Mercury IT Governance Center™
Mercury Change Management Extension for Siebel® eBusiness Applications™ Guide

Version: 6.0.0

MERCURY™

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

Table of Contents

List of Figures

List of Tables

In This Chapter:

- *About This Document*
- *Who Should Read This Document*
- *Prerequisite Documents*
- *Related Documents*
- *Overview of the Extension*
  - *General Overview of Extension for Siebel eBusiness Applications*
  - *What's New in Release 6.0.0*
  - *Prerequisite Knowledge and Experience*

About This Document

This document provides information about installing or upgrading to release 6.0.0 of Mercury Change Management Extension for Siebel eBusiness Applications™ (often referred to hereafter as "the Extension"), and it provides conceptual, procedural, and reference information about the product.

You can access this document from the Mercury IT Governance Center™ Documentation home page and elsewhere, as described in the *Guide to Documentation*.

This document is organized as follows:

● Chapter 1, *Introduction,* on page 11 (this chapter) describes who should read this document, includes information about prerequisite and related documents, and provides an overview of the Extension for this release.

● Chapter 2, *Installing or Upgrading the Extension,* on page 17 provides overview and detailed information about installing and upgrading the Extension.

● Chapter 3, *Configuring the Extension,* on page 27 provides information related to defining the Extension environments and an overview of Siebel entity migration.

● Chapter 4, *Extension Entities,* on page 35 provides information about the object types, request types, workflows, report types, special commands, validations, and tokens that are unique to the Extension.

Who Should Read This Document

This document is intended for the following audiences:

● People responsible for installing, upgrading, configuring, and maintaining the system and database environments where Mercury Change Management Extension for Siebel eBusiness Applications resides, and setting up and maintaining the Mercury IT Governance schema

● People responsible for maintaining access and security for, or supporting use of, Mercury Change Management Extension for Siebel eBusiness Applications

For More Information

For more information about audience types as defined for Mercury IT Governance Center, see the *Guide to Documentation*.

Prerequisite Documents

You should be familiar with the information in the following documents:

- *What's New in Release 6.0*

- *Release Notes*

- *Key Concepts*

- *Getting Started*

- *System Administration Guide and Reference*

- *Mercury Change Management: Configuring a Deployment System*

- *Mercury Change Management User's Guide*

- *Security Model Guide and Reference*

Before performing an installation or upgrade, check the latest version of the *Release Notes* for release 6.0 on the Mercury IT Governance Download Center for any last-minute information or issues you need to understand.

Related Documents

See the *Guide to Documentation* for:

- Descriptions of all of the Mercury documents for Mercury IT Governance Center and Mercury Change Management Extensions, and how to access each document

- Definitions of audience types, reflecting the roles played by various personnel in using Mercury IT Governance Center and Mercury Change Management Extensions

- Descriptions of typographical conventions used in the documentation set

- Additional resources for support in using Mercury IT Governance Center and Mercury Change Management Extensions

Overview of the Extension

Mercury offers prepackaged change management products to automate migrations, upgrades, and other changes in a variety of leading database, system, and application environments. These products are called Mercury Change Management Extensions.

Extensions are installed on the Mercury IT Governance Server.

General Overview of Extension for Siebel eBusiness Applications

Mercury Change Management Extension for Siebel eBusiness Applications helps to automate change management in the Siebel environment.

The Extension enhances the functionality of Mercury Change Management™ by providing predefined entities that are unique to Siebel eBusiness Applications environments, such as specialized object types, request types, workflows, report types, commands, and tokens.

This manual documents release 6.0.0 of Mercury Change Management Extension for Siebel eBusiness Applications.

Before using Mercury IT Governance Center release 6.0 or the Extension at release 6.0.0 for productive work, you must do one of the following:

- Install Mercury IT Governance Center release 6.0, then install Extension release 6.0.0

- Upgrade Mercury IT Governance Center to release 6.0, then upgrade the Extension to release 6.0.0

What's New in Release 6.0.0

Release 6.0.0 of the Extension introduces the Extension functionality described in the following sections.

For More Information

For more information about Mercury IT Governance Center release 6.0, see the *What's New in Release 6.0* document.

For descriptions of supported upgrade paths, see *Installing or Upgrading to Release 6.0.0* on page 19. For general information about the impacts of upgrading from any release to release 6.0.0, see *General Upgrade Impacts and Guidelines* on page 19. For specific information about the impacts of upgrading from release 5.0.0, see *Impacts of Upgrading from Release 5.0.0* on page 20. There are no specific impacts of upgrading from release 5.5.0.

Support for Siebel Release 7.7

Release 6.0.0 of the Extension adds support for Siebel eBusiness Applications Release 7.7. No Extension entities have been added or removed in release 6.0.0, but some have been revised as necessary. The impact for upgrades is described in *General Upgrade Impacts and Guidelines* on page 19 and *No Specific Impacts of Upgrading from Release 5.5.0* on page 20.

Use of EIM Interface for Siebel Pre-Defined Query Object Type

The object type previously called Siebel Pre-Defined Query (COM) has been renamed Siebel Pre-Defined Query. This reflects the fact that the object type now uses the Enterprise Integration Migration (EIM) interface rather than the COM interface.

Reports Accessible Only from Standard Interface

As with Mercury IT Governance Center release 6.0, Extension-related reports are accessible only from the standard interface.

Licensing

Extension licensing has been updated, consistent with the new Mercury IT Governance Center release 6.0 licensing strategy. For more information, see the *Security Model Guide and Reference*.

Prerequisite Knowledge and Experience

To install, upgrade, configure, maintain, or use release 6.0.0 of the Extension, you need to understand the following:

- Particular key concepts described in the *Key Concepts* document:

  - Change management

  - Environments

  - Mercury IT Governance Workbench

  - Object types

  - Packages

  - Portlets

  - Software deployment

  - Tokens

  - Workflows and workflow steps

  - Request types

  - Report types

- The Mercury IT Governance Center entities installed by the Extension

In addition, you must have practical experience installing, upgrading, configuring, and using Mercury IT Governance Center, if you are responsible for configuring the Extension.

You should have experience using Siebel eBusiness Applications.

For More Information

For more information about documents and resources that help you gain the required knowledge and experience, see the *Guide to Documentation.*

Installing or Upgrading the Extension

In This Chapter:

Overview of Installation

This section discusses the following overview subjects:

- System requirements

- Installing or upgrading to release 6.0.0

- General upgrade impacts and guidelines

- Impacts of upgrading from release 5.0.0

- No specific impacts of upgrading from release 5.5.0

In addition, this chapter discusses the following:

- Preparing for installation

- Downloading and installing the Extension

- Post-installation procedures

System Requirements

Mercury IT Governance Center release 6.0 and Mercury Change Management Extension for Siebel eBusiness Applications release 6.0.0 are installed on the same system and have the same system requirements. For more information about version level and other general system requirements, see the *System Requirements and Compatibility Matrix* document, which is available from the Mercury IT Governance Download Center:

http://itg.merc-int.com/support/download/login.jsp

Installing or Upgrading to Release 6.0.0

Before installing or upgrading the Extension, you must install (or upgrade to) and configure Mercury IT Governance Center release 6.0. For information about those procedures, see the *System Administration Guide and Reference.*

After yo have installed Mercury IT Governance Center release 6.0, you can install Mercury Change Management Extension for Siebel eBusiness Applications release 6.0.0 or upgrade to release 6.0.0 from either of the following releases:

- Mercury Change Management Extension for Siebel eBusiness Applications release 5.5.0

- Kintana™ Accelerator for Siebel eBusiness Applications release 5.0.0

The Mercury IT Governance Center release 6.0 upgrade process detects the existence of either release of the Extension in your environment and advises you that you will need to upgrade the Extension. This chapter provides information about upgrading the Extension from either release 5.5.0 or release 5.0.0.

If you are upgrading the Extension from any release earlier than Kintana Accelerator for Siebel eBusiness Applications release 5.0.0, you must first upgrade it to release 5.0.0, then upgrade it to release 6.0.0. The Mercury IT Governance Center upgrade process enforces this rule. For information about upgrades to (and features of) releases earlier than release 5.0.0, see the documentation for all of the releases from your current product release through release 5.0.0 as necessary.

Upgrade the Extension before using Mercury IT Governance Center release 6.0 for productive work.

General Upgrade Impacts and Guidelines

The Extension upgrade process replaces code entities installed in earlier releases, including all reference (REFERENCE) versions of the Extension entities described in Chapter 4, *Extension Entities,* on page 35.

During the upgrade process, existing non-reference entities, including those that you have copied, renamed, or customized for your specific environment, are not modified in any way. In other words, these non-reference entities retain all your customizations and they do not reflect any changes made in release 6.0.0, so they might not be compatible with Mercury IT Governance Center release 6.0.

Therefore, after upgrading to release 6.0.0, you should review the new reference entities. If necessary, copy and rename them, then apply the same customizations that were previously made to the corresponding non-reference entities for the earlier release.

Impacts of Upgrading from Release 5.0.0

The only significant impact seen in the Extension for an upgrade from release 5.0.0 is that some product names changed in release 5.5.0 as shown in *Table 2-1*.

Table 2-1. Product name changes in release 5.5.0

Product Name in Release 5.0.0	Product Name in Release 5.5.0
Kintana Accelerator	Mercury Change Management Extension
Kintana Deliver	Mercury Change Management
Kintana Drive	Mercury Project Management™
Kintana Create	Mercury Demand Management™

No Specific Impacts of Upgrading from Release 5.5.0

Release 6.0.0 does not have any release-specific impact on operation of the Extension after an upgrade from release 5.5.0.

Preparing for Installation

To prepare for installation of the Extension:

1. Purchase the Extension, if you have not already done so.

   You will not be allowed to download the Extension installation files from the Mercury IT Governance Download Center unless you have the username and password given to you at the time of purchase.

2. Decide where to put the installation file (`mitg-600-Siebel.jar`).

   The installation script looks in one of these locations for this file:

   - *ITG_Home* directory (this is the recommended location)

   - If the ITG_DEPLOYMENT_HOME environment variable is set: *ITG_Home/Deployment_Home* directory

   - If you use the `-D` option on the `sh` command to start the installation procedure: *Deployment_Home* directory

3. Collect the following information, which you will need to supply during the installation procedure:

   - The username and password you were given at the time you purchased the Extension.

   - The database password for the Mercury IT Governance Center schema

   - A Mercury IT Governance Center logon username

     The username (typically "admin") must belong to a security group that has the following access grants:

     - SysAdmin: Migrate ITG Objects

     - SysAdmin: Server Administrator

   - The password for the Mercury IT Governance Center logon username

4. Stop the Mercury IT Governance Server.

5. Set the Mercury IT Governance Server to restricted mode by running the following script:

   ```
   setServerMode.sh Restricted
   ```

   For more information about this script, see the *System Administration Guide and Reference*.

6. Start the Mercury IT Governance Server.

Downloading and Installing the Extension

This section describes the download and installation procedures to install or upgrade the Extension.

Downloading the Installation Files

To download the installation files for Mercury Change Management Extension for Siebel eBusiness Applications:

1. Go to the Mercury IT Governance Download Center:

   http://itg.merc-int.com/support/download/login.jsp

2. Log on using the username and password your company was given when it purchased the Extension.

3. In the ITG Download > License window:

   a. Read the Software End User License Agreement.

   b. Select the **I accept these terms** radio button.

   c. Click **Submit**.

4. In the ITG Download > Main window, click the **Installs** link under the name.

5. To download the Extension, click **mitg-600-Siebel.jar**.

Installation Procedure

To install the Extension:

1. Be sure you have completed all the steps in *Preparing for Installation on page 20*.

   In particular, be sure Mercury IT Governance Center is running in restricted mode. This procedure is documented in step 5 on page 21.

2. Be sure you have completed all the steps in *Downloading the Installation Files on page 22*.

3. Copy the `mitg-600-Siebel.jar` file to the *ITG_Home* directory.

   For information about using another location for this file, see step 2 on page 21.

Note	You do not need to unpack the installation file. The installation process automatically unpacks it.

4. Navigate to the *ITG_Home*/bin directory.

5. Run the script:

   ```
   sh kDeploy.sh -i Siebel
   ```

Note	The `kDeploy` script can upgrade one Extension. For more information, see the *System Administration Guide and Reference*. If you have upgraded Mercury IT Governance Center to release 6.0 and if one or more Mercury Change Management Extensions were previously installed on your server at the release 5.0.0 or 5.5.0 level, you can upgrade multiple Extensions by running one script. To do so, download one `.jar` file for each Extension to the *ITG_Home* directory, and run the `sh mitg_extensions.sh` upgrade script from the `upgrade_600` directory.

6. As `kDeploy.sh` runs, respond to prompts for the information you collected in step 3 on page 21 (for example, the database password for the Mercury IT Governance Center schema, and the Mercury IT Governance Center logon name and password).

   Files are installed in various subdirectories under *ITG_Home*. Data is also placed in the Mercury IT Governance database.

   When the installation procedure is complete, the following message appears:

   ```
   Deployment Siebel has been successfully installed.
   ```

7. Using a Web browser, view and check the installation summary report, which is located in:

   ITG_Home/logs/deploy/600/Siebel/*log_x*/installLog.html

   where *log_x* is a random number generated by kDeploy.sh to make each log file name unique. The number increments by one each time the installation script is run, so the log file for the most recent run is the one with the highest log number.

   The summary report lists all Mercury entities installed as part of the Extension installation process.

   Each entity that was installed correctly is marked as "Completed." If there is an error for a particular entity, the report contains a direct link to another log file (HTML page) with additional information.

   Installation of the Extension generates the logs that are listed and described in *Table 2-2*.

   Correct the errors, if necessary, and repeat the installation procedure. Otherwise, proceed to step 8.

8. (Optional) To verify that Mercury Change Management Extension for Siebel eBusiness Applications is listed among the installed release 6.0.0 Mercury Extensions, issue the following command:

   sh kDeploy.sh -l

   The name Siebel should appear in the list of installed Mercury Extensions.

   For example, if you have both Mercury Change Management Extension for Siebel eBusiness Applications and Mercury Change Management Extension for E-Commerce Technologies™ installed, you could see a table similar to the following:

Deployment	Version	Deployed	Description
Siebel	600	2005-02-15	Siebel Extension
eCommerce	600	2005-03-22	eCommerce Extension

9. (Optional) Install or upgrade other Mercury Extensions you have purchased, if you have not already done so.

For specific installation procedures for the other Mercury Extensions, see the appropriate Extension document described in the *Guide to Documentation*.

10. Stop and restart the Mercury IT Governance Center in normal mode, as follows.

    a. Stop the Mercury IT Governance Server.

    b. Run the script:

    ```
    setServerMode.sh NORMAL
    ```

    For more information about this script, see the *System Administration Guide and Reference*.

    c. Start the Mercury IT Governance Server.

Logs Generated During Installation

The logs listed and described in *Table 2-2* are generated during installation and saved in the `ITG_Home`/logs/deploy/600/Siebel directory. The log number (#####) is a random number generated by `kDeploy.sh` to make each log file name unique.

Table 2-2. Logs generated during installation

Directory or File	Description
ddlDriver.#####.log	Contains information about data model changes made during installation.
jarxvf.#####.log	Contains information from the procedure that unpacks the `jar` file.
packageDriver.#####.log	Contains information about the installation of database code, for example, reports.
postXMLDriver.#####.log	Contains information about the application of SQL scripts required after the installation of Mercury-supplied data.
preXMLDriver.#####.log	Contains information about the application of SQL scripts required prior to the installation of Mercury-supplied data, such as the definition for Mercury Change Management Extension for Siebel eBusiness Applications.

Post-Installation Procedures

After you have finished installing Mercury Change Management Extension for Siebel eBusiness Applications, you need to:

- Review the object types

- Define environments for each application environment that will be a source or destination of migration, for example:

    - Host connection information

    - Database connection information

- If required, customize the logic in the various installed entities.

After the Extension has been running successfully for a substantial period of time, you can optionally delete all of the installation or upgrade files. However, Mercury recommends that you retain (or copy) the log files.

Configuring the Extension

In This Chapter:

Configuring Environments

Before you can use the Extension, you need to define the following environments, using Mercury IT Governance Workbench:

● The Siebel system and database

You specify these environments in the **Extension Data** tab, **Siebel** subtab and in the **Host** tab of Mercury IT Governance Workbench Environment window.

● The Mercury IT Governance Server you intend to use for migrating the Siebel entities

Configuring the Extension Data Tab

The environment for Mercury Change Management Extension for Siebel eBusiness Applications is specified in the Mercury IT Governance Workbench, **Extension Data** tab, **Siebel** subtab. *Figure 3-1* shows a sample environment. *Table 3-1* lists the field names and their definitions.
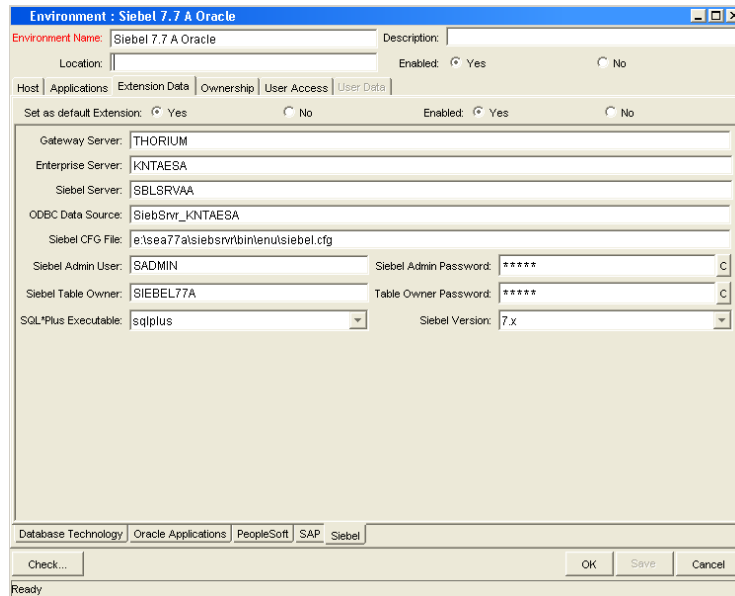
Figure 3-1. Environment window, Extension Data tab, Siebel subtab, sample data

Table 3-1. Environment window, Extension Data tab, Siebel subtab field definitions

Field Name	Definition
Set as default Extension	Whether or not **Siebel** is the default subtab to show for this window's **Extension Data** tab.
Enabled	Whether or not the fields on this tab are available to be configured.
Gateway Server	Name of the Siebel Gateway Server that controls access to the Siebel Server instance. If the Siebel Server instance is one of a group managed by an Enterprise Server, the Gateway Server may also provide load balancing and failover among the instances in the group.
Enterprise Server	Name of the Siebel Enterprise Server to which the Siebel Server instance belongs. An Enterprise Server is a logical group of Siebel Servers that support the same group of users accessing a common database server.
Siebel Server	Name of the Siebel Server instance.
ODBC Data Source	ODBC data source name. The ODBC data source should have been generated on the Windows machine that hosts the Siebel Server instance, and it should point to the database in which it was installed.
Siebel CFG File	A Siebel configuration (`.cfg`) file on the server that can be used by the EAI interface. Typically, this file is located in the Siebel Server `bin` directory and is created during Siebel Server installation.
Siebel Admin User	User who has Siebel Administrator rights.
Siebel Admin Password	Password for the Siebel Admin user.
Siebel Table Owner	User name of the database user who owns the Siebel tables (usually **SIEBEL**, not **SADMIN**).
Table Owner Password	Password for the Siebel table owner.
SQL*Plus Executable	Executable used to connect to the Oracle database.
Siebel Version	Siebel application version.

To access the **Siebel** subtab on the **Extension Data** tab in the Workbench Environment window:

1. Open the Workbench.

2. From the shortcut bar, click the **Environments** button and click the **Environments** icon.

3. Open a new environment or, if the environment of interest already exists, open it. The Environment window opens.

4. For this Extension, click the **Extension Data** tab and then, at the bottom of the window, click the **Siebel** subtab.

Configuring the Host Tab

To configure the **Host** tab of the Environment window (see *Figure 3-2* for a sample environment):

1. Complete the applicable data in the **Server** section with information for connection to the physical server that hosts the Siebel server.

2. Complete the **Database** section for the Siebel database.

   The User Login field for a Server Type of **SQL Server** or the Username field for a Server Type of **Oracle Server** should usually be set to **SADMIN**.

3. Complete the applicable data in the **Client** section with information for connection to the physical server that hosts the Siebel Tools client. For example, for the Base Path field, specify the Siebel Tools directory. See *Siebel Project Object Type* on page 53.
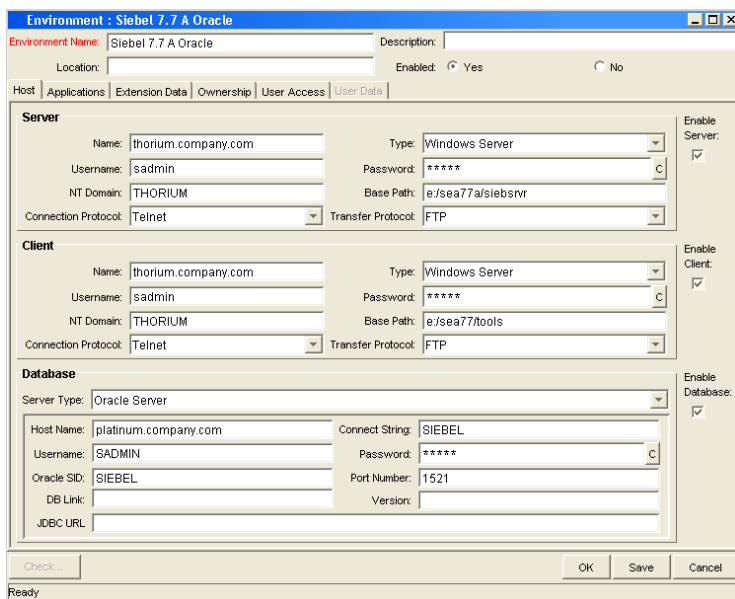
Figure 3-2. Environment window, Host tab sample data

Overview of Siebel Entity Migration

This following sections discuss the following subjects:

- Siebel Enterprise Integration Manager (EIM) interface

- Siebel eBusiness Application Integration (EAI) interface

Siebel EIM Interface

EIM is a Siebel process that is used to export and import data into Siebel Applications. Siebel EIM consists of a large number of interface tables known as EIM tables, SQL procedures to manipulate them, and a server management program from which the entire process is controlled. EIM also requires a user-provided configuration (`ifb`) file that is used to supply various parameters to the EIM process, such as whether the current operation is an import or export, and which interface table(s) it involves.

Most of the Extension's packaged object types utilize the EIM architecture for migrating data. Each EIM migration uses the following general process (see *Figure 3-3*):

1. Connect to the Siebel source environment.

2. Construct the EIM configuration file for the export, using selection criteria specified by the user (that is, a list of the specific objects that should be exported).

3. Invoke EIM, passing it a reference to the configuration file generated in step 2.

4. For Siebel 2000 and Siebel 7, run an additional SQL script to populate the Business Unit details.

5. Export the interface table contents into a datafile (`.dat`).

6. FTP the resulting data file from the source environment to the destination environment.

7. Connect to the Siebel destination environment.

8. Import the contents of the datafile (`.dat`) into the interface table.

9. Construct the EIM configuration file for the import.

10. Invoke EIM, passing it a reference to the configuration file generated in step 9.

11. Attach EIM log files to the Mercury IT Governance package log.

12. Print the import summary and attach the logs.
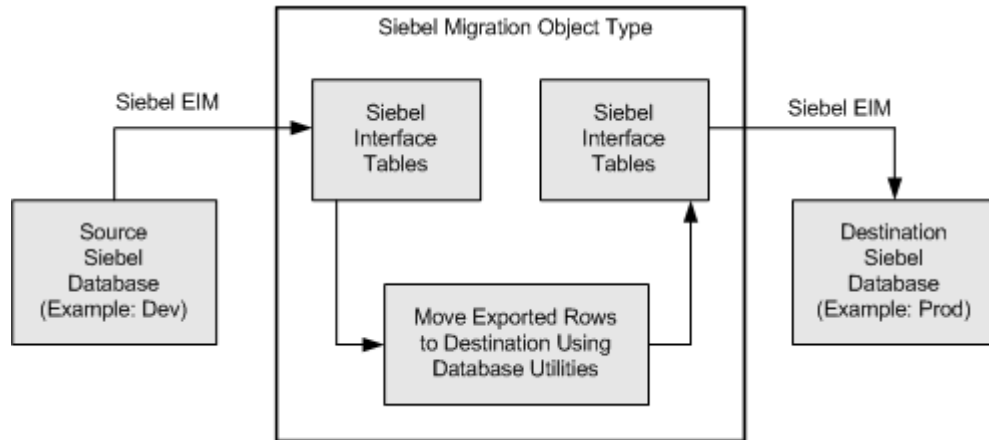
Figure 3-3. Overview of Siebel entity migration

Mercury Change Management Extension for Siebel eBusiness Applications allows users of Mercury IT Governance products to initiate and track the Siebel entity migrations using special object types and workflows, which are discussed in Chapter 4, *Extension Entities,* on page 35.

Siebel EAI Interface

The Siebel EAI architecture offers a comprehensive integration strategy. Siebel EAI is a framework for building integration between Siebel applications and external applications and technologies. This architecture and the components of the Siebel EAI infrastructure offer a comprehensive interface to migrate complex data structures involving multiple business components.

Each EAI migration uses the following general steps:

1. Connects to the Siebel source environment.

2. Uses EAI to export the entity into an XML file.

3. FTPs the resulting XML file from the source environment to the destination environment.

4. Connects to the destination environment.

5. Uses EAI to import the contents of the XML file.

To invoke EAI services, the Extension uses the `ksacom` utility, a console-based application built using C++. It provides a command-line interface to invoke workflows in a Siebel Application and it allows passing parameters to the workflows. It is used to invoke the Siebel workflows provided with the Extension as XML files.

Table 3-2 lists and describes the options for the `ksacom` utility. Only one option can be specified at a time. The options and arguments are case sensitive, and the arguments required depend on the option chosen.

Table 3-2. ksacom utility process options

Option	Description	Required Arguments
-h	Displays the command options for Help.	None
-t	Tests the connection.	**-U**, **-P**, and **-C**
-b	Invokes the Siebel workflow specified by the arguments.	**-U**, **-P**, **-C**, **-W**, **-F**, and **-O**

Table 3-3 lists and describes the arguments for the `ksacom` utility.

Table 3-3. ksacom utility arguments

Argument	Description
-U	Siebel Administrator user login. Defaults to **SADMIN**.
-P	Siebel Administrator login password.
-C	Complete path of the Siebel CFG file.
-F	XML file name.
-W	Workflow process name that needs to be triggered.
-O	Object ID parameter for the Siebel workflow process. Required only when the Object ID parameter of the workflow process needs to be passed.

Configuring the Extension for the EAI Interface

To use the `ksacom` utility, configure the Extension as follows:

1. Copy the `ksacom` utility from the Mercury IT Governance Server located in the *ITG_Home*`/deploy/600/Siebel/utilities` directory to the `TEMP` directory under the Siebel Server base path (for example, `c:\Siebel\ Siebsvr\TEMP`).

   Repeat for each server to be used with EAI interface object types.

2. Import the Siebel workflows located in the *ITG_Home*`/deploy/600/ Siebel/utilities` directory into Siebel for EAI interface object types as follows, depending on the Siebel release.

   For Siebel releases earlier than Release 7.7, use the Workflow Process Definition View in the Siebel Application to import and activate the following workflows:

   - Kintana Export Employee.xml

   - Kintana Export Workflow.xml

   - Kintana Import Employee.xml

   - Kintana Import Workflow.xml

   For Siebel Release 7.7, use Siebel Tools to import and deploy the following files:

   - Kintana Export Employee77.xml

   - Kintana Import Employee77.xml

   Then, for Release 7.7, use the Repository Workflow Process Definition View in the Siebel Application to activate the workflows.

Note | You can also use VBScripts to invoke EAI services. VBScripts can be called from the Extension using the command:
`cscript //Nologo scriptname`

Chapter
4

Extension Entities

In This Chapter:

- *Validations*
- *Tokens*

Object Types

This section provides reference information about the following categories of Siebel-specific object types provided in the Extension:

- Siebel Configuration Data Migration object types

- Siebel Repository Data Migration object types

Overview of Object Types

Migration and compilation of object types are driven by commands included within the object types. For more information about commands in the Mercury IT Governance Center environment, see the *Commands, Tokens, and Validations Guide and Reference*. For more information about using object types in packages, see the *Mercury Change Management User's Guide*.

You can view or modify an object type as follows:

1. Open the Workbench.

2. From the shortcut bar, click **Change Mgmt** and click the **Object Types** icon.

3. If you want, select a particular Extension from the Extension drop-down list.

4. In the Object Type Workbench window, click **List**.

5. Select the object type of interest and click **Open**.

6. Edit the object type. (On the **Fields** tab, the list of fields in the Prompts column is alphabetized.)

Subsequent figures in this chapter show screens you can use to revise object types in conjunction with adding package lines. You can access these screens as follows:

1. From the shortcut bar, click **Change Mgmt** and click the **Packages** icon.

2. Add a new or open an existing package, as necessary.

3. Select a workflow.

4. Add a line.

5. Select the object type of interest.

Reference Object Types

Reference (REFERENCE) object types cannot be edited, but you can copy and rename them and edit the copies to meet your needs. You can also use existing non-reference object types as is or configure them further to meet your needs.

Siebel Configuration Data Migration Object Types

The following sections describe the following categories of Siebel configuration data migration object types:

- Object types that use Siebel EIM

- Object types that use the Siebel EAI interface

Object Types That Use Siebel EIM

The various object types that use Siebel EIM configuration data migration capture specific information needed for migration from one instance to another.

These object types can have some or all of the common fields shown in *Figure 4-1* and listed in *Table 4-1*. (The Assignment Rule field in *Figure 4-1* is not a common field.) Some of the common fields are not present for particular object types. You can use the Object Type Workbench to view all the fields for any object type.

Table 4-2 lists and defines all of these object types, indicates which ones are pre-configured with which options (and therefore display the associated common fields), and which Siebel versions are supported by each object type.

Figure 4-1. Common fields for Siebel EIM configuration data object types

Table 4-1. Definitions for common fields for Siebel EIM configuration data object types

Field Name	Definition
Selection Type	Set of objects to migrate: **Single Value**, **Range of Objects**, **Wild Card Match**, or **All**.
Start of Range	Starting value for range of values to be moved.
End of Range	Ending value for range of values to be moved.
Wildcard Match	Wildcard match to determine values to be moved. For example, S% will migrate all values starting with an S.
Import Base Tables	Specifies whether to migrate data to all base tables or to only the selected base table identified in the Base Table field.
Base Table List	List of base tables for the object type's associated interface table. If the Migrate Base Tables option is set to Specify, the base table selected from the drop-down list will be the only base table to which data is migrated.
Valid Siebel Versions (Display only)	Siebel versions that are supported by this object type.

Table 4-2. Extension object types using Siebel EIM

Object Type Name	Definition	Option to Migrate to Specific Base Tables?	Option to Migrate Range?	Option to Migrate Wildcard Set?	Supported Versions
Siebel Agreement	Migrates Siebel Agreements	Y	Y	Y	Siebel 5.x Siebel 6.x Siebel 7.x
Siebel Assignment Rule	Migrates Siebel Assignment Rules	Y	Y	Y	Siebel 5.x Siebel 6.x Siebel 7.x
Siebel Assignment Rule Group	Migrates Siebel Assignment Rule Groups	Y	Y	Y	Siebel 7.x
Siebel Assignment Workload	Migrates Siebel Assignment Workloads	Y	Y	Y	Siebel 5.x Siebel 6.x Siebel 7.x
Siebel Competitive Metric	Migrates Siebel Competitive Metrics	Y	N	N	Siebel 5.x Siebel 6.x Siebel 7.x

Table 4-2. Extension object types using Siebel EIM [continued]

Object Type Name	Definition	Option to Migrate to Specific Base Tables?	Option to Migrate Range?	Option to Migrate Wildcard Set?	Supported Versions
Siebel Decision Issue	Migrates Siebel Decision Issues	Y	Y	Y	Siebel 5.x Siebel 6.x Siebel 7.x
Siebel Employee	Migrates Siebel Employee Data	Y	Y	Y	Siebel 5.x Siebel 6.x
Siebel Employee Detail	Migrates Siebel Employee Details	Y	Y	Y	Siebel 5.x Siebel 6.x
Siebel Entitlement	Migrates Siebel Entitlements	Y	N	N	Siebel 5.x Siebel 6.x Siebel 7.x
Siebel Exchange Rate	Migrates Siebel Exchange Rates	N	N	N	Siebel 5.x Siebel 6.x Siebel 7.x
Siebel Industry	Migrates Siebel Industry Codes	N	Y	Y	Siebel 5.x Siebel 6.x Siebel 7.x
Siebel List of Values	Migrates Siebel List of Values	N	Y	Y	Siebel 5.x Siebel 6.x Siebel 7.x
Siebel Organization	Migrates Siebel Organization	N	Y	Y	Siebel 5.x Siebel 6.x
Siebel Position	Migrates Siebel Positions	Y	N	N	Siebel 5.x Siebel 6.x
Siebel Pre-Defined Query *	Migrates Siebel Pre-Defined Query *	N	Y	Y	Siebel 7.x
Siebel Price List	Migrates Siebel Prices Lists	Y	N	N	Siebel 5.x Siebel 6.x Siebel 7.x
Siebel Product	Migrates Siebel Products	Y	N	N	Siebel 5.x Siebel 6.x Siebel 7.x
Siebel Product Line	Migrates Siebel Product Lines	Y	Y	Y	Siebel 5.x Siebel 6.x Siebel 7.x

Table 4-2. Extension object types using Siebel EIM [continued]

Object Type Name	Definition	Option to Migrate to Specific Base Tables?	Option to Migrate Range?	Option to Migrate Wildcard Set?	Supported Versions
Siebel Responsibility	Migrates Siebel Responsibilities	Y	Y	Y	Siebel 5.x Siebel 6.x Siebel 7.x
Siebel State Model	Migrates Siebel State Models	N	Y	Y	Siebel 5.x Siebel 6.x Siebel 7.x
Siebel States and Transitions	Migrates Siebel States and Transitions	Y	N	N	Siebel 5.x Siebel 6.x Siebel 7.x
Siebel View	Migrates Siebel Views	N	Y	Y	Siebel 5.x Siebel 6.x Siebel 7.x
Siebel Workflow Action	Siebel Workflow Actions	N	Y	Y	Siebel 5.x Siebel 6.x Siebel 7.x
Siebel Workflow Group	Siebel Workflow Groups	N	Y	Y	Siebel 6.x Siebel 7.x
Siebel Workflow Rule	Siebel Workflow Rules	N	Y	Y	Siebel 6.x Siebel 7.x

* When migrating a single Siebel Pre-Defined Query, private queries are not listed in the validation and cannot be migrated. However, when migrating queries using Range or Wildcard migration, private queries with matching criteria are also migrated.

Recommended Migration Order for Siebel Entities That Use EIM

The following recommended import order is based on information located in the default `ifb` file:

1. Reference Data

   - List of Values (migrate LOV_TYPE first)

   - Industry Codes

   - Competitive Metrics

   - Decision Requests

   - Exchange Rates

2. Views

3. Responsibilities

4. Employees

5. Internal Divisions (Org)

6. Positions

7. Products

8. State Models

9. Workflows

10. Assignments

Error Processing

Mercury products are equipped to report the standard errors that can occur during a Siebel migration. Common errors encountered during import might include:

- `DUP_ROW_EXISTS`

- `Failed to resolve foreign key value`

- `Invalid values`

You can view errors in the package line execution log.

Creating New Siebel Object Types That Use EIM

If an object type does not exist for an interface table you want to use in your migration, generate a new object type as follows.

1. Copy an existing Siebel object type (for example, Siebel View) to a new object type (for example, Siebel Payment Term).

2. Generate a new validation to retrieve the appropriate values from the Siebel database (for example, copy SBL - Views to SBL - Payment Terms). Change the command to the appropriate command name and command steps. For example, change List of Siebel Views to List of Payment Terms. Change the command step query to retrieve the appropriate values. For example, change:

   ```
   select NAME, NAME
   from S_APP_VIEW
   ```

   to:

   ```
   select NAME, NAME
   from S_PAYMENT_TERM
   ```

3. Change the validation column headers to the appropriate names and widths.

4. Save the validation.

5. Edit the object type generated by the copy in step 1. Change the Field Prompt for the value to be the appropriate name (for example, change Application View to Payment Term) and change the validation to the validation generated in step 2 (for example, SBL - Payment Term).

6. Change the value of the Interface Table Key field to the appropriate interface table key as identified in the Siebel Interface Table Mapping documentation. For example, the interface table key for S_PAYMENT_ TERM is TRM_NAME.

7. Change the value of the Interface Table field to the interface table name. For example, the value for Siebel Schedule Calendar is EIM_PAY_TERM.

8. Change the value of the Target Table field to the name of the target table for the interface table. For example, the value for the Payment Term table is S_ PAYMENT_TERM.

9. Change the value of the Target Field field to the primary key column name of the base table. For example, the value for the Payment Term is NAME.

10. Save the object type.

11. Determine whether an additional SQL script needs to be run to populate the Business Unit details in the source interface tables. To do this, look for the EIM table in the following script in the ADMIN folder of the Siebel server directory:

    `eim_export_lookup_bu_name.sql`

    Copy any additional SQL statement into the `sc_sbl2000_export_bu_name` special command, depending on the Siebel version. Follow the conventions in the `ksc_sbl2000_export_bu_name` special command. Be sure to use the `sc_sbl2000_export_bu_name` special command in your object type.

    For Payment Terms, you are not required to run any additional SQL.

Note

> The `sc_sbl2000_export_bu_name` special command is provided because the `ksc_sbl2000_export_bu_name` special command cannot be edited.

The new object type can now be processed using standard Mercury IT Governance Center package procedures. For more information on creating new object types, see the *System Administration Guide and Reference*.

Object Types That Use the EAI Interface

The Siebel eBusiness Application Integration (EAI) architecture and the components of the Siebel EAI infrastructure offer a comprehensive interface to migrate complex data structures involving multiple business components.

Using this interface, you can migrate entities such as Siebel Workflow Processes, which constitute multiple business components and do not have supporting interface tables for migration purposes.

Using the EAI interface, you can migrate the contents of Siebel base tables associated with an object type from one environment to another.

For more information, see *Configuring the Extension for the EAI Interface on page 34*.

The EAI object types and the Siebel versions that support them are listed in *Table 4-3*. They are discussed further in the following sections.

Table 4-3. Extension object types using EAI interface

Object Type	Valid for Siebel Versions
Siebel Workflow Processes (EAI)	6.x, 7.0, 7.5
Siebel Employee (EAI)	6.x, 7.x

Siebel Workflow Processes (EAI) Object Type

Figure 4-2 shows sample data for the Siebel Workflow Processes (EAI) object type when adding a package line. *Table 4-4* contains field definitions for the object type.

Note	In Siebel Release 7.7, Workflow Processes are part of the repository and are migrated with Siebel Repository object types.

Warning	Do not use the Siebel Workflow Process (EAI) object type to migrate more than 100 Workflow Processes in one execution.
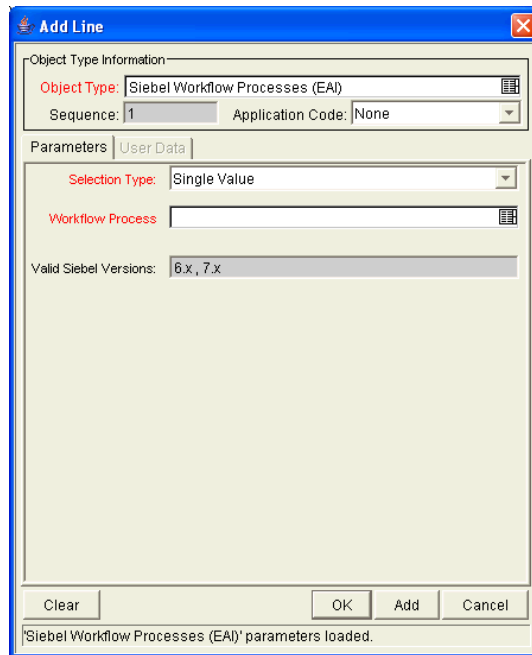
Figure 4-2. Siebel Workflow Processes (EAI) object type sample data

Table 4-4. Siebel Workflow Processes (EAI) object type field definitions

Field Name	Definition
Selection Type	Set of objects to migrate: **Single Value** or **All**.
Workflow Process	Name of the Siebel workflow process to be migrated.
Valid Siebel Versions (Display only)	Siebel versions that are supported by this object type. Value displayed is **6.x, 7.x**, however in Siebel Release 7.7, Workflow Processes are part of the repository and are migrated with Siebel Repository object types

Siebel Employee (EAI) Object Type

Figure 4-3 shows sample data for the Siebel Employee (EAI) object type when adding a package line. *Table 4-5* contains field definitions for the object type.

Figure 4-3. Siebel Employee (EAI) object type sample data

Table 4-5. Siebel Employee (EAI) object type field definitions

Field Name	Definition
Selection Type	Set of objects to migrate: **Single Value** or **All**.
Employee	Name of the employee to be migrated.
Valid Siebel Versions (Display only)	Siebel versions that are supported by this object type.

Creating a New Siebel Object Type Using the EAI Interface

To create a new object type based on the EAI interface (in this example, to migrate Accounts):

1. Identify the Integration Object related to the data you need to move.

   For example, Accounts are migrated with a predefined Integration Object named Accounts in Siebel Tools. If the Integration Object is not present, you can create your own Integration Object using Siebel Wizards.

2. Create workflow processes in Siebel to import and export accounts. You can copy the Kintana Export Employee and the Kintana Import Employee workflows (in Siebel) and configure them to use the Account Integration Object. Example names for the workflow processes are Mercury Export Account and Mercury Import Account. These two workflow processes need to exist on both source and destination Siebel instances.

3. Create a new validation in Mercury IT Governance Center to retrieve accounts from the Siebel database. You could copy the SBL - Employee (EAI) validation and rename it as SBL - Accounts (EAI). Change the commands and command steps to select accounts.

4. Copy the existing Siebel Employee (EAI) to a new object type named Siebel Account (EAI).

5. Edit the object type generated. Change the field prompt for the value to be the appropriate name (such as Account), and change the validation created in step 3.

6. Go to the **Commands** tab for the new object type. Edit the command steps named Export Data from Source and Insert Data at Destination to use the workflows renamed in step 2.

7. Save the object type.

Siebel Repository Data Migration Object Types

The following sections discuss the following object types for Siebel repository data migration:

- Siebel Repository object type

- Siebel Repository - Multi-Step object type

- Siebel Project object type

- Siebel Extract Mobile Client DB object type

Siebel Repository Object Type

The Siebel Repository object type migrates a Siebel repository and schema from one Siebel instance to another.

The Siebel Repository object type performs the following tasks:

- Exports the Siebel repository from a specified Siebel instance

- Imports the Siebel repository into a specified instance

- Determines the schema differences between new and old repositories

- Imports the Siebel schema into a specified Siebel instance

- Applies schema changes to the destination instance

Note   Old repositories can accumulate at the destination instance if they are not periodically deleted. After successful migration and verification of a repository at the destination, delete old repositories if their contents are no longer needed.

Figure 4-4 shows sample data for the Siebel Repository object type when adding a package line. *Table 4-6* contains field definitions for the object type.

Figure 4-4. Siebel Repository object type sample data

Table 4-6. Siebel Repository object type field definitions

Field Name	Definition
Source Repository	Name of the Siebel Repository in the source instance (defaults to **Siebel Repository**).
Destination Repository	Name under which the Siebel Repository should be installed in the destination instance (defaults to **Siebel Repository**).
Destination Data Area	(Oracle databases only) Name of the data tablespace in the destination Siebel instance.
Destination Index Area	(Oracle databases only) Name of the index tablespace in the destination Siebel instance.
Valid Siebel Versions (Display only)	Siebel versions that are supported by this object type.

Siebel Repository - Multi-Step Object Type

The Siebel Repository - Multi-Step object type migrates a Siebel Repository and schema from one Siebel instance to another. This object type provides the same functionality as the Siebel Repository object type, but because Repository migration is a very time-consuming process and involves two or more Siebel servers, the Siebel Repository - Multi-Step object type provides a way to rerun parts of the Repository migration process, without rerunning the entire process.

The commands in the Siebel Repository - Multi-Step object type are divided into the components of the Repository migration and conditioned on the appropriate workflow step in the Siebel Migrate Repository subworkflow. The commands are conditioned as shown in *Table 4-7*.

Table 4-7. Siebel Repository - Multi-Step object type commands

Command	Condition
Connect to source instance	'[WFS.STEP_NAME]'='Export Repository'
Export repository	'[WFS.STEP_NAME]'='Export Repository'
FTP repository data file to destination instance	'[WFS.STEP_NAME]'='Import Repository'
Connect to destination instance	'[WFS.STEP_NAME]'='Import Repository'
Import repository	'[WFS.STEP_NAME]'='Import Repository'
Export schema definition	'[WFS.STEP_NAME]'='Export Schema Definition'
Apply schema changes (synchronization)	'[WFS.STEP_NAME]'='Apply Schema Changes'

Note   Old repositories can accumulate at the destination instance if they are not periodically deleted. After successful migration and verification of a repository at the destination, delete old repositories if their contents are no longer needed.

Figure 4-5 shows sample data for the Siebel Repository - Multi-Step object type when adding a package line. *Table 4-8* contains field definitions for the object type.
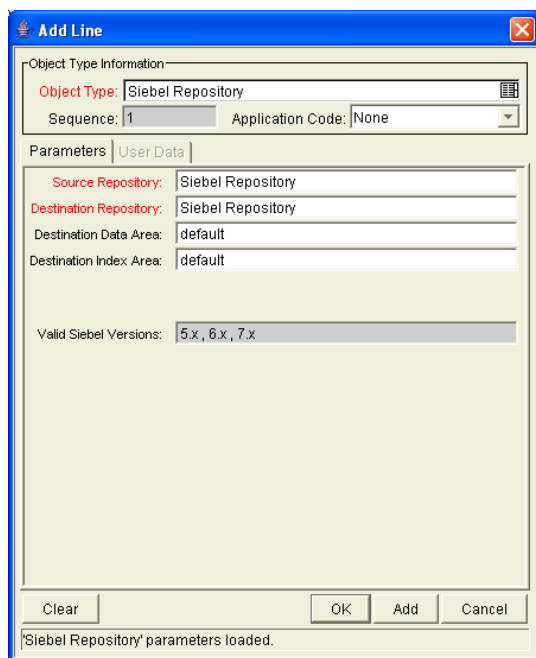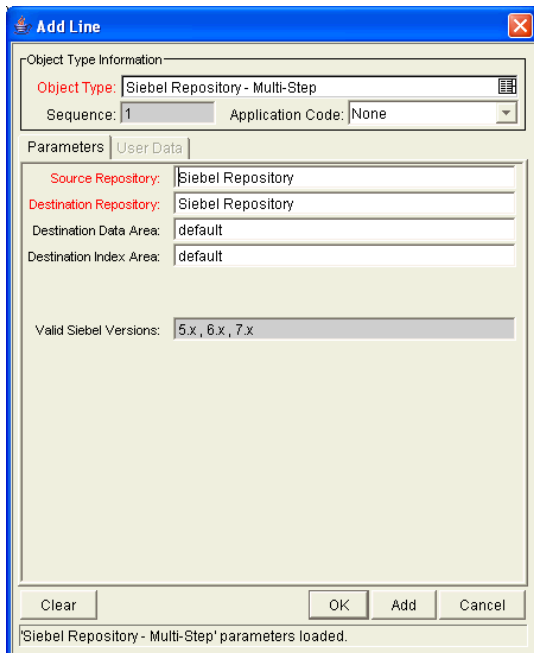
Figure 4-5. Siebel Repository - Multi-Step object type sample data

Table 4-8. Siebel Repository - Multi-Step object type field definitions

Field Name	Definition
Source Repository	Name of the Siebel Repository in the source instance (defaults to **Siebel Repository**).
Destination Repository	The name with which the Siebel Repository should be installed in the destination instance (defaults to **Siebel Repository**).
Destination Data Area	(Oracle databases only) Name of the data tablespace in the destination Siebel instance.
Destination Index Area	(Oracle databases only) Name of the index tablespace in the destination Siebel instance.
Valid Siebel Versions (Display only)	Siebel versions that are supported by this object type.

Note

The Siebel Migrate Repository and Data - Multi-Step workflow, the Siebel Migrate Repository Subworkflow, and the Siebel Repository - Multi-Step object type are designed to be used together. Leaving out one or more of these components will result in incomplete executions. The SIEBEL_HOME environment definition is set every time there is a new connection formed in the Siebel Repository - Multi-Step object type.

Siebel Project Object Type

The Siebel Project object type migrates Siebel Projects from one Siebel instance to another. Siebel Projects are named sets of object definitions that reside in the Siebel repository. Projects are created so the objects can be worked on by teams of application developers.

Siebel Tools connects to the Siebel database using the `tools.cfg` file. This file needs to be configured to enable connections to the source and destination Siebel databases. The naming convention for the datasource is *Siebel_ Enterprise_Server_Name_*`Datasrc`.

The connection information for the Siebel Tools Client is specified in the **Client** section on the **Host** tab. Set the base path for the client to the home directory of Siebel Tools. It is not necessary for all Siebel instances to have a separate Siebel Tools installation. The Siebel Tools application can connect to different datasources using the entries specified in the `tool.cfg` configuration file.

The Siebel Project object type is used with the Siebel Project Deployment workflow. The object type will export and import the project in the form of a `sif` file using the command line interface for tools. The import of the project is possible only when the project is locked at the destination, so the object type verifies the lock status at the destination. If the project is not locked, it terminates the execution with an error message.

After the required projects have been migrated to the destination, the workflow calls a subworkflow, which:

- Compiles the new `srf`.

- Stops the destination service.

- Replaces the `srf`.

- Restarts the destination service.

Figure 4-6 shows sample data for the Siebel Project object type when adding a package line. *Table 4-9* contains field definitions for the object type.
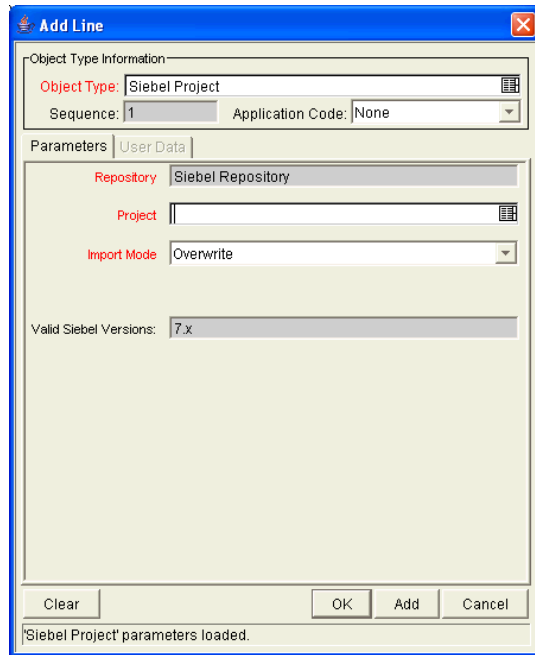
Figure 4-6. Siebel Project object type sample data

Table 4-9. Siebel Project object type field definitions

Field Name	Definition
Repository	Name of the Siebel Repository in the source instance (defaults to **Siebel Repository**).
Project	Name of the project that needs to be migrated.
Import Mode	Import Mode option for importing the project in the destination instance.
Valid Siebel Versions (Display only)	Siebel versions that are supported by this object type.

Siebel Extract Mobile Client DB Object Type

The database extract process retrieves data visible to a specific mobile client from the server database. It creates a compressed file containing data to be loaded into a local database when a mobile client initializes the database. If a mobile user's position changes, you must re-extract the mobile client's database to delete records that should no longer be visible to the user based on the user's new position.

The first step in creating a local database for a new mobile user is to extract the database on the Siebel Remote Server. If you are extracting a mobile user for the first time, you will also need to generate a new database template. Extract mobile users by running the Database Extract component on the Siebel Server for which the Mobile Client has been defined. Database Extract extracts all visible data for the mobile client into a snapshot file in the mobile client's outbox directory.

The Database Extract component creates a database snapshot file for a given mobile user. The file contains all the data required to initialize the user's local database. Database Extract retrieves data according to routing rules that define the level of access to information for each mobile client.

When database extracts take place after a Repository Migration, an `srf` Upgrade Kit should be created prior to running the DB Extract component so that both the DB extract and the `srf` pieces will be waiting in the mobile client's inbox simultaneously. `srf` Upgrade Kits are manually created through Siebel Anywhere.

The Siebel Extract Mobile Client DB object type does the following:

- Creates a list of Mobile Clients whose databases need to be extracted.

- Extracts the Mobile Client's databases using the Siebel `srvrmgr` utility.

Figure 4-7 shows sample data for the Siebel Extract Mobile Client DB object type when adding a package line. *Table 4-10* contains field definitions for the object type.
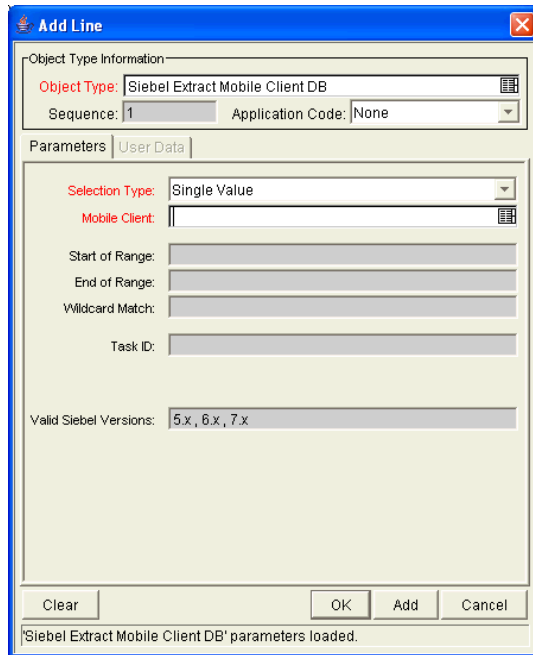

Figure 4-7. Siebel Extract Mobile Client DB object type sample data

Table 4-10. Siebel Extract Mobile Client DB object type field definitions

Field Name	Definition
Selection Type	Whether to extract a **Single Value**, a **Range**, a **Wildcard Set**, or **All Databases** for Mobile Clients.
Mobile Client	Identifies a single Mobile Client database to be extracted.
Start of Range	Starting value for range of values to be extracted.
End of Range	Ending value for range of values to be extracted.
Wildcard Match	Wildcard match to determine values to be extracted. For example, S% entered in the Wildcard in combination with a Selection Type of Wildcard will extract all values starting with an S.
Task ID (Display only)	Siebel server task initiated to extract the mobile clients.
Valid Siebel Versions (Display only)	Siebel versions that are supported by this object type.

Request Type

This section provides reference information about the only Siebel-specific request type provided by the Extension, which is the Siebel Enhancement request type.

Overview of Request Types

Migration and compilation of request types can be driven by commands included within the request types. For more information about commands in the Mercury IT Governance Center environment, see the *Commands, Tokens, and Validations Guide and Reference*. For more information about using request types in packages, see the *Mercury Change Management User's Guide*.

As an example, *Figure 4-8* shows a screen you can use to create requests from an existing Siebel Enhancement request type. You can access this screen as follows:

1. Make sure that the desired request type is enabled. From the Workbench shortcut bar, click **Demand Mgmt** and click the **Request Types** icon. If the desired request type needs to be enabled, open and enable it.

2. In the standard interface, select **Create > Request**. The Create New Request page appears.

3. Select the Request Type of interest from the drop-down list and click **Create**.

Reference Request Types

Reference (REFERENCE) request types cannot be edited, but you can copy and rename them and edit the copies to meet your needs. You can also use existing non-reference request types as is or configure them further to meet your needs.

Siebel Enhancement Request Type

The Siebel Enhancement request type gathers information required to log and track changes to Siebel instances.

Figure 4-8 shows the screen for creating a Siebel Enhancement request based on the provided request type. *Table 4-11* contains field definitions for the request type.

Figure 4-8. Siebel Enhancement request type creation

Table 4-11. Siebel Enhancement request type field definitions

Field Name	Definition
Summary tab (not expanded in *Figure 4-8*)	
Workflow	Siebel Enhancement/Bug
Siebel Enhancement tab	
Modification Type	Whether the enhancement is associated with **New** or **Existing** functionality.
Siebel Application	Siebel Application associated with the request.
Justification	Reason the request is being generated.
Difficulty	Relative difficulty associated with implementing the enhancement.
Estimated Time to Complete	Approximate time it will take to perform the enhancement.
Resolution	Final outcome of the enhancement (for example, **Fixed**, **Not Fixed**, or **Duplicate**).
Duplicate ID	Indicate any identical request(s) that has been logged.
Resolution Summary	Summary of the final outcome of the enhancement.

The Siebel Enhancement request type can be used in conjunction with the Siebel Enhancement/Bug workflow to log and track Siebel enhancements. This request type generates a Siebel Service Request when the request status is Ready for External Processing.

Note

When configuring the request type, the administrator needs to perform the analysis to determine which data should be transferred to the Siebel Service Request. Once the data is identified, the administrator needs to modify the associated request type (using the reference Siebel Enhancement/Bug request type as a basis for the request type).

Workflows

This section provides reference information about the Siebel-specific workflows provided in the Extension. These workflows are listed and defined in *Table 4-12*.

Overview of Workflows

Some workflows are associated with object types and Mercury Change Management, and include package execution. Other workflows are associated with request types and Mercury Demand Management, and describe decision-making processes. For more information about Mercury Change Management and Mercury Demand Management, see the *Mercury Change Management: Configuring a Deployment System* document and the *Mercury Demand Management: Configuring a Request Resolution System* document.

Migration and compilation of entities using workflows are driven by commands included within the entities or workflow steps. For more information about commands in the Mercury IT Governance Center environment, see the *Commands, Tokens, and Validations Guide and Reference*.

Reference Workflows

Reference (REFERENCE) workflows cannot be edited, but you can copy and rename them and edit the copies to meet your needs. You can also use existing non-reference workflows as is or configure them further to meet your needs.

List of Workflows

Table 4-12 lists and defines the workflows included in the Extension. Each is described in subsequent sections.

Table 4-12. Workflows included in the Extension

Workflow Name	Product Scope	Definition
Siebel Enhancement/Bug	Mercury Demand Management	Logs and tracks configuration changes to Siebel instances.
Siebel Migrate Repository and Data	Mercury Change Management	Migrates Siebel repository and configuration data from one Siebel instance to another.
Siebel Migrate Repository and Data - Multi-Step	Mercury Change Management	Migrates Siebel repository and configuration data from one Siebel instance to another.
Siebel Project Deployment	Mercury Change Management	Migrates Siebel projects.

Siebel Enhancement/Bug Workflow

The Siebel Enhancement/Bug workflow logs and tracks changes to Siebel instances. *Figure 4-9* shows the workflow diagram. Steps in the workflow are described in the procedure that follows.
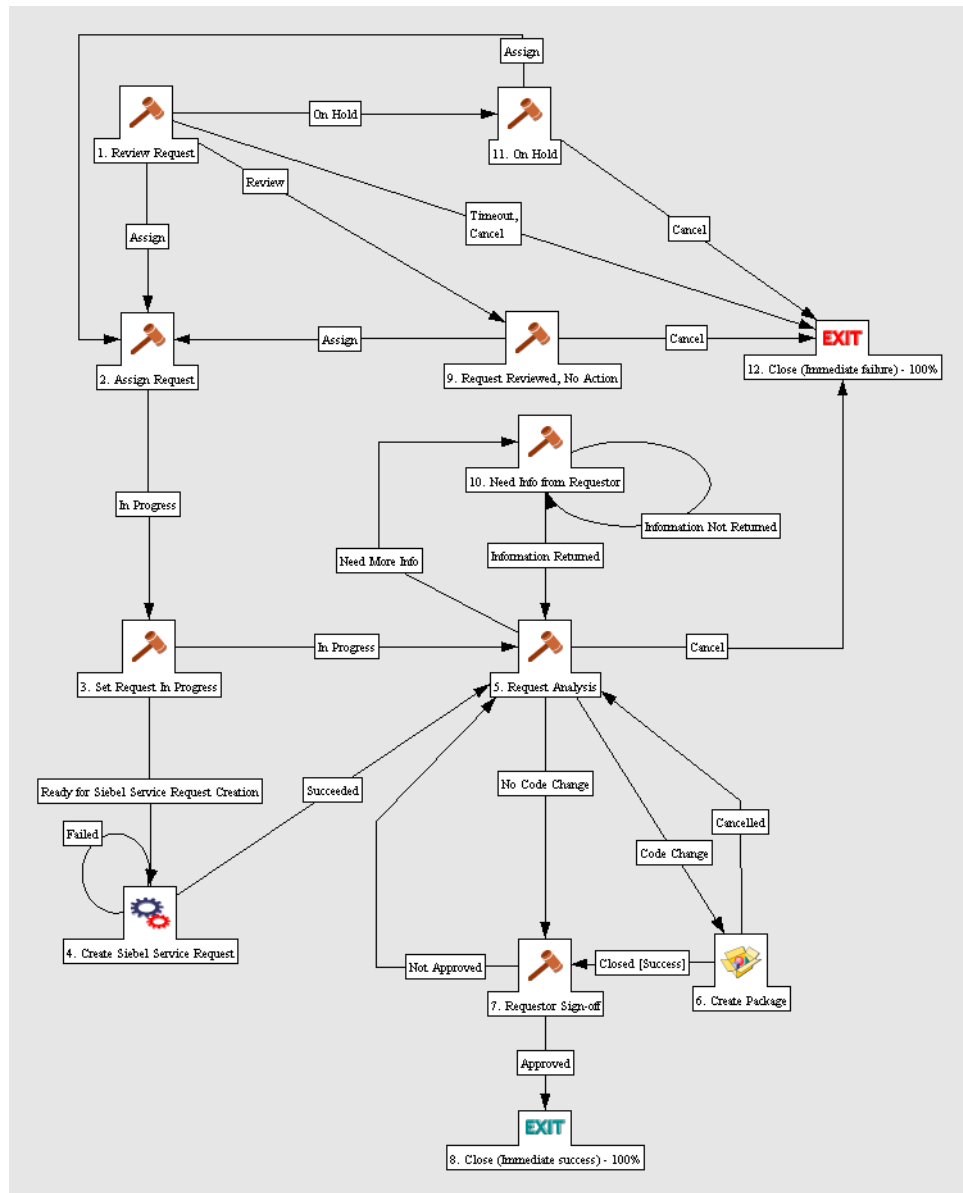

Figure 4-9. Siebel Enhancement/Bug workflow

After the Set Request in Progress step, users have an option to transition using Ready for Siebel Service Request Creation to Create Siebel Service Request. Create Siebel Service Request uses the Siebel COM interface to generate a service request from a Create Request. Using data in the Create Request, a service request is created in the source environment on the Create Siebel Service Request step.

Note | Administrators should make sure the associated request type is configured to generate the Siebel Service Request as intended.

Siebel Migrate Repository and Data Workflow

The Siebel Migrate Repository and Data workflow is used to migrate Siebel repository and configuration data from one Siebel instance to another.

Before performing a Siebel Repository migration, Mercury recommends shutting down the destination Siebel Server so that no server tasks or components are running that might conflict with the migration.

Starting and stopping the Siebel Servers is not the same as starting and stopping Siebel System Services.

Siebel Servers should not be shut down during Configuration Data migration. Siebel Configuration Data migration is performed using the EIM Server Task. If the servers are shut down during this migration, the migration fails.

The commands to start and stop the Siebel Servers are included in the Extension as available workflow execution steps. Commands are included in the workflow step commands and in Mercury special commands. There is one available workflow step for starting the Siebel Servers and one for stopping them. Each of these workflow steps executes commands on both the source and destination Siebel environments (or environment groups, in the case of multiple Siebel Servers).

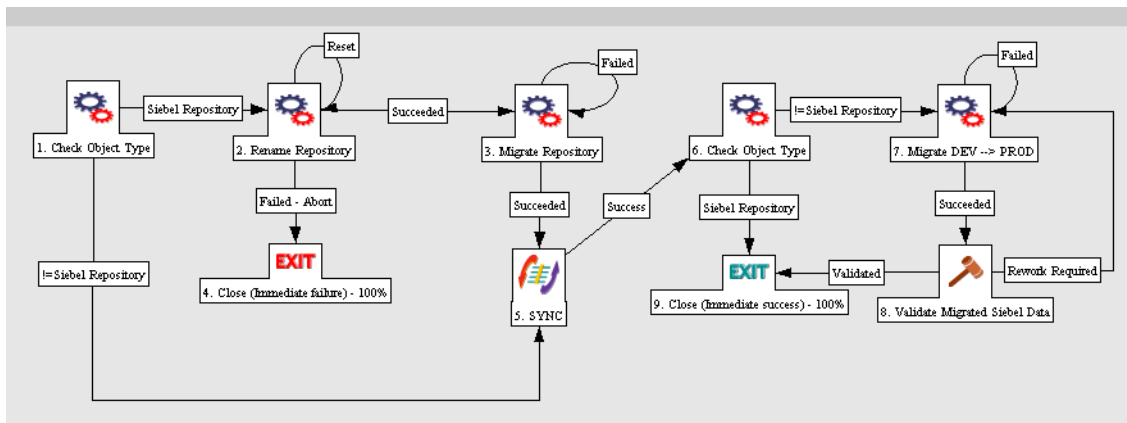*Figure 4-10* shows the workflow diagram. Steps in the workflow are described in the procedure that follows.

Figure 4-10. Siebel Migrate Repository and Data Workflow

1. Check Object Type—This is an immediate execution step that evaluates the package line's object type. If the object type is Siebel Repository, the

package line proceeds to the Rename Repository step. For all other Siebel object types, the workflow branches to the SYNC step.

2. Rename Repository—This step renames the destination repository. Siebel eBusiness Applications do not allow overwriting of repositories. Existing repositories must be renamed before a new, identically named version is imported.

3. Migrate Repository—Once the Siebel repository has been renamed, this step becomes eligible. This step has a reset loop so that the repository migration can be resubmitted in the event of failure. Once the migration has succeeded, the workflow proceeds to the SYNC step.

4. Close (Immediate failure)—If the Rename Repository step failed, the workflow branches to this step.

5. SYNC—After all package lines meet at this step, indicating successful migration of the Siebel Repository, the Siebel configuration data can be migrated.

6. Check Object Type—This step again determines the object type of each package line. Package lines using the Siebel Repository object type automatically proceed to the Close (Immediate success) step. All other Siebel object types proceed to the Migrate DEV --> PROD execution step.

7. Migrate DEV --> PROD—This step migrates the Siebel configuration data from the development environment to the production environment. This step has a reset loop so that the migration can be resubmitted in the event of failure. Once the migration has succeeded, the workflow proceeds to the Validate Migrated Siebel Data step

8. Validate Migrated Siebel Data—When the Validate Migrated Siebel Data step becomes eligible, users should verify the integrity of the migrated data by checking the Siebel EIM task logs (which are attached to the Mercury Change Management execution log) and logging into the application to verify the data using the Siebel GUI.

If the data appears to be complete and correct, users should select the Validated transition from this step. This causes the package line to proceed to the Close (Immediate success) step.

Otherwise, users should select the Rework Required transition. This causes the workflow to return to the Migrate DEV --> PROD execution step, where necessary adjustments can be made and the data can be migrated again.

9. Close (Immediate success)—This step represents successful completion of the workflow.

Siebel Migrate Repository and Data - Multi-Step Workflow

The Siebel Migrate Repository and Data - Multi-Step workflow is similar to the Siebel Migrate Repository and Data workflow. The only difference is inclusion of a subworkflow to define the repository migration process. This subworkflow divides the repository migration process into several steps and includes some recommended pre- and post-migration tasks.

> **Note**
>
> The Siebel Migrate Repository and Data - Multi-Step workflow, the Siebel Migrate Repository subworkflow, and the Siebel Repository - Multi-Step object type are designed to be used together. Leaving out one or more of these components will result in incomplete executions.

Figure 4-11 shows the workflow diagram. *Figure 4-12* shows the subworkflow diagram. Steps in the workflow and the subworkflow are described in the procedure that follows.
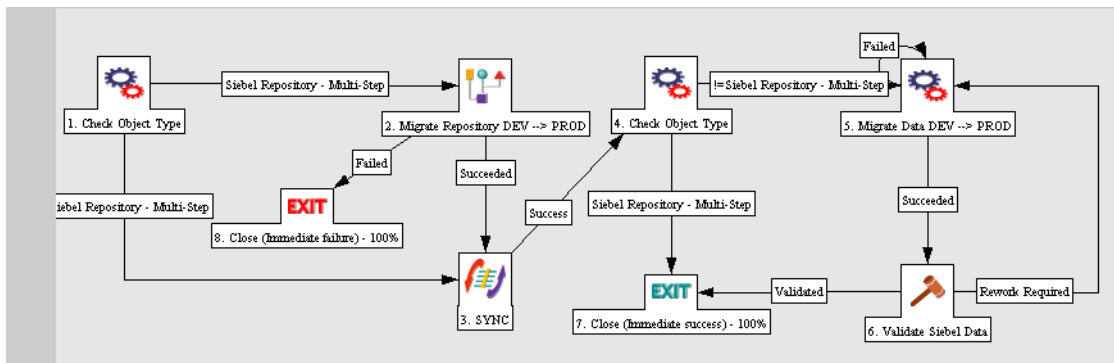

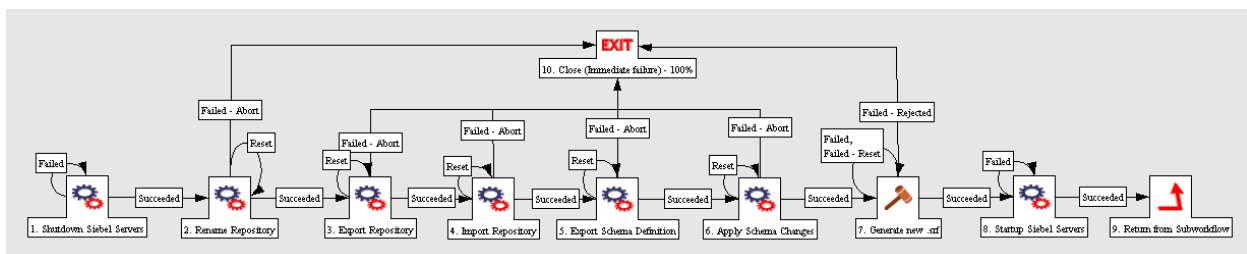*Figure 4-11. Siebel Migrate Repository and Data Multi-Step workflow*

Figure 4-12. Siebel Migrate Repository Subworkflow

1. Check Object Type—This is an immediate execution step that evaluates the package line's object type. If the object type is Siebel Repository - Multi-Step, the package line proceeds to the Migrate Repository DEV --> PROD step. For all other Siebel object types, the workflow branches to the SYNC step

2. Migrate Repository DEV --> PROD—For Siebel Repository - Multi-Step object types, initiate the Siebel Migrate Repository Subworkflow.

Siebel Migrate Repository Subworkflow steps:

2.1. Shutdown Siebel Servers—This is an execution step that shuts down both the source and destination Siebel Servers. An environment group with all Siebel Servers can be used to make sure that all Siebel Servers are shut down prior to a Repository migration.

2.2. Rename Repository—This step renames the destination repository. Siebel eBusiness Applications do not allow overwriting of repositories. Existing repositories must be renamed before a new, identically named version is imported.

2.3. Export Repository—This step exports the Siebel Repository from the source Siebel instance.

2.4. Import Repository—This step imports the Siebel Repository into the destination Siebel instance.

2.5. Export Schema Definition—This step determines schema differences between the new and old repositories.

2.6. Apply Schema Changes—This step applies data model changes to the destination schema.

2.7. Generate new .srf—This step reports whether or not a new `srf` has been generated for the destination Siebel Repository. You must have an `srf` file on the destination server that matches the latest version of the Siebel Repository. There are several ways to manage `srf` files following a Repository migration:

- Generate a new `srf` file through Siebel Tools on the destination environment.

- Migrate the source `srf` file to the destination environment.

- Incorporate version control into the migration process and check out the `srf` file from a version control system.

2.8. Startup Siebel Servers—This step starts both the source and destination Siebel Servers.

2.9. Return from Subworkflow—Return to the main workflow. If migration was successful, proceed to the SYNC step. If not, proceed to the Close (Immediate failure) step.

2.10. Close (Immediate failure)—If failure occurred at any point prior to the Startup Siebel Servers subworkflow step, exit with failure status.

3. SYNC—After all package lines meet at this step, indicating successful migration of the Siebel Repository, the Siebel configuration data can be migrated.

4. Check Object Type—This step again determines the object type of each package line. Package lines using the Siebel Repository - Multi-Step object type automatically proceed to the Close (Immediate success) step. All other Siebel object types proceed to the Migrate DEV --> PROD execution step.

5. Migrate Data DEV --> PROD—This step migrates the Siebel configuration data from the development environment to the production environment. This step has a reset loop so that the migration can be resubmitted in the event of failure. Once the migration has succeeded, the workflow proceeds to the Validate Siebel Data step.

6. Validate Siebel Data—When the Validate Siebel Data step becomes eligible, users should verify the integrity of the migrated data by checking the Siebel EIM task logs (which are attached to the Mercury Change Management execution log) and logging into the application to verify the data using the Siebel GUI.

If the data appears to be complete and correct, users should select the Validated transition from this step. This causes the package line to proceed to the Close (Immediate success) step.

Otherwise, users should select the Rework Required transition. This causes the workflow to return to the Migrate Data DEV --> PROD execution step, where necessary adjustments can be made and the data can be migrated again.

7. Close (Immediate success)—This step represents successful completion of the workflow.

8. Close (Immediate failure)—If the Migrate Repository DEV --> PROD step (calling the subworkflow) failed, the workflow branches to this step.

Siebel Project Deployment Workflow

The Siebel Project Deployment workflow is used for migrations of projects. After each project migration, it uses the Siebel Project Compile Subworkflow to compile a new `srf` file and replace the `srf` file on the destination server.

Figure 4-13 shows the workflow diagram. *Figure 4-14* shows the subworkflow diagram. Steps in the workflow and the subworkflow are described in the procedure that follows.
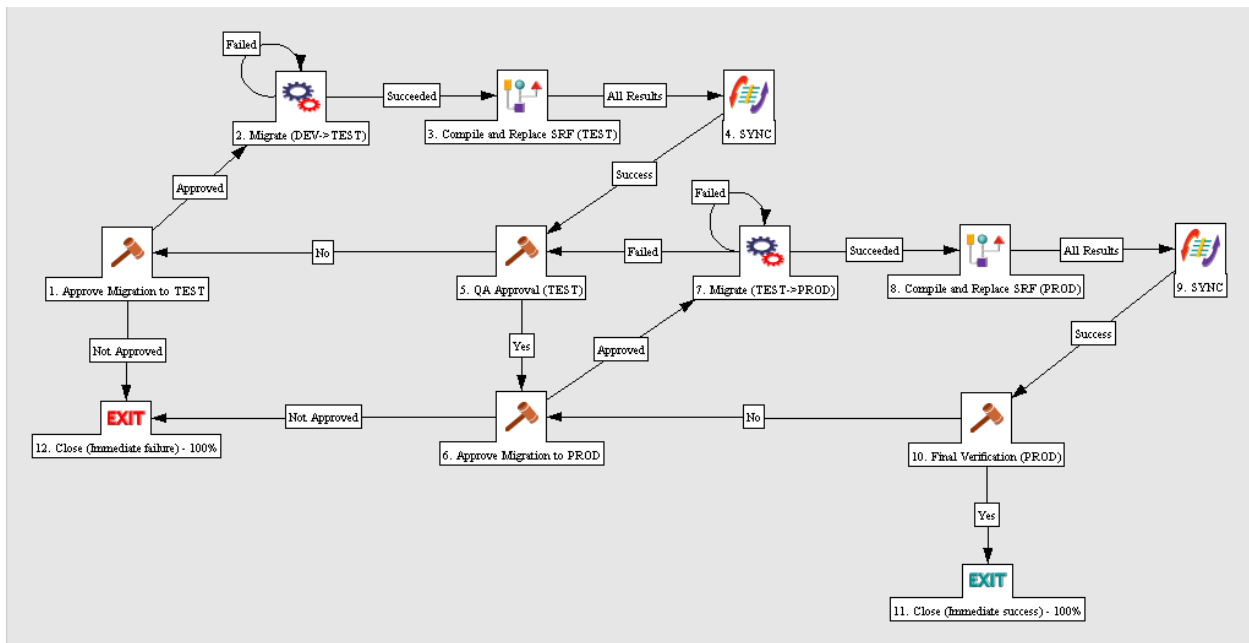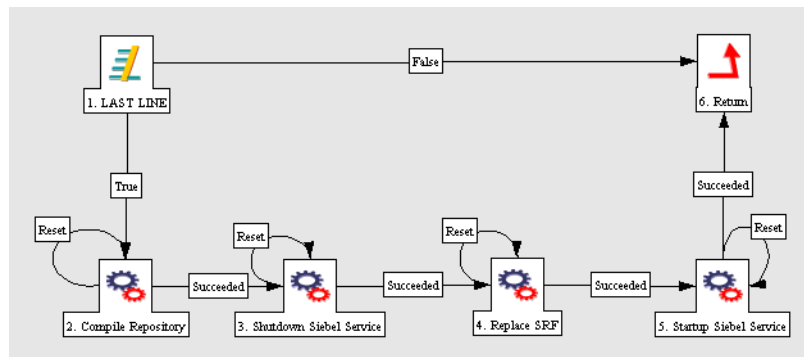

Figure 4-13. Siebel Project Deployment workflow

Figure 4-14. Siebel Project Compile Subworkflow

1. Approve Migration to Test— In this step, the user approves the migration of projects to the test instance.

2. Migrate (DEV -> TEST)—Migrate the projects from the development instance to the test instance.

3. Compile and Replace SRF (TEST)—When all the projects have migrated, then invoke the Siebel Project Compile Subworkflow to compile the new `srf` file and replace the `srf` at the destination server.

Siebel Project Compile Subworkflow steps:

3.1. LAST LINE—This step checks if the line is the last package line. If so, proceed to the Compile Repository step; if not, return to the main workflow. The compile is required only once for the project being migrated.

3.2. Compile Repository—Compile the Siebel Repository.

3.3. Shutdown Siebel Service—Shut down the Siebel Service.

3.4. Replace SRF—Replace the destination `srf` file.

3.5. Startup Siebel Service—Start up the Siebel Service.

3.6. Return—Return to the main workflow.

4. SYNC—All package lines wait for the `srf` to be compiled and deployed.

5. QA Approval (TEST)—In this step, the user approves the changes on the test instance.

6. Approve Migration to PROD—In this step, the user approves the migration of projects to the production instance.

7. Migrate (TEST -> PROD)—Migrate the projects from the test instance to the production instance.

8. Compile and Replace SRF (PROD)—This step calls the Siebel Project Compile Subworkflow to compile the new `srf` file and replace the `srf` at the destination server. See the details of this subworkflow in the preceding Compile and Replace SRF (TEST) step.

9. SYNC—All package lines wait for the `srf` to be compiled and deployed.

10. Final Verification (PROD)—In this step, the user approves the changes on the production instance.

11. Close (Immediate success)—This step represents successful completion of the workflow.

12. Close (Immediate failure)—This step reflects cancellation of the migration by the user.

Report Types

This section provides reference information about the Siebel-specific report types provided in the Extension. These report types are listed and defined in *Table 4-13*.

Overview of Report Types

The content or output of a report is controlled by what you specify in the fields for the report type, as described in this section. Some fields allow multiple entries. Some fields are hidden by default, but can be enabled to allow further control of the report output.

Execution of reports is driven by commands included within the report types. For more information about commands in the Mercury IT Governance Center environment, see the *Commands, Tokens, and Validations Guide and Reference*.

Report types are configured in the Workbench (in the shortcut bar, click **Configuration** and click the **Report Types** icon). Reports are run (submitted) from the standard interface (**Reports** menu).

For More Information

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

Reference Report Types

Reference (REFERENCE) report types cannot be edited, but you can copy and rename them and edit the copies to meet your needs. You can also use existing non-reference report types as is or configure them further to meet your needs.

List of Report Types

Table 4-13 lists and defines the report types included in the Extension. Each is described in subsequent sections.

Table 4-13. Report types included in the Extension

Report Type Name	Definition
Siebel Object Migrations Report	Lists migrations that were executed using any of the Siebel object types.
Siebel Request Detail Report	Lists details of requests that use Siebel request types, based on criteria you specify.

Siebel Object Migrations Report

The Siebel Object Migrations Report provides a list of migrations that were executed using any of the Siebel object types based on criteria you specify. The report lists the following fields:

- Package Number
- Package Seq
- Object Name
- App Code
- Workflow Step Name
- Last Migration Date
- User
- Destination environment

Figure 4-15 shows the Submit Report window you use to specify which object types and other criteria will be used to generate the Siebel Object Migrations Report. *Table 4-14* provides definitions for the displayed fields.

Figure 4-15. Siebel Object Migrations Report

Table 4-14. Siebel Object Migrations Report field definitions

Field Name (*Required)	Definition
*Siebel Object Type	Name of the object type.
Siebel Object Name	Name of the migrated Siebel Object.
App Code	Application Code, if used in your implementation.
Destination Environment	Destination environment to which objects were migrated.
Transaction Date From	Beginning of date range when migration took place.
Transaction Date To	End of date range when migration took place.
Performed by	User who performed the migration.
Order by	Criterion to use to sort the objects in the report: **Siebel Object Name**, **Destination Environment**, or **Package**.

Siebel Request Detail Report

The Siebel Request Detail Report provides a detailed list of Mercury IT Governance Center requests that use the Siebel request types, based on criteria you specify.

Figure 4-16 shows the Submit Report window you use to specify which requests you want to appear in the Siebel Request Detail Report and which information you want to appear for each request. *Table 4-15* provides definitions for the displayed fields.

Figure 4-16. Siebel Request Detail Report

Table 4-15. Siebel Request Detail Report field definitions

Field Name (*Required)	Definition
Request Numbers	Request numbers.
*Include Closed Requests	Whether or not to include requests that have been closed or cancelled.
Siebel Request Type	Request type on which the report can be run.
Status	Requests having this status.
Priority	Requests having this priority.
Assigned to	Requests assigned to this user.
Assigned to Group	Requests assigned to this group.
Created By	Requests created by this user.
Request Sub Type	Requests having this sub-type.
Department	Requests logged against this department.
Application	Requests that are associated with this application.
Workflow	Requests that are associated with this workflow.
Request Group	Requests using request group(s) **Customization**, **Upgrade**, or **Setup**.
Contact	Requests having this contact name.
Company Name	Requests having this company name.
Creation Date From	Requests created on this date or later.
Creation Date To	Requests created on this date or earlier.
Last Update Date From	Requests last updated on this date or later.
Last Update Date To	Requests last updated on this date or earlier.
Description Contains	Requests having this description text string.
Report Title	Title of the report.
*Order By	Criterion to use to sort the requests in the report: **Request Number**, **Application**, **Assigned To**, **Created By**, **Creation Date**, **Department**, **Last Update Date**, **Priority**, **Request Group**, **Request Sub Type**, **Request Type** or **Status**.
*Show Header Fields	Whether or not the report will show the full header fields of each request.
*Show Detail Fields	Whether or not the report will show the detail fields of each request.

Table 4-15. Siebel Request Detail Report field definitions [continued]

Field Name (*Required)	Definition
*Hide Prompts for Empty Fields	Whether or not the report will hide prompts that have empty fields.
*Show Contents of Table Fields	Whether or not the report will show table fields for requests that have them.
*Show Field Audit History	Whether or not the report will show the transaction history of each request.
*Show Notes	Whether or not the report will show the notes attached to each request.
*Filter Notes	Whether the report will show all notes or only user notes.
*Show Status	Whether or not the report will show the workflow steps and current step status for each request.
*Show References	Whether or not the report will show the references associated with each request.

Special Commands

Table 4-16 lists and defines the special commands included with Mercury Change Management Extension for Siebel eBusiness Applications.

Note	When working with the special commands, verify that you are using the command that corresponds to your Siebel product version. For example, special commands for use with Siebel 2000 are labeled with `ksc_sbl2000_`.

Table 4-16. Special commands included in the Extension

Special Command Name	Called By	Description
ksc_sbl_attach_log_ref	Object type	Attaches the EIM log to the Mercury IT Governance package line log.
ksc_sbl_check_task_stat	ksc_sbl_export_process ksc_sbl_import_process	Checks EIM task status for completion.
ksc_sbl_copy_log	Object type	Copies the EIM log to Mercury IT Governance Server.
ksc_sbl_create_export_config	ksc_sbl_export_process	Generates Siebel ifb file and input file to export data.
ksc_sbl_create_import_config	ksc_sbl_import_process	Generates Siebel ifb file and input file to import data.
ksc_sbl_export_process	Object type	Generates export ifb file, runs export, and extracts data from an interface table.
ksc_sbl_export_repository		Exports data into a repository data file at the Siebel source environment using repimexp.exe.
ksc_sbl_get_eim_task_id	ksc_sbl_export_process ksc_sbl_import_process	Obtains an EIM task id for a submitted task.

Table 4-16. Special commands included in the Extension [continued]

Special Command Name	Called By	Description
ksc_sbl_import_process	Object type	Copies an export data file from ksc_sbl_create_ export_process, generates an import ifb file, runs import, checks for errors, and prints a summary of the import.
ksc_sbl_import_repository	For custom use	Imports data from the repository data file at the Siebel destination environment using repimexp.exe.
ksc_sbl_print_log_header	Object type	Prints an EIM task header in the Mercury Change Management execution log.
ksc_sbl_print_summary	ksc_sbl_import_process	Prints a summary of an import based on the data and status of a Siebel interface table.
ksc_sbl_rename_repository	Workflow step source	Renames the destination repository.
ksc_sbl_run_export	ksc_sbl_export_process	Runs an EIM task to export data. A Siebel input file must exist with a name in the following format: `PKG.ID PKG.PKGL.SEQ_ P.INTERFACE_TABLE_ input.txt` Typically, this file is generated as a part of ksc_ sbl_export_process.

Table 4-16. Special commands included in the Extension [continued]

Special Command Name	Called By	Description
ksc_sbl_run_import	ksc_sbl_import_process	Runs an EIM task to export data. A Siebel input file must exist with a name in the following format: *PKG.ID PKG.PKGL.SEQ_P.INTERFACE_TABLE_*input.txt Typically, this file is generated as a part of ksc_sbl_import_process.
ksc_sbl_run_sql	Validations and object types	Runs an SQL statement against a Siebel database.
ksc_sbl2000_attach_log_ref	Object type	Attaches an EIM log to a Mercury IT Governance package line log.
ksc_sbl2000_check_DBXTRACT_status	Siebel Extract Mobile Client DB object type	Checks the status of the client database extraction process.
ksc_sbl2000_check_task_stat	ksc_sbl2000_export_process ksc_sbl2000_import_process	Polls the EIM task status for completion.
ksc_sbl2000_copy_eim_table	Object type	Copies EIM table rows from source to destination database.
ksc_sbl2000_copy_log	Object type	Copies an EIM log to Mercury IT Governance Server.
ksc_sbl2000_create_export_config	ksc_sbl2000_export_process	Generates a Siebel ifb file and input file to export data.
ksc_sbl2000_create_import_config	ksc_sbl2000_import_process	Generates a Siebel ifb file and input file to import data.
ksc_sbl2000_create_mobile_client_file	Siebel Extract Mobile Client DB object type	Creates a list of mobile clients whose databases need extraction.
ksc_sbl2000_DBXTRACT	Siebel Extract Mobile Client DB object type	Performs an extract of client databases.
ksc_sbl2000_export_bu_name	Object type sc_sbl2000_export_bu_name	Creates an SQL script for populating the Business Unit name in EIM tables.

Table 4-16. Special commands included in the Extension [continued]

Special Command Name	Called By	Description
ksc_sbl2000_export_process	Object type	Generates export ifb file, runs export, and checks for errors in export.
ksc_sbl2000_export_repository		Exports the Siebel repository into a datafile (.dat)
ksc_sbl2000_Generate_DB_Template	SBL - Generate New DB Template workflow step source	Generates a new database template. See *Using the ksc_sbl2000_Generate_DB_Template Special Command* on page 84.
ksc_sbl2000_get_eim_task_id	ksc_sbl2000_export_process ksc_sbl2000_import_process	Obtains an EIM task id for submitted task.
ksc_sbl2000_import_process	Object type	Generates the import ifb file, runs import, checks for errors, and prints a summary of import.
ksc_sbl2000_import_repository		Imports the Siebel repository from the .dat file.
ksc_sbl2000_print_log_header	Object type	Prints EIM task header in the Mercury IT Governance execution log.
ksc_sbl2000_print_summary	ksc_sbl2000_import_process	Prints a summary of import based on data and status in the Siebel interface table.
ksc_sbl2000_run_export	ksc_sbl2000_export_process	Runs EIM task to export data.
ksc_sbl2000_run_export_sql	ksc_sbl2000_export_bu_name	Runs the SQL script for populating the Business Unit name in the EIM tables. A Siebel input file must exist with a name in the following format: `PKG.ID PKG.PKGL.SEQ P.INTERFACE_TABLE_ input.txt` Typically, this file is generated as a part of ksc_ sbl_export_process.

Table 4-16. Special commands included in the Extension [continued]

Special Command Name	Called By	Description
ksc_sbl2000_run_import	ksc_sbl2000_import_process	Runs an EIM task to import data. A Siebel input file must exist with a name in the following format: `PKG.ID PKG.PKGL.SEQ_P.INTERFACE_TABLE_input.txt` Typically, this file is generated as a part of ksc_sbl_import_process.
ksc_sbl2000_shutdown_siebel_server	SBL - Shutdown Siebel Servers workflow step source	Calls the srvrmgr utility to shut down a Siebel Server.
ksc_sbl2000_startup_siebel_server	SBL - Startup Siebel Servers workflow step source	Calls the srvrmgr utility to start a Siebel Server.
ksc_sbl7_export_bu_name	Object type	Creates a SQL script for populating the Business Unit name in EIM tables for Siebel 7.
sc_sbl2000_export_bu_name	Object type	Calls the ksc_sbl2000_export_bu_name special command. Add new SQL to this command.

Using the ksc_sbl2000_Generate_DB_Template Special Command

The `ksc_sbl2000_Generate_DB_Template` special command is used for the administration of Siebel Remote Clients. A new database must be generated whenever any schema changes are made. The `ksc_sbl2000_Generate_DB_Template` command creates the local database template for a given database schema version. It reads the database schema definition from the Siebel Repository. It then creates Siebel tables and indexes in a database template file stored on the Siebel Remote Server in the `dbtempl/` subdirectory. The Local Database Initialization program uses the local database template when initializing a new database on the mobile client.

You must generate a new database template whenever the Siebel database schema changes as soon as you:

- Install the Siebel database server

- Upgrade to a new version of Siebel applications

- Extend the database schema using Siebel Tools

In general, the database template must be created after a repository migration only if there are changes made to the database. However, because creating a database template does not take very long and is automated, Mercury suggests doing so after any repository migration.

The `ksc_sbl2000_Generate_DB_Template` command should be added to a new execution workflow step source (see *Figure 4-17*), which can then be added to the Siebel Migrate Repository Subworkflow.
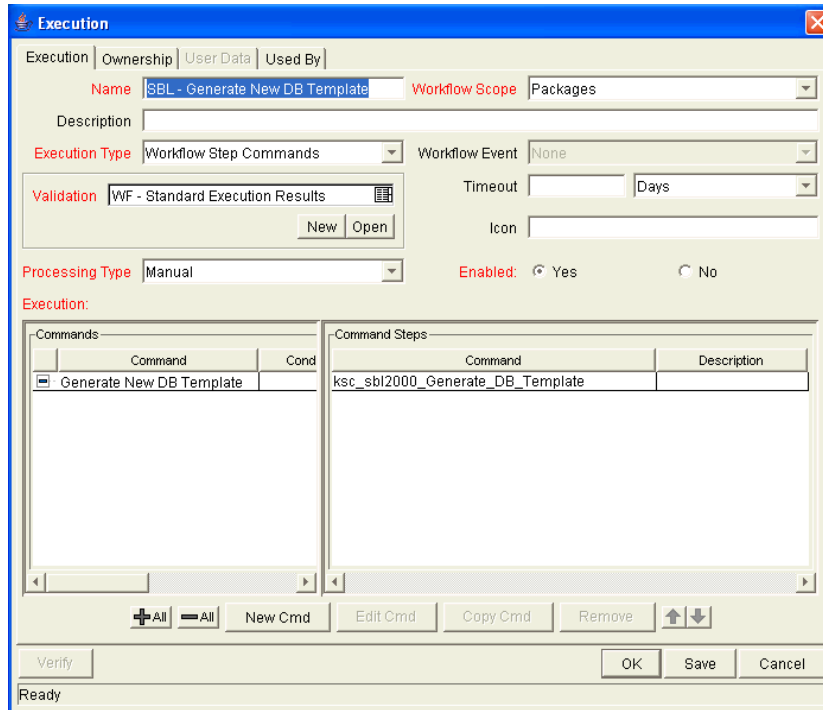
Figure 4-17. Special command added to execution workflow step source

Validations

The Extension includes a number of common validations specifically related to Mercury IT Governance entities such as object types, request types, and report types. *Table 4-17* lists and defines these validations.

For More Information

For information about using and configuring validations, see the document titled *Commands, Tokens, and Validations Guide and Reference*.

Table 4-17. Validations included in the Extension

Validation Name	Component Type	Description
SBL - Agreements	Auto Complete List	List of Siebel Agreements
SBL - Application Version	Drop Down List	List of Siebel Application Versions
SBL - Assignment Rule Groups	Auto Complete List	List of Siebel Assignment Rule Groups
SBL - Assignment Rules	Auto Complete List	List of Siebel Assignment Rules
SBL - Assignment Workloads	Auto Complete List	List of Assignment Workloads
SBL - Base Tables	Drop Down List	Base tables to import
SBL - Competitive Metrics	Auto Complete List	List of Competitive Metrics
SBL - Decision Requests	Auto Complete List	List of Decision Requests
SBL - Employee	Auto Complete List	List of Employees
SBL - Employee (EAI)	Auto Complete List	List of Employees
SBL - Entitlements	Auto Complete List	List of Siebel Entitlements
SBL - Environments	Auto Complete List	List of Siebel Environments
SBL - Exchange Rates	Auto Complete List	List of Exchange Rates
SBL - Industries	Auto Complete List	List of Industries
SBL - List of Values	Auto Complete List	List of Value Types
SBL - Mobile Clients	Auto Complete List	List of Siebel Mobile clients
SBL - Object Migrations Report Order By	Drop Down List	List of ways the report can be sorted.
SBL - Object Types	Auto Complete List	List of Siebel Object Types
SBL - Organizations	Auto Complete List	List of Internal Organizations
SBL - Positions	Auto Complete List	List of Positions

Table 4-17. Validations included in the Extension [continued]

Validation Name	Component Type	Description
SBL - Pre-Defined Query	Auto Complete List	List of Pre-Defined Queries
SBL - Price Lists	Auto Complete List	List of Price Lists
SBL - Product Lines	Auto Complete List	List of Product Lines
SBL - Products	Drop Down List	List of Products
SBL - Project Import Mode	Drop Down List	List of Siebel Project Import Modes
SBL - Projects	Auto Complete List	List of Siebel Projects
SBL - Request Assigned	Drop Down List	List of Siebel Requests Assigned
SBL - Request Numbers	Auto Complete List	List of Siebel Request Numbers
SBL - Request Types	Auto Complete List	List of Siebel Request Types
SBL - Responsibilities	Auto Complete List	List of Responsibilities
SBL - Selection Type	Drop Down List	List of methods for selecting values to be exported and imported: **Single**, **Range**, **Wildcard**, or **All**.
SBL - Selection Type (restricted)	Drop Down List	List of methods for selecting values to be exported and imported: **Single** or **All**.
SBL - SQL*Plus Executables	Drop Down List	List of Siebel SQL*Plus Executables
SBL - State Models	Auto Complete List	List of State Models
SBL - States	Auto Complete List	List of States
SBL - Static Application List	Drop Down List	List of Siebel Applications
SBL - Target Table List	Auto Complete List	List of Siebel base tables corresponding to a given interface table
SBL - Validation Step	Auto Complete List	List of states associated with validating data after migration
SBL - Views	Auto Complete List	List of Views
SBL - Workflow Actions	Auto Complete List	List of Workflow Actions
SBL - Workflow Groups	Auto Complete List	List of Workflow Groups
SBL - Workflow Processes	Auto Complete List	List of Workflow Processes
SBL - Workflow Rules	Auto Complete List	List of Workflow Rules

Tokens

While configuring certain features in Mercury IT Governance Center, it is often necessary to refer to information that is undefined until it is used in a specific context. Instead of generating objects that are valid only in specific contexts, variables can be used to create general objects that can be applied to a variety of contexts. These variables are called tokens.

Tokens can be used in (but are not limited to) the following Mercury IT Governance entities:

- Object types

- Request types

- Validations and SQL statements

- Report types

- Workflow executions and notifications

- Workflow steps

Extension-specific tokens are available to be used only after you install the Extension. *Table 4-18* describes the standard environment tokens included with the Extension.

For More Information

For information about using and configuring tokens, see the *Commands, Tokens, and Validations Guide and Reference*.

Table 4-18. Tokens included in the Extension

Prefix	Token	Description
ENV.AC	SBL_ADMIN_PASSWORD	Siebel Admin Password
ENV.AC	SBL_ADMIN_USER	Siebel Admin User
ENV.AC	SBL_APP_VER	Siebel Version
ENV.AC	SBL_CFG_FILE	Siebel CFG File
ENV.AC	SBL_ENTERPRISE_SERVER	Siebel Enterprise Server Name
ENV.AC	SBL_GATEWAY_SERVER	Siebel Gateway Server Name
ENV.AC	SBL_ODBC_DATA_SRC	Siebel ODBC Data Source
ENV.AC	SBL_PASSWORD	Siebel Table Owner Password
ENV.AC	SBL_SERVER_NAME	Siebel Server Name
ENV.AC	SBL_SQLPLUS	SQL*Plus Executable for Oracle Database
ENV.AC	SBL_TBL_OWNER	Siebel Table Owner

Index

K

kDeploy.sh script **23**, **24**

ksacom utility **33**

ksc_sbl2000_export_bu_name Special
Command **44**

L

license agreement for the Extension **22**

logs **25**
    ddlDriver.#####.log **25**
    jarxvf.#####.log **25**
    packageDriver.#####.log **25**
    postXMLDriver.#####.log **25**
    preXMLDriver.#####.lo **25**

M

migrating entities **31**

mitg_extensions.sh script **23**

mitg-600-Siebel.jar file **21**, **22**, **23**

O

object types
    Siebel Agreement **39**
    Siebel Assignment Rule **39**
    Siebel Assignment Rule Group **39**
    Siebel Assignment Workload **39**
    Siebel Competitive Metrics **39**
    Siebel Decision Issues **40**
    Siebel Employee **40**
    Siebel Employee (EAI) **45**
    Siebel Employee Detail **40**
    Siebel Entitlements **40**
    Siebel Exchange Rate **40**
    Siebel Industry **40**
    Siebel List of Values **40**
    Siebel Organization **40**
    Siebel Position **40**
    Siebel Pre-Defined Query **40**
    Siebel Price List **40**
    Siebel Product **40**
    Siebel Product Lines **40**

Siebel Project **53**
Siebel Repository **49**
Siebel Repository - Multi-Step **51**
Siebel Responsibility **41**
Siebel State Model **41**
Siebel States and Transitions **41**
Siebel Views **41**
Siebel Workflow Action **41**
Siebel Workflow Group **41**
Siebel Workflow Process (EAI) **45**
Siebel Workflow Rule **41**

overview of the Extension **14**

P

packageDriver.#####.log **25**

postXMLDriver.#####.log **25**

prerequisites
    documents **13**
    knowledge and experience **16**

preXMLDriver.#####.log **25**

R

release 5.0.0, upgrading from **19**

release 5.5.0, upgrading from **19**

release 6.0.0, what's new in **15**

reports
    Siebel Object Migration **74**
    Siebel Request Detail **76**

request type, Siebel Enhancement **57**

S

sc_sbl2000_export_bu_name Special
Command **44**

scripts
    kDeploy.sh **23**, **24**
    mitg_extensions.sh **23**

Siebel Agreement object type **39**

Siebel Assignment Rule Group object type **39**

Siebel Assignment Rule object type **39**

Siebel Assignment Workload object type **39**