDAP server. It is mapped to a unique attribute that determines a user.	
LDAP_USER_OBJECTCLASS	Objectclass attribute for a user on the LDAP server.	Default: person
USER_DATA	Indicates that USER_DATA should be mapped. Always specify this attribute.	
VISIBLE_USER_DATA	Indicates that VISIBLE_USER_DATA should be mapped. Always specify this attribute.	



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.

Server Directory Structure and Server Tools

In This Chapter:

- *Directory Structure Overview*
- *mitg600/system Directory*
 - *CreateKintanaUser.sql*
 - *CreateRMLUser.sql*
- *ITG_Home/bin Directory*
 - *kBuildStats.sh*
 - *kCancelStop.sh*
 - *kConvertToLog4j.sh*
 - *kDeploy.sh*
 - *kDeploy.sh*
 - *kEncrypt.sh*
 - *kGenPeriods.sh*
 - *kGenPeriods.sh*
 - *kGenTimeMgmtPeriods.sh*
 - *kJSPCompiler.sh*
 - *kKeygen.sh*
 - *kMigratorExtract.sh*
 - *kMigratorImport.sh*
 - *kRunServerAdminReport.sh*
 - *kStart.sh*
 - *kStatus.sh*
 - *kStop.sh*
 - *kUpdateHtml.sh*

- *kUpdateHtml.sh*
 - *kWall.sh*
 - *setServerMode.sh*
 - *ITG_Home/docs Directory*
 - *ITG_Home/integration Directory*
 - *ITG_Home/logs Directory*
 - *ITG_Home/reports Directory*
 - *ITG_Home/server Directory*
 - *ITG_Home/sql Directory*
 - *ITG_Home/transfers Directory*
 - *Other Directories*
-

Directory Structure Overview

The `mitg600` directory (the installation bundle directory) contains two subdirectories that relate to the Oracle database schemas: `mitg600/sys` and `mitg600/system`.

`ITG_Home`, the directory where Mercury IT Governance Center is installed, contains a number of subdirectories (for example, `bin`, `docs`, `logs`, and `reports`) that contain server- and system-oriented information and administrative tools that perform tasks like starting, stopping, and reporting on the Mercury IT Governance Server or system.

This chapter lists and describes these directories, and the scripts and other tools located in them.

mitg600/system Directory

This section describes two scripts:

- `CreateKintanaUser.sql`
- `CreateRMLUser.sql`

CreateKintanaUser.sql

Table B-1 lists and describes the `CreateKintanaUser.sql` script variables.

Table B-1. CreateKintanaUser.sql variables

Variable	Description
<i>ITG_User</i>	Username of the new database schema.
<i>ITG_Password</i>	Password of the new database schema.
<i>Data_Tablespace</i>	Tablespace used to store Mercury IT Governance Center tables.
<i>Index_Tablespace</i>	Tablespace used to store Mercury IT Governance Center indexes.
<i>Temp_Tablespace</i>	Temporary tablespace.
<i>Clob_Tablespace</i>	Tablespace used to store large data (CLOB).

CreateRMLUser.sql

Table B-2 lists and describes the `CreateRMLUser.sql` script variables.

Table B-2. CreateRMLUser.sql variables

Variable	Description
<i>Rml_User</i>	Username of the new RML database schema.
<i>Rml_Password</i>	Password of the new RML database schema.
<i>Rml_data_tablespace</i>	Tablespace used to store Mercury IT Governance Center database tables.
<i>Rml_temp_tablepace</i>	Temporary tablespace.

ITG_Home/bin Directory

The `bin` subdirectory of `ITG_Home` contains all of the scripts necessary to configure and administer the server. These scripts are discussed in the following sections.

kBuildStats.sh

The `kBuildStats.sh` script collects system statistics. Usage of this script is discussed in *Collecting Additional Statistics Using Scripts* on page 180.

kCancelStop.sh

If the server has been scheduled to stop by a command such as `kStop.sh -delay 10` (which stops the server in 10 minutes), the stop request can be cancelled by running this script. Authentication may be required to do this, which works in the same way as for `kStop.sh`. Use the `-user username` flag.

kConvertToLog4j.sh

Converts the JDBC log, Web log, or server log to the log4j XML format.

You can view logs in this format with a tool like Chainsaw (which is a GUI-based log viewer available from <http://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 redirect them to a result log:

```
sh kConvertToLog4j.sh -serverLog serverLog.txt -jdbcLog  
jdbc.kintana.log -webLogiisLog.txt
```

For More Information

To get more information about usage type:

```
sh kConvertToLog4j.sh -help
```

kConfig.sh

Sets the server mode.

For More Information

For more information about server mode, see [Setting Server Modes on page 80](#).

kDeploy.sh

`kDeploy.sh` is a command-line tool used to install Mercury Change Management Extensions and Mercury IT Governance Center product patches. This software is distributed as a deployment, which is a software bundle containing files, and are in the format:

```
mitg-itg_server_version-deployment_id.jar
```

where `itg_server_version` is the Mercury IT Governance Server version number and `deployment_id` (a variable-length name) is a unique identifier for the deployment.

For example, to install product patch PL9:

1. Untar the deployment jar file.
2. Run run the script; for example (to apply the PL9 patch):

```
sh kDeploy.sh -i PL9
```

The key command-line parameters for `kDeploy.sh` are shown in [Table B-3 on page 284](#). For a complete list of parameters, issue the command:

```
sh kDeploy.sh -h
```

Table B-3. Selected command-line parameters for `kDeploy.sh`

Parameter	Description
-i	Installs deployments. For example, the command to install Mercury Change Management Extension for Databases would be: <pre>sh kDeploy.sh -i OracleApps</pre> The command to install a Mercury IT Governance Center patch (PL14) might be: <pre>sh kDeploy.sh -i PL14</pre>
-l	Lists the deployments that are installed in an instance. For example: <pre>sh kDeploy.sh -l</pre> results in: <pre>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)</pre>
-D	Searches for bundles in a given directory. For example, to search for a file in the <code>DIR</code> directory: <pre>sh kDeploy.sh -D DIR</pre>
-h	Provides help for <code>kDeploy.sh</code> . 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 unneeded deployment files.
-skip -database	Specifies that database changes should not be applied if they already exist.

kEncrypt.sh

Utility for encrypting strings that can be inserted into `server.conf`.

kGenPeriods.sh

Generates the period information and seeds the data in the database tables that contain this information: `knta_periods` and `knta_period_groups`. It takes in the start year and end year parameters, then generates the monthly periods and period groups from the start year till the end year. If any of the periods between the specified years already exist, then these periods and period groups will not be regenerated. Only 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 — will be created.

kGenTimeMgmtPeriods.sh

(Used in Mercury Time Management™) Seeds data in the `KTMG_PERIODS` table. This script takes the number of periods to be seeded and the start date from which the periods need to be seeded.

Usage:

```
kGenTimePeriods.sh num start_date
```

The `num` value is the number of time periods required. The `start_date` is the start date from which the periods will be seeded. For a new installation, running this script is optional. Running `kGenTimePeriods.sh` with no arguments will default the number of `time_periods` to 24.

kJSPCompiler.sh

Precompiles all JSP files in the Mercury IT Governance Server. Precompiling JSP files should result in performance improvements in the standard interface.

kKeygen.sh

Generates new security keys.

kMigratorExtract.sh

Used in migrating Mercury IT Governance Center entities.

kMigratorImport.sh

Used in migrating Mercury IT Governance Center entities.

kRunServerAdminReport.sh

Runs diagnostic reports on the Mercury IT Governance Server. Run `kRunServerAdminReport.sh` to view a list of reports from which to choose. This utility provides a summary of how much activity is currently on the system and how many database connections are being made.

kStart.sh

This 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 80](#).

kStatus.sh

Run this script to check the state of the Mercury IT Governance Server at any time. The load value refers to the number of active user sessions.

kStop.sh

Use this script to stop the Mercury IT Governance Server. This script requires some arguments. When stopping the server, it is possible to choose to stop it now with the `-now` flag, or after a delay of a certain number of minutes with the flag `-delay #minutes`. Using the `-delay` option results in a message automatically being sent to all currently-connected Mercury IT Governance Center users suggesting that the server will stop after the specified delay. In addition, this script may require authentication (if the server parameter `REMOTE_ADMIN_REQUIRE_AUTH` has been set to `True`). In this case, the flag `-user username` is also required. For more information on available flags, run `kStop.sh` without any options to show the usage notes.

For More Information

For more information about stopping the server, see [Stopping the Server on page 82](#).

kSupport.sh

Gathers information useful to Mercury Support in diagnosing system problems, and creates a zip file with a timestamp in the `support/zipfiles` directory.

You can use the script to automatically gather:

- Install and upgrade 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

When collecting server logs or JDBC logs, the script concatenates all the files into one `serverLog.txt` file.

You can run `kSupport.sh` in GUI, console, or silent mode. Silent mode will automatically capture 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

Updates the Logon HTML files with the latest configuration. This script is automatically run by the server configuration utility. If any changes are made to `server.conf` by hand, this script should be run to make sure the changes are propagated.

kWall.sh

Sends out a message to all users currently logged on to the Mercury IT Governance Workbench. When this script is run, it prompts for the Mercury IT Governance Center username and password and the desired message text. The message is displayed in a dialog on the monitor screen of anyone who is logged onto Mercury IT Governance Center at that time.

setServerMode.sh

Sets the mode of the Mercury IT Governance Server.

For More Information

For more information about server modes, see [Setting Server Modes on page 80](#).

ITG_Home/docs Directory

The `docs` subdirectory contains all documentation files for Mercury IT Governance Center, in PDF format (you need Adobe Reader to view them). At the time you install or upgrade Mercury Change Management Extensions and Migrators (which is a different installation or upgrade procedure than for Mercury IT Governance Center), the installation or upgrade script also installs the appropriate 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 Directory

The `integration` subdirectory contains several directories used for integration purposes. For example, the `ITG_Home/integration/webserver` directory contains folders for each of the external Web server that can be integrated 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

There are two log directories in the server directory structure. The first log directory is *ITG_Home/logs*. This 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 Change Management packages and Request Management requests. The *number* variable in the directory name corresponds to the ID of the package or request that is being executed.

The other log directory is *ITG_Home/server/kintana/log*. This directory contains all logs generated by the Mercury IT Governance Server. As the server runs, logging messages are generated and written into the file *serverLog.txt*. 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. In addition, the Java servlets used to serve the Web pages generate their own log files, named *servletLog.txt*. The amount of information present 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 it is necessary to obtain more information in the logs, log on to the Workbench as Administrator and set the server debug level to high from the menu **Edit > Server Settings**.

ITG_Home/reports Directory

The *reports* subdirectory contains the HTML files for all reports that have been run through the client.

ITG_Home/server Directory

The `ITG_Home/server` directory contains subdirectories that define the nodes of a Mercury IT Governance Server instance.

For More Information

For more information about nodes in a server cluster, see [Configuring a Server Cluster on page 137](#).

ITG_Home/sql Directory

The `sql` subdirectory contains source code for the built-in Mercury IT Governance Center reports.

ITG_Home/transfers Directory

The `transfers` subdirectory is used as temporary storage for files being transferred between the server and remote computers.

For More Information

For more information about the use of the `transfers` directory in entity migration, see [Basic Parameters on page 216](#).

Other Directories

Other directories contain reference files as indicated by their name, and it is not likely that they need to be accessed.

Symbols

@node directives 138

A

access grants

Ownership Override 224

Sys Admin: Migrate Mercury ITG Objects 219

Sys Admin: Migrate Mercury ITG objects 218

SysAdmin: Server Tools: Execute Admin Tools 155

SysAdmin: Server Tools: Execute SQL Runner 155

SysAdmin: View Server Tools 155

Admin Tools window 156

administration tools

accessing 151

overview 154

AIX platform, running Mercury IT Governance Center on 24

AJP13 communication protocol 24, 29, 33, 34

ALLOW_SAVE_REQUEST_DRAFT parameter 255

Apache Web server 24

application server tier, system architecture 24

architecture overview 22

ATTACHMENT_DIRNAME parameter 197, 199, 200, 255

audience for this document 17

AUTHENTICATION_MODE parameter 255

AUTOCOMPLETE_STATUS_REFRESH_RATE parameter 255

B

backing up

instances 171

the database schema 68

the file system 66

the system before upgrading 66

BACKUP directory 72

BASE_LOG_DIR parameter 138

BASE_PATH parameter 138, 196, 199, 200, 255

BASE_URL parameter 89, 135, 138, 197, 199, 200, 255

batch executions in progress, report providing information about 158

batch file to run the Workbench 113

batches pending execution, report providing information about 158

Best Practices

- installing 42, 110
- upgrading 42
- bin directory 282
- Broker Connection report 158
- Broker In Use Sessions report 158
- Broker Performance report 158

C

- cache, report providing information about 159
- CacheManager Sizes report 159
- CacheManager Statistics report 159
- CATCH_SYSTEM_ERR parameter 272
- CATCH_SYSTEM_OUT parameter 272
- Client Alive report 157
- client environment, report providing information about 158
- Client Font report 158
- Client Property report 158
- client tier, system architecture 23
- Client Timezone report 158
- CLIENT_TIMEOUT parameter 189, 255
- cloning instances 192
- commands, migrating 210
- CONC_LOG_TRANSFER_PROTOCOL parameter 255
- CONC_REQUEST_PASSWORD parameter 170, 256
- CONC_REQUEST_USER parameter 256
- configuration parameters 252
- Configure Server prompt, installation procedure 51
- configuring the server 43
- console mode, installing or upgrading in 59, 74
- content bundles, entity migration 216
- CreateKintanaUser.sql script 56, 68, 202, 281
- CreateRMLUser.sql script 56, 202, 203, 281
- Currency Code prompt
 - installation procedure 51
 - upgrade procedure 70

- custom parameters 85

D

- Data Source migrator 226
- database
 - configuring 90
 - maintaining 169
 - reconfiguring 90
- Database Access Information prompt, installation procedure 50
- database configuration examples 97
- database connection pool 24
- database links, generating 103
- database parameters 90
- database pool connections, report providing information about 158
- database schema 55
 - collecting statistics about 179
 - creating automatically 44
 - exporting before upgrading 68
 - giving grants to 43
 - migrating 201
- DATE_NOTIFICATION_INTERVAL parameter 256
- DAYS_TO_KEEP_APPLET_KEYS parameter 256
- DAYS_TO_KEEP_COMMAND_ROWS parameter 256
- DAYS_TO_KEEP_INTERFACE_ROWS parameter 187, 256
- DAYS_TO_KEEP_LOGON_ATTEMPT_ROWS parameter 171, 187, 256
- DB_BLOCK_SIZE parameter 90
- DB_CACHE_SIZE parameter 91
- DB_CONNECTION_STRING parameter 86, 256
- DB_LOGIN_TIMEOUT parameter 189, 256
- DB_PASSWORD parameter 88, 170, 256
- DB_USERNAME parameter 88, 256
- DBMS_PROFILER package (Oracle) 163

-
- dbms_stats package 180
 - DBMS_TRACE package (Oracle) 164
 - DEFAULT_COMMAND_TIMEOUT parameter 189, 257
 - DEFAULT_SERVER_LOGGING_LEVEL parameter 164, 165, 272
 - DEFAULT_USER_DEBUG_LEVEL parameter 165, 188, 273
 - DEFAULT_USER_LOGGING_LEVEL parameter 163
 - Destination Password field, entity migration 218
 - directories
 - BACKUP 72
 - bin 282
 - docs 289
 - integration 289
 - ITG_Home 48, 65
 - logs 171, 290
 - mitg600/sys 280
 - mitg600/system 280, 281
 - PKG_number 290
 - reports 290
 - REQ_number 290
 - server 291
 - specifying path names 253
 - sql 291
 - TO_BE_CONFIGURED 73
 - transfer 291
 - upgrade_600 74
 - disabled mode, Mercury IT Governance Server 80
 - docs directory 289
 - document management module, migrating 194
 - documents
 - prerequisite 17
 - related 18
 - Documentum Content Server 40
 - downloading
 - the installation files 51
 - the upgrade files 70
 - E**
 - EMAIL_NOTIFICATION_CHECK_INTERVAL parameter 189, 257
 - EMAIL_NOTIFICATION_SENDER parameter 257
 - EMC/Documentum Content Server 40
 - Enable Profiler setting, Server Setting window 163
 - ENABLE_CONSOLE_LOGGING parameter 273
 - ENABLE_DB_SESSION_TRACKING parameter 188, 257
 - ENABLE_EXCEPTION_ENGINE parameter 257
 - ENABLE_INTERFACE_CLEANUP parameter 187
 - ENABLE_JDBC_LOGGING parameter 188, 257
 - ENABLE_LOGGING parameter 188
 - ENABLE_PENDING_PROJECT_CHANGE parameter 189, 258
 - ENABLE_SQL_TRACE parameter 188, 258
 - ENABLE_STATISTICS_CALCULATION parameter 179, 258
 - ENABLE_TIMESTAMP_LOGGING parameter 188
 - ENABLE_WEB_ACCESS_LOG parameter 166
 - ENABLE_WEB_ACCESS_LOGGING parameter 273
 - entities, migrating 210
 - entity migrators
 - defining 215
 - object types 226
 - events, report providing information about 159
 - EXCEPTION_ENGINE_INTERVAL parameter 189, 258
 - EXCEPTION_ENGINE_WAKE_UP_CHECK_FREQUENCY parameter 189, 258
 - EXCEPTION_ENGINE_WAKE_UP_TIME
-

parameter 189, 258
exe_debug_log.txt file 168
Execution Dispatcher Manager report 158
Execution Dispatcher Pending Batch report 158
Execution Dispatcher Pending Group report 158
execution engine 24
EXECUTION_DEBUGGING parameter 188
executions
 viewing interrupted 152
 viewing running 152
exp command 202, 207
Extensions, installing 41
EXTERNAL_WEB_PORT parameter 118, 120, 135, 139

F

file path names, separator characters in 84
FILE_RECHECK_INTERVAL parameter 273
files
 httpd.conf 133
 install.exe 57
 kintana.bat 113
 knta_classes.jar 112
 libraries.jar 112
 mitg-600-install.zip 57, 59
 mitg-600-install-zip 51
 mitg-600-upgrade.zip 64, 70
 oracle-jdbc.jar 112
 private_key.txt 88
 public_key.txt 88
 serverLog.txt 206
 upgrade.exe 74
 workers property 119
fonts supported in the installation environment, report providing information about 158
forward slashes in directory path names 253
FTP server, configuring 61, 75

G

GLOBAL_NAMES parameter 91
grants to the database schema 43
GrantSysPrivs.sql script 44, 68, 202, 204
graphic mode, installing or upgrading in 44

H

Holiday Schedule prompt
 installation procedure 51
 upgrade procedure 69
HOURS_TO_KEEP_DEBUG_MESSAGE_ROWS parameter 187, 259
HOURS_TO_KEEP_MESSAGE_ROWS parameter 171
HP-UX platform, running Mercury IT Governance Center on 24
HTTP communication protocol 23, 27, 28, 29, 33, 36
HTTP listener 30
HTTP_PORT parameter 139, 145, 259
httpd.conf file 133
HTTPS communication protocol 23, 28, 29, 33, 34, 36

I

I18N_CARAT_DIRECTION parameter 259
I18N_ENCODING parameter 259
I18N_LAYOUT_DIRECTION parameter 259
I18N_REPORT_HTML_CHARSET parameter 259
I18N_REPORTS_ENCODING parameter 259
I18N_SECTION_DIRECTION parameter 259
IBM AIX platform, running Mercury IT Governance Center on 24
IIS Web server 24
imp command 203, 208
import behavior fields, entity migration 217
install.exe file 57
install.sh script 59

-
- INSTALLATION_LOCALE** parameter 260
- installing
- Best Practices 42, 110
 - configuring the FTP server 61
 - creating a Mercury IT Governance Center user 53
 - creating the database schemas 55
 - document management 40
 - downloading the files 51
 - Extensions 41, 111
 - GL Migrator 41
 - license keys used in 41
 - Object Migrator 41
 - overview of steps to follow 38
 - patches 111
 - preparing for 48
 - running the installation script 57
 - SDK 53
 - system requirements 41
 - unzipping the files 52
 - verifying that the **JAVA_HOME** parameter is set 52
 - verifying the installation 62
- instances
- backing up 171
 - migrating 192
- integration directory 289
- Internationalization field, entity migration 219
- interrupted executions, viewing 152
- iPlanet Web server 24
- ITG_Home** directory 48, 65
- J**
- J2EE application server 22, 23, 24
- Java plug-in 23
- configuring 105
 - setting the correct version of 105
- JAVA_HOME** parameter 48, 52, 64, 71
- JAVA_HOME** prompt
- installation procedure 49
 - upgrade procedure 69
- JAVA_PLUGIN_PATH_IE** parameter 106
- JAVA_PLUGIN_PATH_NS** parameter 106
- JAVA_PLUGIN_VERSION** parameter 106
- JBoss Application Server 24
- JDBC communication protocol 24, 27, 28, 30, 33
- JDBC_DEBUGGING** parameter 188
- JDBC_URL** parameter 88, 260
- JSP files
- compiling 43
 - Mercury IT Governance Center standard interface 23
- JSP_RECOMPILE_ENABLED** parameter 260
- JVM
- problems, troubleshooting 106
 - running in interpreted mode 183
- JVM Memory report 159
- K**
- kBuildStats.sh script 180, 282
- kCancelStop.sh script 282
- kConfig.sh script 80, 142, 170, 196, 197, 198, 205, 283
- kConvertToLog4j.sh script 282
- kDeploy.sh script 110, 111, 283
- kEncrypt.sh script 284
- KEY_STORE_FILE** parameter 87
- KEY_STORE_PASSWORD** parameter 87
- keytool application 87
- kGenPeriods.sh script 285
- kGenTimeMgmtPeriods.sh script 285
- Kintana RMI Detail report 158
- kintana.bat file 113
- kintana.sh script 114
- KINTANA_LDAP_ID** parameter 116, 260
- KINTANA_LDAP_PASSWORD** parameter 116, 261
- KINTANA_SERVER** parameter 212
- KINTANA_SERVER_NAME** parameter 73,
-

123, 137, 138, 142, 261
KINTANA_SESSION_TIMEOUT parameter 261
kJSPCompiler.sh script 285
kKeygen.sh script 88, 285
kMigratorExtract.sh script 285
kMigratorImport.sh script 285
knta_classes.jar file 112
KNTA_DEBUG_MESSAGES table 171
KNTA_LOGON_ATTEMPTS table 171
KNTA_USERS_INT parameter 275
KRSC_ORG_UNITS_INT table 277
kRunServerAdminReport.sh script 286
kStart.sh script 81, 146, 183, 199, 254, 286
kStatus.sh script 147, 286
kStop.sh script 146, 196, 200, 254, 286
kSupport.sh script 287
kUpdateHtml.sh script 141, 166, 200, 254, 287
kWall.sh script 288

L

LDAP server, integrating with 116
LDAP_BASE_DN parameter 261
LDAP_DYNAMIC_GROUP_MEMBERS parameter 275
LDAP_GROUP_NAME parameter 275
LDAP_GROUP_OBJECTCLASS parameter 275
LDAP_GROUP_RECURSION_LIMIT parameter 261
LDAP_KEYSTORE parameter 117
LDAP_KEYSTORE_PASSWORD parameter 117
LDAP_LOGON_ID parameter 276
LDAP_MODIFY_TIMESTAMP parameter 276
LDAP_OBJECTCLASS parameter 276
LDAP_ORG_UNIT_NAME parameter 276
LDAP_SSL_PORT parameter 117, 261

LDAP_STATIC_GROUP_MEMBERS parameter 276
LDAP_TIME_FORMAT parameter 276
LDAP_URL parameter 116, 261
LDAP_USER_ID parameter 276
LDAP_USER_OBJECTCLASS parameter 276
LdapAttribute.conf parameters 275
libraries.jar file 112
License Configuration File prompt
 installation procedure 49
 upgrade procedure 69
license keys 41
Linux platform, running Mercury IT
 Governance Center on 24
load balancing 33
LOCAL_IP parameter 262
log files 165
 execution 168
 report 167
 server 165
 temporary 168
LOG_BUFFER parameter 92
LOG_LAYOUT parameter 273
logging.conf parameters 272
LOGON_TRIES_INTERVAL parameter 262
logs directory 171, 290

M

maintaining the system 150
MAX_BACKUP_INDEX parameter 273
MAX_COMMIT_PROPAGATION_DELAY parameter 92
MAX_DB_CONNECTION_IDLE_TIME parameter 190, 262
MAX_DB_CONNECTION_LIFE_TIME parameter 190, 262
MAX_DB_CONNECTIONS parameter 190, 263
MAX_EXECUTION_MANAGERS

parameter 182, 189, 263
MAX_LOGON_TRIES parameter 263
MAX_RELEASE_EXECUTION_ MANAGERS parameter 263
MAX_STATEMENT_CACHE_SIZE parameter 190, 263
MAX_WORKER_THREADS parameter 181, 189, 263
Mercury ITG Schema prompt
 installation procedure 50
 upgrade procedure 69
Microsoft IIS Web server 24
Microsoft Project data, updating after upgrade 76
Microsoft Windows platform, running Mercury IT Governance Center on 24
migrating
 entities 210
 instances 192
 the database schema 201
 the document management module 194
 the server 196
Migrator action field 215
migrators
 Data Source 226
 Module 227
 Object Type 228
 Overview Page Section 230
 Portlet 231
 Project Template 232
 Report Type 234
 Request Header Type 236
 Request Type 238
 Special Command 241
 User Data Context 242
 Validation 243
 Workflow 245
mitg600/sys directory 280
mitg600/system directory 280, 281
mitg-600-install.zip file 51, 57, 59
mitg-600-upgrade.zip file 64, 70

Module migrator 227
MULTICAST_CLUSTER_NAME parameter 264
MULTICAST_DEBUG parameter 264
MULTICAST_IP parameter 264
MULTICAST_LEASE_MILLIS parameter 264
MULTICAST_PORT parameter 264

N

Name Display Format, upgrade procedure 69
NCSA Common format, internal HTTP logging 166
Netegrity SiteMinder, integration with 33
NON_DOMAIN_FTP_SERVICES parameter 86
normal mode, Mercury IT Governance Server 80
notification engine 24
NOTIFICATIONS_CLEANUP_PERIOD parameter 187, 264

O

Object Migrator, running on a single database schema 44
Object Type migrator 228
object types
 entity migrator 226
 migrating 210
OPEN_CURSORS parameter 92
OPEN_LINKS parameter 93
OPTIMIZER_MODE parameter 93
Oracle
 database 22
 RAC (Real Application Cluster)
 configuration 25
 stored procedures 25
ORACLE_APPS_ENABLED parameter 264
ORACLE_HOME parameter 138, 196, 199, 200, 264

ORACLE_HOME prompt, installation procedure **49**
oracle-jdbc.jar file **112**
ORG_UNIT_NAME parameter **277**
Overview Page Section migrator **230**
ownership groups, and entity migration **224**
Ownership Override access grant **224**

P

PACKAGE_LOG_DIR parameter **139**
parameters
 cleanup **187**
 configuration **252**
 custom **85**
 database connection **190**
 debug **187**
 LdapAttribute.conf **275**
 logging **190**
 logging.conf **272**
 scheduler **189**
 server.conf **254**
 services **189**
 special **85**
 thread **189**
 timeout **189**
parameters in effect for active servers, report providing information about **157**
PARENT_ORG_UNIT_NAME parameter **277**
password security, generating **88**
PASSWORD_EXPIRATION_DAYS parameter **264**
PASSWORD_REUSE_RESTRICTION_DAYS parameter **265**
passwords (database schema), changing **169**
patches, installing **111**
path names, directories **253**
PENDING_COST_EV_UPDATE_SERVICE_DELAY parameter **265**
PENDING_COST_EV_UPDATE_SERVICE_ENABLED parameter **265**

PENDING_PROJECT_CHANGE_INTERVAL parameter **265**
performance
 improving **173, 183**
 improving during advanced searches **186**
 improving throughput **185**
 JVM **183**
 tuning server cluster **184**
performance problems
 identifying **174**
 isolating **174**
 troubleshooting **181**
PGA_AGGREGATE_TARGET parameter **93**
pinging
 the database **161**
 the server **161**
PKG_number directory **290**
PL/SQL packages **25**
Portlet migrator **231**
PORTLET_EXEC_TIMEOUT parameter **189, 265**
portlets, migrating **210**
Preview Import field, entity migration **217**
Primary Object Migrator Host **201**
Primary Object Migrator Host definition **204**
private_key.txt file **88**
PROCESSES parameter **94**
Project Template migrator **232**
project templates, migrating **210**
public_key.txt file **88**

R

RAC (Real Application Cluster) configuration **25**
RecompileInvalid.sql script **204**
Recreate_customer_link.sql script **204**
Recreate_db_links.sql script **204**
Red Hat Linux platform, running Mercury IT Governance Center on **24**
Region Name prompt

-
- installation procedure 51
 - upgrade procedure 70
 - REMOTE_ADMIN_REQUIRE_AUTH parameter 265, 286
 - Report Type migrator 234
 - report types, migrating 210
 - REPORT_DIR parameter 139
 - Reporting Meta Layer Schema prompt
 - installation procedure 50
 - REPORTING_STATUS_REFRESH_RATE parameter 189, 265
 - reports 156
 - Broker Connection 158
 - Broker In Use Sessions 158
 - Broker Performance 158
 - CacheManager Sizes 159
 - CacheManager Statistics 159
 - Client Alive 157
 - Client Font 158
 - Client Property 158
 - Client Timezone 158
 - Execution Dispatcher Manager 158
 - Execution Dispatcher Pending Batch 158
 - Execution Dispatcher Pending Group 158
 - JVM memory 159
 - Kintana RMI Detail 158
 - running, viewing and cancelling 151
 - Server Cache Status 159
 - Server Configuration 157
 - Server Event Listener 159
 - Server Logon 157
 - Server Status 157
 - Server Thread 157
 - Service Controller 158, 187
 - reports directory 290
 - REQ_number directory 290
 - Request Header Type migrator 236
 - request header types, migrating 210
 - Request Type migrator 238
 - request types, migrating 210
 - REQUEST_LOG_DIR parameter 139
 - resource information, updating users with after upgrade 76
 - RESOURCE_CACHE_SIZE parameter 266
 - RESTRICT_BYPASS_EXECUTION_TO_MANAGERS parameter 266
 - restricted mode, Mercury IT Governance Server 80
 - RMI communication protocol 23, 28, 30, 33, 36
 - RMI connection threads, report providing information about 158
 - RMI_DEBUGGING parameter 165, 166
 - RMI_URL parameter 87, 139, 197, 199, 200, 266
 - RML_PASSWORD parameter 170, 266
 - RML_USERNAME parameter 266
 - RMLSetupInITGSchema.sql script 202
 - RMLSetupInRMLSchema.sql script 202
 - Rollback Segment prompt, upgrade procedure 69
 - ROTATE_LOG_SIZE parameter 165, 274
 - running executions, viewing 152
 - RunServerAdminReport.sh script 159
- ## S
- SCHEDULER_INTERVAL parameter 190, 266
 - scheduling engine 24
 - SCPCLIENT_TIMEOUT parameter 266
 - scripts
 - CreateKintanaUser.sql 56, 68, 202, 281
 - CreateRMLUser.sql 56, 202, 203, 281
 - GrantSysPrivs.sql 44, 68, 202, 204
 - install.sh 59
 - kBuildStats.sh 180, 282
 - kCancelStop.sh 282
 - kConfig.sh 80, 142, 170, 196, 197, 198, 205, 283
 - kConvertToLog4j.sh 282
 - kDeploy.sh 110, 111, 283
 - kEncrypt.sh 284
-

- kGenPeriods.sh 285
- kGenTimeMgmtperiods.sh 285
- kintana.sh 114
- kJSPCompiler.sh 285
- kKeygen.sh 88, 285
- kMigratorExtract.sh 285
- kMigratorImport.sh 285
- kRunServerAdminReport.sh 286
- kStart.sh 81, 146, 183, 199, 254, 286
- kStatus.sh 147, 286
- kStop.sh 146, 196, 200, 254, 286
- kSupport.sh 287
- kUpdate.Html.sh 141
- kUpdateHtml.sh 166, 200, 254, 287
- kWall.sh 288
- RecompileInvalid.sql 204
- Recreate_customer_link.sql 204
- Recreate_db_links.sql 204
- RMLSetupInITGSchema.sql 202
- RMLSetupInRMLSchema.sql 202
- RunServerAdminReport.sh 159
- setServerMode.sh 80, 288
- upgrade.sh 74
- SDK (Software Developer Kit) 53
- SEARCH_TIMEOUT parameter 189, 267
- SECURE_RMI parameter 267
- security, generating password 88
- separator characters in file paths 84
- server
 - configuring 83
 - directory 291
 - log files 165, 167, 168
 - migrating 196
 - modes 80
 - reconfiguring 83
 - starting 80
 - stopping 80
 - stopping and restarting for maintenance 169
 - verifying client access to 89
- Server Cache Status report 159
- server clusters
 - configuring 31, 137
 - starting and stopping 146
- Server Configuration report 157
- Server Event Listener report 159
- Server Logon report 157
- server reports 156
- Server Status report 157
- Server Thread report 157
- Server Tools window 154
 - access grants required to use 155
 - accessing 154
- server.conf parameters 254
- SERVER_DEBUG_LEVEL parameter 168, 188, 274
- SERVER_ENV_NAME parameter 223, 267
- SERVER_LOG_DIR parameter 138
- SERVER_MODE parameter 267
- SERVER_NAME parameter 197, 199, 200, 267
- SERVER_TYPE_CODE parameter 200, 267
- serverLog.txt file 165, 206
- serverLog_timestamp.txt file 165
- Service Controller report 158, 187
- services enabled for the server, report providing information about 158
- sessions active on the server, report providing information about 157
- setServerMode.sh script 80, 288
- SHARED_POOL_RESERVED_SIZE parameter 95
- SHARED_POOL_SIZE parameter 95
- SHOW_BASE_URL_ON_NOTIFICATION parameter 267
- single sign-on integration 33
- SiteMinder, integration with 33
- SMTP_SERVER parameter 267
- Software Developer Kit (SDK) 53
- Solaris platform, running Mercury IT Governance Center on 24

-
- SORT_AREA_SIZE parameter 93
 - Source Password field, entity migration 218
 - Special Command migrator 241
 - special commands, migrating 210
 - Special Ops solutions, backing up before upgrading 64
 - special parameters 85
 - sql directory 291
 - SQL Runner window 160
 - SQL statements, running using SQL Runner 161
 - SQL*PLUS prompt, installation procedure 49
 - SQL*Plus utility 49
 - SQLPLUS parameter 267
 - SRMI communication protocol 23, 28, 30, 36
 - SRMI, enabling 87
 - SSL accelerator, integrating with 34
 - standard interface, Mercury IT Governance Center 23
 - starting
 - servers in a cluster 146
 - the server 80
 - statistics, setting the database to gather 179
 - STATS_CALC_DAY_OF_WEEK parameter 179, 268
 - STATS_CALC_INTERVAL parameter 179
 - STATS_CALC_WAKE_UP_TIME parameter 179, 268
 - STATS_CALC_WEEK_INTERVAL parameter 179, 268
 - status of the server, report providing information about 157
 - stopping
 - servers in a cluster 146
 - the server 80
 - Sun Java plug-in 23
 - Sun Java System Web Server 24
 - Sun ONE Web Server 24
 - Sun Solaris platform, running Mercury IT Governance Center on 24
 - swing mode, installing or upgrading in 44, 59, 74
 - SYNC_EXEC_INIT_WAIT_TIME parameter 268
 - SYNC_EXEC_MAX_POLL_TRIES parameter 268
 - SYNC_EXEC_POLL_INTERVAL parameter 268
 - Sys Admin
 - Migrate Mercury ITG Objects access grant 219
 - Migrate Mercury ITG objects access grant 218
 - Server Tools: Execute Admin Tools access grant 155
 - Server Tools: Execute SQL Runner access grant 155
 - View Server Tools access grant 155
 - System Calendar prompt
 - installation procedure 51
 - upgrade procedure 69
 - system overview 22
 - System Password prompt
 - installation procedure 49
 - upgrade procedure 69
 - system requirements 41
- ## T
- tables
 - KRSC_ORG_UNITS_INT 277
 - tables (temporary), maintaining 170
 - tablespaces, naming during the installation procedure 50
 - telnet server 42
 - TEMP_DIR parameter 86
 - temporary tables, maintaining 170
 - THREAD_POOL_MAX_THREADS parameter 190, 268
 - THREAD_POOL_MIN_THREADS parameter 190, 269
 - threads running in the server, report providing
-

information about 157
throughput, improving 185
time zones recognized by the client, report providing information about 158
TIME_ZONE parameter 269
TIMED_STATISTICS parameter 96
TO_BE_CONFIGURED directory 73
Trace Call Stack setting, Server Setting window 164
Trace Exception setting, Server Setting window 164
Trace SQL setting, Server Setting window 164
transfer directory 291
TRANSFER_PATH parameter 139
TURN_ON_IF_TIMEOUT_REAPER parameter 190
TURN_ON_NOTIFICATIONS parameter 190, 269
TURN_ON_SCHEDULER parameter 190, 269
TURN_ON_WF_TIMEOUT_REAPER parameter 269

U

UNIX emulator 42
upgrade.exe file 74
upgrade.sh script 74
upgrade_600 directory 74
upgrading

- backing up Special Ops solutions before 64
- backing up the file system before 66
- backing up the system before 66
- Best Practices 42
- cleaning up the server directory 72
- configuring the FTP server 75
- downloading the files 70
- exporting the database schema before 68
- Extensions 111
- license keys used in 41
- overview of steps to follow 39

preparing for 64
procedure summary 66
supported upgrades 40
system requirements 41
unzipping the files 70
verifying 75

URL for Mercury IT Governance Center 89
User Data Context migrator 242
user data contexts, migrating 210
USER_DATA parameter 276
USER_DEBUG_LEVEL parameter 168
USER_PASSWORD_MAX_LENGTH parameter 269
USER_PASSWORD_MIN_DIGITS parameter 270
USER_PASSWORD_MIN_LENGTH parameter 270
USER_PASSWORD_MIN_SPECIAL parameter 270
users logged on to the server, report providing information about 157

V

v_\$session, granting select privileges to 72, 103
Validation migrator 243
validations, migrating 211
VISIBLE_USER_DATA parameter 276
VISUALIZATION_EXEC_TIMEOUT parameter 270

W

Web browser, setting the default 115
Web port 118
Web servers

- Apache 24
- configuring 118, 124
- integrating with the Mercury IT Governance Server 135
- iPlanet 24

- Microsoft IIS **24**
- Sun Java System **24**
- Sun ONE **24**
- WEB_SESSION_TIMEOUT parameter **189**
- WF_SCHEDULED_TASK_INTERVAL parameter **190, 270**
- WF_SCHEDULED_TASK_PRIORITY parameter **190, 270**
- WF_TIMEOUT_REAPER_INTERVAL parameter **190, 270**
- Windows platform, running Mercury IT Governance Center on **24**
- WORK_ITEM_BREAKDOWN_SERVICE_DELAY parameter **270**
- WORK_ITEM_BREAKDOWN_SERVICE_ENABLED parameter **270**
- WORK_ITEM_UPDATE_SERVICE_DELAY parameter **271**
- WORK_ITEM_UPDATE_SERVICE_ENABLED parameter **271**
- WORKAREA_SIZE_POLICY parameter **96**
- Workbench
 - batch file to run **113**
 - configuring as a Java application **112**
 - interface, configuring **105**
 - launching **106**
- WORKBENCH_SESSION_TIMEOUT parameter **189**
- worker.list parameter **120**
- workers property file **119**
- workers.properties file **119**
- workers2.properties file **121**
- workflow engine **24**
- Workflow migrator **245**
- workflows
 - deprecating **248**
 - migrating **211**

