Mercury IT Governance Center™

Mercury Object Migrator Guide™

Version: 6.0

MERCURY™

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

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

Publication Number: ITG60ObjectMigrator1104A

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

In This Chapter:

About This Document

Mercury Object Migrator™ automates the deployment of AOL (Application Object Library) setups between Oracle E-Business Suite instances, potentially saving you time, money, and reducing errors.

This document provides the following information about Object Migrator (OM).

- This chapter provides an overview of the Mercury Object Migrator and lists the chapters found in this document.
- Chapter 2, *System and Functional Overview*, on page 19 provides an overview of the product and components as well as the fundamental mechanisms for its use.
- Chapter 3, *Installation and Upgrade Overview*, on page 29 describes the requirements, considerations, and recommendations for the installation and use of Mercury Object Migrator.
- Chapter 4, *Installing Object Migrator*, on page 35 details the installation and configuration procedures.
- Chapter 5, *Upgrading Object Migrator*, on page 61 details the procedure to upgrade from an earlier release of Object Migrator.
- Chapter 6, *Optional Configurations for Object Migrator*, on page 99 describes additional configuration options.
- Chapter 7, *Maintaining Object Migrator*, on page 113 describes how to maintain the various Object Migrator components.
- Chapter 8, *Migrating Objects*, on page 127 details how to migrate objects with the product.
- Chapter 9, *Comparing Objects*, on page 139 explains how to use Object Migrator to compare objects.
- Chapter 10, *Using Version Control*, on page 147 details the Mercury Object Migrator version control functionality.
- Chapter 11, *Using Object Migrator with Mercury Change Management*, on page 155 describes how to use the product with the Mercury Change Management™ component of the Mercury IT Governance Center™.
- Appendix A, *AOL-Dependent Objects*, on page 161 lists the objects supported by Object Migrator.

- Appendix B, *Object Type-Specific Migrators and Migration Rules*, on page 165 provides the rules governing object migration.
- Appendix C, *Exception Messages*, on page 235 lists the exception messages, typical causes, and suggested remedial actions.

Who Should Read This Document

This document is for the following users:

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

For More Information

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

Prerequisite Documents

Prerequisite documents include:

- *Getting Started*
- *Key Concepts*
- *System Administration*
- *Security Model Guide and Reference*

For More Information

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

Related Documents

Supplemental documents include:

- *Commands, Tokens, and Validations Guide and Reference*
- *Mercury Change Management User's Guide*
- *Mercury Change Management: Configuring a Deployment System*

For More Information

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

Overview of the Object Migrator

When business processes change, Oracle E-Business Suite customers often have to reconfigure their systems to accommodate the change. AOL configuration changes often result from such business process changes and, without Mercury Object Migrator, AOL configuration changes typically have do be done manually.

Every Oracle E-Business Suite customer (regardless of whether they are currently implementing, upgrading, or maintaining Oracle E-Business Suite) has the challenge of configuring AOL (Application Object Library) objects to meet its needs. As each object is configured, the object must be migrated to the multiple environments (Development, QA, Staging, Production, and so forth) the customer uses to manage its E-Business Suite implementation. Mercury Object Migrator can help you in the following ways:

- With Object Migrator, each AOL setup deployment does not require the re-keying of the setup information—a complex, repetitive, and error prone manual task requiring hours of database administrator, system administrator, and developer time.
- Identifying if a migration was successful requires manual verification, as does fixing an improper migration. With Object Migrator, comparison reports list the differences between objects in different environments, and validations minimize errors.
- If an improper object is still migrated, Object Migrator's version control functionality enables the change to be automatically rolled back.

As business needs change or applications are patched or upgraded, AOL setups change and need to be deployed to affected instances. With Object Migrator you'll know what has changed, what needs to change, and save hundreds of hours implementing and maintaining the AOL setups that support your critical business processes.

Details about the Mercury Object Migrator architecture and features are discussed in the next chapter, *System and Functional Overview*.

Chapter

2

System and Functional Overview

In This Chapter:

Architecture Overview

Mercury Object Migrator is a custom application installed on an Oracle E-Business Suite instance. Its default application short name is CLM, and its full application name is Mercury Object Migrator.

Object Migrator has its own database schema. This schema contains the Object Migrator database objects and communicates with the APPS schema. These are required in order to migrate Application Object Library (AOL) setup data, as well as retain archived versions of objects and capture migration audit data.

Object Migrator programs are registered as concurrent programs. These programs utilize the standard security and concurrent processing features available within the Oracle E-Business Suite, such as concurrent program registrations, request groups, and responsibilities, to control their use.

Object Migrator has the flexibility to be installed on either the source or destination instance of data migrations, or on a completely independent instance. This is because it communicates with other instances using SQL*Net and database links created in the Object Migrator schema. The database links are defined to provide a link from the Object Migrator schema to the APPS schema of a given instance. If Object Migrator is installed in an instance that is the source or destination of a migration, the Object Migrator schema will contain a database link pointing to the APPS schema of the local database.

Figure 2-1 illustrates the relationship between the Object Migrator schema, APPS, source, and destination databases.

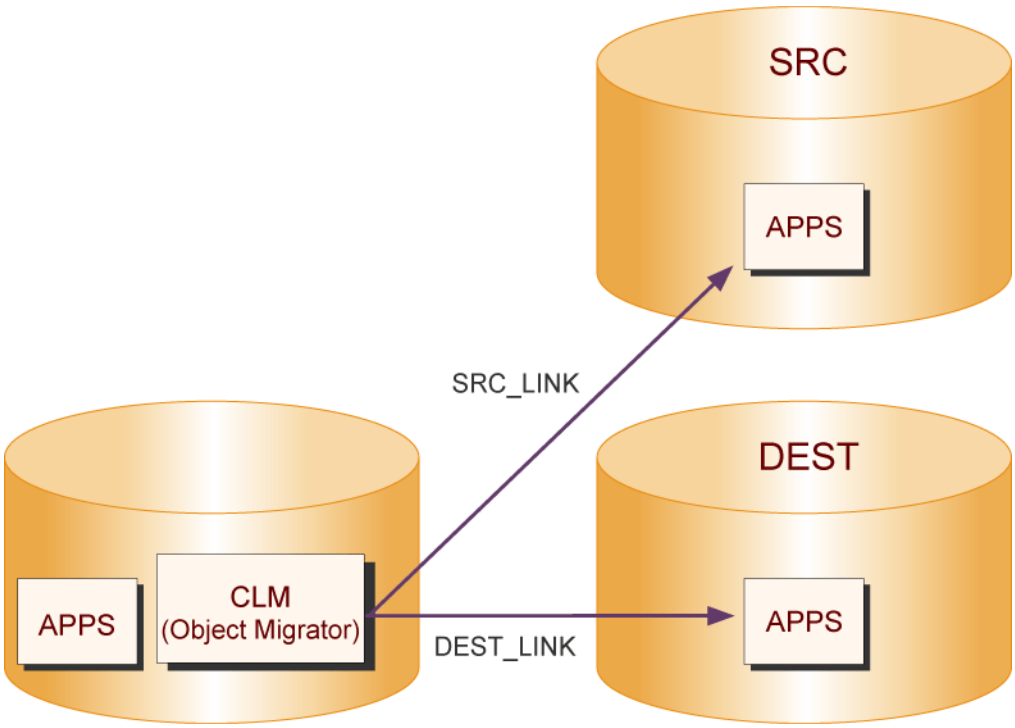


Figure 2-1. Object Migrator architecture

Product Components

Mercury Object Migrator relies on three main components to install properly. The following sections discuss these components:

- Reports
- SQL scripts
- Shell scripts

Reports

Report files are executed as concurrent programs launched from the Submit Requests form (also known as the Run Reports form). Oracle reports files have `rdf` file extensions. There is one report file for each of the Object Migrator modules. Some Object Migrator modules are not applicable to all Oracle E-Business Suite releases.

Table 2-1 describes the reports used to perform AOL migrations. *Table 2-2* describes the reports used to perform system administration tasks.

Table 2-1. Reports used to perform AOL migrations

Report Name	Used to Migrate
CLMRMCL1.rdf	QuickCodes
CLMRMCM1.rdf	Concurrent Manager Definitions
CLMRMCP1.rdf	Concurrent Programs
CLMRMDF1.rdf	Descriptive Flexfields
CLMRMFD1.rdf	Folders
CLMRMFN1.rdf	Functions
CLMRMFS1.rdf	FSG Row/Column Sets
CLMRMGM1.rdf	GUI Menus
CLMRMHT1.rdf	Help Text
CLMRMMN1.rdf	Menus
CLMRMMS1.rdf	Messages
CLMRMNS1.rdf	Named SQL
CLMRMPO1.rdf	Profile Options
CLMRMPT1.rdf	Printer Definitions
CLMRMRG1.rdf	Report Groups
CLMRMRP1.rdf	Responsibilities
CLMRMRS1.rdf	Report Sets
CLMRMUS1.rdf	Users
CLMRMVS1.rdf	Value Sets
CLMRMZM1.rdf	Zooms

Table 2-2. Reports used in system maintenance and administration

Report Name	Usage
`CLMRDPG1.rdf`	Purge Object Migrator Interface Tables
`CLMRDPV1.rdf`	Purge Object Archive Versions
`CLMRMAD1.rdf`	Mercury Object Migrator Audit Report
`CLMRMMC1.rdf`	Migration Comparison Report
`CLMRMVR1.rdf`	Object Archive Version Detail Report
`CLMRDVW1.rdf`	Maintain Object Migrator Views

SQL Scripts

SQL Script files have `sql` file extensions. These files are used to perform automatic setup of the internal objects, such as applications, tables, indexes, sequences, concurrent programs, flexfields, and value sets necessary for each of the modules to execute.

Shell Scripts

Shell scripts do not have file extensions and are used to drive the installation process for Object Migrator.

Migration Overview

The sections below discuss the following topics:

- Deploying objects across database instances
- Supported Oracle object types
- Migration capabilities

Deploying Objects Across Database Instances

Individual applications in the Oracle E-Business Suite rely on the configuration of common objects (AOL objects) within their database instances. When customizing or configuring Oracle E-Business Suite applications, it is often necessary to deploy these objects from one instance to another. For example, you may have one instance for customizing your Oracle Applications (DEV), one instance for testing your customizations (TEST), and one production instance (PROD). The AOL objects in this example would be deployed from DEV to TEST to PROD, as shown in *Figure 2-2*. Manually re-keying these configurations would be both time-consuming and error prone.

Figure 2-2. Migration overview

Mercury Object Migrator automates the deployment of these AOL objects between Oracle E-Business Suite instances, increasing both the accuracy and efficiency of object deployments. Object Migrator also supports archiving and restoring configurations. Object Migrator is installed as an application within Oracle E-Business Suite applications, and is typically installed only once in an instance that will not get refreshed.

Object Migrator consists of a series of concurrent programs that run through the standard Oracle Submit Requests form, making it easy and familiar to use. Users launch an Object Migrator program for each AOL object type (such as concurrent program, value set, or menu). For each program, users specify information regarding the object(s) to migrate as well as parameters specifying source and target data locations. There can also be parameters specific to an individual object type.

Based on the concurrent request parameters, Object Migrator:

- Deploys the objects to the destination
- Validates and transforms data values as required
- Identifies business rule violations and reporting migration outcome

The results of each Object Migrator execution can be queried from the standard Oracle View Requests form. Each execution run produces an output report that lists the parameters passed to the program, the objects queried for migration, and the detailed results of the migration.

Supported Oracle Object Types

Mercury Object Migrator currently supports migration of the following object types (each of these object types has a separate concurrent program):

- Concurrent Manager Definitions
- Concurrent Programs
- Descriptive Flexfields
- FSG Row/Column Sets
- Folders
- Functions
- GUI Menus
- Help Text (used in release 10.7 character mode only)
- Menus (used in release 10.7 character mode only)
- Messages
- Named SQL (used in release 10.7 only)
- Printer Definitions
- Profile Options
- QuickCodes
- Report Groups
- Report Sets
- Responsibilities
- Users
- Value Sets
- Zooms (used in release 10.7 only)

Migration Capabilities

Mercury Object Migrator supports the following features when migrating each object type:

- Migrate a single object, a range of objects, or objects matching a string with wild cards
- Rename an object as it is migrated into the destination instance
- Create new objects and update existing objects in the destination instance
- Retain referential integrity when creating or updating existing objects just as if performed using the application forms
- Migrate across releases of Oracle Applications (from 10.7 to 11i)
- Migration simulation to identify issues with data setups without actually migrating data (can run a migration simulation as a “dry run” before an actual migration)

Version Control Overview

You can use Mercury Object Migrator to save data to special archive tables as objects are migrated across Oracle E-Business Suite instances. This feature can be used to store new versions of objects. Object Migrator also allows you to use this archived data as your source information, thus enabling you to revert your objects back to previous versions. Finally, you can use Object Migrator reports to view complete version histories of a specific object or a group of objects.

The object archive is the version control repository used by Mercury Object Migrator. Each object type that Object Migrator migrates can be archived, and sets of objects of the same or different objects types can be grouped together into a single logical archive. The object archive resides in the instance where Object Migrator is installed, allowing AOL object versions from any instance to be stored in one central location.

Version labels are used by the object archive to identify groupings of objects into a single logical archive. The user enters the version label during the execution of Object Migrator. It is also used to identify objects to retrieve from the archive, and to report on objects in the archive.

Within each archive (as specified by a version label), it is possible to store multiple objects of multiple object types. When saving an object to the object archive, Object Migrator copies the entire definition of the object into the archive. Object Migrator gives the new object a revision number that tracks the number of times a particular object has been archived. A particular object (such as a value set called My Sample Value Set) can be archived only once within a given version label.

Object Migrator also makes it possible to purge data from the object archive as it becomes obsolete.

For an example of version label and revision number usage, see *Figure 2-3*.

Object Migrator Executions

Run #	Object Type	Application	Object	Version Label
1	Conc Program	WIP	WICDOL	Label_1
2	Conc Program	WIP	WICDOL	Label_2
3	Value Set		Yes_No	Label_1
4	Conc Program	WIP	WIPDJPCK	Label_2
5	Conc Program	WIP	WICDOL	Label_3

Status of Object Archive after Object Migrator executions

Version Label	Object Type	Application	Object	Rev
Label_1	Conc Program	WIP	WICDOL	1
Label_1	Value Set		Yes_No	1
Label_2	Conc Program	WIP	WICDOL	2
Label_2	Conc Program	WIP	WIPDJPCK	1
Label_3	Conc Program	WIP	WICDOL	3

Figure 2-3. Example of version label and revision number usage

Reports Overview

Two kinds of reports are available in Mercury Object Migrator:

- Comparison reports
- Migration audit reports

Comparison Reports

Before migrating a new version of a specific AOL object from one database to another, it is possible to compare the version that will be migrated to the version already present in the destination database. Mercury Object Migrator generates a comparison report showing the differences between the two versions.

Long fields and translated data (for instances running under multi-language support [MLS]) are not compared in the comparison reports.

Some child entities are not compared in the comparison reports. Major exceptions are noted in Appendix B, *Object Type-Specific Migrators and Migration Rules*, on page 165.

Migration Audit Reports

Mercury Object Migrator provides an audit report with every batch of objects being migrated. This report provides an audit trail of successfully migrated objects as well as a detailed exception listing for objects that failed migration. The report also lists key information about each object.

Chapter

3

Installation and Upgrade Overview

In This Chapter:

File System Requirements

Mercury Object Migrator requires a minimum amount of disk space to be successfully installed. Table 3-1 lists the space requirements for the Object Migrator file system.

Table 3-1. Object Migrator file system space requirements

Object	Required Space
Install Bundle	30–35 MB (temporary)
Install Scripts (shell and SQL scripts)	5.0 MB
Programs (`rdf` files)	1.5 MB per program 30 MB total

Tablespace Requirements

Object Migrator requires a minimum amount of tablespace in order to function properly. Table 3-2 lists the tablespace requirements for Object Migrator:

Table 3-2. Object Migrator tablespace requirements

Object	Required Space	Recommended Space
Interface Tables	30 MB	60 MB
Indexes on Interface Tables	20 MB	30 MB
Archive Tables and Indexes	30 MB*	60 MB*

* Requirements directly relate to number of objects placed under version control (see *Version Control Considerations* on page 32 for details). In addition, sizing may depend on your database version and configuration.

Platform Requirements

Since Object Migrator works within Oracle E-Business Suite and uses standard Oracle tools, it can run on any platform running Oracle E-Business Suite.

Installation of Mercury Object Migrator utilizes a shell script to drive the installation. The location in which Object Migrator is being installed must support Bourne shell executions. The MKS Toolkit fulfills this requirement on Windows environments.

Instance Requirements

Mercury Object Migrator runs as concurrent programs using Oracle Application's Concurrent Manager. The user accesses the Oracle Submit Requests form (also known as the Standard Request Submission form) to launch the migration programs. The instance on which Object Migrator is installed must be able to run a concurrent program.

Object Migrator requires only an installation of the Applications Object Library (AOL). Object Migrator does not need any other application (such as INV, HR, or GL) to be installed, though having these applications installed will not cause any compatibility problems.

Mercury recommends that you also review *Special Considerations for Release 6.0* on page 63.

Since Object Migrator resides as a custom application within the Oracle E-Business Suite instance, it should be installed in an instance that will not be refreshed. Normally, this will be either a production instance or an instance dedicated to hosting Object Migrator.

Distributed Database Option - Recommended

Mercury Object Migrator uses database links to access all databases. The Distributed Database Option is required to insert, delete, or update data on a remote database. It is strongly recommended that this option be active on all the databases being used as sources and destinations. For information on how to install Object Migrator with the Distributed Database Option off, see *Using Object Migrator Without the Distributed Database Option* on page 112.

If using Object Migrator in conjunction with Mercury Change Management™, the Distributed Database Option is required.

The database where Object Migrator resides must be configured such that Oracle Applications databases that will be the source or destination of a migration can be accessed using database links (that is, configurations like tnsnames, listener, or global names).

Application Naming Requirements

Mercury Object Migrator requires that existing applications with the short name CLM do not exist on the instance in which the Object Migrator is being installed. If a CLM application does exist, contact Mercury Support for assistance (http://support.mercury.com).

Version Control Considerations

Mercury Object Migrator uses archive tables to store version information for AOL objects. These tables reside on the instance where Object Migrator is installed. If Object Migrator is installed on multiple databases and uses the version control functionality on all the installations, multiple AOL object repositories will exist. To prevent confusion, it is recommended that the version control functionality be used on only one installation.

Since Object Migrator uses database tables to store object information, install it on a database instance that will not be periodically refreshed.

If the database does need to be refreshed for some reason, export the data in the Object Migrator tables (all migrator table names begin with CLM) before the refresh and import the data once the refresh is completed.

Sizing Requirements for Object Archive Tables

When saving an object to the object archive, Object Migrator takes the entire definition of the object and stores it in special archive tables. The growth of these archive tables directly corresponds to the number of objects and the number of versions put into these tables. For some objects, object complexity and size may also be a factor.

If periodic storage of versions of entire groups of objects is planned rather than only objects that change, then more tablespace will be required than the recommended 30 megabytes. It is recommended to start with a 30 megabyte allocation and then add tablespace as needed.

Installing Object Migrator

Before using Mercury Object Migrator to migrate AOL objects, it must be installed and configured on one or more Oracle E-Business Suite instances at your site. Configuration includes determining which Oracle E-Business Suite instances objects can be migrated from and to instances, and defining security setups to control user access to the different programs included in Object Migrator.

See Chapter 4, *Installing Object Migrator*, on page 35 for detailed installation instructions. Installation is typically the responsibility of the database administrator or system administrator.

Obtaining a User Name and Password

Each user who is going to run Mercury Object Migrator must have an Oracle Applications user ID and password, and be given access to run one or more Object Migrator programs using the Submit Request form.

You should obtain this information from your system administrator. Your administrator may set up a specific responsibility for Object Migrator usage, or may incorporate Object Migrator functions into an existing responsibility.

Opening Oracle Applications

Object Migrator programs are available as concurrent programs within Oracle Applications, and are accessed using the Submit Request form within Oracle Applications (in some cases, the form name may be Run Requests).

To migrate an AOL object between Oracle Applications instances:

1. Log on to Oracle Applications.
2. Open the Submit Requests window.
3. In the Name field, select the supported AOL object type you want to migrate.

For instructions on using Object Migrator to migrate data, see Chapter 8, *Migrating Objects*, on page 127.

Only migrator programs that your system administrator has given you access to will appear in the Name list.

Chapter

4

Installing Object Migrator

In This Chapter:

Example Scenario

The ACME Corporation is installing Mercury Object Migrator at a site with four database instances named STAGE, DVLP, PILOT, and PROD. Carl, their database administrator, has to select one of the instances on which to install Object Migrator. Although he could select any of the instances, he chooses the one that is refreshed least frequently. DVLP and PILOT, which are refreshed frequently, are not preferred. PROD is a good candidate on which to install Object Migrator; however, the operations staff prefers that Carl not install any external tools in the PROD database. STAGE is a small staging database that only has Application Object Library installed and is never refreshed. STAGE is thus selected as the instance on which Object Migrator is to be installed.

Figure 4-1 shows the sample setup schematic.

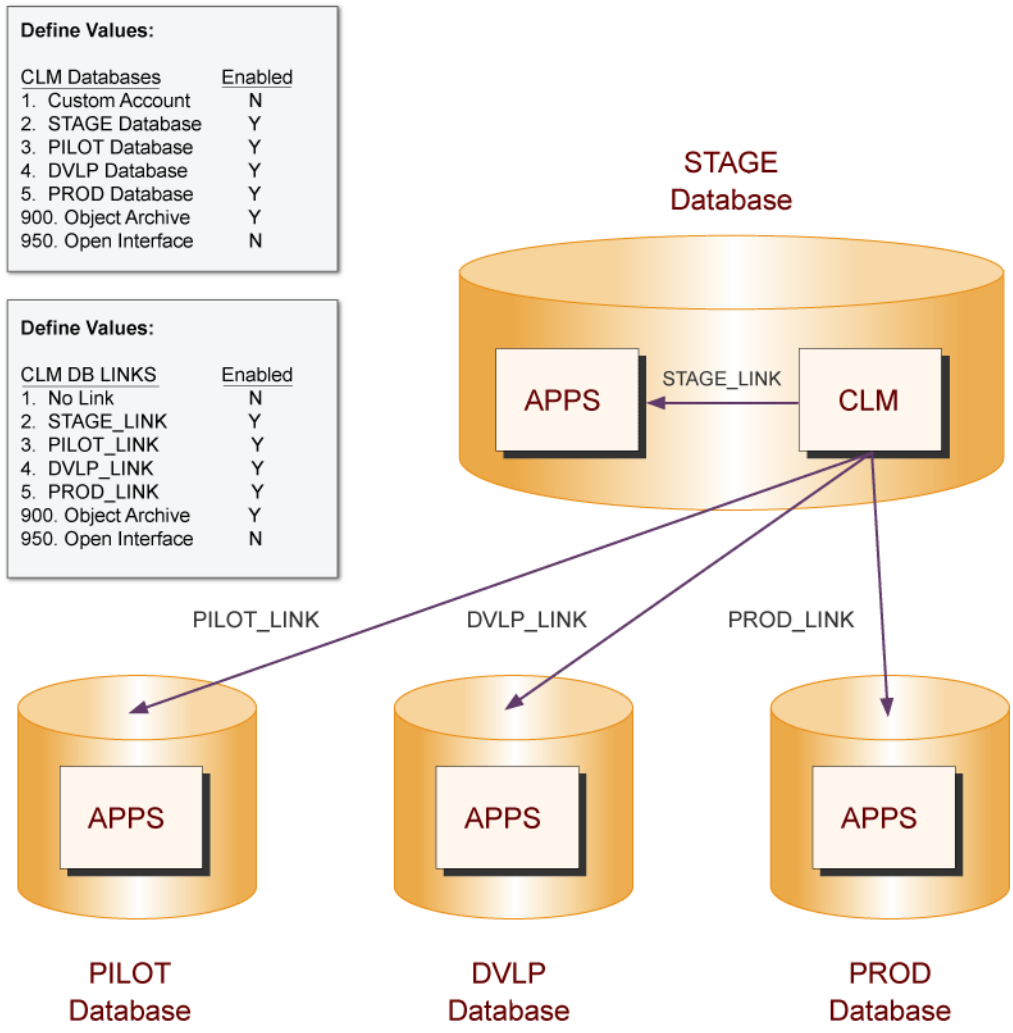


Figure 4-1. Sample setup schematic

Pre-Installation Setup

Before beginning the Mercury Object Migrator installation, you need to complete the following setup procedure after reviewing these sections of Chapter 5:

- *Overview* on page 62
- *Supportability* on page 63.
- *Special Considerations for Release 6.0* on page 63
- *Upgrade Impacts for Object Migrator Release 6.0* on page 64

The syntax used in this section reflects a UNIX environment. In a Windows environment syntax will be different (for example, `%APPLOUT%` versus `$APPLOUT`) and environment information is usually contained in `cmd` files, not in `env` files.

To prepare to install Object Migrator:

1. Decide where Object Migrator will be installed and create the requisite directory structure.

   Object Migrator can be installed in its own <*PROD*>_TOP directory (for example, CLM_TOP), or it can share a <*PROD*>_TOP directory with an existing application (for example, FND_TOP). A separate <*PROD*>_TOP is recommended. In this document, CLM_TOP will represent the location where Object Migrator is installed.

   a. If Object Migrator will use its own <*PROD*>_TOP, create the directory structure according to applications standards.

      A subdirectory will be needed for the installation, plus another subdirectory for the Oracle*Reports files. This file structure should be created on every tier in the instance. The directories should be owned by the user who owns the TOP directories (usually applmgr).

      For example, Carl creates a TOP directory for Object Migrator

      - On UNIX:

        ```
        mkdir /u1/stageappl/CLM
        ```

      - On Windows:

```
mkdir d:\stageappl\CLM
```

The directories should be owned by the user who owns the other `TOP` directories (usually `applmgr`).

The subdirectory name for Oracle*Reports files varies by release.

- Release 10 - `CLM_TOP/srw`
- Release 11 - `CLM_TOP/reports`
- Release 11i - `CLM_TOP/reports/US`

b. If this instance is set up to place log and output files for concurrent processing under individual applications rather than in a central location, create a log (`$APPLLOG`) and out (`$APPLOUT`) directory under `CLM_TOP`.

   For example, Carl creates the directories

   - On UNIX:

     ```
     mkdir /u1/stageappl/CLM/install
     mkdir /u1/stageappl/CLM/reports/US
     mkdir /u1/stageappl/CLM/$APPLLOG
     mkdir /u1/stageappl/CLM/$APPLOUT
     ```

   - On Windows:

     ```
     mkdir d:\stageappl\CLM\install
     mkdir d:\stageappl\CLM\reports\US
     mkdir d:\stageappl\CLM\%APPLLOG%
     mkdir d:\stageappl\CLM\%APPLOUT%
     ```

c. Add the `CLM_TOP` environment variable to the instance context. Often, this is the customization section of the appropriate file where environment variables are set.

   For release 11i, this file is usually the `$APPL_TOP/admin/adovars.env` file for the instance (`adovars.cmd` on Windows).

   Your instance may have additional requirements or different procedures. On Windows, the value must be populated to the registry as well.

For example, Carl would set

- On UNIX:

```
CLM_TOP="//u1/stageappl/CLM"
export CLM_TOP
```

- On Windows:

```
set CLM_TOP=d:\stageappl\CLM\
```

d. Ensure that the environment has been updated.

- On UNIX:

  Source the file and then bounce the Oracle E-Business Suite applications processes.

- On Windows:

  If the installation is done as a custom application on Windows, update the windows registry with CLM_TOP.

2. Decide which tablespaces Object Migrator will use for tables and indexes.

   It is recommended that Object Migrator use its own tablespaces, especially if Object Migrator will reside in a production Oracle E-Business Suite instance.

   a. If you want new tablespaces, create them. See the sizing requirements for the tablespaces in *Tablespace Requirements* on page 30.

   b. Consider creating additional rollback segments to support Object Migrator migration transactions, especially if Object Migrator does not reside in a dedicated instance. Adding at least one new rollback segment for each of the new tablespaces is recommended. These rollback segments should reside in a separate tablespace reserved for rollback segments. They should be generated with the OPTIMAL size constraint to make sure that the rollback segments automatically deallocate space as it becomes free.

3. Create a custom SQL account in which to install Object Migrator.

   An existing SQL account may be used, but this is not recommended unless the account was originally created for Object Migrator. This account needs privileges to create tables, indexes, sequences, database links, and stored procedures. It should also have access to the V$DATABASE view.

Default tablespaces should be defined for the user.

For example, Carl chooses to call his schema CLM with a password of CLM. He has also set up new tablespaces of CLM_DATA and CLM_NDX to hold the tables and indexes for CLM.

```
create user CLM identified by CLM
default tablespace CLM_DATA
quota unlimited on CLM_DATA
quota unlimited on CLM_NDX;
grant connect, resource, unlimited tablespace to CLM;
```

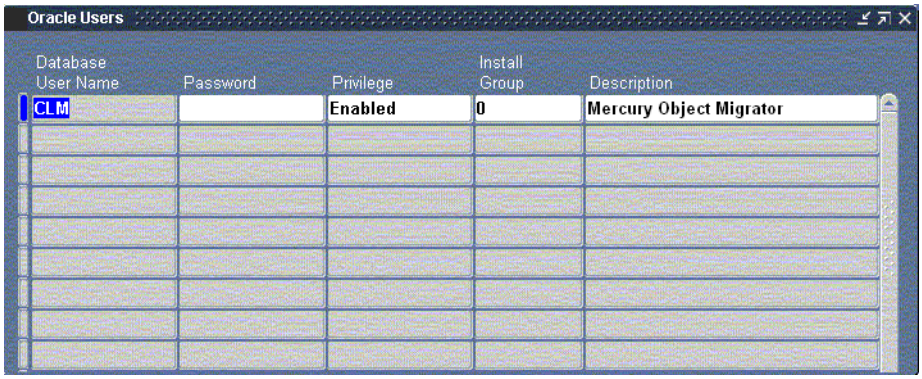
4. Register the SQL account created for Object Migrator with the Oracle E-Business Suite.

   a. Log on to the Oracle E-Business Suite for this database instance.

   b. Through the system administrator responsibility, navigate to the Register form. (Security:ORACLE:Register)

   c. Add an entry for the new SQL account.

      For example, Carl chose to call his schema CLM, so he registers that schema in the STAGE instance.



Note

For some Oracle E-Business Suite releases, this will automatically submit the Make Foundation Grants and Synonyms program for an Oracle user. Wait for the program to finish before continuing. If the program does not get submitted, continue with the installation.

Installing on Windows

The following procedure describes how to install Object Migrator in a Windows environment.

To install Object Migrator on Windows:

1. Decide where Object Migrator will be installed and create the requisite directory (folder) structure. For more information, see *Pre-Installation Setup* on page 38.

   For example, Carl has signed onto his database server, hostname Cobra, as the user who owns the Oracle Applications files, `applmgr`.

2. Download the Object Migrator bundle from the Mercury IT Governance Download Center at http://itg.merc-int.com/support/download/login.jsp.

   The executables are compressed into one zip file for Windows named `CLM_60.zip`. Place this file into the folder where the Mercury Object Migrator installation is to be staged.

   For example, in Carl's case, the folder is `d:\stageappl\CLM\install`.

3. Extract the Object Migrator bundle using an unzip utility.

   This creates an installation folder, `CLM_6_0`, with `install` and `srw` folders beneath it.

   - The `install` folder contains all the installation scripts necessary to install Object Migrator.
   - The `srw` folder contains all of the Object Migrator executables.

4. Navigate to the `install` folder created by the unzip process, `CLM_6_0\install`.

   Under the `install` folder are two folders, one for release 10 (`r10`) and another for release 11 (`r11`). When installing Object Migrator on a release 10 instance, navigate to the `r10` folder. When using a release 11 or 11i instance, navigate to the `r11` folder.

   a. Start a command shell and set the Oracle E-Business Suite context, if required.

   b. Navigate to the `install` folder.

For example, Carl is installing Object Migrator on a release 11i instance.

```
cmd d:\stageappl\envshell.cmd
cd d:\stageappl\CLM\install\CLM_6_0\install\r11
```

5. Make sure the custom SQL account discussed in *Pre-Installation Setup on page 38* has been created and registered.

6. Run the `CLM_INSTALL_NT` script.

   a. Start a bash shell.

      In environments where MKS Toolkit is in use (not required), MKS must be in the user's path.

   b. Run the `CLM_INSTALL_NT` script.

      For example:

      ```
      sh CLM_INSTALL_NT
      ```

      `CLM_INSTALL_NT` is an interactive program that prompts for the information in *Table 4-1*. In each case, the program verifies the information entered and proceeds only if the information is valid (for example, the install program will try to connect to the database using the user name and password you have entered). Additionally, the install process can be stopped by entering a period (.) at any user prompt.

Table 4-1. CLM_INSTALL_NT script parameters

Parameter	Description
SQL Account registration	Confirm (Y/N) whether the Object Migrator SQL account has been registered in the Oracle E-Business Suite.
sql executable	The executable to use in order to make a command-line SQL*Plus connection. This could be plus80 or sqlplus, depending on the toolset in use.
Username of Object Migrator sql account	The user name for the Object Migrator SQL account that was created and registered in Oracle E-Business Suite.
Password for CLM sql acct	The password for this SQL account.

Table 4-1. CLM_INSTALL_NT script parameters [continued]

Parameter	Description
Connect string for Object Migrator database	Specifies the database on which Object Migrator is being installed. • When using SQL*Net 1.0, enter the entire two-task database connect string. (For example, T:dev_machine:MIGRATE) • When using SQL*Net 2.0 or higher, enter the database identifier (usually the ORACLE_SID).
Tablespace for Object Migrator tables	All Object Migrator tables will go into the specified tablespace. Note: The SQL account must have permissions to write to this tablespace.
Tablespace for Object Migrator indexes	All Object Migrator indexes will go into the specified tablespace. Note: The SQL account must have permissions to write to this tablespace.
Username of APPS user	The user name for the Oracle E-Business Suite APPS account.
APPS user password	The password for this SQL account.

For example:

```
bash
sh CLM_INSTALL_NT
```

Confirm whether the SQL account has been registered: **Y** (Carl registered the SQL account per the steps in *Pre-Installation Setup* on page 38)

Command name used for invoking SQL*Plus: **plus80**

Username of Object Migrator sql account: **CLM**

Password for CLM sql account: **CLM** (twice)

Connect String for Object Migrator Database: **STAGE** (if Carl had been using SQL*Net 1.x, he would have entered **t:cobra:STAGE** as the connect string)

Tablespace for Object Migrator tables: **CLM_DATA**

Tablespace for Object Migrator indexes: **CLM_NDX**

Username of APPS user: **APPS**

Password for APPS user: **APPS** (twice)

Carl had already created the `CLM_DATA` and `CLM_NDX` tablespaces before the install process. You can use an existing tablespace.

7. The script outputs log messages as standard output on your screen. It will also save the output to a log file named `clm_log.out` in the current folder.

   The command file `CLM_INSTALL_NT` will run all the Object Migrator install scripts. These scripts perform the following actions:

   - Create Mercury Object Migrator Application (application code = **CLM**)
   - Add the new application to all data groups that contain the FND application (using the OracleID in which Object Migrator was installed)
   - Create database objects used by Object Migrator
   - Create value sets and registers concurrent programs used to run migrators, system maintenance, and administration reports
   - Create a limited number of grants and synonyms to the APPS account to facilitate request submission
   - Define incompatibilities between different Object Migrator programs

All object names created by the install scripts begin with CLM. If an application with a name of CLM already exists, please contact Mercury Support (http://support.mercury.com).

8. Move the `rdf` files from the `srw` folder of the Object Migrator bundle (for example, `CLM_TOP\install\CLM_6_0\srw`) to their permanent location under `CLM_TOP`.

This location varies by release:

- `CLM_TOP\srw` for release 10
- `CLM_TOP\reports` for release 11
- `CLM_TOP\reports\us` for release 11i

For example, Carl is using the `CLM_TOP` folder structure to store the Object Migrator programs and files, so he is copying from `%CLM_TOP%\install\CLM_6_0\srw` to `%CLM_TOP%\reports\US`.

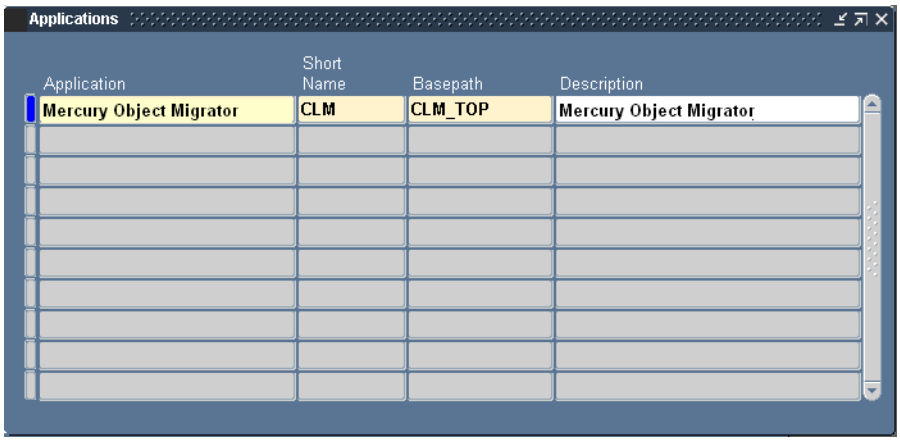
(Note: If Carl were using a release 11 instance, he would copy the files to `$CLM_TOP\reports`. For release 10.7, he would copy the files to `$CLM_TOP\srw\10.7`.)

The installation files are not needed for continued use of Object Migrator, but it is recommended that they be saved.

9. Point Object Migrator to the correct basepath.

   The installation program defaults to point the CLM application to the `FND_TOP` area. Update this value to reference the <*PROD*>`_TOP` where Object Migrator is installed.

   To change the CLM application to point to an existing folder structure, navigate to the Register Application form. Query the application named Mercury Object Migrator. Change the Basepath field to point to <*PROD*>`_TOP` environment variable for the CLM application.



At this point, Object Migrator must be configured before it can be used to migrate data. For these instructions, see *Configuring Your System to Work with Object Migrator* on page 52.

Installing on UNIX

The following procedure describes how to install Object Migrator in a UNIX environment.

To install Object Migrator on UNIX:

1. Decide where Object Migrator will be installed and create the requisite directory structure. For more information, *Pre-Installation Setup* on page 38.

   For example, Carl has signed on to his database server, hostname Cobra, as `applmgr`, the user who owns the Oracle Applications files.

2. Download the Object Migrator bundle from the Mercury IT Governance Download Center at http://itg.merc-int.com/support/download/login.jsp.

   The executables are compressed into a tar file for UNIX named `CLM_60_tar.Z`. Place this file into the directory where Object Migrator installation is to be staged.

   For example, in Carl's case, the directory is `/u1/stageappl/CLM/install`.

3. Extract the files from the Object Migrator bundle.

   For example, the files are uncompressed and unarchived.

   ```
   uncompress CLM_60_tar.Z
   tar -xvf CLM_60_tar
   ```

4. Navigate to the install subdirectory created by the untar process, `CLM_6_0/install`.

   Under the install directory are two subdirectories, one for Oracle release 10 (`r10`) and another for release 11 (`r11`). When installing Object Migrator on a release 10 instance, navigate to the `r10` directory. When using a release 11 or 11i instance, navigate to the `r11` directory.

For example, Carl is installing Object Migrator on a release 11i instance.

```
cd CLM_6_0/install/r11
```

5. Make sure the custom SQL account discussed in *Pre-Installation Setup* on page 38 has been created and registered.

6. Set the environment context to the Oracle E-Business Suite instance, if not already set.

7. Run the `CLM_INSTALL` script.

   For example, `CLM_INSTALL`.

   Regardless of the shell that is being run, the installation script will spawn a Bourne shell and run under it.

   `CLM_INSTALL` is an interactive program that prompts for the parameters listed in *Table 4-2* before installing Object Migrator. In each case, the program will verify the information entered and only proceed if the information is valid (for example, the install program will try to connect to the database using the user name and password you have entered). Additionally, the installation process can be stopped by entering a period (.) at any user prompt.

Table 4-2. CLM_INSTALL script parameters

Parameter	Description
SQL Account registration	Confirm (Y/N) whether the Object Migrator SQL account has been registered in the Oracle E-Business Suite.
Username of Object Migrator sql account	The user name for the Object Migrator SQL account that was created and registered in Oracle E-Business Suite.
Password for CLM sql acct	The password for this SQL account.
Connect string for Object Migrator database	Specifies the database on which Object Migrator is being installed. • When using SQL*Net 1.0, enter the entire two-task database connect string. (For example, T:dev_machine:MIGRATE.) • When using SQL*Net 2.0 or higher, enter the database identifier (usually the ORACLE_SID).

Table 4-2. CLM_INSTALL script parameters [continued]

Parameter	Description
Tablespace for Object Migrator tables	All Object Migrator tables will go into the specified tablespace. Note: The SQL account must have permissions to write to this tablespace.
Tablespace for Object Migrator indexes	All Object Migrator indexes will go into the specified tablespace. Note: The SQL account must have permissions to write to this tablespace.
Username of APPS user	The user name for the Oracle E-Business Suite APPS account.
APPS user password	The password for this SQL account.

For example:

```
CLM_INSTALL
```

Confirm whether the SQL account has been registered: **Y** (Carl registered the SQL account using the steps in *Pre-Installation Setup* on page 38)

Username of Object Migrator sql account: **CLM**

Password for CLM sql account: **CLM** (twice)

Connect String for Object Migrator Database: **STAGE** (if Carl had been using SQL*Net 1.x, he would have entered **t:cobra:STAGE** as the connect string)

Tablespace for Object Migrator tables: **CLM_DATA**

Tablespace for Object Migrator indexes: **CLM_NDX**

Username of APPS user: **APPS**

Password for APPS user: **APPS** (twice)

Carl had already created the CLM_DATA and CLM_NDX tablespaces before he began the installation process. You can use an existing tablespace.

8. The script will output log messages as standard output on the screen. It will also save the output to a log file named clm_log.out in the current directory.

   The command file CLM_INSTALL will run all the Object Migrator install scripts. These scripts perform the following actions:

   - Creates Mercury Object Migrator Application (application code = **CLM**)
   - Adds the new application to all data groups that contain the FND application (using the OracleID in which Object Migrator was installed)
   - Creates database objects used by Object Migrator
   - Creates value sets and registers the concurrent programs used to run migrators, system maintenance, and administration reports
   - Creates a limited number of grants and synonyms to the APPS account to facilitate request submission
   - Defines incompatibilities between different Object Migrator programs

All object names created by the install scripts begin with CLM. If an application with a name of CLM already exists, please contact Mercury Support (http://support.mercury.com).

9. Move the `rdf` files from the `srw` subdirectory of the extracted Object Migrator bundle (for example, `CLM_TOP/install/CLM_6_0/srw`) to their permanent location under `CLM_TOP`.

   This location varies by release:

   - `CLM_TOP/srw` for release 10
   - `CLM_TOP/reports` for release 11
   - `CLM_TOP/reports/us` for release 11i

   For example, Carl is using the `CLM_TOP` directory structure to store the Object Migrator programs and files.

   ```
   cp $CLM_TOP/install/CLM_6_0/srw/*.rdf $CLM_TOP/reports/US/
   ```

(Note: If Carl were using a release 11 instance, he would copy the files to `$CLM_TOP/reports`. For release 10.7, he would copy the files to `$CLM_TOP/srw/10.7`)

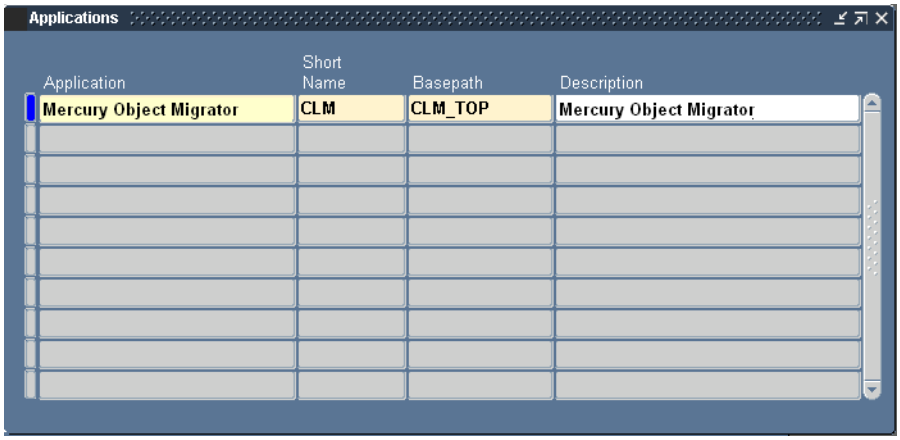
Note

The installation files are not needed for continued use of Object Migrator, but it is recommended that they be saved.

10. Point Object Migrator to the correct basepath.

    The installation program defaults to point the CLM application to the `FND_TOP` area. Update this value to reference the <*PROD*>`_TOP` where Object Migrator is installed.

    To change the CLM application to point to an existing directory structure, navigate to the Register Application form. Query the application named Mercury Object Migrator. Change the Basepath field to point to <*PROD*>`_TOP` environment variable for the CLM application.



    At this point, Object Migrator must be configured before it can be used to migrate data. For instructions on how to do that, *Configuring Your System to Work with Object Migrator*.

Configuring Your System to Work with Object Migrator

After installing Mercury Object Migrator, you need to configure Oracle E-Business Suite system to work with Object Migrator. Configuration consists of the following major steps, which are discussed below:

- Create database links for each source and destination database.
- Register these database links with Object Migrator.
- Define security to access Object Migrator programs, including users, responsibilities, report groups, and so forth.
- Run the Maintain Object Migrator Views report.
- Run a test migration with Report Only set to **Yes** to validate the installation.

The example given is a simplified case that assumes the system administrator will be able to access all Object Migrator functions. See *Setting Up Object Migrator Security* on page 124 for more information about security options.

To configure your system:

1. Create database links to all valid source and destination databases. Note that the database configurations (for example, `tnsnames.ora`) must support connection using the database link.

Prior to this step, you need to have configured the database to see other databases.

Perform this step in the instance containing Object Migrator using the SQL account under which Object Migrator was installed. Create a database link to the `APPS` account of each remote database that will be a source or destination of migrations.

For example, Carl signs on to the STAGE database as CLM and creates a link to each database he will use.

Create database link called STAGE_LINK from STAGE to STAGE:

```
SQL> create database link STAGE_LINK
2 connect to APPS identified by APPS <Password for APPS
account in STAGE>
3 using 'STAGE' ; <Connect string for the STAGE database>
```

Create database link called DVLP_LINK from STAGE to DVLP:

```
SQL> create database link DVLP_LINK
2 connect to APPS identified by APPS <Password for APPS
account in DVLP>
3 using 'DVLP' ; <Connect string for the DVLP database>
```

Create database link called PILOT_LINK from STAGE to PILOT:

```
SQL> create database link PILOT_LINK
2 connect to APPS identified by APPS <Password for APPS
account in PILOT>
3 using 'PILOT' ; <Connect string for the PILOT database>
```

Create database link called PROD_LINK from STAGE to PROD:

```
SQL> create database link PROD_LINK
2 connect to APPS identified by APPS <Password for APPS
account in PROD>
3 using 'PROD' ; <Connect string for the PROD database>
```

Test each link after creating it by executing the following in sql*plus:

```
SQL> select count(*) from fnd_user@<db_link_name> where
rownum = 1;
```

2. Add a value to the CLM_DATABASES value set for each valid source or destination database.

   This includes a value for the database in which Object Migrator is installed. Disable the Value **1** if this value is not already disabled.

The values need to be pure numeric values (integers) with no leading zeroes.

   a. Select the **System Administrator** responsibility. Go to the Define Value Set Values window (**Application:Validation:Values**).

   b. Enter **CLM_DATABASES** in the Name field of the Find section.

   c. Click **Find.**

d. Query all the records in the Values zone.

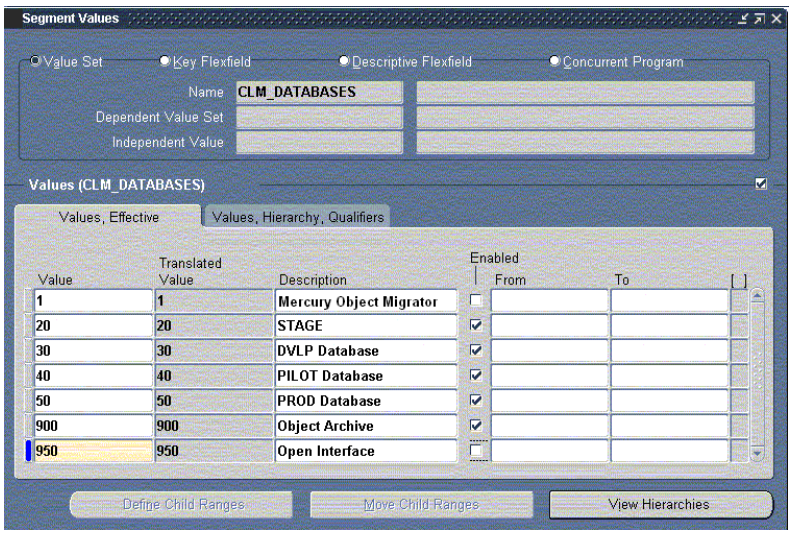
This should bring up three records:

- One representing the Object Migrator account (this value is initially set to **Disabled**)
- One representing the object archive (for version control)
- One representing the Open Interface (this value is initially set as **Disabled**)

This value set represents all the database instances between which objects will be migrated.

e. Add additional records in this region, one per database instance, as shown below.

This includes a value for the current database. The Value field needs to be a pure numeric value (integer). The Description value should be a value that will identify the instance to users.



3. For each database link, add a value to the CLM_DB_LINKS value set. For the descriptions, use the exact database link names.

This value set represents all the database links that have been created.

a. Enter **CLM_DB_LINKS** in the Name field of the Find Block.

b. Click **Find.**

c. Query all the records in the Values zone.

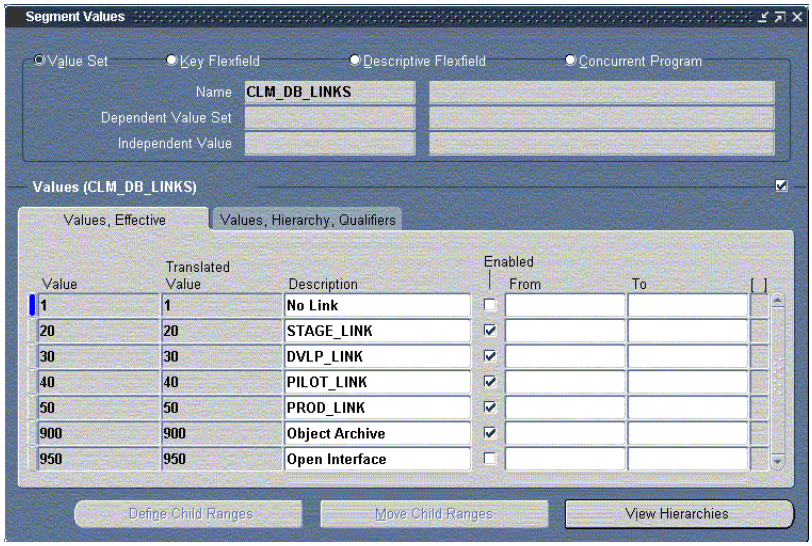
This should bring up three records.

d. Add all the database links created in the prior steps as shown below.

Make sure that the Value column matches with the corresponding Value column in the CLM_DATABASES value set. The values need to be pure numeric values (integers).

Note

Use the same number in the Value field that were used for the corresponding database in the CLM_DATABASES value set.



4. Add the concurrent programs defined to execute the program files for each of the Object Migrator modules (and the system administrator reports) to the appropriate request security group(s).

Add reports one by one or attach the entire CLM application to the Request Security Group. *Table 4-3*provides a complete list of concurrent program names that can be included; check off each one as they are included.

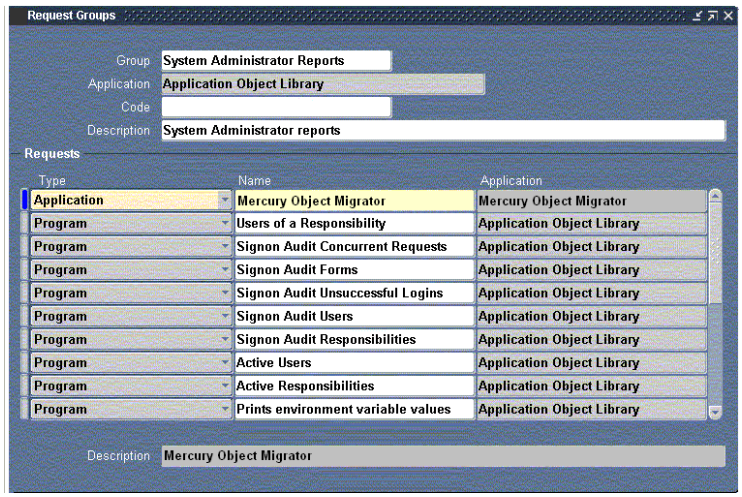
Table 4-3. Object Migrator concurrent program names

Concurrent Program Name	Included?
Mercury Object Archive Version Detail Report	
Mercury Object Migrator Audit Report	
Maintain Object Migrator Views	
Migrate Concurrent Manager Definitions	
Migrate Concurrent Programs	
Migrate Descriptive Flexfields	
Migrate FSG Row/Column Sets	
Migrate Folders	
Migrate Functions	
Migrate GUI Menus	
Migrate Help Text	
Migrate Menus	
Migrate Messages	
Migrate Named SQL	
Migrate Profile Options	
Migrate Printer Definitions	
Migrate Responsibilities	
Migrate Report Groups	
Migrate Report Sets	
Migrate QuickCodes	
Migrate Value Sets	
Migrate Users	
Migrate Zooms	
Migration Comparison Report	
Purge Object Archive Versions	
Purge Object Migrator Interface Tables	

For optional setups, see Chapter 6, *Optional Configurations for Object Migrator,* on page 99.

For example, Carl is adding the Object Migrator concurrent programs to the System Administrator Reports request group. This allows the system administrator to run all the Object Migrator programs.

- Carl navigates to the Define Request Security Group window (**Security:Responsibility:Request**).
- Carl queries a request group called **System Administrator Reports.**
- Carl goes to the Requests zone.
- Carl inserts a record by using the **Add Row** button from the toolbar.
- Carl picks **Application** in the Type field.
- Carl selects **Mercury Object Migrator** as the application.



Alternatively, Carl could have added individual programs rather than the Object Migrator application.

10. Run the Maintain Object Migrator Views program with Report Only set to **2** (**No**).

    This program defines the views used by some Object Migrator parameters when submitting Object Migrator migrations.

This program should be scheduled to run periodically (Mercury recommends once a day) to keep the Object Migrator views current with the number and statuses of your databases. For details regarding the support of these views, see *Maintaining Object Migrator Views* on page 114.

If you have more than four databases that you will be migrating to or from, increase the value for the open_links database parameter (the suggested value is 20) and bounce the database before executing this step. For more information, see *open_links Database Parameter* on page 118.

Make sure to view the report output of this program. The output lists all the enabled databases for use with Object Migrator and any errors encountered while trying to connect to these databases.

For example, since Carl has only four enabled databases, he does not need to change the open_links database parameter. Therefore, he can now run this concurrent program with Report Only set to **2** (**No**). He runs this concurrent program with reschedule options to run one day after completion. For information regarding the support of these views, see *Maintaining Object Migrator Views* on page 114.

11. To validate your installation, launch one of the Object Migrator programs with Report Only set to **Yes.**

    For example, Carl is validating his installation as recommended.

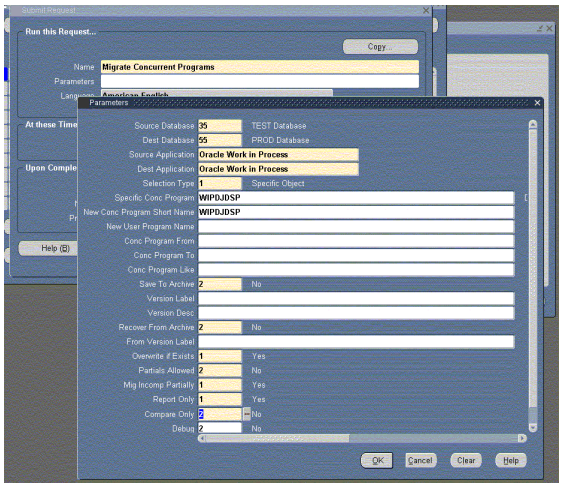
    - Carl launches Object Migrator programs.
    - Carl logs on to the Oracle E-Business Suite in the STAGE database and selects the **System Administrator** responsibility.
    - Carl navigates to the Submit Requests window **Other:Requests:Run.**
    - Carl picks **Request** for the Type field.
    - Carl enters **Migrate%** in the Name field.

      This brings up all the Mercury Object Migrator programs.
    - Carl selects **Migrate Concurrent Programs.**

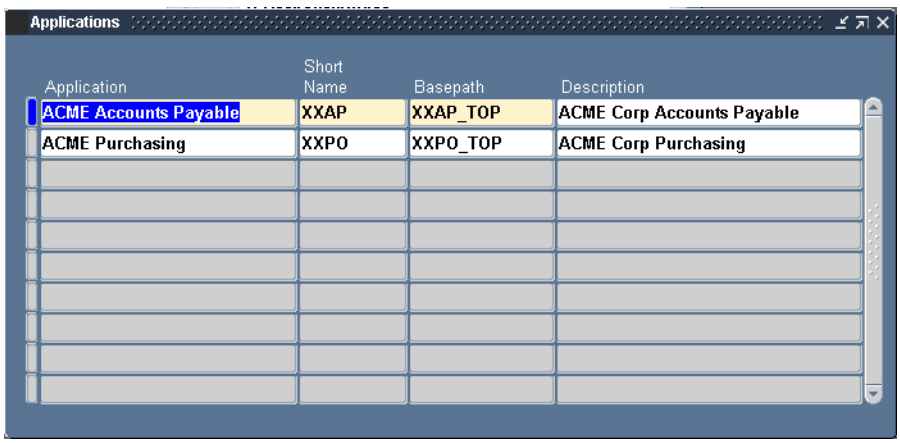
      This opens up the Parameter window.

- Carl selects source and destination databases and fills in other information as shown in the illustration below.

  In this example, Object Migrator migrates a concurrent program called WIPDJDSP from the DVLP database to the PROD database with Compare Only set to **Yes.** Carl commits the information and monitors the concurrent program. At the end of the run, Object Migrator produces an easy-to-read report that gives him the results of the program comparison.



12. In the Object Migrator instance, define any custom applications for which data will be migrated. For more information, see *Managing Custom Applications* on page 119.



See Chapter 6, *Optional Configurations for Object Migrator*, on page 99 for additional optional configurations.

Chapter

5

Upgrading Object Migrator

In This Chapter:

New Features

Overview

Release 6.0 of Mercury Object Migrator includes the following new features, most of which provide support for Oracle Applications release 11.5.10:

- **Concurrent Program Migrator.** This migrator now supports migration of new columns which are included in 11.5.10 and related to Concurrent Request Submission. The migratore also includes improved comparison logic for all columns consisting of important functional data.
- **Value Sets Migrator.** This migrator now supports migration of the Attribute Sort Order information added in 11.5.10 as well as the option to switch the query optimizer used by the migrator.
- **Responsibilities Migrator and Users Migrator**. These migrators support synchronization of workflow local tables for all Oracle Application instances having intermediate patch levels prior to release 11.5.10.
- **GUI Menus Migrator.** Apart from 11.5.10 support, improvements have been made in the following areas:
  - Multi-level migration for GUI menus using range and wildcard options
  - Comparison reporting support between archive and destination instances that have Security information.
- **Report Sets Migrator.** Improved Multi Language Support that now fully migrates the Translations and is able to archieve the Translations.
- **Concurrent Manager Definitions Migrator and GUI Menus Migrator.** These migrators now extend support for Oracle Applications release 11.5.10.
- **Descriptive Flexfields.** This migrator now supports migration of Segment View data added in 11.5.10.
- **Messages Migrator.** This migrator now supports migration of Log Severity information added in 11.5.10.
- **Mercury Object Migrator.** The product architecture extends support for the Oracle Applications instances, which use custom schema names for Application Object Library (FND) tables.

- **Product name.** The product name has been changed from Kintana Object*Migrator (abbreviated as O*M) to Mercury Object Migrator (abbreviated as OM).

Supportability

This section lists the Oracle and Mercury product releases supported by Object Migrator release 6.0 as well as upgrade compatibility requirements.

- Sites running Object Migrator release 3.1 or higher can upgrade directly to Object Migrator release 6.0.
- If you are running a version of Object Migrator less than release 3.1, then you must first upgrade to Object Migrator release 3.1 before upgrading to release 6.0. Upgrade instructions for updating to Mercury Object Migrator release 3.1 are available from the Mercury IT Governance Download Center at http://itg.merc-net.com/support/download/login.jsp.
- To run the upgrade, your system environment must support a bash- or Bourne-style shell.
- Object Migrator release 6.0 is designed to integrate with Mercury Extension for Oracle E-Business Suite release 5.5 (or later). Object Migrator release 6.0 can integrate with earlier releases of the Mercury Extension for Oracle E-Business Suite but some manual configuration is required. Mercury Extension for Oracle E-Business Suite and the Mercury IT Governance Center applications should be at the most current patch level.
- Mercury Object Migrator release 6.0 supports Oracle E-Business Suite releases 10, 11, and 11i (11.5.1 through 11.5.10 and the FND.H Minipack). As the Oracle E-Business Suite evolves with new and modified functionality, changes are reviewed for incorporation into Object Migrator. More detailed information regarding point release and patch level support, as well as known issues, is available from the Mercury IT Governance Download Center at http://itg.merc-net.com/support/download/login.jsp.

Special Considerations for Release 6.0

Please pay special attention to the following:

- You should not personalize any forms related to Mercury Object Migrator including Concurrent Programs Definitions and Submit Concurrent Request Parameters. Customizations to these forms is not supported.

- You should not convert Mercury Object Migrator custom schema (CLM) using the Oracle Applications Tablespace Model (OATM) now available with Oracle Applications release 11.5.10. If you wish to migrate to OATM, use the OATM migration utility available and exclude the Mercury Object Migrator schema.

Upgrade Impacts for Object Migrator Release 6.0

- **Concurrent Programs Migrator.** The migrator ensures that additional Concurrent Request Submission information captured in 11.5.10 for concurrent processing is retained when refreshing concurrent program definitions using OM.
- **Descriptive Flexfields Migrator.** In 11.5.10, Oracle has added Concatenated Segment View Name to the Descriptive Flexfields table. The migrator supports the migration of this column. The migrator derives the value for the View Name and makes sure that the value is non-empty and unique in the 11.5.10 destination for new Custom Descriptive Fields. For the existing Descriptive fields at the destination, the value is retained. Actual view is created using this name when the flexfield is compiled. The migrator logic is consistent with the Oracle's upgrade logic for 11.5.10.
- **Messages Migrator.** In 11.5.10, the Messages form also stores Log Severity information. The migrator defaults the value to be "Error" for this column if the Message type is "Error," and the source des not support this column or have the value as null. Otherwise the value is migrated from Source to Destination. This is parallel to Oracle's upgrade logic for 11.5.10.
- **Users and Responsibilities Migrator.** Both the migrators now have enhanced capability to accurately identify the Oracle Applications instances which have intermediate patch levels up to 11.5.10. The migrators synchronize the workflow local tables at the Destination only if the Destination supports this functionality and the Synchronize the Workflow tables parameter is set to "Yes."
- **Support for Custom APPLSYS schema name.** Object Migrator architecture now extends support for the Oracle Applications instances, which use custom schema names for Application Object Library tables (FND tables). Users having this setup do not need to make any specific configuration modifications for this—the Object Migrator architecture automatically takes care of this.
- **Value Sets Migrator.** In 11.5.10, Oracle has added Attribute Sort Order information to Value Sets. The migrator supports the migration of this

information. The migrator now also has a new hidden parameter called Default Optimizer which the users can use to switch the query optimizer being used by the Value Sets Migrator. The upgrade script adds new columns to automatically support this functionality.

- **New 11.5.10 column support.** The installation script automatically adds new columns in support of the following migrators:
  - Concurrent Programs Migrator
  - Descriptive Flexfields Migrator
  - Report Sets Migrator
  - Messages Migrator

Features from Release 5.1

Described in the following sections are the key new features for Mercury Object Migrator 5.1:

Synchronization of Workflow Local Tables

The Responsibilities Migrator and the Users Migrator now support the synchronization of workflow local tables for Oracle 11.5.9 databases.

In release 11.5.9, Oracle has added functionality to synchronize workflow local tables for use with Oracle workflow's new directory service model. The local tables store denormalized information regarding roles for better performance. The local tables are either populated individually by the Define Responsibilities or Define Users form or by batch using the Synchronize WF Local Tables program.

Object Migrator's Responsibilities Migrator and Users Migrator can synchronize the workflow local table information for each responsibility or user migrated, but users retains the option to defer the synchronization, and then initiate it in bulk themselves after migrations are complete.

Additional parameters (both called Synchronize Workflow Tables) have been added to the Responsibilities Migrator and Users Migrator for this purpose. These parameters are described in *Responsibilities* on page 220 and *Users* on page 225.

Enforcement of Type and Size Information

When migrating table-validated value sets that specify ID or meaning columns to an 11i instance, the Object Migrator now enforces that the related type and size information be included in the value set definition. Oracle began enforcing this business rule in release 11.5.8, but the functional error resulting from the missing data occurs in all 11i instances. The Object Migrator enforces this rule for all release 11i instances.

Error messages (7311 and 7312) that are generated if the source instance does not have the complete definition are documented in Appendix C, *Exception Messages*, on page 235.

Migration of Category and Severity Information

In release 11.5.9, Oracle added two new fields to the messages, allowing users to specify a message category as well as the severity for the message. Object Migrator's Messages Migrator now migrates this category and severity information to destinations that support them.

The error messages (3003 and 3004) that support this new feature are documented in Appendix C, *Exception Messages*, on page 235.

New Features for the Profile Options Migrator

The Profile Options Migrator now:

- Supports hierarchy types, including server and organization hierarchies.

  In release 11.5.9, Oracle added a concept of hierarchy type to profile options and added two new hierarchies (server and organization) to the existing security hierarchy. Object Migrator now supports migrating profile options and values for any of the three hierarchies. Server and organization profiles and values can be migrated only to instances that support those hierarchies; this is validated by Object Migrator. Profiles migrated from instances without hierarchy types to instances with hierarchy types are created as security hierarchy profiles.

  Error messages (2107 through 2110) supporting this new feature are documented in Appendix C, *Exception Messages*, on page 235.

- Supports integration with the Business Event System when adding, changing, or deleting profile option values in Oracle applications instances that include this integration (11.5.9).

  Object Migrator now raises the oracle.apps.fnd.profile.value.update Business Event in the destination for each Profile for which values have been migrated to or removed from the destination. This occurs only if the destination supports the 11.5.9 functionality and Migrate Profile Option Values is set to **Yes**.

- Supports single quotation marks in the profile option name and user profile option name.

  An error message (2111) added to support these profile option changes is documented in Appendix C, *Exception Messages*, on page 235.

New Features for the Printer Definitions Migrator

The Printer Definitions Migrator includes changes for 11.5.9 compatibility.

Additional Features

This section lists additional features from Mercury Object Migrator release 5.0.

- Management of additional user level data

  The Users Migrator now supports populating additional person party data added in Oracle 11.5.9. This feature is applicable only when creating new users.

- Capture of statistical data for concurrent programs

  The Concurrent Programs Migrator now retains the additional statistical data captured for concurrent programs in Oracle 11.5.9.

Features from Release 5.0

Mercury Object Migrator release 5.0 provided expanded functionality and features to support the evolution of the Oracle E-Business Suite release 11i.

This section lists the features introduced in Object Migrator 5.0. Features from previous product releases were incorporated into Object Migrator 5.0.

This includes the following topics:

Features for the Responsibilities Migrator

This section lists the major features for Object Migrator release 5.0 Responsibilities Migrator.

- Securing attribute values

  The Responsibilities Migrator now supports the optional migration of securing attribute values assigned to the responsibility. For additional information, see *Responsibilities Migrator Parameter Additions, Extension Integration* on page 78.

- Responsibility name

  The Responsibilities Migrator now supports migrating responsibilities containing single quotes in the responsibility name.

Features for the Concurrent Programs Migrator

This section lists the features for the Object Migrator release 5.0 Concurrent Programs migrations.

- Incompatibility types

  The Concurrent Programs Migrator now supports migration of incompatibility type information as part of concurrent program incompatibility definitions to instances where this functionality is in place.

- Queue verification

  The Concurrent Programs Migrator now submits a queue verification request in Oracle release 11i destination instances if incompatibilities have been migrated successfully.

- Existing statistical data now preserved

  The Concurrent Programs Migrator now preserves existing statistical data in the destination instance when concurrent program definitions are refreshed.

Features for the Functions Migrator

This section lists the features for the Object Migrator release 5.0 Functions Migrator.

- Maintenance mode support and context dependence

  Maintenance mode support and context dependence information is now included in functions migrations if the Oracle E-Business Suite instances support it. When migrating from an instance without this functionality to an instance with this functionality, the maintenance mode support value is defaulted to **None** and the context dependence is defaulted to **Responsibility**.

- User name uniqueness

  User function names no longer need to be unique for some Oracle E-Business Suite application patch levels. The Object Migrator takes into consideration the rules appropriate to the instance.

- Migration of JRAD reference path

  JRAD reference path information is now migrated to instances that support it.

- Function type

  The Object Migrator requires the function type for the function exist in the destination if the destination instance enforces this rule.

Additional Features

This section lists additional features from Object Migrator release 5.0.

- Predefined migrator incompatibilities

  The Object Migrator upgrade optionally predefines incompatibilities between Object Migrator programs to assure safer migrations. If this option is selected, data migrations are not allowed while any purge program is running.

- Value Sets Migrator

  Migration of value sets using hierarchical security is now supported. The Object Migrator submits the compile value set hierarchies program to compile the hierarchies after migration. Value sets using hierarchical security cannot be migrated to instances not supporting hierarchical security.

- Users Migrator performance

  For the Users Migrator, performance has been improved for instances where large numbers of customers are defined.

- Descriptive Flexfields Migrator

  The handling of column information has been improved. Columns used by the descriptive flexfield must be registered in the destination instance, and the columns must not be used by a different flexfield. Handling of name changes is also improved.

- Security group support

  Migration of security group information to instances supporting security group information is added for the following migrators:

  - Descriptive Flexfields
  - Users
  - GUI Menus
  - Concurrent Programs
  - Concurrent Manager Definitions
  - Value Sets
  - Responsibilities

- Concurrent Manager Definitions Migrator manager type

  For the Concurrent Manager Definitions Migrator, the manager type must be supported in order to migrate the manager. Due to a change in the derivation and type of data stored, migration from a release with the new definition to an older release with the old definition is prevented for certain predefined services, such as Apache JServer.

Features from Releases Prior to 5.0

This section lists important features added in releases after Object Migrator release 3.1 and prior to Object Migrator release 5.0. All features from these previous releases were incorporated into Object Migrator 5.0 and have also been incorporated into release 5.1.

- GUI Menus Migrator (release 4.2)

  The GUI Menus Migrator now submits the Compile Security program in the destination instance if the destination instance includes this functionality. The Compile Security program compiles the menus added or changed by the Object Migrator.

- Descriptive Flexfields Migrator (release 4.2)

  The Descriptive Flexfields Migrator supports context override value set specifications and accommodates certain table name changes of standard application tables between releases. For additional information, see *Descriptive Flexfields Migrator Parameter Additions, Extension Integration* on page 80.

- Users Migrator (release 4.2)

  The Migrate Securing Attributes parameter was added to the Users Migrator. For additional information, see *Users Migrator Parameter Additions, Extension Integration* on page 79.

- Responsibilities Migrator (release 4.2)

  The Responsibilities Migrator includes logic improvements for migrations involving a change to the responsibility name or a cross-release migration from Oracle E-Business Suite applications release 10 to a higher release. The improvements eliminate the potential of incorrectly finding a match for the responsibility in the destination instance.

- Concurrent Programs Migrator (release 4.2)

  The Concurrent Programs Migrator now has more accurate handling of security groups, MLS application, and executable data.

- Concurrent Manager Definitions Migrator (release 4.2)

  Improvements were made when archiving concurrent manager data.

- Request set stage function improvements (release 4.2)

  Improvements were made in the treatment of request set stage function information.

- Accommodation for an Oracle RDBMS bug (release 4.2)

  Changes were made to the Object Migrator to accommodate an Oracle RDBMS bug which produced the ORA-00902 (invalid relational operator) error message when using valid SQL syntax in some Oracle 8i releases prior to 8.1.7.2.3.

- Descriptive Flexfields Migrator (release 4.1)

  Two new parameters (Disabled Contexts and Disabled Segments) were added to the Descriptive Flexfields Migrator. These new parameters allow users the option of excluding disabled flexfield contexts and/or segments from their descriptive flexfields migrations.

- Users Migrator (release 4.1)

  Modifications were made in the handling of Customer Contact information associated with a user to comply with data model changes made in the Oracle E-Business Suite. Due to changes made to the data profile, users with Customer Contacts cannot be migrated from release 11.5.3 or later backwards to a release prior to 11.5.3 if the user does not exist in the destination.

- Release 4.0 added initial support for Oracle E-Business Suite release 11i.

Upgrade Impacts for Object Migrator Release 5.1

Mercury Object Migrator release 5.1 makes the following changes to existing Object Migrator installations. The following impacts are the result of the release 5.1 upgrade:

- rdf files replaced

  All of the of the Object Migrator programs (`rdf` files) are replaced with new versions.

- Packages and procedure replaced

  The Object Migrator packages and procedures are refreshed with new versions.

- Temporary tables re-created

  All temporary tables used by the Object Migrator are re-created. The Execution History for the Object Migrator is retained. Existing object archive tables are retained ensuring no loss of version controlled data.

- Chain Link Object Migrator™ name change

  If upgrading from Object Migrator release 3.1, the name of the Object Migrator application changes from Chain Link Object Migrator to Mercury Object Migrator.

- Archive data structure upgrade

  Tables used to store archived objects are updated with recent changes. Be sure to back up the Object Migrator schema before upgrading.

- Archive data upgrade

  As required by Object Migrator logic changes, data in the Archive is upgraded to correspond to the new logic. Usually, this occurs automatically during the upgrade, but manual steps might be required depending on the release currently installed. Be sure to back up the Object Migrator schema before upgrading.

- Incompatibilities between Object Migrator programs

  Optionally, the upgrade defines incompatibilities between Object Migrator programs to make sure migrations will not run at the same time as purge programs.

- Additional parameter for Responsibilities Migrator

  An additional parameter, Synchronize Workflow Tables, has been added to the Responsibilities Migrator. (For more information, see *Synchronization of Workflow Local Tables* on page 66.)

- Additional parameter for Users Migrator

  An additional parameter, Synchronize Workflow Tables, has been added to the Users Migrator. (For more information, see *Synchronization of Workflow Local Tables* on page 66.)

- Archive update for profile hierarchy type

  All existing archived profile options will be identified as security hierarchy profiles.

- Additional parameter for Responsibilities Migrator (if you are upgrading from a release prior to 5.0)

  A new parameter, Migrate Sec Attrib Values, is added to the Responsibilities Migrator to determine whether securing attributes values associated with the responsibility should be migrated. For more information, see *Responsibilities Migrator Parameter Additions, Extension Integration* on page 78.

Prior to using the Responsibilities Migrator, review the Responsibilities Migrator information in *Responsibilities* on page 220.

- Additional parameter for Users Migrator

  When upgrading from Object Migrator release 4.1 or earlier, a new parameter, Migrate Securing Attributes, is added to the Users Migrator. For more information, see *Users Migrator Parameter Additions, Extension Integration* on page 79.

Prior to using the Users Migrator, review the Users Migrator information in *Users* on page 225.

- Additional parameters for the Descriptive Flexfields Migrator

  When upgrading from release 4.0 or earlier, additional parameters have been added to the Descriptive Flexfields Migrator. For more information, see *Descriptive Flexfields Migrator Parameter Additions, Extension Integration* on page 80.

Prior to using the Descriptive Flexfields Migrator, review the Descriptive Flexfields Migrator information in *Descriptive Flexfields* on page 182.

- Manual update for users archive data

  When upgrading from Object Migrator release 4.1 or earlier, all Users archive data must be manually upgraded prior to using the Users Migrator. For more information, see *Manual Data Upgrade for Users Archive* on page 82.

Manual Update Information

For Object Migrator release 6.0 and some earlier releases, new features or logic require manual updates, which are described in the following sections.

Synchronize Workflow Tables Parameter, Extension Integration

(Release 5.1) A new parameter, Synchronize Workflow Tables, has been added to the Responsibilities Migrator and also to the Users Migrator. If you are integrating Object Migrator 5.1 with Mercury Change Management 5.0 or earlier, you need to add this parameter to your Change Management object types (which are usually AOL:Resp and AOL:User) manually.

Be sure to make a backup copy of these object types before incorporating the changes listed below.

This parameter indicates whether workflow local tables are to be populated in the destination. The default is **Yes**.

The parameter values are shown in *Table 5-1*.

Table 5-1. Synchronize Workflow Tables parameter attributes

Attributes	Value
Prompt	Synchronize Workflow Tables:
Token	P_SYNCH Note: You must use this token value.
Display	Yes
Editable	Yes
Display Only	Never
Required	Always
Validation	Radio Buttons (Y/N)
Default (Constant)	Yes

Responsibilities Migrator Parameter Additions, Extension Integration

(Release 5.0) When upgrading from Object Migrator releases prior to 5.0, the upgrade procedure adds a new parameter, Migrate Sec Attrib Values, to the Responsibilities Migrator. When integrating Mercury Object Migrator with Mercury Extension for Oracle E-Business Suite versions prior to release 5.0, you need to manually add the Migrate Sec Attrib Values parameter to the Mercury Change Management object type used for migrating responsibilities (usually, AOL:Resp). Before incorporating the change, make a backup copy of the existing object type.

The attributes for the Migrate Secur Attrib Values parameter are listed in Table 3-2 on page 27.

In order for the parameter to function correctly, the token value must be exactly as listed in Table 3-2 on page 27. This parameter indicates whether securing attribute values defined on the responsibility should be migrated to the destination. The default value for Migrate Sec Attrib Values is **No**.

Do not set Migrate Sec Attrib Values to **Yes** unless both the source and destination Oracle E-Business Suite application instances use version 11.5.7 or higher of the Responsibilities form. If the parameter is not defined, the migrator assumes a value of **No**.

Prior to using the Responsibilities Migrator, review the Responsibilities Migrator information in *Responsibilities* on page 220.

Table 5-2. Migrate Sec Attrib Values parameter attributes

Attributes	Value
Prompt	Migrate Sec Attrib Values:
Token	P_MIGRATE_ATTR_VALUES
Display	Yes
Editable	Yes
Display Only	Never
Required	Always
Validation	Radio Buttons (Y/N)
Default (Constant)	No

Users Migrator Parameter Additions, Extension Integration

(Release 4.2) When upgrading from Object Migrator releases prior to 4.2, the upgrade adds a new parameter, Migrate Securing Attributes, to the Users Migrator. When integrating Mercury Object Migrator with Mercury Extension for Oracle E-Business Suite versions prior to release 4.5, you need to manually add the Migrate Securing Attributes parameter to the Mercury Change Management object type used for migrating Users (usually, AOL:User). Before incorporating the change, make a backup copy of the existing object type.

The attributes for the Migrate Securing Attributes parameter are listed in Table 3-3 on page 28.

In order for the parameter to function correctly, the token value must be exactly as listed in Table 3-3 on page 28. This parameter indicates how securing attributes attached to the User should be handled during migration. The default value for Migrate Securing Attributes is **No**.

Prior to using the Users Migrator, review the Users Migrator information in *Users* on page 225.

Table 5-3. Migrate Securing Attributes parameter attributes

Attribute	Value
Prompt	Migrate Securing Attributes
Token	P_MIGRATE_ATTRIBS
Display	Yes
Editable	Yes
Display Only	Never
Required	Yes
Validation	Radio Buttons (Y/N)
Default (Constant)	No

Descriptive Flexfields Migrator Parameter Additions, Extension Integration

(Release 4.1) When upgrading from Object Migrator releases prior to 4.1, the upgrade adds two new parameters (Disabled Contexts and Disabled Segments) to the Descriptive Flexfields Migrator. When integrating Mercury Object Migrator with Mercury Extension for Oracle E-Business Suite versions prior to release 4.0, manually add these parameters to the Mercury Change Management object type used for migrating descriptive flexfields (usually, AOL:Desc Flex). A backup copy of the existing object type should be made before incorporating the change. The attributes for the Disabled Contexts and Disabled Segments parameter are listed in Table 3-4 on page 29 and Table 3-5 on page 29.

In order for the parameters to function correctly, the token values must be exactly as listed in Table 3-4 on page 29 and Table 3-5 on page 29. These parameters govern how segments and contexts disabled in the source instance are treated during migration.

.

Prior to using the Descriptive Flexfields Migrator, review the Descriptive Flexfields Migrator information in *Descriptive Flexfields* on page 182.

Table 5-4. Disabled Contexts parameter attributes

Attribute	Value
Prompt	Disabled Contexts
Token	P_DISABLED_CONTEXTS
Display	Yes
Editable	Yes
Display Only	Never
Required	Yes
Validation	Yes No Radio Buttons
Default (Constant)	Yes

Table 5-5. Disabled Segments parameter attributes

Attribute	Value
Prompt	Disabled Segments
Token	P_DISABLED_SEGMENTS
Display	Yes
Editable	Yes
Display Only	Never
Required	Yes
Validation	Yes No Radio Buttons
Default (Constant)	Yes

Manual Data Upgrade for 11.5.9 Responsibilities and Users

The Object Migrator release 5.1 upgrade includes changes to automatically populate the Workflow Local Tables for responsibilities and users after migration to an 11.5.9 destination.

Who Needs to Upgrade the 11.5.9 Responsibilities and Users Data?

Upgrade responsibilities and users definitions if:

- You are upgrading from an Object Migrator release prior to 5.1
- You have migrated to an 11.5.9 instance using an Object Migrator release prior to 5.1

How to Upgrade the Users Archive Data

Run the Synchronize Workflow Tables program in the 11.5.9 instance to which the data was migrated.

Manual Data Upgrade for Users Archive

The Object Migrator release 5.1 upgrade includes changes to the archive for the Users object type. If upgrading from a release prior to 4.2, existing archived Users data must be upgraded in order to function properly with Object Migrator release 5.1. Read the following section carefully to determine whether to upgrade archive data, and how to use the `CLM_UPG_US_DATA.sql` script.

Who Needs to Upgrade the Users Archive Data?

Upgrade archive User definitions if:

- Upgrading from an Object Migrator release prior to 4.2
- Data has ever been archived for the Users object type

If these criteria are met, complete the process of upgrading archive data before using the Users Migrator to extract data from the archive. If upgrading the archive data is not completed, some data might not be handled correctly.

How to Upgrade the Users Archive Data

The upgrade script itself contains details regarding the actions taken during execution. Table 3-6 on page 31 lists the information prompted for during the running of the `CLM_UPS_US_DATA.sql` script.

The upgrade script contains additional details regarding the actions taken during execution.

If archive data was derived from multiple source instances, run the upgrade script for each archive (version label) separately.

Table 5-6. CLM_UPS_US_DATA script parameters

Parameter	Required/ Optional	Description
source database link	Required	Enter the database link name defined to access the Oracle E-Business Suite applications instance to be used as a source instance for defaulting new archive data. Do not include the @ sign.
version label	Optional	Enter the exact version label for the archive records to update with data from the given source instance. If archive data was derived from multiple source instances, run the upgrade script for each archive (version label) separately.

If the Archive contains only data which will never be used, the data can be purged instead of running the `CLM_UPG_US_DATA.sql` script.

To Run the Manual Data Upgrade Script

To run the `Manual Data Upgrade` script:

1. Sign on to the server where Object Migrator is currently installed as the owner of the Oracle E-Business Suite applications file system. Typically, the owner of the Oracle E-Business Suite applications file system is **applmgr**.
2. Go to the `$CLM_TOP/upg_60` directory.
3. Locate the `CLM_UPG_US_DATA.sql` script. This script defaults data into the archive tables for columns newly created.
4. Connect to the Object Migrator schema via SQL*Plus.
5. Run the `CLM_UPG_US_DATA.sql` script.

   For example:

   ```
   % SQL > @CLM_UPG_US_DATA.sql
   ```

6. The `CLM_UPG_US_DATA.sql` prompts for the source database link and the version label. For information concerning the source database link and version label, see *Table 5-1* on page 77.

7. Review the output messages and commit the changes.

Upgrade Requirements

This section lists all requirements before upgrading to Object Migrator release 5.1.

Table 5-7 lists the disk space requirements for Object Migrator release 5.1.

Table 5-7. Object Migrator release 5.1 file system space requirements

Object	Required Space
Upgrade Bundle	30 MB (temporary)
Upgrade Scripts	5.0 MB
Programs (`rdf` files)	1.5 MB per program or 30 MB total

- If the application short name in Oracle E-Business Suite applications for Object Migrator is not CLM, contact Mercury Support (http://support.mercury.com) for assistance.
- Using a custom SQL account to run the Object Migrator allows for better separation between Object Migrator objects and standard Oracle E-Business Suite application objects. Some early versions of Object Migrator installed the Object Migrator in the standard **applsys** SQL account. If the Object Migrator is in the **applsys** account, contact Mercury Support (http://support.mercury.com) for complete instructions regarding the migration of the existing Object Migrator to a new SQL account.
- The Object Migrator upgrade utilizes a shell script to drive the upgrade actions. The location from which Object Migrator is being upgraded must support Bourne or bash shell executions. On Windows platforms running release 11i, the MKS Toolkit meets this requirement.
- Release 5.1 of Object Migrator backs up the data for archived users during the upgrade process. Make sure there is adequate database space before beginning the upgrade.

- Only releases of Object Migrator release 3.1 or later can upgrade directly to Object Migrator release 5.1.

Versions of Object Migrator prior to release 3.1 must upgrade to Object Migrator release 3.1 first. Upgrade instructions for Object Migrator release 3.1 are available from the Mercury IT Governance Download Center at http://itg.merc-net.com/support/download/login.jsp.

Before You Begin

This section lists information to consider or obtain before upgrading to Object Migrator release 6.0.

- Before upgrading the current version of Object Migrator, back up the current version of the Object Migrator files (such as `rdf`) and the database where Object Migrator is installed.

Before initiating an upgrade to Object Migrator release 6.0, read the entire section regarding the upgrade process.

- For UNIX upgrades, see *Upgrading Object Migrator on UNIX* on page 93.
- For Windows upgrades, see *Upgrading Object Migrator on Windows* on page 88.

Object Migrator Upgrade Directories

The main directories associated with upgrading Object Migrator to release 6.0 include:

- The current Object Migrator directory

  The current Object Migrator directory is the directory where the current version of Object Migrator resides. Object Migrator might be installed in its own <*PROD*>_TOP directory (such as CLM_TOP), or Object Migrator might share a <*PROD*>_TOP directory with an existing application. For this guide, $CLM_TOP indicates the current Object Migrator directory.

- The staging Object Migrator directory

  The staging Object Migrator directory is where the Object Migrator release 6.0 bundle is downloaded from the Mercury IT Governance Download Center. The staging Object Migrator directory should be a different directory than the current Object Migrator directory, usually a subdirectory under `$CLM_TOP`. For this guide, `$CLM_TOP/clm_upg` is the staging Object Migrator directory.

Create the `$CLM_TOP/clm_upg` directory before downloading the Object Migrator release 6.0 upgrade bundle.

- The upgrade Object Migrator directory

  The upgrade Object Migrator directory is where most of the Object Migrator upgrade scripts and libraries reside. The upgrade Object Migrator directory is created by the untar process for UNIX and the unzip process for Windows. For this guide, the upgrade Object Migrator directory is `$CLM_TOP/clm_upg/upg_60`. In addition to the upgrade scripts and libraries, the `$CLM_TOP/clm_upg/upg_60` directory creates the following subdirectories:

  - `srw` subdirectory contains all of the Object Migrator executables for Object Migrator release 6.0.
  - `r10` and `r11` subdirectories contain release-specific scripts used by the upgrade.

The Object Migrator Upgrade Script

This section details information concerning the Object Migrator upgrade script.

- For UNIX, the upgrade script is `CLM_UPG_60`
- For Windows, the upgrade script is `CLM_UPG_60_NT`

The upgrade scripts are in the `$CLM_TOP/upg_60` directory created by the untar process for UNIX or unzip process for Windows.

The upgrade scripts are interactive programs prompting for the parameters detailed in Table 4-2. Regardless of what shell is running, the installation script spawns a Bourne shell and runs under it. In each case, the interactive program

verifies the information entered and proceeds if the information is valid. For example, the upgrade program attempts to connect to the database using the user name and password entered.

To stop the upgrade process, enter a period (.) at any of the prompts.

The upgrade scripts output log messages as standard output to the screen. The upgrade scripts also saves the output to a log file named `clm_log.out` in the current directory.

Table 5-8. Upgrade script parameters

Parameter	Description
executable name for SQL*Plus	The executable used to initiate an SQL*Plus session: sqlplus or plus80.
OM acct_name	The user name for the Object Migrator SQL account already created and registered in Oracle E-Business Suite applications.
OM acct password	The password for the Object Migrator SQL account.
database connect string	Specifies the database where Object Migrator is installed. • If using SQL*Net 1.0, enter the entire two-task database connect string. (such as T:dev_machine:MIGRATE) • If using SQL*Net 2.0 or higher, enter the database identified (usually the ORACLE_SID of the database).
name of tablespace for OM tables	All Object Migrator tables goes into the specified tablespace. **The SQL account must have permissions to write to this tablespace.**
name of tablespace for OM indexes	All Object Migrator indexes goes into the specified tablespace. **The SQL account must have permissions to write to this tablespace.**
APPS acct_name	The user name for the Oracle E-Business Suite applications APPS account.
APPS acct password	The password for the APPS acct_name SQL account.
incompatibilities option	**Y** or **N**. The Object Migrator upgrade can define standard incompatibilities between Object Migrator programs. Enter **Y** to create these incompatibilities. Existing incompatibilities are not changed.

Upgrading Object Migrator on Windows

This section details how to upgrade to Object Migrator release 6.0.

Before upgrading Object Migrator, back up the current version of the Object Migrator files (such as `rdf`) and the database where Object Migrator is installed.

To upgrade Object Migrator release 6.0 on Windows:

1. Download the Object Migrator release 6.0 upgrade bundle. The Object Migrator release 6.0 upgrade bundle is a zip file.

   a. Log on to the Mercury IT Governance Download Center at http://itg.merc-net.com/support/download/login.jsp.

   b. Download the Object Migrator release 6.0 upgrade bundle to the staging Object Migrator directory, `$CLM_TOP/clm_upg` (see *Object Migrator Upgrade Directories* on page 85). The name of the Object Migrator release 6.0 upgrade bundle (file) is `CLM_UPG60.zip`.

2. Sign on to the server where Object Migrator is currently installed as the owner of the Oracle file system. Typically, the owner of the Oracle E-Business Suite applications file is **applmgr**. Set the Oracle E-Business Suite applications context, is required.

3. From the staging Object Migrator directory, extract the Object Migrator release 6.0 upgrade bundle files in the `$CLM_TOP/clm_upg` directory using the unzip utility. This process creates the `$CLM_TOP/clm_upg/upg_60` subdirectory containing the upgrade files.

4. Determine the current Object Migrator release version.

   a. In the command shell, navigate to the `$CLM_TOP/clm_upg/upg_60` subdirectory.

   b. Connect to the database containing the current installation of Object Migrator as the Oracle user owning Object Migrator (for example, CLM).

   c. Run the `CLM_REL_INFO.sql` script in the `$CLM_TOP/clm_upg/upg_60` directory. This script outputs the current Object Migrator release.

      For example:

      ```
      % SQL > @CLM_REL_INFO.sql
      ```

If the currently installed version of Object Migrator is less than release 3.1, STOP. DO NOT PROCEED. Upgrade Object Migrator to release 3.1 prior to upgrading to release 6.0. Obtain Object Migrator release 3.1 upgrade from the Mercury IT Governance Download Center at http://itg.merc-net.com/support/download/login.jsp.

If the application short name for Object Migrator in Oracle E-Business Suite applications is not CLM, contact Mercury Support (http://support.mercury.com). DO NOT PROCEED with the upgrade.

5. Run the upgrade script.

   From the `$CLM_TOP/upg_60` subdirectory, run the upgrade script `CLM_UPG_60_NT`. For information concerning the `CLM_UPG_60_NT` upgrade script, see *The Object Migrator Upgrade Script* on page 86. The upgrade script prompts for information found in Table 4-2 on page 36. The following is only an example of the `CLM_UPG_60_NT` upgrade script. For this example, the following is true:

   - CLM is the Object Migrator SQL account name and password
   - STAGE is the database connect string
   - CLM_DATA is the tablespace for Object Migrator tables
   - CLM_INDEX is the tablespace for Object Migrator indexes
   - APPS is the APPS account SQL user name and password
   - Enter **y** (or **Y**) to create default incompatibilities

   a. Start a command shell and set the applications context, if required.

   b. Navigate to the `$CLM_TOP/clm_upg/upg_60` directory.

   c. If the environment supports bash, start a bash shell. In environments where MKS sToolkit is in use, this is not required, but MKS must be in the user's path.

d. Run the `CLM_UPG_60_NT` script.

For example:

```
prompt> sh CLM_UPG_60_NT
Enter the command name(sqlplus/plus80/plus33) used for
invoking SQL*Plus (. to exit): sqlplus
Enter username of Object Migrator sql account (. to
exit):
CLM
Enter the password for CLM sql account (. to exit):
CLM <<does not print>>
Please reenter for verification:
CLM  <<does not print>>
Please enter the connect string (ex. PROD, DEV, etc.)
for the Object Migrator database.  :
STAGE
Verifying Username and Password, Please
wait....successful
Enter the tablespace for the Object Migrator tables (. to
exit):
CLM_DATA
Verifying tablespace, Please wait....successful
Enter the tablespace for the Object Migrator indexes (.
to exit):
CLM_INDEX
Verifying tablespace, Please wait....successful
Enter username of APPS user - usually APPS (. to exit):
APPS
Enter the password for APPS user (. to exit):
APPS <<does not print>>
Please reenter for verification:
APPS <<does not print>>
Verifying Username and Password, Please
wait....successful
Do you want to get the standard incompatibilies for
Object Migrators ? (y/n)(. to exit): y
```

The upgrade runs, outputting log messages as standard output to the screen. The upgrade also saves the output to a log file in the current directory. The log file name is `clm_upg_60.out`.

6. Review the upgrade output log file for errors. If unexpected errors are found, contact Mercury Support (http://support.mercury.com) for information on how to proceed.

7. Run the `Manual Data Upgrade` script.

   - For systems upgrading from Object Migrator release 4.2 or higher, proceed to the next step.
   - For systems upgrade from releases of Object Migrator earlier than release 4.2, Object Migrator release 6.0 requires updates to archive data for the Users object type that cannot be made during the upgrade. If upgrading from a release earlier than release 4.2, follow the instructions in *Manual Update Information* on page 77.

If upgrading from an Object Migrator release earlier than release 4.2, complete the process of upgrading archive data before using the Users Migrator to extract data from the archive. If an upgrade of the archive data is not completed, some data might not be handled correctly.

8. Install the Object Migrator executables required for the Object Migrator release 6.0.

   a. Using your favorite browser, copy all the files under the `$CLM_TOP/clm_upg/upg_60/srw` subdirectory (the files all have the `rdf` file extension) to the area where the current Object Migrator executables are stored. These new files replace all existing Object Migrator `rdf` files. Typically, these files are stored in the following:

      - `$CLM_TOP/srw` for release 10.7 instances
      - `$CLM_TOP/reports` for release 11 instances
      - `$CLM_TOP/reports/US` for release 11i instances

9. Verify the Database Settings.

   Select the open_links setting of the database parameter (select value from v$parameter where the name is open_links). If this value (the default value is 4) is smaller than the maximum number of databases being migrated, increase this setting to a number larger than the maximum number of databases (suggested is 20). This parameter can be changed through the init.ora file (add the line `open_links = 20`). If this parameter is changed, shut down and restart the database before continuing.

10. Re-create the Object Migrator Views.

    In Oracle E-Business Suite applications, run the Maintain Object Migrator Views program with Report Only set to **2** (**No**) to refresh the view definition. Schedule this program to run periodically (suggested is once a day) to keep the Object Migrator views current with the number and status of databases. For information regarding the support of these views, see *Maintaining Object Migrator Views* on page 114.

Be sure to view the report output of this program. The output lists all the enabled databases for use with the Object Migrator, and any errors encountered while trying to connect to these databases.

11. Recompile the descriptive flexfields (recommended).

    Recompile all descriptive flexfields. Oracle E-Business Suite applications recommends a periodic recompile the descriptive flexfields. To recompile the descriptive flexfields:

    a. Sign on to the server where Object Migrator is currently installed as the owner of the Oracle E-Business Suite applications file system, which is typically **applmgr**.

    b. Run the `fdfcmp` command using the syntax appropriate for the Oracle E-Business Suite applications release.

       For example:

       ```
       $FND_TOP/bin/fdfcmp apps/[apps passwd] 0 Y
       ```

12. Review the sections *Overview* on page 62 and *Upgrade Impacts for Object Migrator Release 5.1* on page 74 for additional information, changes, and features of Object Migrator release 6.0. See *Manual Update Information* on page 77 for additional procedures that might be required to support Object Migrator release 6.0.

    If using Object Migrator with Mercury Change Management Extension for Oracle E-Business Suite, make any configuration changes required for the Extension.

    Object Migrator release 6.0 is now ready. Contact Mercury Support (http://support.mercury.com) if any questions or issues arise.

Upgrading Object Migrator on UNIX

This section details how to upgrade to Object Migrator release 6.0. Before starting the upgrade, review the following sections:

- *Upgrade Requirements*
- *Before You Begin*
- This entire section

Before upgrading the current version of Object Migrator, backup the current version of the Object Migrator files (such as `rdf`) and the database where Object Migrator is installed.

To upgrade Object Migrator release 6.0 on UNIX:

1. Download the Object Migrator release 6.0 upgrade bundle. The Object Migrator release 6.0 upgrade bundle is a compressed tar file.

   a. Log on to the Mercury IT Governance Download Center at:

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

   b. Download the Object Migrator release 6.0 upgrade bundle to the staging Object Migrator directory, `$CLM_TOP/clm_upg` (for more information, see *Object Migrator Upgrade Directories* on page 85). The name of the Object Migrator release 6.0 upgrade bundle (file) is `CLM_UPG60_tar.Z`.

2. Sign on to the server where Object Migrator is currently installed as the owner of the Oracle file system, typically **applmgr**. Set the Oracle E-Business Suite applications context.

3. From the staging Object Migrator directory, uncompress and untar the Object Migrator release 6.0 upgrade bundle using the uncompress command and the tar command. This process creates the `$CLM_TOP/clm_upg/upg_60` subdirectory containing the upgrade files.

   For example:

```
% cd $CLM_TOP/clm_upg
% uncompress CLM_UPG60_tar.Z
% tar -xvf CLM_UPG60_tar
```

4. Determine the current Object Migrator release version.

   c. Navigate to the `$CLM_TOP/clm_upg/upg_60` subdirectory created by the untar process.

   d. Connect to the database containing the current installation of Object Migrator as the Oracle user owning Object Migrator (such as CLM).

   e. Run the `CLM_REL_INFO.sql` script in the `$CLM_TOP/clm_upg/upg_60` directory. This script outputs the current Object Migrator release.

      For example:

      ```
      % SQL > @CLM_REL_INFO.sql
      ```

If the currently installed version of Object Migrator is less than release 3.1, STOP. DO NOT PROCEED. Object Migrator must be upgraded to at least release 3.1 prior to upgrading to release 6.0. Obtain the Object Migrator release 3.1 upgrade from the Mercury IT Governance Download Center at http:/itg.merc-net.com/support/download/login.jsp.

If the application short name for Object Migrator in Oracle E-Business Suite applications is not CLM, contact Mercury Support (http://support.mercury.com). DO NOT PROCEED with the upgrade.

5. Run the upgrade script.

   From the `$CLM_TOP/upg_60` subdirectory, run the upgrade script `CLM_UPG_60`. For information concerning the `CLM_UPG_60` upgrade script, see *The Object Migrator Upgrade Script* on page 86. The upgrade script prompts for information found in Table 4-2 on page 36. The following is only an example of the `CLM_UPG_60` upgrade script. For this example, the following is true:

   - CLM is the Object Migrator SQL account name and password
   - STAGE is the database connect string
   - CLM_DATA is the tablespace for Object Migrator tables
   - CLM_INDEX is the tablespace for Object Migrator indexes
   - APPS is the APPS account SQL user name and password

- Enter **y** to create default incompatibilities

  For example:

```
prompt% CLM_UPG_60
Enter the command name(sqlplus/plus80/plus33) used for
invoking SQL*Plus (. to exit): sqlplus
Enter username of Object Migrator sql account (. to
exit): CLM
Enter the password for CLM sql account (. to exit): CLM
<<does not print>>
Please reenter for verification: CLM  <<does not print>>
Please enter the connect string (ex. PROD, DEV, etc.)
for the Object Migrator database.  : STAGE
Verifying Username and Password, Please
wait....successful
Enter the tablespace for the Object Migrator tables (. to
exit): CLM_DATA
Verifying tablespace, Please wait....successful
Enter the tablespace for the Object Migrator indexes (.
to exit): CLM_INDEX
Verifying tablespace, Please wait....successful
Enter username of APPS user - usually APPS (. to exit):
APPS
Enter the password for APPS user (. to exit): APPS <<does
not print>>
Please reenter for verification: APPS <<does not print>>
Verifying Username and Password, Please
wait....successful
Do you want to get the standard incompatibilies for
Object Migrators ? (y/n)(. to exit): y
```

  The upgrade runs, outputting log messages as standard output to the screen. The upgrade also saves the output to a log file in the current directory. The log file name is `clm_upg_60.out`.

6. Review the upgrade output log file for errors. If unexpected errors are found, contact Mercury Support (http://support.mercury.com) for information on how to proceed.

7. Run the `Manual Data Upgrade` script.

   - For systems upgrading from Object Migrator release 4.2 or higher, proceed to the next step.

   - For systems upgrade from releases of Object Migrator earlier than release 4.2, Object Migrator release 6.0 requires updates to archive data for the Users object type that cannot be made during the upgrade. If upgrading from a release earlier than release 4.2, follow the instructions in *Manual Data Upgrade for Users Archive* on page 82.

.

If upgrading from an Object Migrator release earlier than release 4.2, complete the process of upgrading archive data before using the Users Migrator to extract data from the archive. If an upgrade of the archive data is not completed, some data might not be handled correctly.

8. Install the Object Migrator executables required for the Object Migrator release 6.0.

   a. Navigate to the `$CLM_TOP/clm_upg/upg_60/srw` subdirectory created by the untar process. The `srw` subdirectory contains all of the current version of Object Migrator programs.

   b. Copy all the files under the `srw` subdirectory (the files all have the `rdf` file extension) to the area where the current Object Migrator executables are stored. These new files replace all existing Object Migrator `rdf` files, usually:

      - `$CLM_TOP/srw` for release 10.7 instances
      - `$CLM_TOP/reports` for release 11 instances
      - `$CLM_TOP/reports/US` for release 11i instances

      For example:

      ```
      % cp
      $CLM_TOP/clm_upg/upg_60/srw/*.rdf $CLM_TOP/reports/US/
      ```

9. Verify the Database Settings.

   Select the open_links setting of the database parameter (select value from v$parameter where the name is open_links). If this value (the default value is 4) is smaller than the maximum number of databases being migrated, increase this setting to a number larger than the maximum number of databases (suggested is 20). This parameter can be changed through the init.ora file (add the line `open_links = 20`). If this parameter is changed, shut down and restart the database before continuing.

10. Re-create the Object Migrator Views.

    In Oracle E-Business Suite applications, run the Maintain Object Migrator Views program with Report Only set to **2 (No)** to refresh the view definition. This program should be scheduled to run periodically (suggested is once a day) to keep the Object Migrator views current with the number and statuses or databases. See Chapter 7, *Maintaining Object Migrator,* on page 113 for details regarding the support of these views.

Make sure to view the report output of this program. The output lists all the enabled databases for use with the Object Migrator, and any errors encountered while trying to connect to these databases.

11. Recompile the descriptive flexfields (recommended).

    Recompile all descriptive flexfields. Oracle E-Business Suite applications recommends a periodic recompile the descriptive flexfields. To recompile the descriptive flexfields:

    a. Sign on to the server where Object Migrator is currently installed as the owner of the Oracle E-Business Suite applications file system, typically **applmgr**.

    b. Run the **fdfcmp** command using the syntax appropriate for the installed Oracle E-Business Suite applications release.

       For example,

       ```
       $FND_TOP/bin/fdfcmp apps/[apps passwd] 0 Y
       ```

12. Review the sections *Upgrade Impacts for Object Migrator Release 5.1* on page 74 and *Features from Release 5.1* on page 65 for additional information, changes and features of Object Migrator release 6.0. See *Manual Update Information* on page 77 for additional procedures that might be required to support Object Migrator release 6.0.

13. If using Object Migrator with Mercury Change Management, make any configuration changes required to the installed version of the Mercury Extension for Oracle E-Business Suite.

14. Object Migrator release 6.0 is now ready. Contact Mercury Support (http://support.mercury.com) if any questions or issues arise.

Chapter

6

Optional Configurations for Object Migrator

In This Chapter:

Setting Up a Separate Responsibility for Migrations

It is possible to set up a separate responsibility solely intended for Migrations and then assign this responsibility to specific users. The following example illustrates one implementation possibility.

To set up a separate responsibility for migrations:

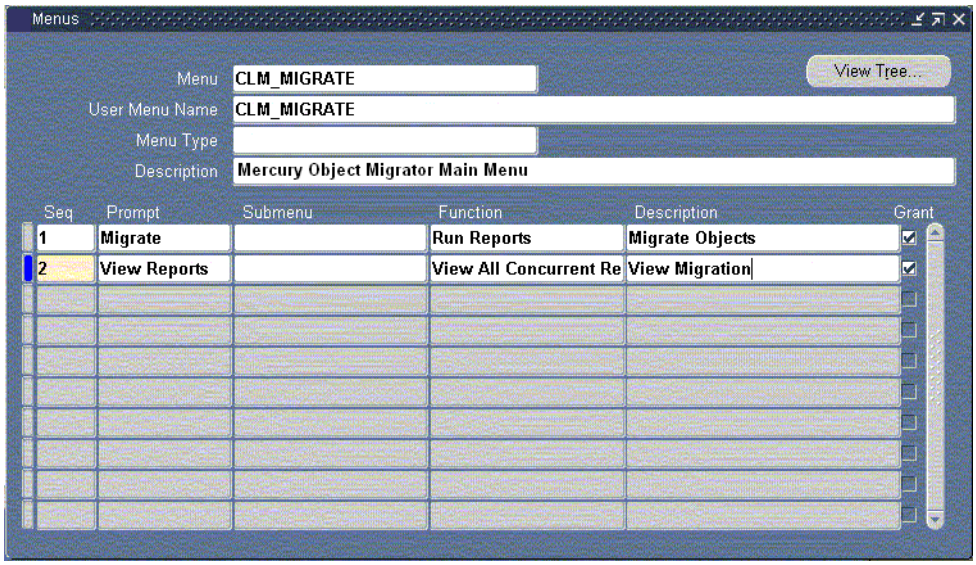
1. Sign on to the Oracle Applications on the instance in which the Mercury Object Migrator is installed.

2. Select the **System Administrator** responsibility.

3. Navigate to the Define Menu form. (**Application:Menu**).

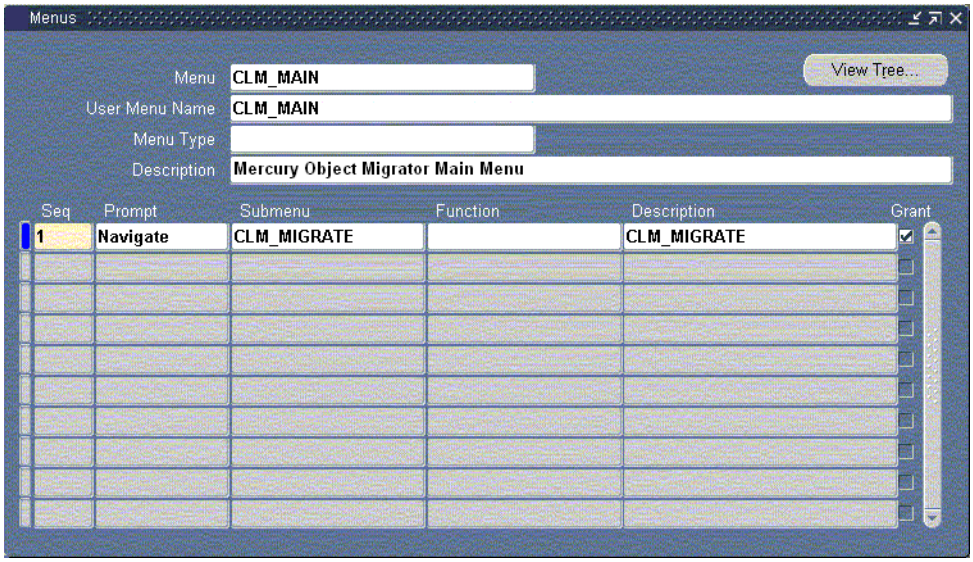
4. Create a new menu called **CLM_MIGRATE.**

   Add the Run Reports and View All Concurrent Requests forms to the menu. This allows users to submit Object Migrator requests and view the results.



5. Create another menu called **CLM_MAIN.**

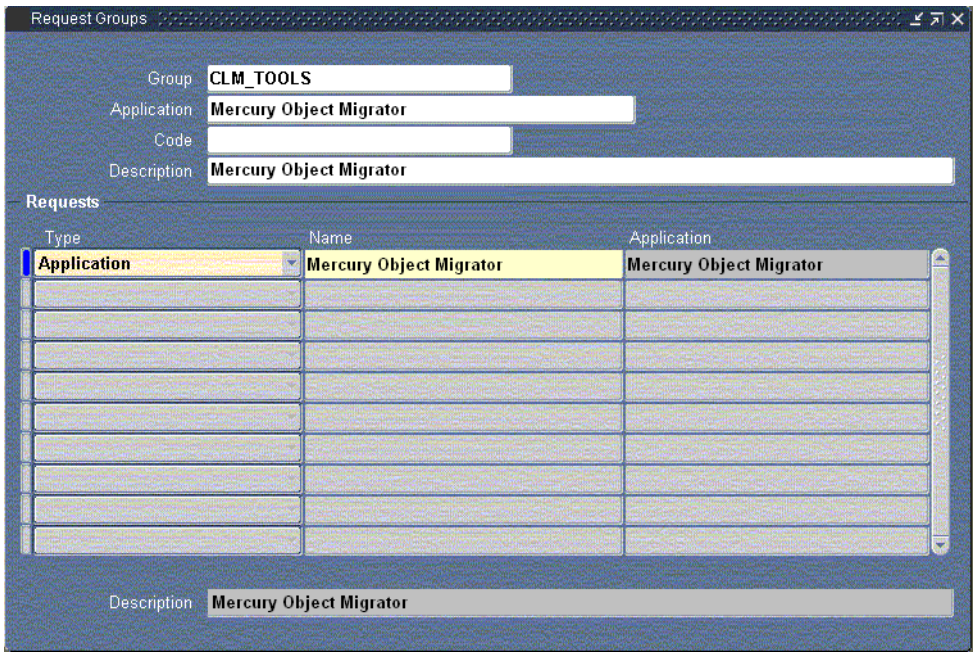
   Add the **CLM_MIGRATE** menu as a menu entry. This become the main navigation menu for the new responsibility.



6. Navigate to the Define Request Group form.
   (**Security:Responsibility:Request**)

7. Create a new request group called **CLM_TOOLS.**

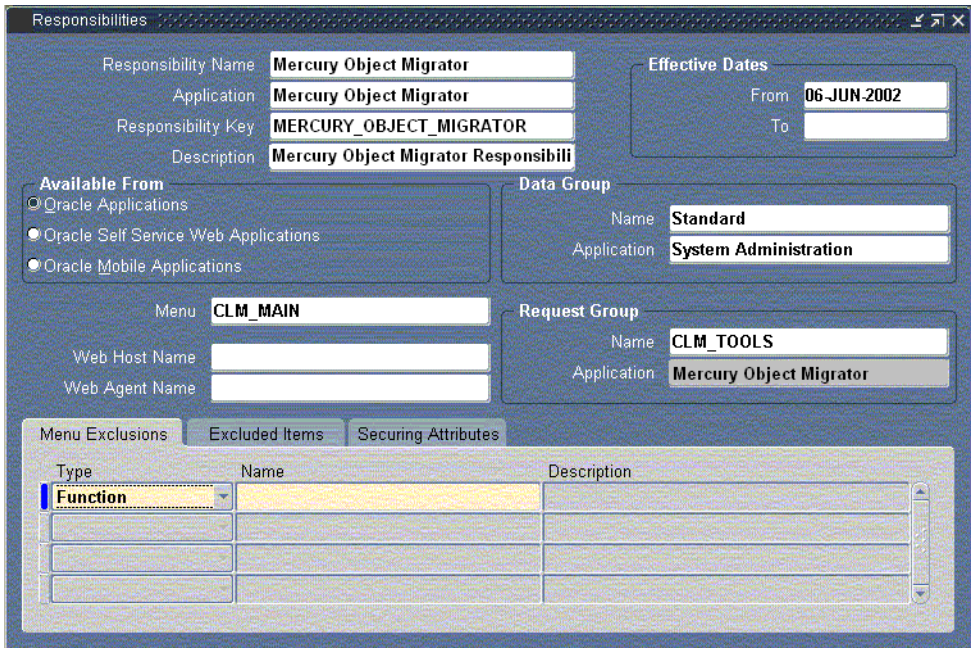
   Add Object Migrator programs to the request group. Either add individual programs if the group should include limited functionality, or the Object Migrator application if all programs will be allowed.



8. Navigate to the Define Responsibility form. (**Security:Responsibility:Define**)

9. Define a new responsibility called **Mercury Object Migrator.**

   Add the request group and menu defined above. (Alternatively, it is possible to include a standard menu and define explicit menu exclusions.)



10. Navigate to the Define Concurrent Programs window. (**Concurrent:Program:Define**)

    a. Modify default values as desired.

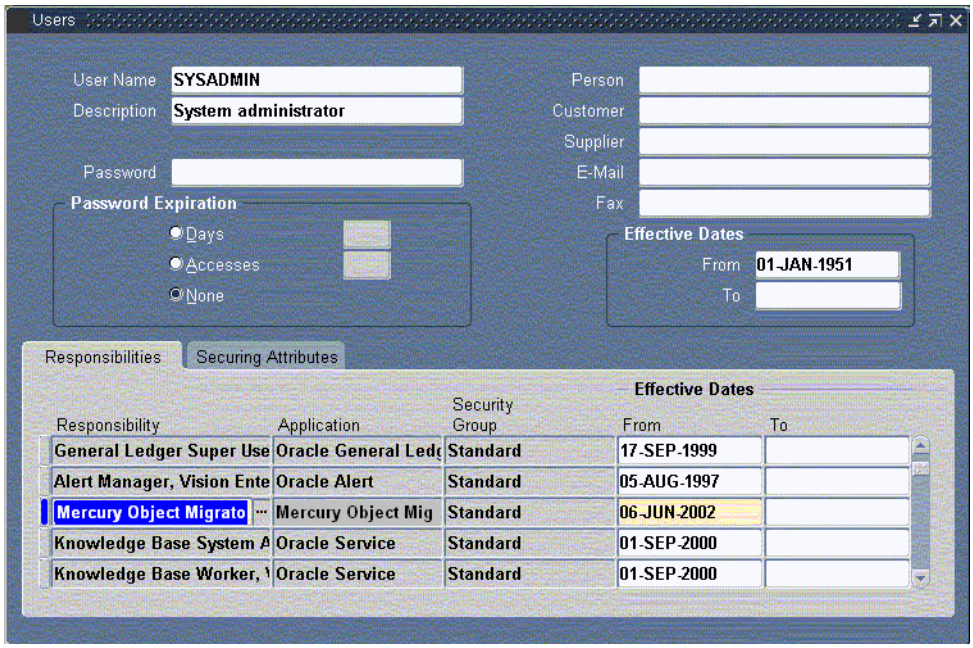
    b. If there are Object Migrator programs not in use, disable them to prevent them from appearing in select lists.

    c. If different groups of users should have different access to a program, copy the program, and modify it appropriately.

    See *Setting Up an Unrestricted Migrator* on page 104 for a sample of these types of changes.

11. Navigate to the Define Application User window (**Security:User:Define**).

12. Assign the custom responsibility defined in step 9 to the users who would be doing the migrations.



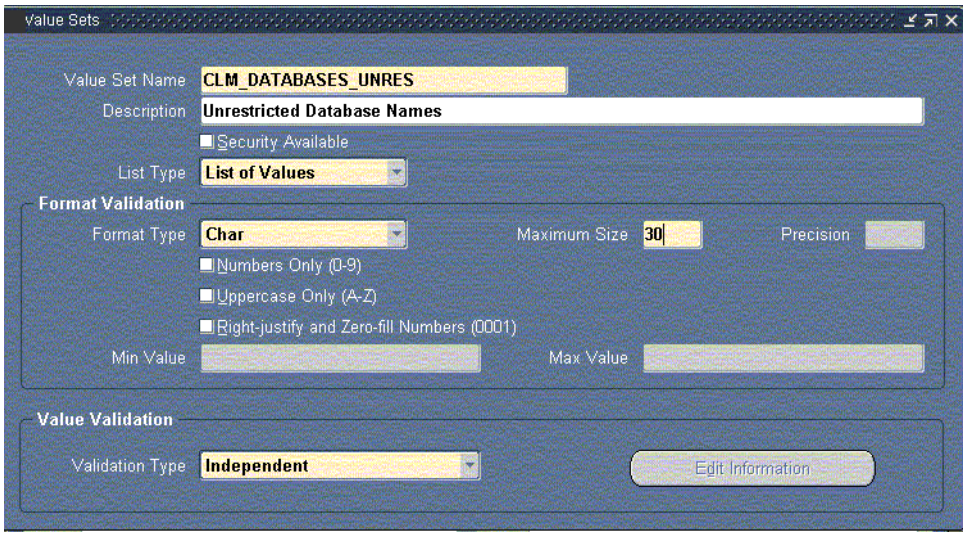
Setting Up an Unrestricted Migrator

Setting up an unrestricted migrator allows users to migrate objects among all the non-critical database instances while restricting their ability to migrate objects into critical database instances such as a production database instance.

To accomplish this, create a new set of concurrent programs by copying the existing Object Migrator programs and changing the value sets behind the parameters for each of the programs.

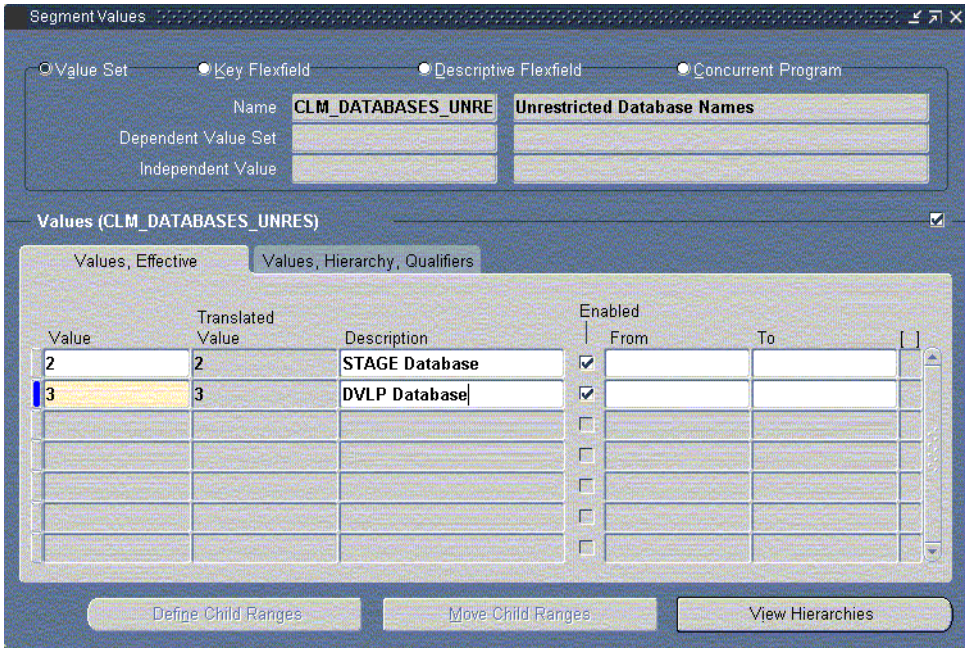
The following is an example case of setting up an unrestricted migrator:

1. Create a new value set called **CLM_DATABASES_UNRES** by copying the existing value set **CLM_DATABASES.**



2. Remove the databases that are considered critical from the **CLM_DATABASES_UNRES** value set.

   After this step, the value set should look like the ones in the following window. In this example, assume that **PILOT** and **PROD** are critical databases. Access to the object archive is also not allowed (the unrestricted migrator will not have version control ability).



Note

The values in this new value set must match the values in the original value set for each database entered.

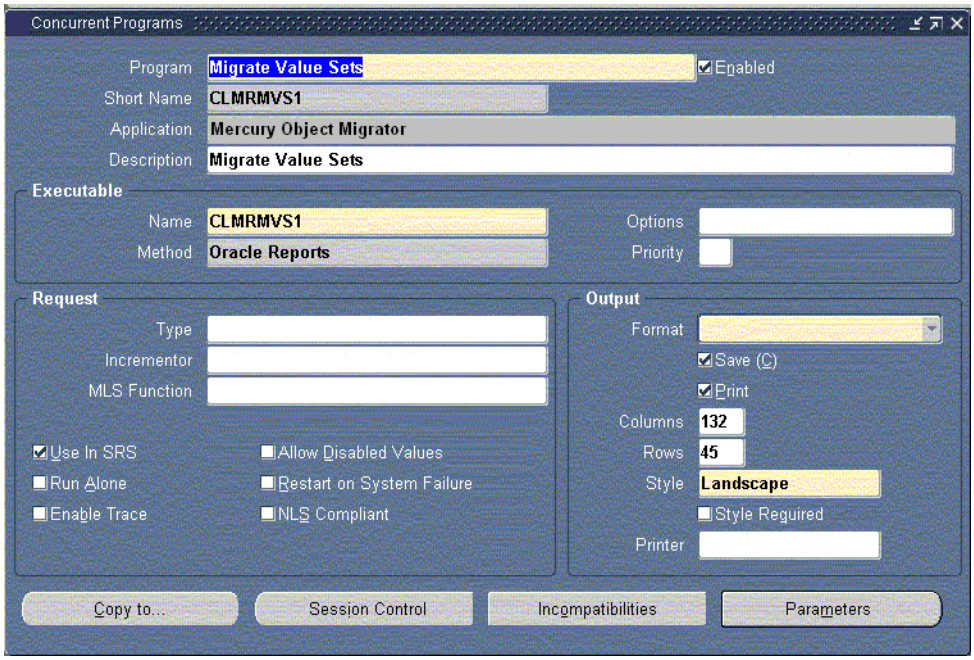
3. Copy all the existing Object Migrator concurrent programs to create identical concurrent programs for the unrestricted migrator.

   a. Navigate to the Define Concurrent Programs window. (**Concurrent:Program:Define**)

   b. Query **Mercury Object Migrator** in the Application field.

   c. Use the Down Arrow key to review all the concurrent programs that belong to Object Migrator.

      This example uses the Migrate Value Sets program (short name CLMRMVS1) to set up a Migrate Value Sets - Unrestricted program (short name CLMRMVS1_UNRES).



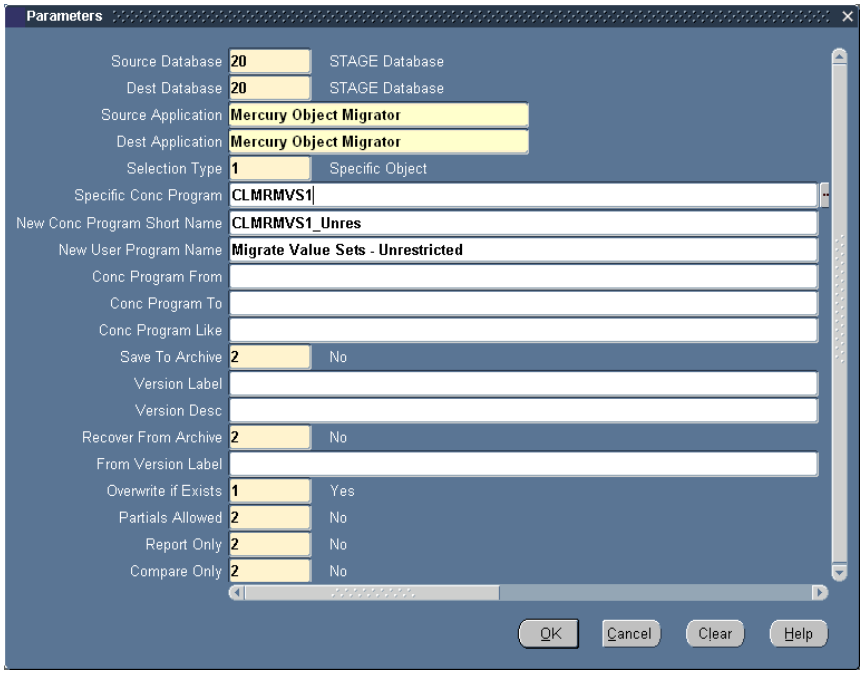
- If your Oracle E-Business Suite release supports it, use the **Copy to** button to create a copy of CLMRMVS1 named CLMRMVS1_UNRES.

- If your Oracle E-Business Suite release does not support Copy To functionality, copy the program using the Concurrent Programs Migrator.

  i. Navigate to the Submit Requests window. (**Other:Requests:Run**)

  ii. Launch the Concurrent Programs Migrator as shown in the following window to copy CLMRMVS1 and create a new program called CLMRMVS1_UNRES.

  The figure below shows the parameters to launch the Migrate Concurrent Programs report to copy CLMRMVS1 to CLMRMVS1_UNRES.



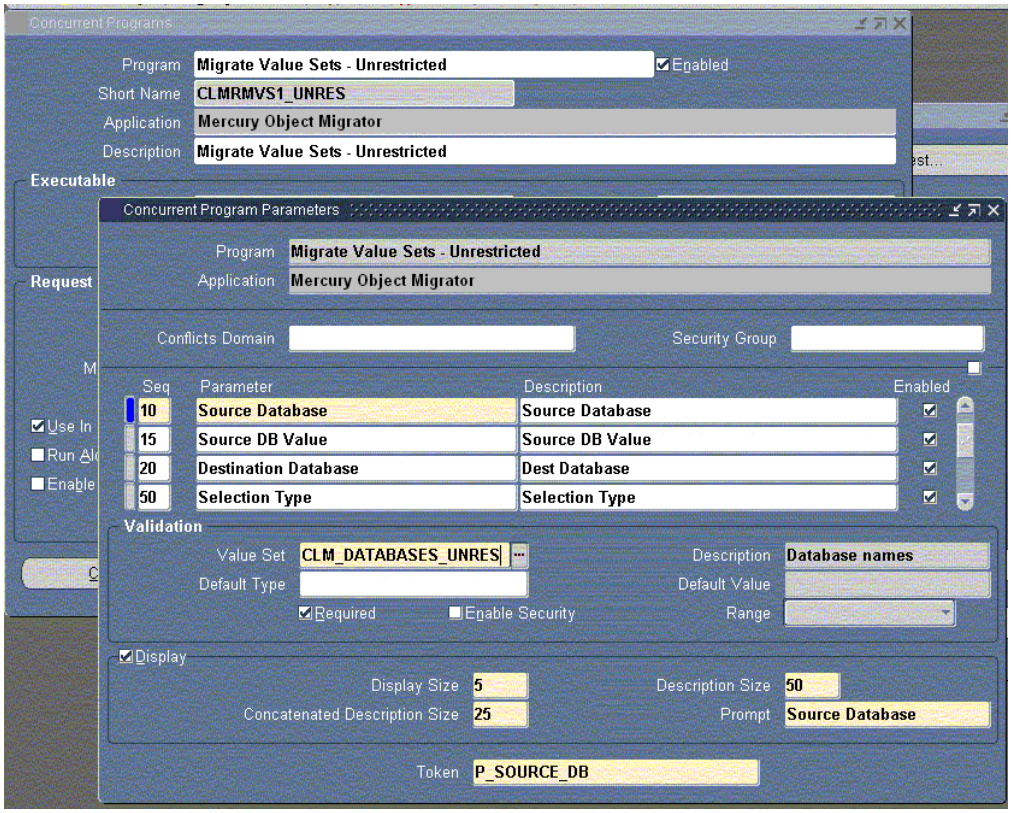
4. Assign the new value set to the new concurrent programs.

   a. Navigate to the Define Concurrent Programs window. (**Concurrent:Program:Define**)

   b. Query for the new concurrent program, CLMRMVS1_UNRES.

   c. Click **Parameters.**

   d. Change the value set associated with Destination Database from **CLM_DATABASES** to **CLM_DATABASES_UNRES.**



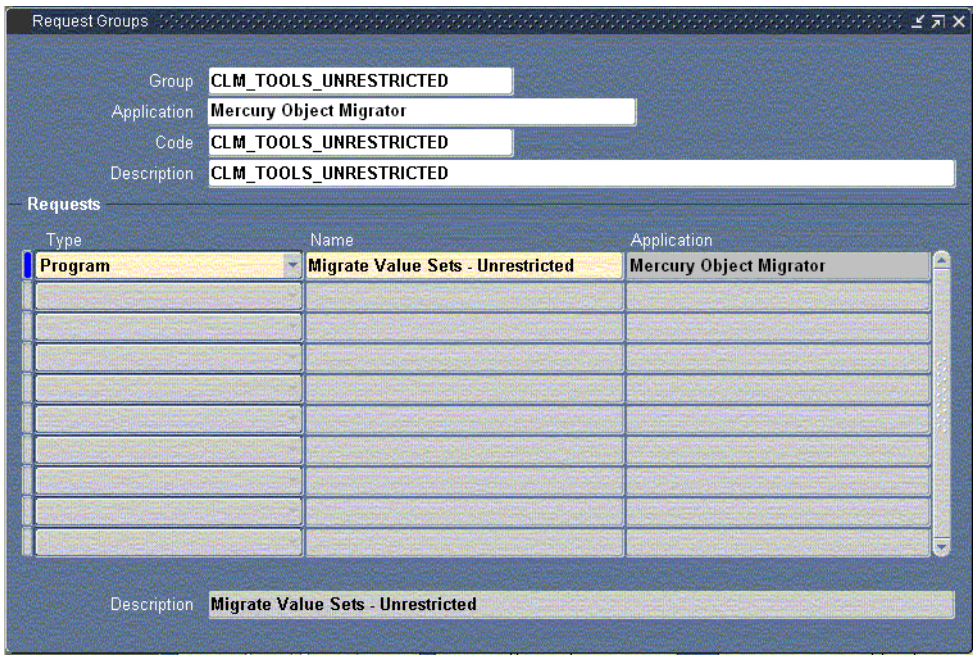
5. Create a custom menu. See *Setting Up a Separate Responsibility for Migrations* on page 100 for more information.

6. Create a request group for the unrestricted migrator.

   a. Navigate to the Define Request Groups window. (**Security:Responsibility:Request**)

   b. Create a new request group called **CLM_TOOLS_UNRESTRICTED.**

   c. Associate all the unrestricted migrator concurrent programs to this request group.

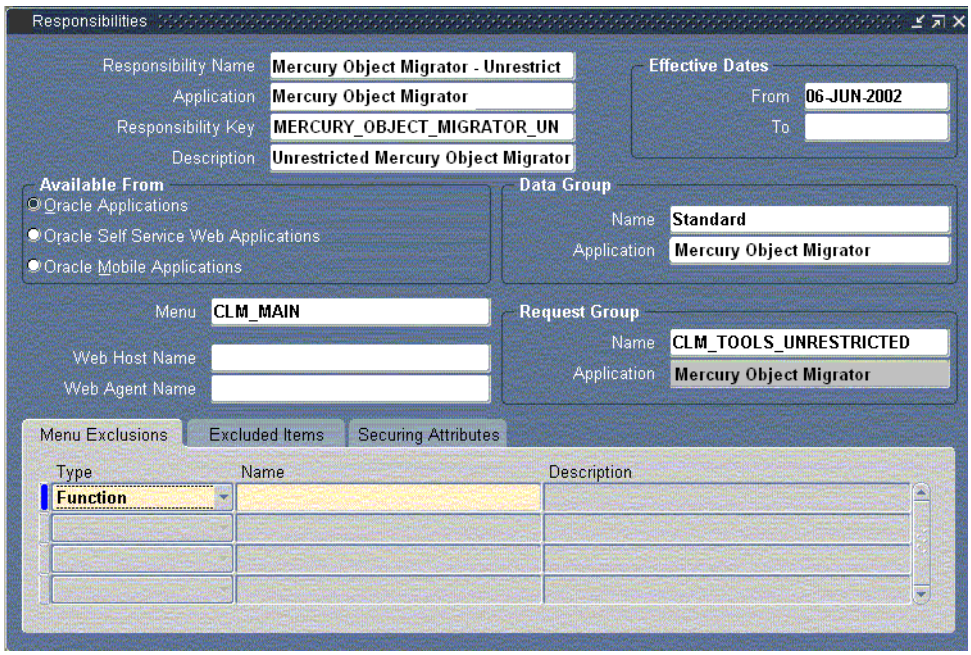


7. Create a new responsibility for the unrestricted migrator.

   a. Navigate to the Define Responsibility window. (**Security:Responsibility:Define**)

   b. Create a new responsibility called **Mercury Migrator - Unrestricted.**

   c. Associate the new request group CLM_TOOLS_UNRESTRICTED to this responsibility as shown in the following window.



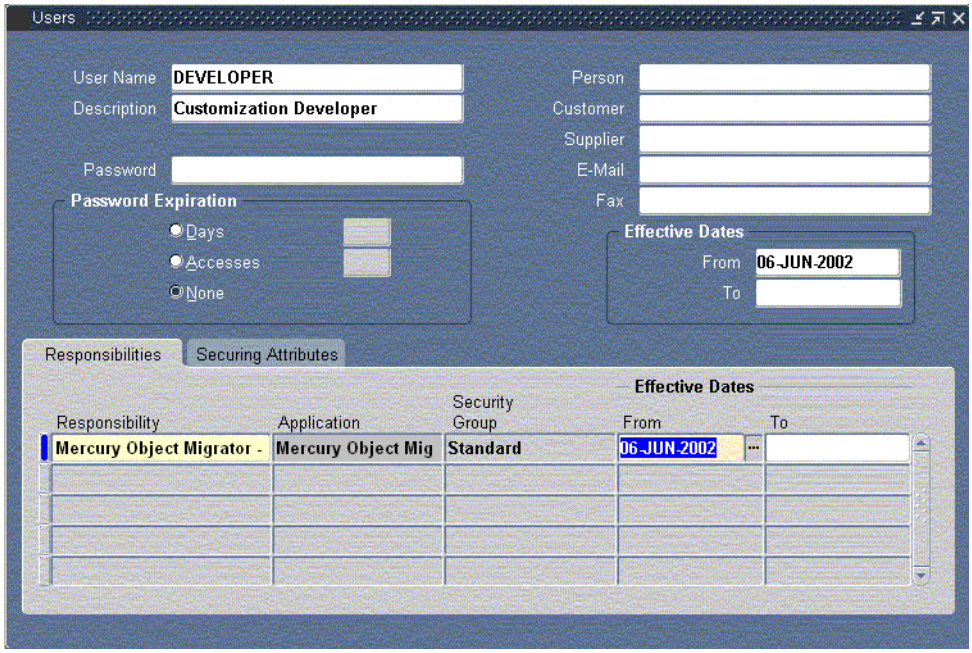
8. Assign the new responsibility to users.

   a. Navigate to the Define Application User window. (**Security:Use:Define**)

   b. Search for the user to whom the responsibility will be assigned.

   c. Add the **Mercury Migrator - Unrestricted** responsibility to the user.

      This user now can migrate objects between all the non-critical database instances while the users who have the Mercury Migrator responsibility can migrate objects among all the database instances, including the critical instances.


Users
User Name
DEVELOPER
Description
Customization Developer
Password
Password Expiration
Days
Accesses
None
Person
Customer
Supplier
E-Mail
Fax
Effective Dates
From
06-JUN-2002
To
Responsibilities
Securing Attributes
Effective Dates
Responsibility
Application
Security Group
From
To
Mercury Object Migrator -
Mercury Object Mig
Standard
06-JUN-2002

Using Object Migrator Without the Distributed Database Option

It is possible, though not recommended, to use Mercury Object Migrator without the Distributed Database option.

When Object Migrator is used in conjunction with Mercury Change Management, the Distributed Database Option is required.

When running Object Migrator with a Destination Database other than the current instance, Object Migrator will attempt to insert, delete, and update data using database links. This is not allowed by Oracle without the Distributed Database Option.

To work around this restriction, install Object Migrator on each of the instances that can potentially be a Destination Database. Then, whenever an object is to be moved to a given database, log on to the database and run Object Migrator there, setting the Destination Database to be the current database. The Source Database can be a remote database because Object Migrator queries data from only the source database (that is, no inserts, updates, or deletes) and Oracle allows queries across database links even without the Distributed Database Option.

Since the Distributed Database Option is required to execute database link inserts, updates, or deletes even on the same database, do one of the following when installing Object Migrator on an instance:

- Install Object Migrator into the APPS account, enable value **1** in CLM_DATABASES (when selecting this value, Object Migrator will not use any database link) and use **Current Database** for its description.

The privileges of the APPS account may need to be extended to allow creating tables and indexes.

- Install Object Migrator in a custom account, and instead of creating a link to the APPS account, grant all the appropriate objects from APPS to the custom account [contact Mercury Support (http://support.mercury.com) for a sample script to execute these grants]. Then, enable value **1** in CLM_DATABASES and rename it to **Current Database.**

Chapter

7

Maintaining Object Migrator

In This Chapter:

Maintaining Database Links

You can change passwords as often as daily. Keep Object Migrator database links updated whenever you change passwords by dropping the appropriate database link and recreating it using the new password. Run the Maintain Object Migrator Views program after recreating the database links.

Maintaining Value Sets

You can change databases and database links as often as daily. Keep the CLM_DATABASES and CLM_DB_LINKS value sets updated whenever you change databases and database links. For more information, see *Managing Database Instances* on page 125.

Maintaining Object Migrator Views

The Maintain Object Migrator Views program validates and rebuilds the Object Migrator views based on the current values in the CLM_DB_LINKS and CLM_DATABASES value sets, as well as the current status of your remote databases. The program also reports the current definition of these views.

These views are used during parameter selection when running Object Migrator concurrent programs. If any source or destination database enabled for Object Migrator use is referenced in the view and is unavailable (for example, when it is offline for backup), the Parameter form will return an error and users will be unable to submit requests for the Migrator/Object type.

For more details on the Object Migrator views, see *Using Object Migrator Custom Views* on page 116.

This program should schedule to run periodically, as well as after any changes are made to the CLM_DB_LINKS or CLM_DATABASES value sets, or when an Object Migrator source or destination becomes available or unavailable.

Make sure to view the report output of this program. The report will list all the enabled databases for use with Object Migrator and any errors encountered while trying to connect to these databases.

To run the Maintain Object Migrator Views program:

1. Open the Submit Requests form. (**Other:Requests:Run**)

   Use this form to launch the Maintain Object Migrator Views program. This prompts the user for report parameters, allowing the user to run the program in **Report Only** mode.

2. Enter the parameters listed in *Table 7-1*.

Table 7-1. Maintain Object Migrator Views program parameters

Parameter	Entry
Type	**Quickpick Request.**
Name	**Quickpick Maintain Object Migrator Views.**
Report Only	Select one of the following options: • **Yes:** Do not modify the Object Migrator views. Just report on their current definition. • **No:** Re-create the Object Migrator views based on the current enabled values in the CLM_DATABASES and CLM_DB_LINKS value sets and the current statuses of your remote databases.

Some objects, such as character mode menus, are not applicable to all releases of Oracle Applications. If a particular object is not applicable to a given Oracle instance, the instance will be omitted from the view definition.

Using Object Migrator Custom Views

Object Migrator uses views to provide a list of values when selecting a specific object to migrate. This list of values is taken from the source database of the migration. These lists of values are based on custom views created and maintained by the Maintain Object Migrator Views program.

These custom views are discussed in more detail in the following sections:

- *About the Custom Views*
- *Drawbacks to the Custom Views*
- *Switching to Non-Validated Mode*
- *open_links Database Parameter*

About the Custom Views

The custom views (one per object type) are a series of unions that select from each database defined in the CLM_DATABASES value set. To get to the remote databases, the views utilize the database links defined in the CLM_DB_LINKS value set. Object Migrator retrieves data from the remote database in real time.

For example, suppose CLM_DATABASES has the following enabled values:

```
Value     Meaning
2         DEV
3         QA
4         PROD
900       Object Archive
```

and CLM_DB_LINKS has the enabled values:

```
Value     Meaning
2         DEV_LINK
3         A_LINK
4         PROD_LINK
900       Object Archive
```

The view for the Users Migrator would be:

```
SELECT   2     db_value,
         user_name object_name
FROM     fnd_user@DEV_LINK
UNION
SELECT   3     db_value,
         user_name object_name
FROM     fnd_user@QA_LINK
UNION
SELECT   4     db_value,
user_name object_name
FROM     fnd_user@PROD_LINK
UNION
<select clause to query the Object Archive tables against
the local database>
```

When retrieving a list of values from this view, the Object Migrator program will check at the selected source database and return the values from that database. Where appropriate, the list of values will be limited by the source application.

Drawbacks to the Custom Views

Although there are advantages to users to be able to select the object name as it exists on the source database, there are also drawbacks to using these views:

- While the list of values based on these views queries only the selected database, it validates each database link when it first parses the query. If it takes a long time to access one or more remote databases using a database link, then there might be a slight delay when the list of values is first executed.

  This should happen only when the list of values appears in the application session for the first time. Subsequent uses of the list of values should get better performance.

- Due to the way the database handles unions, if one database link in the union is invalid (the link has the wrong user name or password, the remote database is down, or there is a problem with SQL*Net), the entire view is invalid. This results in an error when the Object Migrator program is submitted.

  To alleviate this problem, run the Maintain Object Migrator Views program periodically.

When Report Only is set to **No,** this program looks at all the databases as defined in CLM_DATABASES and CLM_DB_LINKS, and checks each view against each database. It then rebuilds each view omitting the problem databases. The

report output of this program reports on any problems that occurred. If Report Only is set to **Yes**, the program merely reports on the current definition of the views.

To keep the Object Migrator views current with the current database status, schedule the Maintain Object Migrator Views program to run periodically. The greater the amount of database activity (in terms of changes in passwords, database status, and SQL*Net status), the more often this program should be run. If your level of activity is high, run the program daily. Always run the program after making any modifications to CLM_DATABASES or CLM_DB_LINKS.

Switching to Non-Validated Mode

If significant problems are encountered with the Object Migrator views, switch to the non-validated object name mode. To do this, execute the `CLMSMBS9.sql` script (located with the rest of the Mercury Object Migrator installation scripts) against the Object Migrator SQL account. When you are prompted for inst_value_sets, enter **Y.** After you run this script, the object name parameters will have no validation or list of values.

A sample execution of `CLMSMBS9.sql` is shown below:

```
sqlplus <OM acct>/<OM password>
SQL> @CLMSMBS9.sql
Enter value for inst_value_sets: Y
Enter value for inst_value_sets: Y
SQL> exit;
```

To revert back to validated parameters with list of values, run the `CLMSMBS8.sql` script against the Object Migrator SQL account.When you are prompted for inst_value_sets, enter **Y.** A sample execution of `CLMSMBS8.sql` is shown below:

```
sqlplus <OM acct>/<OM password>
SQL> @CLMSMBS8.sql
Enter value for inst_value_sets: Y
Enter value for inst_value_sets: Y
SQL> exit;
```

open_links Database Parameter

The open_links database parameter controls the number of different databases that can be accessed through database links in a given session, though it does not control how many times a specific link can be used in that session. The default value for this parameter is **4.** If there are (or will be) more than four databases defined in CLM_DATABASES (including the source and

destination databases), the value for this parameter needs to be increased in the Object Migrator instance. The suggested value is 20.

To change the open_links parameter:

1. Shut down the database.
2. Modify the `init.ora` file by adding (or modifying) the line `open_links =20`.
3. Restart the database.

Managing Custom Applications

The parameter selections for application parameters of Object Migrator concurrent programs (for example, source application or destination application) are taken from the instance where Object Migrator is installed, not directly from the source or destination instance.

In the Object Migrator instance, use the Applications form (**Application:Register**) to register each custom application for which data will be migrated. Otherwise, Object Migrator will be unable to migrate data for the application.

Object Migrator identifies applications by their application short name, not by the user application name.

Figure 7-1 shows an example.

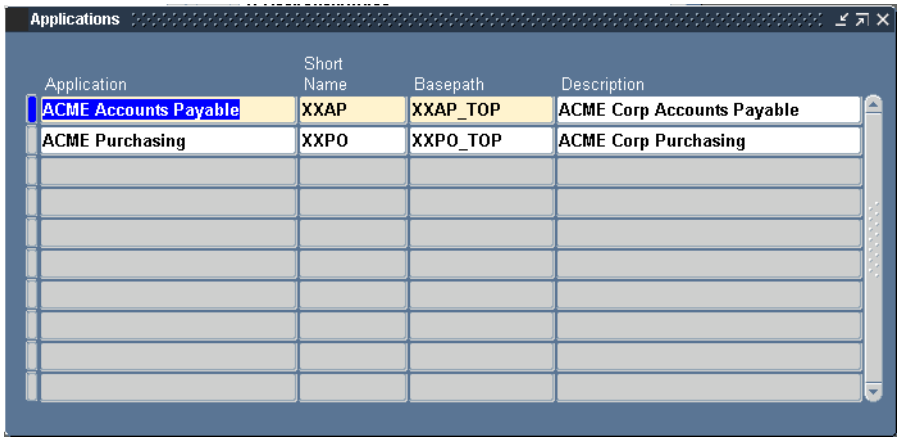


Figure 7-1. Registering custom applications

Purging the Object Migrator Interface Tables

To maintain data processing efficiency, periodically purge the temporary Object Migrator tables using the Purge Object Migrator Interface Tables program. You can:

- Truncate the Object Migrator tables
- Delete records older than a certain date
- Purge particular object types

Note

This clean up is only necessary for executions of the Object Migrator in Debug mode. If Debug is set to **No,** Object Migrator removes all records inserted into its temporary tables during the execution of the program.

To remove any fragmentation created through the insertion and deletion of records in the temporary tables, run this program at least once a month with Truncate Table set to **Yes** and all migrators selected. To protect data integrity during migrations, the Object Migrator installation program automatically defines incompatibilities such that migrations cannot run while interface tables are being purged.

To run the Purge Object Migrator Interface Tables program:

1. Open the Submit Requests window. (**Other:Requests:Run**)

   Use this form to launch the Purge Object Migrator Interface Tables program. This will prompt the user for report parameters to restrict the tables to purge.

2. Enter the parameters shown in *Table 7-2*.

Table 7-2. Purge Object Migrator Interface Tables program parameters

Parameter	Entry
Type	**Quickpick Request.**
Name	**Quickpick Purge Object Migrator Interface Tables.**
Truncate Tables	Select one of the following options: • **Yes:** Purge the tables by truncating them. This will remove all the records from the table and remove and table fragmentation. Note that records cannot be selectively deleted based on age if this parameter is set to **Yes.** Unless there are reasons to retain records, set this parameter to **Yes.** • **No:** Purge the tables by deleting rows from the table.
Days Old	Enter the least number of days old the records need to be in order to be eligible for deletion. This parameter is only relevant if Truncate Tables is set to **No.**
Conc Prog Migrator Tables	Select whether or not the tables used by the Concurrent Programs Migrator will be purged.
Rep Set Migrator Tables	Select whether or not the tables used by the Report Sets Migrator will be purged.
Value Set Migrator Tables	Select whether or not the tables used by the Value Sets Migrator will be purged.
Desc Flex Migrator Tables	Select whether or not the tables used by the Descriptive Flexfields Migrator will be purged.
Request Group Migrator Tables	Select whether or not the tables used by the Request Groups Migrator will be purged.
Zoom Migrator Tables	Select whether or not the tables used by the Zooms Migrator will be purged.
Menu Migrator Tables	Select whether or not the tables used by the Menus Migrator will be purged.

Table 7-2. Purge Object Migrator Interface Tables program parameters

Parameter	Entry
Responsibility Migrator Tables	Select whether or not the tables used by the Responsibilities Migrator will be purged.
User Migrator Tables	Select whether or not the tables used by the Users Migrator will be purged.
Profile Options Migrator Tables	Select whether or not the tables used by the Profile Options Migrator will be purged.
Help Text Migrator Tables	Select whether or not the tables used by the Help Text Migrator will be purged.
FSG Row/Column Set Migrator Tables	Select whether or not the tables used by the FSG Row/ Column Sets Migrator will be purged.
QuickCode Migrator Tables	Select whether or not the tables used by the QuickCodes Migrator will be purged.
Named SQL Migrator Tables	Select whether or not the tables used by the Named SQL Migrator will be purged.
GUI Menu Migrator Tables	Select whether or not the tables used by the GUI Menus Migrator will be purged.
Function Migrator Tables	Select whether or not the tables used by the Functions Migrator will be purged.
Printer Migrator Tables	Select whether or not the tables used by the Printer Definitions Migrator will be purged.
Folder Migrator Tables	Select whether or not the tables used by the Folders Migrator will be purged.
Message Migrator Tables	Select whether or not the tables used by the Messages Migrator will be purged.
Concurrent Manager Migrator Tables	Select whether or not the tables used by the Concurrent Manager Definitions Migrator will be purged.

Purging the Object Migrator Archive Details

Data accumulates in the Object Migrator archive tables as different versions of various object types are archived. Over time, the volume of this data can become considerable.

To save storage space and increase report efficiency, maintain the revision history of the objects, but purge the actual details of the older versions of objects that will never be used for migration or comparison. Archive data can also be purged to remove object versions that were incorrectly archived.

Use the Purge Object Archive Version program to purge entire archives (by Archive label) or subsets of an archive, selecting by object type or object name. The Object Migrator installation program automatically defines incompatibilities such that migrations cannot run while interface tables are being purged. This is to protect data integrity during migrations. For additional details on running this program, see *Purging the Object Archive* on page 152.

The ability to compare against or revert to these purged versions will be lost after the purge program is executed.

Recompiling the Descriptive Flexfields

Oracle recommends that the descriptive flexfields be recompiled periodically. To do this, at the file system level, run the following command:

```
$FND_TOP/bin/fdfcmp apps/<apps passwd> 0 Y
```

The syntax may vary in different Oracle E-Business Suite releases.

Maintaining the Data Model

If the Object Migrator database uses cost-based SQL optimization, you should periodically gather statistics on the Object Migrator schema. The required frequency depends on the volume of data being migrated, especially archivals, and the number of submissions requested. In addition, you should adjust table sizing to reflect your style of usage of Object Migrator.

Setting Up Object Migrator Security

Mercury Object Migrator uses standard Oracle E-Business Suite security functionality to control access to Object Migrator programs. *Configuring Your System to Work with Object Migrator* on page 52 provides some default security setup information. Beyond this default security setup, you may want to consider additional internal security considerations for Object Migrator. Your security tasks might include the following:

- **Request groups.** Set up a request group for each grouping of programs if there are different groups of users who should access different sets of programs.

System administrators might have access to all programs, however business analysts and developers have access only to migrator and archive programs.

- **Responsibilities.** For each grouping of programs, set up a custom responsibility for the user group that will access the group of programs. See Chapter 6, *Optional Configurations for Object Migrator*, on page 99 for detailed examples.
- **Users.** The Oracle E-Business Suite instance where Object Migrator is installed needs to have a logon defined for each user who will submit Object Migrator requests (this is optional if integrating with Mercury Change Management). Create the user and assign the appropriate responsibility that includes Object Migrator programs.

Managing Database Instances

Adding a New Database Instance

New database instances can be registered for use by Mercury Object Migrator, either as a source or destination for object data.

To register a new database instance:

1. On the Object Migrator instance, define a database link from the Object Migrator SQL account to the `APPS` account of the new database instance.

   For examples, see *Configuring Your System to Work with Object Migrator* on page 52. Note that the database configurations (for example, `tnsnames.ora`) must support connection using the database link.

2. Add a value to the CLM_DATABASES value set for the new database instance.

   For examples, see *Configuring Your System to Work with Object Migrator* on page 52.

3. Add a value to the CLM_DB_LINKS value set for the new database link. Use the exact database link name for the description.

Use the same number in the Value field that the database is assigned in the CLM_DATABASES value set. For examples, see *Configuring Your System to Work with Object Migrator* on page 52.

4. Run the Maintain Object Migrator Views Concurrent program. This will re-create all the object views used by Object Migrator to include this new database. It will also perform a validation check for all enabled database names and links.

Removing a Database Instance

Mercury Object Migrator configurations should be updated when an Oracle E-Business Suite instance will no longer be used as a source or destination of migrations (for example, when a test instance is removed).

To remove a database instance:

1. Query the value for the instance in the CLM_DB_LINKS value set and disable it.
2. Query the value for the instance in the CLM_DATABASES value set and disable it.
3. Remove the database link from the Object Migrator database schema.
4. Run the Maintain Object Migrator Views Concurrent program. This re-creates all the object views used by Object Migrator and no longer includes the obsolete Oracle E-Business Suite instance.

Chapter

8

Migrating Objects

In This Chapter:

Running the Object Migrator Request

Mercury Object Migrator migrations are run through the Submit Request window in Oracle Applications. The general procedures for migrating an object between instances is the same for all object types. These procedures are discussed in this section:

- *Reviewing the Prerequisites for the Object*
- *Running the Report to Migrate the Object*

Reviewing the Prerequisites for the Object

Each Oracle AOL object type requires specific information and has unique rules for migrating between instances. See Appendix B, *Object Type-Specific Migrators and Migration Rules*, on page 165 for the prerequisites and rules related to specific object types.

When migrating multiple object types, consider the order in which the objects will be migrated. For a list of object dependencies, see Appendix A, *AOL-Dependent Objects*, on page 161.

In migrating report sets between database instances, the following rules apply:

- The concurrent programs referenced by the report set must exist in the destination database.
- The concurrent programs parameters for which the report set has default values must exist in the destination database.
- The request set stage function referenced by the report set must exist in the destination database.

Running the Report to Migrate the Object

After you verify that all prerequisites are satisfied to migrate a specific object type, use Mercury Object Migrator to transfer objects. Object Migrator uses the standard Submit Requests form in Oracle Applications.

By default, Object Migrator defines incompatibilities between migration programs and archive/interface purge programs. This protects the integrity of the migrations.

The following procedure reflects Oracle release 11i. Navigation paths may differ for releases 10.7 or 11.

To run the migration program:

1. Log on to Oracle Applications.

   You need access to a user ID and password for a user account that has the responsibility (access) to use Oracle's Submit Requests window and the Object Migrator programs.

2. Open the Submit Requests window.

   a. From the menu, select **View > Request.**

      The Find Requests window opens.

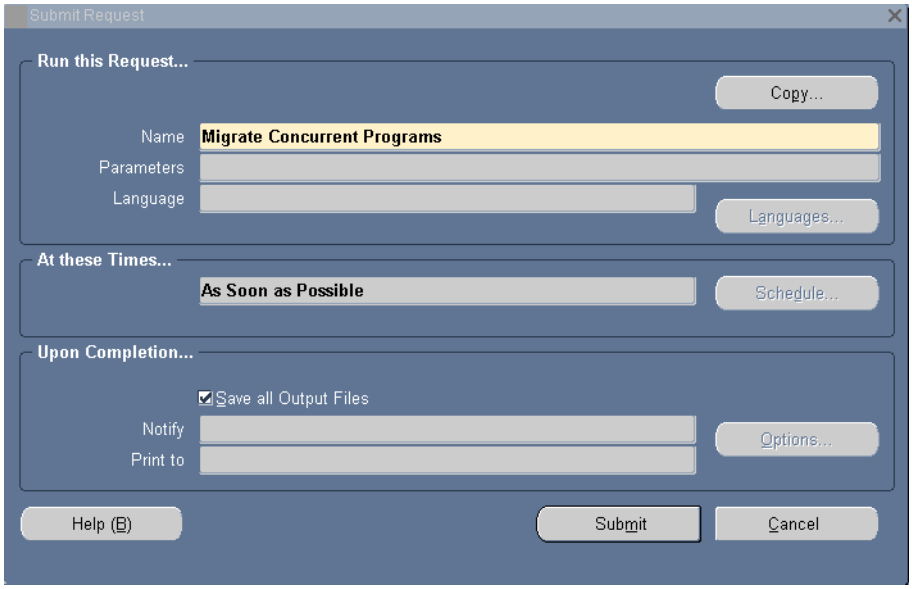
   b. Click **Submit a New Request.**

      The Submit a New Request window opens.

   c. Select the **Single Request** radio button.

   d. Click **OK.**

      The Submit Request window opens.



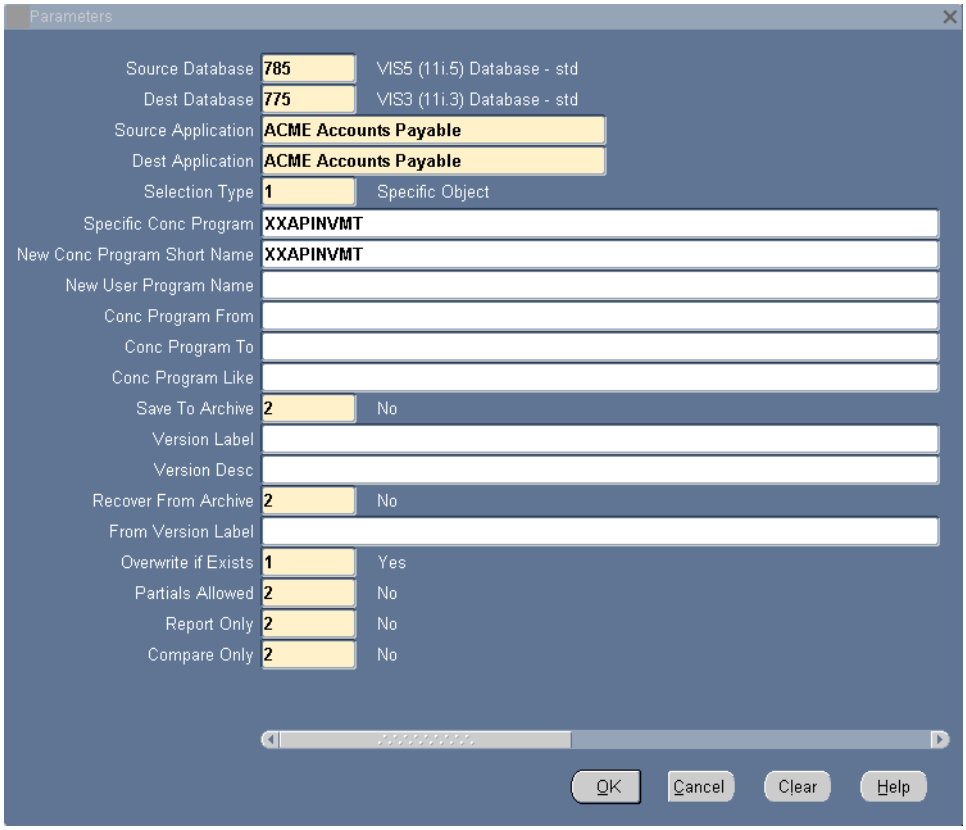
3. In the Name field, select the object to migrate.

   Object Migrator requests follow the naming convention Migrate <*ObjectTypeName*>. For example, the object for moving concurrent programs is migrate concurrent programs.

Note

Only Object Migrator programs to which you have been given access by the application system administrator will be displayed.

   The Parameters window opens for the selected object. For example, if you select **Migrate Concurrent Programs,** the window will display the fields required to migrate concurrent programs.



4. Enter the parameters for the migration.

   Enter information to define the migration. Specify the Source Database and Dest Database for each migration. Also, specify additional information about which objects to migrate, the type of migration, and archiving information. For example, you can limit which concurrent programs are migrated by specifying them in the Conc Program From and Conc Program To fields.

   For a detailed description of each common and object-specific parameter, see Appendix A, *AOL-Dependent Objects*, on page 161.

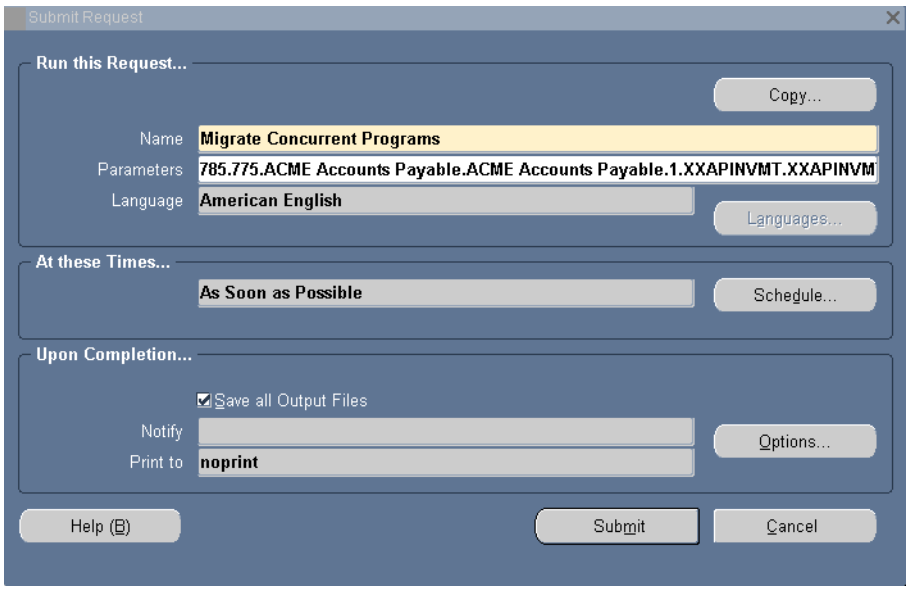
Note

It is possible to migrate one or multiple objects at a time. You do this by changing the Selection Type field on the Parameter window. The available options are:

- **Specific Object.** One object is selected for migration at this time. The specific object name needs to be entered.
- **Range of Objects.** A range of objects is selected for migration at this time. The alphabetic range to use needs to be specified.
- **Wild Card Match.** Multiple objects are migrated with object names matching a specified string. The string to match, including the percent (%) sign, needs to be specified.

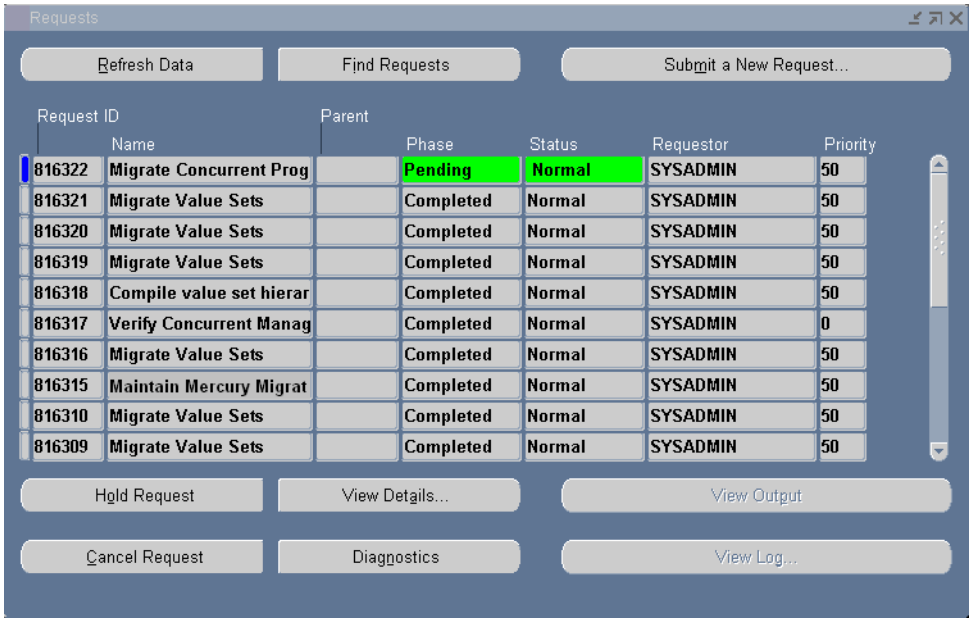
5. Click **OK.**

   The Parameter window closes. The Submit Requests window is updated with the selected parameters.



6. Click **Submit.**

   The migration will proceed based on the selected scheduling options. Either submit another request or view the results of the migration in the Requests window.


Requests
Refresh Data
Find Requests
Submit a New Request...
Request ID
Parent
Name
Phase
Status
Requestor
Priority
816322 Migrate Concurrent Prog Pending Normal SYSADMIN 50
816321 Migrate Value Sets Completed Normal SYSADMIN 50
816320 Migrate Value Sets Completed Normal SYSADMIN 50
816319 Migrate Value Sets Completed Normal SYSADMIN 50
816318 Compile value set hierar Completed Normal SYSADMIN 50
816317 Verify Concurrent Manag Completed Normal SYSADMIN 0
816316 Migrate Value Sets Completed Normal SYSADMIN 50
816315 Maintain Mercury Migrat Completed Normal SYSADMIN 50
816310 Migrate Value Sets Completed Normal SYSADMIN 50
816309 Migrate Value Sets Completed Normal SYSADMIN 50
Hold Request
View Details...
View Output
Cancel Request
Diagnostics
View Log...

Viewing the Migration Results

You can view the results of the migration using the standard Oracle Applications request result window. After a migration is complete, the Request window will be displayed containing the details and logs related to the migration.

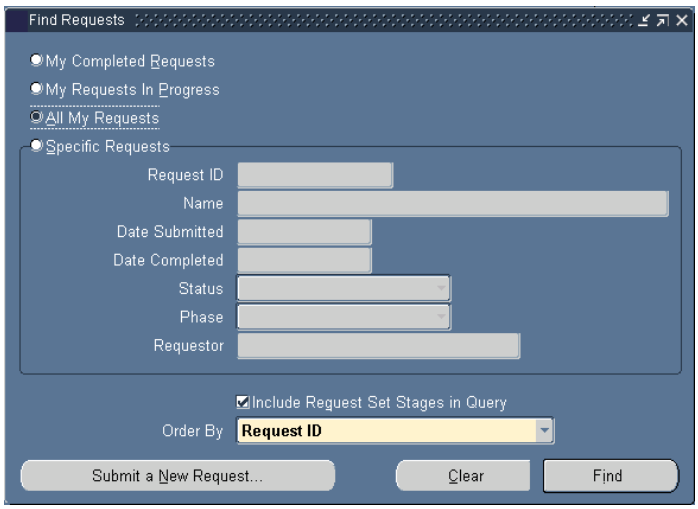
Note

When migrating multiple objects (range or wild card), the migration will fail if no objects can be migrated successfully, but will succeed if even one object is successfully migrated. For this reason, always review the request output carefully.

To open the Request window to view migration details and logs:

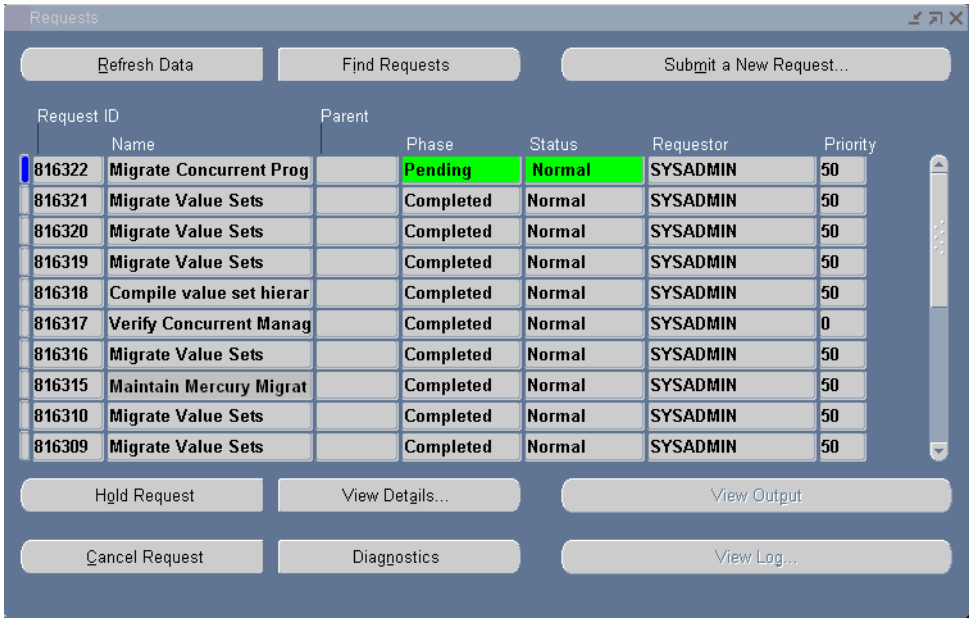
1. From the **Oracle Applications** menu, access the Request window.
2. From the menu, select **View > Requests.**

   The Find Requests window opens.



3. Select the criteria to locate the request.
4. Click **Find.**

   The Requests window opens.



View information for a specific request by clicking in a row and clicking one of the following buttons:

- **View Details.** Opens a window that displays the details related to the request submission. This includes the name of the program, parameters used, and scheduling options.
- **Diagnostics.** Opens a window that displays a summary of the migration results. This includes information on whether the migration succeeded with or without errors. Migrations with errors will include text indicating the cause or instructions on obtaining more detailed information.
- **View Output.** Displays the migration execution report detailing migrated objects and error messages, if any. Errors preventing migration are listed here.
- **View Log.** Displays the execution log. Errors preventing execution are listed here.

The migration output and logs may contain some error messages indicating specific problems. See Appendix C, *Exception Messages*, on page 235 for a complete list of error messages and related troubleshooting tips.

Auditing Migration Executions

Use the Mercury Object Migrator Audit Report program to audit executions of Object Migrator concurrent programs, as well as audit objects that have been migrated. Every time an Object Migrator concurrent program is run, Object Migrator logs information about the request and the specific objects migrated. This information remains available even after concurrent requests have been purged from Oracle E-Business Suite thereby allowing Object Migrator activity to be successfully audited over time.

To run Mercury Object Migrator Audit Report:

1. Open the Submit Requests window. (**Other:Requests:Run**)

   Use this form to launch the Mercury Object Migrator Audit Report program. This prompts the user for report parameters allowing the user to restrict the audit data.

2. Enter the parameters shown in *Table 8-1*.

Table 8-1. Mercury Object Migrator Audit Report program parameters

Parameter	Entry
Type	**Quickpick Request.**
Name	**Quickpick Mercury Object Migrator Audit Report.**
Order By	Select one of the following options: **Run Date** — Order the Audit Report by the Run Date of the migrator program. **Object Type** — Order the Audit Report by the object type that was migrated. Within a specific object type, order the report by the program Run Date.
Run Date From	Enter the starting Run Date of the programs to include in the audit report.
To	Enter the ending Run Date of the programs to include in the audit report.
Specific Object Type	Enter a specific object type to limit the audit report to a specific Object Migrator object type.
Specific Application	Enter a specific application to limit the audit report to a specific application (this refers to the destination application in a given migration).

Table 8-1. Mercury Object Migrator Audit Report program parameters

Parameter	Entry
Specific Object	Enter a specific object name to restrict the audit report to a specific object (this refers to the destination object name).
Destination Database	Enter a specific database to limit the audit report to migrations to a specific database instance.
Ignore Report Only	Select whether or not to include executions of Object Migrator that were run in Report Only mode.

Chapter

9

Comparing Objects

In This Chapter:

Object Comparison Overview

Object comparisons are performed using the same program as migrations. Each Object Migrator request includes Compare Only that can be used to compare objects. If Compare Only is specified, no migration is executed.

When Compare Only is set to **Yes**, Object Migrator extracts the object information from the Source and Destination Databases into temporary tables. It then compares the detailed attributes of the object(s) and lists the differences in the output report.

If there are differences between the objects in the source and destination databases, the report will categorize the differences which are discussed in the following sections:

- *Only in Source Database*
- *Only in Dest Database*
- *Differences Exist*

Only in Source Database

The object or object detail exists only in the source database. There is nothing in the destination database with the corresponding object name or object detail name. For example, if a new parameter was added to a concurrent program in the source database, the comparison report would indicate that the parameter exists only in the source database.

Only in Dest Database

The object or object detail exists only in the destination database. There is nothing in the source database with the corresponding object name or object detail name. An example of this would be if a new parameter was added to a concurrent program on the destination database. When comparing the concurrent program between the databases, the comparison report would list this parameter, indicating that it existed only in the destination database.

Differences Exist

Objects or object details with this status either have differences with their specific attributes (for example, the titles for the object could be different) or have details where some sort of differences exist. For example, a concurrent program that has the same parameters across two databases but has different parameters attributes (such as required or not) would have a status of Differences Exist. The output report would then show the specific differences between the source and destination for the parameter.

If an object or object detail has a status of Only in Source DB or Only in Dest DB, then Object Migrator does not report on any details below that object. For example, suppose Object Migrator is comparing a range of concurrent programs on one database against the range on another database. If Object Migrator finds that a concurrent program exists on the source database and not the destination database, it does not report on the parameters belonging to concurrent programs.

Long fields and translated data are not compared in the comparison reports.

Some child entities are not compared. Generally, these are noted in the Appendix B, *Object Type-Specific Migrators and Migration Rules*, on page 165.

Running the Object Comparison

Object comparisons are performed using the same concurrent programs as migrations. This section provides instructions for performing object comparisons in a few different situations.

This section covers the following topics:

- *Comparing Objects from Two Oracle Instances*
- *Comparing Objects in a Database Instance and the Object Archive*
- *Comparing Two Versions of an Object in the Object Archive*

Comparing Objects from Two Oracle Instances

The following procedure describes how to compare an object as it appears in two instances.

To compare an object in separate instances:

1. In the Submit Request window, create a new request for an object migration.

   The Parameter window opens for the selected object type. For example, if **Migrate Descriptive Flexfields** is selected, the window will display the fields required to compare descriptive flexfields.

2. In the Source Database instance, select the first instance containing the object to compare.

3. In the Dest Database field, select the second instance containing the object to compare.

4. Select **1 (Yes)** for Compare Only.

   With Compare Only set to **Yes**, one Oracle Applications instance will be specified as the Source Database and the other instance as the Destination Database.

5. Enter any other parameters required to define the object or objects to compare. It is possible to compare by individual objects, range of objects, or by wild card.

   To compare by individual objects, enter the object to compare (Object Migrator will ignore the value in New Object Name and will instead use the

value of Specific Object Name when retrieving object information for both databases).

To compare by a range of objects, enter the range using Object From and Object To.

If you specify a range, objects that do not exist on one database will be shown versus another database, as well as the differences for the objects that do exist on both databases.

When comparing data between releases, you may encounter differences due to changes in the AOL data model between these releases.

For a description of each parameter, see Appendix B, *Object Type-Specific Migrators and Migration Rules*, on page 165.

6. Click **OK.**

   Once the objects are compared, the results are displayed in the output report.

Comparing Objects in a Database Instance and the Object Archive

The following procedure describes how to compare an object in a database instance to an archived object.

To compare an object in a database instance with an archived object:

1. In the Submit Request window, create a new request for an object migration.

   The Parameter window opens for the selected object type. For example, if **Migrate Descriptive Flexfields** is selected, the window will display the fields required to compare descriptive flexfields.

2. In the Source Database instance, select **Object Archive.**

3. In the Dest Database field, select the name of the Oracle Applications instance containing the object to compare.

4. In the Recover From Archive field, select **Yes.**

5. In the From Version Label field, select the appropriate label.

6. Select **1 (Yes)** for Compare Only.

7. Enter any other parameters required to select the object or objects to be compared.

   To compare by individual objects, enter the object to compare (Object Migrator will ignore the value in New Object Name and will instead use the value of Specific Object Name when retrieving object information for both databases).

   To compare by a range of objects, enter the range using Object From and Object To.

   For a description of each parameter, see Appendix B, *Object Type-Specific Migrators and Migration Rules*, on page 165.

8. Click **OK.**

   Once the objects in the archive are compared to the objects in the Oracle Applications instance, the results are displayed in the output report.

Comparing Two Versions of an Object in the Object Archive

The following procedure describes how to compare two archived object versions.

To compare two archived objects:

1. In the Submit Request window, create a new request for an object migration.

   The Parameter window opens for the selected object type. For example, if you select **Migrate Descriptive Flexfields,** the window displays the fields required to compare descriptive flexfields.

2. In the Source Database instance, select **Object Archive.**

3. In the Dest Database field, select **Object Archive.**

4. In the Save to Archive field, select **No.**

5. In the Version Label field, enter the version of the first object to be compared.

6. In the Recover From Archive field, select **Yes.**
7. In the From Version Label field, select the second version label of the object to be compared.
8. Select **1 (Yes)** for Compare Only.
9. Enter any other parameters required to define the object or objects to be compared.

   To compare by individual objects, enter the object to compare (Object Migrator will ignore the value in New Object Name and will instead use the value of Specific Object Name when retrieving object information for both databases).

   To compare by a range of objects, enter the range using Object From and Object To.

   For a description of each parameter, see Appendix B, *Object Type-Specific Migrators and Migration Rules*, on page 165.
10. Click **OK.**

    Once the archived objects are compared, the results are displayed in the output report.

Sample Comparison Report

This section provides a sample report output for an object comparison. The following parameters were used to construct the example:

- The concurrent program (`WIPRELIN`) was compared between two applications instances.
- On the destination instance, the following items were manually altered:
  - The Maximum Columns field value was removed and the description was changed.
  - A new parameter named New Param was entered.
  - The parameter Org ID was removed.
  - The value set for the parameter Dates To was changed from WIP_SRS_DATES_OP to FND_DATE.

The following figure shows the results of comparing these instances.

```
Concurrent Program Differences Report
 ** Concurrent Programs **

Action                  App      Program Name         Description
-----------------       ----     ---------------      --------------
Differences Exist       WIP      WIPRELIN             Repetitive Line Report

Column Name             Value From Source DB          Value From Dest DB
------------            --------------------          ------------
Columns Max             65
Program Long Name       Repetitive Line Report        REPetitive Line Report

** Parameters **

Action                  Seq Num  Parameter
-----------------       -------- ---------------
Only in Source DB       7        Org_id
Only in Dest DB         9        New Param
Differences Exist       6        Dates to

Column Name             Value From Source DB          Value From Dest DB
------------            --------------------          ------------------
Value Set               WIP_SRS_DATES_OPT             FND_DATE
```

Figure 9-1. Sample Migration Comparison Report

Chapter

10

Using Version Control

In This Chapter:

Archiving AOL Object Definitions

Object Migrator stores object information in the object archive. This preserves a version of the object definitions for future uses, including reapplying the definition to a database instance, comparing to other versions, or historical reporting.

This section provides instructions for:

- *Saving an Object to the Object Archive*
- *Retrieving an Object from the Object Archive*

For information on comparing archived objects to objects in other instances, see Chapter 9, *Comparing Objects*, on page 139.

Saving an Object to the Object Archive

You can save Object Migrator supported objects to the object archive using the following procedure.

To save an Object to the object archive:

1. Using the Submit Request window, create a new request for an object migration.

   The Parameter window opens for the selected object type. For example, if you select **Migrate Descriptive Flexfields,** the window will display the fields required to migrate descriptive flexfields.

2. Select the following options related to the object archive:

   - Dest Database = **Object Archive**
   - Save to Archive = **Yes**

3. In the Version Label field, specify the version label for the object.

   If you specify a new version label, Object Migrator will create the label for you. If you select an existing version label, the object will be archived with other objects with the same label.

Only one revision of a given object (such as a particular concurrent program) can exist within a given archive label.

4. In the Version Desc field, enter an optional description of the object version.

   This value will also be the description for the version label if this is a new version label. If this field is left blank and the version label already exists, Object Migrator will use the description of the version label as the description of the object versions.

5. Enter any other parameters required to define the migration. For a description of the parameters, see Appendix B, *Object Type-Specific Migrators and Migration Rules*, on page 165.

6. Click **OK.**

The object is saved to the object archive. To find out if the objects were successfully saved, view the migration results.

Retrieving an Object from the Object Archive

At times it may be desirable to restore an archived object definition to an instance. You can do this by retrieving objects that were previously archived. using the following procedure.

To retrieve an Object from the object archive:

1. Using the Submit Request window, create a new request for an object migration.

   The Parameter window opens for the selected object type. For example, if you select **Migrate Descriptive Flexfields,** the window will display the fields required to migrate descriptive flexfields.

2. Select the following options related to the object archive:

   - Source Database = **Object Archive**
   - Save to Archive = **No**
   - Recover from Archive = **Yes**
   - Dest DB = <*database instance to which the object will be restored*>

3. In the From Version Label field, select the version of the object to retrieve.

4. Enter any other parameters required to define the migration. For a description of the parameters, see Appendix B, *Object Type-Specific Migrators and Migration Rules*, on page 165.

5. Click **OK.**

The previous object definition in the destination is replaced with the definition from the archive. To find out whether the objects were successfully migrated, view the migration results.

Running the Object Archive Version Detail Report

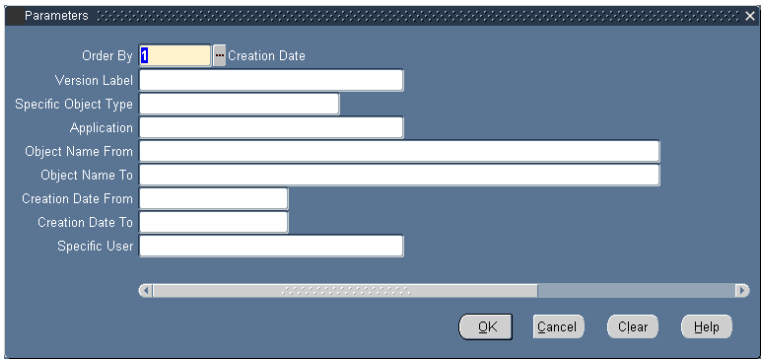
The Object Archive Version Detail Report audits the contents of the object archive and displays the revision history of your AOL objects. You can run this report from the Oracle Applications Submit Requests window.

To run the Object Archive Version Detail Report:

1. Open the Submit Request window.

2. In the Name field, select **Object Archive Version Detail Report.**

   The Parameter window opens for the request.



3. To limit the report results, enter the relevant report parameters (see *Table 10-1*).

4. Click **OK.**

   The report runs according to the specified parameters.

5. To view the report after the request completes, click **View Output.**

Table 10-1 lists and defines the Object Migrator common parameters used with Object Archive Version Detail Report.

Table 10-1. Object Migrator common parameters

Parameter Name	Description
Order By (Required)	Select one of the following: • Date: Order the report by the object creation date. • Version Label: Order the report by version label name. • Object Name: Order the report by object type, object name
Version Label	Limit the report by a specific version label in the object archive.
Object Type	Limit the report by a specific object type (concurrent program, value set, and so forth).
Application	Limit the report by a specific application.
Object Name From	Enter the starting object name to include in the report.
Object Name To	Enter the ending object name to include in the report. Set this equal to the Object Name From to report on a single object.
Creation Date From	Limit the report with object versions saved into the object archive after or equal to this date.
Creation Date To	Limit the report with object versions saved into the object archive before or equal to this date.
Specific User	Limit the report by the user who created the version.

Purging the Object Archive

The Purge Object Archive Versions program purges information from the AOL object archive. Use the purge functionality to remove old versions of objects that will never be used or to removed object versions that were incorrectly saved to the object archive. Run this report from the Oracle Applications Submit Requests window.

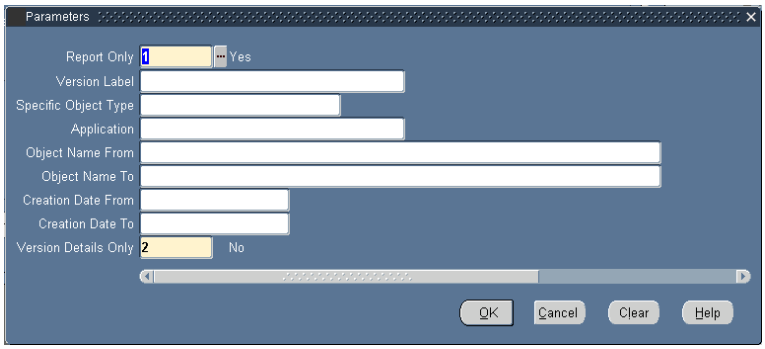
Note

To run this program, you must have access to the program given by the system administrator.

To purge the object archive:

1. Open the Submit Request window.
2. In the Name field, select **Purge Object Archive Versions.**

   The Parameter window opens for the request.



3. To limit the report results, enter the relevant report parameters (see *Table 10-2*).
4. Click **OK.**

   The report runs according to the specified parameters.
5. To view the report after the request completes, click **View Output.**

Table 10-2 lists and defines the Purge Object Archive Versions parameters.

Table 10-2. Purge Object Archive Versions parameters

Parameter Name	Description
Report Only (Required)	Select one of the following options: • **Yes:** Report on the versions that would get purged but do not execute the actual purge process. • **No:** Execute the purge process in addition to reporting on the versions that will get purged.
Version Label	Limit the data to purge by a specific version label in the object archive.
Object Type	Limit the data to purge by a specific object type (concurrent program, value set, and so forth).
Application	Limit the data to purge to a specific application.
Object Name From	Enter the starting object name to include in the purge. This parameter is case sensitive.
Object Name To	Enter the ending object name to include in the purge program. Set this equal to the 'Object Name From' to report on a single object.
Creation Date From	Limit the data to purge to object versions saved into the object archive on or after this date.
Creation Date To	Limit the data to purge to object versions saved into the object archive before or on this date.
Version Details Only (Required)	Select one of the following options: • **Yes:** Purge the details of the object in the object archive but do not purge the version itself. This allows for the reporting on the revision history of an object without having to store all the information for older version you will never recover from or compare against. Once the details have been purged, you will not be able to ever use this object version as a source for migrations or as a version to run comparisons against. • **No:** Remove the object version as well as the details for the object. Use this setting only if you want to remove all record of the object version(s) from the object archive.

Chapter

11

Using Object Migrator with Mercury Change Management

In This Chapter:

Introduction to Mercury Change Management

Object Migrator can be run from Mercury Change Management, another Mercury IT Governance Center product. This section provides an overview of Mercury Change Management and highlights the integration points between Object Migrator and Mercury Change Management.

For more information, see the *Mercury Change Management User's Guide*.

For information about accessing this manual, see the *Guide to Documentation*. You must have already purchased Mercury Change Management to access this document.

Mercury Change Management is designed to automate the deployment of technology solutions. Business system components created or modified by an initiative are grouped into Mercury Change Management packages. These packages have workflows associated with them that automate the process of moving each package through required steps, which typically involve system build, testing, quality assurance, staging, and final deployment. Mercury Change Management automatically deploys the application components necessary for each package, such as XML content, HTML files, Java programs, Oracle Applications configurations, PeopleSoft panels, or Siebel projects.

For example, this software will connect to the development environment, copy SQL scripts from the version control repository and transfer the files to the quality assurance environment. It will then use SQL*Plus to load the stored procedures defined in the SQL script. A built-in scheduler allows the customer to schedule deployments. Mercury Change Management maintains an audit trail for all activities including package information, application components altered, approvals obtained and deployments performed.

Executing Object Migrator from Mercury Change Management

Object Migrator migrations can be run directly through Mercury Change Management once the Mercury Extension for Oracle E-Business Suite has been installed and configured. The Extension includes a pre-defined object type for each type of object migrated using Object Migrator.

As part of the Extension configuration, the Mercury IT Governance Center administrator will have set up the appropriate workflows to control data migrations and enforce promotion rules such as testing approvals. The Extension includes best practice sample workflows for this purpose.

In Mercury Change Management, users create package lines that specify the migration of AOL objects. Package lines that include AOL objects call Object Migrator to execute the migration. Packages in Change Management can contain a mix of AOL and non-AOL package lines, which enables related code to migrate together with its configurations.

To create a package to migrate AOL objects in Mercury Change Management:

1. Open the Mercury IT Governance Workbench. (See the *Getting Started* guide for details.)

2. In the shortcut bar, select **Change Mgmt > Packages.**

3. In the Package Workbench, click **New Package** to create a new package.

   The Package window opens.

4. Complete the fields in the Package Information section.

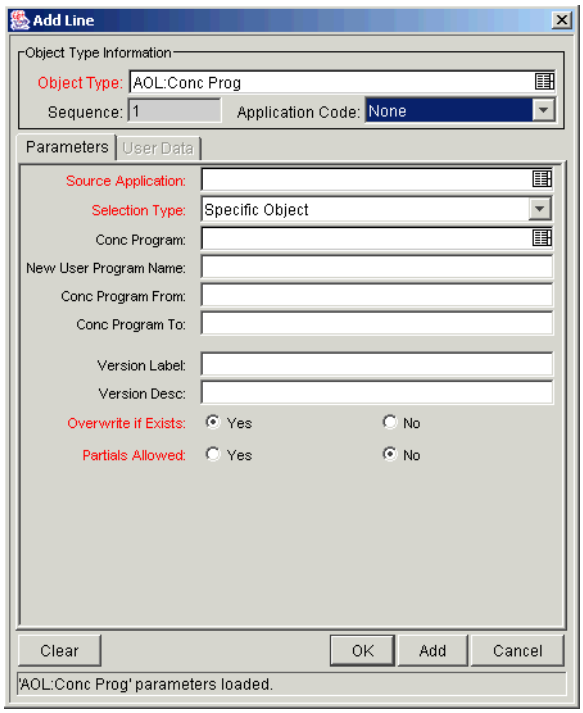
5. Click the **New Line** button on the Package window.

   The Add Line window opens, displaying a list of object types that can be used.

6. In the Object Type field, select the object type to be migrated.

   The Add Line window refreshes to display the parameters required to migrate the specific object. For a description of each parameter, see Appendix B, *Object Type-Specific Migrators and Migration Rules*, on page 165. Both the application name and the specific object name will be validated against the sourcel environment in the workflow. The version label and version description are used only if the workflow is configured to

use the object archive. The source and destination environments are designated by the workflow.



7. Enter the parameters to describe the object.

8. Click **OK** to save the package line.

   Once the package line has been entered and the package submitted, the AOL object moves through the workflow in a similar manner to any other package line in Mercury Change Management.

Processing Packages in Mercury Change Management

Once eligible, users can migrate the selected objects to the destination environment. When an action step in a workflow is executed, Mercury Change Management submits a concurrent request to run Object Migrator. It prints the concurrent request ID for the package line and displays the status of the concurrent request once it completes.

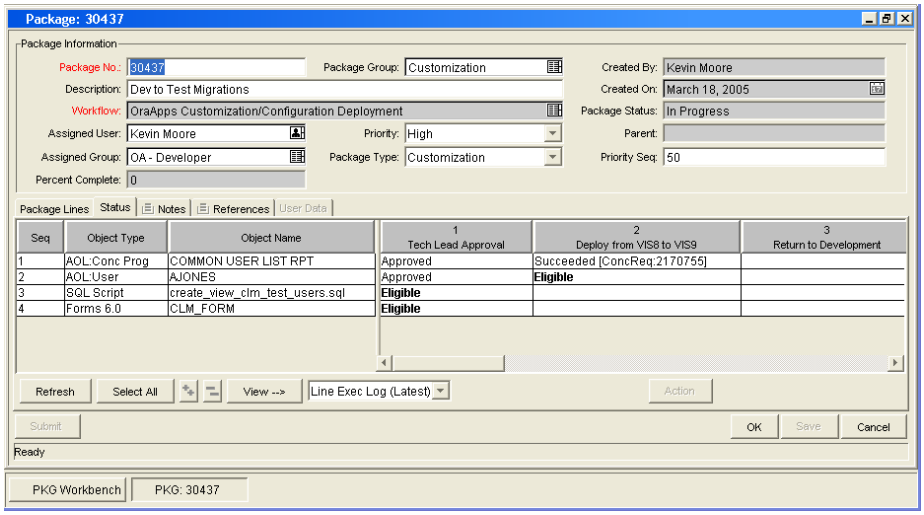


Figure 11-1. Processing a package in Mercury Change Management

When migrating multiple AOL Object request lines, Mercury Change Management automatically runs the concurrent requests serially based on object dependencies. As an additional feature, the report output of the Object Migrator concurrent request can be viewed directly from Mercury Change Management Package window by selecting the log file from the drop-down list in the **Status** tab and clicking **View.**

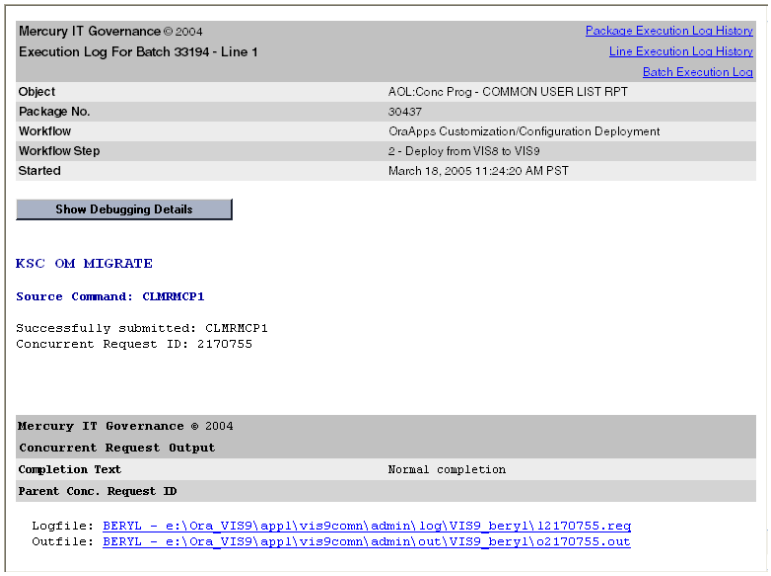


Figure 11-2. Links to concurrent request logs

For More Information

See the *Mercury Change Management User's Guide* for additional details.

Appendix A
AOL-Dependent Objects

Table A-1 lists the dependent object for each of the AOL objects migrated by Object Migrator. The Dependencies (Migrated) column gives the dependent object automatically migrated by the Object Migrator. The Dependencies (Not Migrated) column lists those dependent objects that are not automatically migrated but must be migrated/created by the user before migrating the Object.

Table A-1. AOL object migrated by Object Migrator

No.	Object	Dependencies (Migrated)	Dependencies (Not Migrated)
1	Concurrent Programs	Executables	Value Sets Application # Security Groups # Resource Consumer Groups # Concurrent Request Types # Request Sets Executable Types # Profile Options
2	Concurrent Manager	Work Shifts Specialization Rules	Application # User Oracle User # Conc Program Request Type # Program Library # Security Group #
3	Value Sets	Value Sets (parent)	Security Group #
4	Descriptive Flex Fields		Value Sets Profiles Application # Tables # Security Groups #
5	Request Groups (Report Groups)		Concurrent Program Request Set Application #
6	Menu	Submenus (optionally migrated)	Application # Forms # Subroutines # Macros #
7	Zoom		Forms # Application #

Table A-1. AOL object migrated by Object Migrator [continued]

No.	Object	Dependencies (Migrated)	Dependencies (Not Migrated)
8	Responsibility		Request Group Menu (Char/GUI) Forms (Character only)# Data Groups # Application # Form Functions (for exclusions) Inventory Org # Security Group # Attributes #
9	Users		Responsibilities Employees # Customers # Suppliers # Security Groups # Security Attributes #
10	GUI Menu	Submenus (optionally migrated)	Functions Security Groups #
11	Folders		Users Responsibilities
12	FSG Row/Col Sets		Set of Books # Standard Axes #
13	Functions	Form Definition	Application (form) # Objects # Region Application s# Region Codes #
14	Help Text		Application #
15	Messages		Application #
16	Named SQL		Application #
17	Printer Definitions	Printer Types Printer Drivers Printer Styles	Platforms #

Table A-1. AOL object migrated by Object Migrator [continued]

No.	Object	Dependencies (Migrated)	Dependencies (Not Migrated)
18	Profile Options		Users Responsibilities Application#
19	Quick Codes		Application #
20	Request Sets (Report Sets)		Application # Concurrent Program Request Set Stage Function #

represents items that Object Migrator will not migrate.

Appendix

B

Object Type-Specific Migrators and Migration Rules

Each Oracle AOL object type requires specific information and has unique rules for migrating between instances. Mercury Object Migrator includes a separate concurrent program to migrate each object type. Some information is required for most or all objects; other information is specific to a given object. This section discusses the migration parameters and rules for each supported object type.

For successful migration of the objects described in this section, follow any specific instructions included with the descriptions.

Table B-1 lists (in alphabetical order) the object type-specific migrators along with their associated concurrent program.

Table B-1. Object type-specific migrators

Object Type	Migrator
Common Migrator Parameters	
Concurrent Managers	Migrate Concurrent Managers
Concurrent Programs	Migrate Concurrent Programs
Descriptive Flexfields	Migrate Descriptive Flexfields
Folders	Migrate Folders

Table B-1. Object type-specific migrators [continued]

Object Type	Migrator
FSG Row/Column Sets	Migrate FSG Row/Col Sets
Functions	Migrate Functions
GUI Menus	Migrate GUI Menus
Help Text	Migrate Help Text
Menus (Character Mode)	Migrate Menus
Messages	Migrate Messages
Named SQL	Migrate Named SQL
Printer Definitions	Migrate Printer Definitions
Profile Options	Migrate Profile Options
QuickCodes (AOL)	Migrate QuickCodes
Request Groups (Report Groups)	Migrate Report Groups
Request Sets (Report Sets)	Migrate Report Sets
Responsibilities	Migrate Responsibilities
Users	Migrate Users
Value Sets	Migrate Value Sets
Zooms	Migration Zooms

Common Migrator Parameters

Table B-2 lists (in alphabetical order) and defines the parameters that are common to most Object Migrator object types.

Table B-2. Object Migrator common parameters

Parameter Name	Required?	Description
Compare Only	Yes	Specifies whether to compare objects between instances: Select one of the following options: • **Yes:** Compares a specific object or a range of objects from two instances. Does not migrate any data. • **No:** Does not run the program object comparison mode. For more information, see Chapter 9, *Comparing Objects,* on page 139.
Dest Application	Yes	For object types that are owned by specific applications, the application that is intended to own the object(s) after being migrated (ownership in the Destination Database). This defaults to the application chosen as the source application. Note: Object Migrator uses the application short name to identify applications in the source and destination. The list of applications, however, comes from the database where Object Migrator is installed. If the desired application does not appear on the quickpick, contact your system administrator.

Table B-2. Object Migrator common parameters [continued]

Parameter Name	Required?	Description
Dest Database	Yes	The database to which the object(s) is to be migrated. For object types that allow renaming, this can be the same as the source database, but is typically not the same. If the desired destination database is not an option on the quickpick, contact your system administrator to have the database name added to the appropriate value set. To migrate the object(s) into the object archive rather than to another Oracle Applications instance, select **Object Archive.** Note: In general, the source and destination databases do not need to be on the same Oracle E-Business Suite release: Object Migrator takes care of cross-release migration logic. There are exceptions in some specific migrators where object types become obsolete or where other data model differences preclude cross-reference migration.
From Version Label	No	The version label name of the object(s) that will be recovered from the object archive. This parameter is used in conjunction with Recover From Archive. When trying to recover a specific object, select from all the valid version labels for that object. When recovering a range of objects, select from all version labels that have at least one object of the object type (such as concurrent program and value set) to recover. This parameter is also used when comparing two versions of the same object. For more information, see *Comparing Two Versions of an Object in the Object Archive* on page 144.
New <*Object_Type*> Name	No	The new name of the specific object as it will appear in the destination database after the migration. This parameter defaults to the value from the source database but can be updated to correct the object name or to duplicate the same object to multiple new objects. This parameter is used only when **Specific Object** is selected in the Selection Type field.

Table B-2. Object Migrator common parameters [continued]

Parameter Name	Required?	Description
New User <*Object_Type*> Name/Title	No	Some object types require that the user names/titles be unique within a given application. Object Migrator allows for the changing of the user name/title of the object during the migration process. This is especially useful when duplicating an object. This field is only used if **Specific Object** is selected from the Selection Type field. If left blank, Object Migrator uses the user name from the source database.
<*Object_Name*> From	No	The beginning of the object name range to migrate. This field is used in conjunction with the To field (which specifies the end of the range) when migrating a range of objects. Object Migrator performs an alphabetic search using this parameter so the value does not have to be an exact object name. For example, to migrate all objects for a given application, migrate from A to zzz. Note: The range match is case sensitive. Note: If objects exist that do not begin with characters, then use from exclamation point (!) to tilde (~) to get all objects. This parameter is used only when **Range of Objects** is selected in the Selection Type field.
<*Object_Name*> Like	No	Enter the character string to match against when migrating objects based on a wild card match. Use the percent sign (%) as the multi-character wild card character and the underscore (_) as the single-character wild card character. Object Migrator limits the objects to migrate to object names that match this character string. This wild card match is case sensitive. This parameter is used only when **Wild Card Match** is selected in the Selection Type field.

Table B-2. Object Migrator common parameters [continued]

Parameter Name	Required?	Description
Overwrite if Exists	Yes	Specifies whether to overwrite object definitions that already exist in the destination databases. Select one of the following options: • **Yes:** If the object(s) being migrated already exists in the destination database and the destination application, overwrite the current object(s) definition with the new definition. When migrating and renaming a specific object, use the value of *<Object_Type>* Name as the object name to check against. • **No:** If the object(s) being migrated already exists in the destination database and destination application, then the object will not be overwritten. The migration will fail and indicate the reason in the output report.

Table B-2. Object Migrator common parameters [continued]

Parameter Name	Required?	Description
Partials Allowed	Yes	Specifies whether migrations can proceed when portions of the migration fail. Select one of the following options: • **Yes:** If the object passes validation but any of the components fail validation, then the failed components are logged as failed in the error report, but migrations for the components that pass validation continue. For example, if a parameter of a concurrent program fails validation because it references a value set that does not exist in the destination database, setting Partials Allowed to **Yes** would result in the migration of the concurrent program minus the failing parameter. • **No:** If any of the components of the object fail validation, then the entire object is failed. Additionally, the error report will: ▪ Log the failing components as having failed ▪ Log the object as having failed due to a component failure ▪ Log the components non-failing components as having failed due to an object failure. Example: In the concurrent program example above, Object Migrator will not migrate the concurrent program or any of its components, since one of the parameters could not be migrated. Note: Objects are evaluated individually when migrating multiple objects (using range or wild card selection). The failure of one object will not prevent migration of other objects.
Recover From Archive	Yes	Specifies whether to use the object archive as the source database. Select one of the following options: • **Yes:** Use the object archive as the data source for the migration. • **No:** Do not use the object archive as the data source for the object migration. This parameter should be set to **Yes** if Object Archive is selected in the Source Database and should be set to **No** if the Source Database is not Object Archive.

Table B-2. Object Migrator common parameters [continued]

Parameter Name	Required?	Description
Report Only	Yes	Specifies whether Object Migrator should simulate the migration or perform the actual migration. Select one of the following options: • **Yes:** Does not import the data after being extracted from the source and validated. Reports "would be" successes and exceptions in the same format as the standard report. • **No:** Imports all objects that pass validation. Reports successes and exceptions as normal. Note: Report Only must be set to **No** if Compare Only is set to **Yes** or if the Destination Database is set to Object Archive.
Save To Archive	Yes	Indicates whether to save an object to the object archive, preserving the version for reference or later use. Select one of the following options: • **Yes:** Put the object(s) being migrated under version control and store them in the object archive. You can save to the archive when performing a regular migration to an Oracle Applications instance, or specify the object archive as the destination database. • **No:** Do not store the object(s) being migrated into the object archive. Note: This parameter cannot be set to **Yes** if Report Only is set to **Yes** or Compare Only is set to **Yes.**

Table B-2. Object Migrator common parameters [continued]

Parameter Name	Required?	Description
Selection Type	Yes	Indicates whether to migrate one or multiple objects. Select one of the following options: • **Specific Object.** One object will be selected for migration at this time. Enter the specific object name in the specific *<Object_Name>* field (for example, specific concurrent program). • **Range of Objects.** A range of objects will be selected for migration at this time. Specify the alphanumeric range to use in the *<Object_Name>* name From and To fields (for example, Conc Program Name From and Conc Program Name To). The range parameters are case sensitive. • **Wild Card Match.** Multiple objects will be migrated with object names matching a specified string. The percent sign (%) will be used as a wild card in this string. Specify this character string to match against in the *<Object_Name>* Like field. The wild card parameters are case sensitive. Note: When migrating multiple objects (range or wild card), the migration will fail if no objects can be migrated successfully, but will end succeed if even one object is successfully migrated. For this reason, always review the request output carefully.
Source Application	Yes	For object types that are owned by specific applications, the application that currently owns the object(s) being migrated (ownership in the source database). Note: Object Migrator uses the application short name to identify applications in the source and destination. The list of applications, however, comes from the database where Object Migrator is installed. If the desired application does not appear on the quickpick, contact your system administrator.

Table B-2. Object Migrator common parameters [continued]

Parameter Name	Required?	Description
Source Database	Yes	The database where the object(s) to be migrated currently exists. If the desired source database is not an option on the quickpick, contact your system administrator to have the database name added to the appropriate value set. If using an archived version of the object(s) as the source data for the migration, select **Object Archive** as the source database. Note: In general, the source and destination databases do not need to be on the same Oracle E-Business Suite release: Object Migrator takes care of cross-release migration logic. There are exceptions in some specific migrators where object types become obsolete or where other data model differences preclude cross-reference migration.
Specific <*Object_Type*> Name	When Selection Type = Specific Object	The current name (as named in the source database) of the object to migrate. This value is used (with the application value, if it is defined) to uniquely identify the object to migrate. This name is usually the short name or code for the object used when defining the AOL object. An example is `XXWIPDJPCK` program. This field is required if **Specific Object** is selected in the Selection Type field. If **Range of Objects** is selected in the Selection Type field, do not enter a value in this field.
To	No	The end of the object name range to migrate. This field is used in conjunction with the <*Object_Name*> From field (which specifies the start of the range). Object Migrator performs an alphabetic search using this parameter so the value does not have to be an exact object name. For example, to migrate all objects for a given application, migrate from A to zzz. Note: If objects exist that do not begin with characters, then use from exclamation point (!) to tilde (~) to get all objects. This parameter is only used when **Range of Objects** is selected in the Selection Type field.

Table B-2. Object Migrator common parameters [continued]

Parameter Name	Required?	Description
Version Description	No	The description to attach to the object version(s) as they are stored in the object archive. This value will also be the description for the version label if this is a new version label. If this field is left blank and the version label already exists, the Object Migrator will use the description of the version label as the description of the object versions. This field should be entered only if Save To Archive is set to **Yes.**
Version Label	When Save to Archive = **Yes**	The version label name for object(s) saved into the object archive. If a new version label is specified, Object Migrator will create the label. If an existing version label is specified, Object Migrator will add the object(s) to the existing label. A version label is required if Save To Archive is set to **Yes.** This parameter is also used when comparing two versions of the same object. For more information, see *Comparing Two Versions of an Object in the Object Archive* on page 144.

Concurrent Managers

This migrator migrates the definition of the concurrent manager as well as all associated work shifts and specialization rules.

Before Migrating Concurrent Managers

Before migrating Concurrent Managers, you need to make sure that a number of prerequisite conditions are met.

If the following entities are referenced by the Concurrent Managers in the source database, they must also exist in the destination database prior to the migration.

- Concurrent program, application, or complex rule referenced by a migrated specialization
- User referenced by a migrated specialization rule or complex rule
- Oracle ID referenced by a migrated specialization rule or complex rule
- Request type referenced by a migrated specialization rule or complex rule
- Program library
- Security groups
- Concurrent Manager type

Additionally, the Concurrent Manager being migrated must not be active in the destination database.

Any programs referenced by a migrated complex rule must include both an application and program entry if the destination is release 11 or higher.

The Concurrent Manager Definitions Migrator will migrate only new concurrent managers or managers that are not active on the destination database. If the manager is currently running on the destination database, deactivate it first before running the Object Migrator.

The Concurrent Manager Definitions Migrator will migrate all work shifts and specialization rules attached to the migrated Concurrent Manager. It will overwrite the definition of the work shift or specialization rule if it already exists on the destination database.

The type defined for the Concurrent Manager must be supported in the destination database. Migration from newer to older releases is prevented for certain predefined services, such as Apache JServer, if the data source and the values in the destination are not compatible with those in the source.

Concurrent Managers Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Concurrent Manager request.

Only the object-specific parameters are defined here. For more information about the common Object Migrator parameters, see *Table B-2*.

Table B-3. Concurrent Manager Definitions Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2*.
Dest Database	
Source Application	
Dest Application	
Selection Type	
Specific Conc Manager Name	Enter the current Concurrent Manager name (as named in source database) of the Concurrent Manager to migrate. Query this uppercase-only value from the Define Concurrent Manager form.
New Conc Manager Name	Enter a new name for the Concurrent Manager, if desired. Otherwise, the name from the source instance will be used.
New Description	Enter a new description for the Concurrent Manager, if desired.

Table B-3. Concurrent Manager Definitions Migrator parameters [continued]

Parameter Name	Description/Notes
Conc Manager From	For descriptions of these common parameters, see *Table B-2*.
To	
Conc Manager Like	
Save To Archive	
Version Label	
Version Desc	
Recover from Archive	
From Version Label	
Overwrite if Exists	
Partials Allowed	
Report Only	
Compare Only	

Concurrent Programs

This migrator migrates concurrent program registration as well as the executable definitions referenced by the concurrent program.

This section covers the following topics:

- *Before Migrating Concurrent Managers*
- *Concurrent Managers Migrator Parameters*

Before Migrating Concurrent Programs

Before migrating concurrent programs, you need to make sure that a number of prerequisite conditions are met. If the following entities are referenced by the concurrent program in the source database, they must also exist in the destination database prior to the migration:

- Concurrent request types
- Executable types MLS application and executable
- Profile options
- Request sets
- Resource consumer groups
- Security groups
- Value sets

Also, if there are any incompatible programs defined in the source instance, then those concurrent programs should either be present in the destination instance or, when Mig Incomp Partially is set to **No**, should be migrated together. For details about this parameter, see *Table B-4*.

Some patch levels of the Oracle E-Business Suite 11i require each incompatibility defined for a concurrent program be qualified by an incompatibility type, which determines the scope within which incompatibilities are evaluated by Concurrent Managers. Incompatibilities can be evaluated within a specific conflict domain, or globally across all conflict domains. The Concurrent Programs Migrator will migrate the incompatibility type if the destination supports incompatibility types. If incompatibility types are used in the destination but not in the source, the Object Migrator defaults the incompatibility type for the program incompatibility based on the same logic Oracle uses during upgrades.

Concurrent Programs Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Concurrent Programs request.

Note

Only the object-specific parameters are defined here. For more information about the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-4. Concurrent Programs Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Source Application	The application to which the concurrent program belongs. For a description of this common parameter, see *Table B-2* on page 167.
Dest Application	For a description of this common parameter, see *Table B-2* on page 167.
Selection Type	
Specific Conc Program	Enter the short name (as named in source database) of the concurrent program to migrate. Query this uppercase-only value from the Define Concurrent Programs form. For a description of this common parameter, see *Table B-2* on page 167. Example: FNDSCARU is the short name for the Active Users report.
New Conc Program Short Name	Enter a new short name for the concurrent program if renaming is desired. For a description of this common parameter, see *Table B-2* on page 167.
New User Program Name	Enter the new user program name of the concurrent program as it will appear in the destination database after the migration. This allows for the ability to change the user program name for a concurrent program. If left blank, Object Migrator will use the user name from the source database. For a description of this common parameter, see *Table B-2* on page 167.

Table B-4. Concurrent Programs Migrator parameters [continued]

Parameter Name	Description/Notes
Conc Program From	For a description of this common parameter, see *Table B-2* on page 167.
Conc Program To	
Conc Program Like	Enter a search string to match by concurrent program short name. For a description of this common parameter, see *Table B-2* on page 167.
Save To Archive	For a description of this common parameter, see *Table B-2* on page 167.
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Partials Allowed	
Mig Incomp Partially (Required) (Parameter may not be displayed)	Specifies the treatment of incompatibilities during migration. Select one of the following: • **Yes:** If incompatibilities exist for the program which cannot be mapped to the destination, leave those incompatibilities out without error. • **No:** All incompatibilities defined in the source must exist in the destination in order for the execution to succeed.
Report Only	For a description of this common parameter, see *Table B-2* on page 167.
Compare Only	

Descriptive Flexfields

Before Migrating Descriptive Flexfields

Before migrating descriptive flexfields, you need to make sure that a number of prerequisite conditions are met, as described in this section.

If the following entities are referenced by the descriptive flexfields in the source database, they must also exist in the destination database prior to the migration:

- Profile options
- Value sets
- Security groups

The following must also be true:

- Owning Table of the descriptive flexfield in the source database must exist, and be registered to the same application in the destination database.
- The application owning the table that owns the flexfield in the source database must also exist in the destination database with the same application short name.
- The form that uses the flexfield in the source instance should be the same as the form that uses the flexfield in the destination instance.

This is not validated by the Migrate Descriptive Flexfields program.

- Table structures that own the descriptive flexfield being migrated should be the same in the source and destination databases.

This is not validated by the Migrate Descriptive Flexfields program.

- Any SQL statements used in descriptive flexfields being migrated should be valid in the destination database.

This is not validated by the Migrate Descriptive Flexfields program.

- Columns referenced by the flexfield in the source must be registered in the destination database, and should not be used by a different flexfield in the destination instance. This is validated by the Migrate Descriptive Flexfields program.
- The flexfield in the destination (if it exists) should not be protected. This is validated by the Migrate Descriptive Flexfields program.
- If not migrating disabled contexts, the default context must be enabled in the source.
- The title of flexfields being migrated should not exist on another flexfield for the same application. This is validated by the Migrate Descriptive Flexfields program.
- Be sure that existing transaction data relating to the flexfield in the destination remains valid. The migration replaces the definition in the destination, which can remove or change the value set for existing segments and contexts.

Descriptive Flexfields Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Descriptive Flexfields program.

Only the object-specific parameters are defined here. For more information about the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-5. Descriptive Flexfields Migrator parameters

Parameter Name	Description/Notes
Source Database	For a description of this common parameter, see *Table B-2* on page 167.
Dest Database	
Source Application	
Dest Application	
Selection Type	
Specific Desc Flex	Enter the current name (as named in source database) of the descriptive flexfield to migrate. Query this uppercase-only value from the Register Descriptive Flexfields form. This is not the title from the Define Descriptive Flexfield Segments form. For a description of this common parameter, see *Table B-2* on page 167.
Desc Flex From	For a description of this common parameter, see *Table B-2* on page 167.
To	
Desc Flex Like	
Save To Archive	
Version Label	
Version Description	
Recover From Archive	
From Version Label	

Table B-5. Descriptive Flexfields Migrator parameters [continued]

Parameter Name	Description/Notes
Disabled Contexts	Species whether to migrate descriptive flexfields contexts that are disabled in the source. Select one of the following: • **Yes:** All contexts from the source, including disabled contexts, will be migrated to the destination. • **No:** Flexfield contexts which are disabled in the source environment will not be migrated to the destination environment. Select **No** to prevent contexts still in development from being migrated with the remainder of the flexfield. If the context exists in the destination, it will be removed from the flexfield definition. Caution should be taken when using **No** for disabled contexts to make sure that the transaction data relating to the flexfield in the destination remains valid. If the context exists in the destination it will be removed, even if it is enabled. If the intent is to disable use of the flexfield context in the destination, it should be disabled in the source and migrated with **Yes** above.
Disabled Segments	Specifies whether to migrate disabled segments. Select one of the following: • **Yes:** All segments from the source, including disabled segments, will be migrated to the destination. • **No:** Flexfield segments which are disabled in the source environment will not be migrated to the destination environment. This option is used to prevent segments still in development from being migrated with the remainder of the flexfield. If the segment exists in the destination, it will be removed from the flexfield definition. Caution should be taken when using **No** or disabled contexts and disabled segments to make sure that the transaction data relating to the flexfield in the destination remains valid. If the context or segment exists in the destination it will be removed, even if it is enabled. If the intent is to disable use of the flexfield context or segment in the destination, it should be disabled in the source and migrated with **Yes** above.

Table B-5. Descriptive Flexfields Migrator parameters [continued]

Parameter Name	Description/Notes
Overwrite if Exists	For a description of this common parameter, see *Table B-2* on page 167.
Report Only	
Compare Only	For a description of this common parameter, see *Table B-2* on page 167. Note: The comparison functionality does not compare column information when comparing objects.

Folders

Before Migrating Folders

Before migrating folders, you need to make sure that a number of prerequisite conditions are met. If the following entities are referenced by the folders in the source database, they must also exist in the destination database prior to the migration:

- Owner (application user)
- Segment (form blockname)
- Responsibilities or users referenced by a default folder setting

Folders Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Folders program.

Only the object-specific parameters are defined here. For more information on the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-6. Folders Migrator parameters

Parameter Name	Description/Notes
Source Database	For a description of this common parameter, see *Table B-2* on page 167.
Dest Database	
Selection Type	
Specific Folder Name	Enter the current name (as named in source database) of the folder(s) to migrate. If left blank, the Folders Migrator will not limit by folder name (just by specific owner and specific segment if they are specified).
New Folder Name	Enter a new folder name if a name change is desired.

Table B-6. Folders Migrator parameters [continued]

Parameter Name	Description/Notes
Specific Owner	Enter the owner (as named in source database) of the folder(s) to migrate. The list of values for this parameter is not validated against the specific folder parameter. It is only validated against the application users defined in the source database. If left blank, the Folders Migrator will not limit by folder owner (just by specific folder name and specific segment if they are specified).
New Owner Name	Enter a new owner name if a name change is desired.
Specific Segment Name	Enter the current folder segment (as named in source database) of the folder(s) to migrate. The list of values for this parameter is not validated against the specific folder or the specific owner parameter. It is only validated against all the folder segments in the source database. If left blank, the Folders Migrator will not limit by folder segment (just by specific folder name and specific owner if they are specified).
Folder Name From	For a description of this common parameter, see *Table B-2* on page 167.
To	
Owner Name From	
To	
Segment Name From	
To	
Save To Archive	
Version Label	
Version Desc	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Partials Allowed	
Report Only	
Compare Only	

FSG Row/Column Sets

This section discusses migrating FSG row/column sets, including all of the object-specific parameters on the Submit Request window. Ensure the successful migration of these objects by following the instructions included in this section.

Before Migrating FSG Row/Column Sets

Before migrating FSG row/column sets, make sure that the following prerequisite conditions are met. If the following entities are referenced by the Row Set or Column Set in the source database, they must also exist in the destination database prior to the migration:

- Flexfield structures
- Standard axes
- Set of books

FSG Row/Column Sets Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate FSG Row/Col Sets request.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-7. FSG Row/Column Sets Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Set Type (required)	Enter/Quickpick the type of FSG Set(s) to migrate: • **Column** to migrate FSG Column Set(s) • **Row** to migrate FSG Row Set(s)
Selection Type	For a description of this common parameter, see *Table B-2* on page 167.

Table B-7. FSG Row/Column Sets Migrator parameters [continued]

Parameter Name	Description/Notes
Specific Row/Column Set	Enter the current name (as named in source database) of the FSG Row Set or FSG Column Set to migrate. Query this value from the Define Row Set or Define Column Set forms. The list of values for this parameter does not limit the values by Set Type. For a description of this common parameter, see *Table B-2* on page 167.
New Row/Column Set	For descriptions of these common parameters, see *Table B-2* on page 167.
Row/Column Set From	
To	
Row/Column Set Like	
Partials Allowed	
Save To Archive	
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Report Only	
Compare Only	

Functions

This section discusses migrating functions, including all of the object-specific parameters on the Submit Request window. These are objects that define functions for GUI-based versions of Oracle E-Business Suite: release 11i, release 11, and release 10.7 Smart Client or NCA. Ensure the successful migration of these objects by following the instructions included in this section.

This migrator also registers forms (GUI) if the function(s) being migrated references a form that does not exist in the destination database.

Before Migrating Functions

Before migrating a function, make sure that a number of prerequisite conditions are met. If the following entities are referenced by the function in the source database, they must also exist in the destination database prior to the migration.

- Form applications
- Objects referenced by a function must exist on the destination database. This applies only when the destination instances are 11.5.5 or higher, as object functionality was added in 11.5.5.
- Region applications referenced by a function must exist on the destination database if the destination is 11.5.5 or higher.
- Region codes referenced by a function must exist on the destination database if the destination is 11.5.5 or higher.
- The context dependence value, maintenance mode support value, and type value referenced by a function must exist on the destination database if the destination includes this information. This is validated by the migrate Functions Migrator.

  When migrating from instances that do not include maintenance mode support, the maintenance mode support value will default to **None.**

  When migrating from instances that do not include the Context Dependence field, the context dependence value will default to responsibility.

If a form referenced by a function does not exist on the destination database, it will be created. The form file will not be copied to the destination.

In some patch levels of release 11i, user function names do not need to be unique. The Functions Migrator enforces the appropriate logic based on the rules in place in the destination.

Functions Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Functions program.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-8. Functions Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Selection Type	
Specific Function	Enter the current name (as named in source database) of the function to migrate. Query this uppercase-only value from the Define Form Functions form (using the Smart Client application). For a description of this common parameter, see *Table B-2* on page 167.
New Function Name	For a description of this common parameter, see *Table B-2* on page 167.
New User Function Name	Enter the new user name of the function as it will appear in the destination database after the migration. This gives you the ability to change the user name for a function. If left blank, Object Migrator will use the user name from the source database.

Table B-8. Functions Migrator parameters [continued]

Parameter Name	Description/Notes
Function Name From	For descriptions of these common parameters, see *Table B-2* on page 167.
To	
Function Name Like	
Save To Archive	
Version Label	
Version Desc	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Report Only	
Compare Only	

GUI Menus

This section discusses migrating GUI menus, including all of the object-specific parameters on the Submit Request window. These are objects that define menus for GUI-based versions of Oracle E-Business Suite: release 11i, release 11, and release 10.7 Smart Client or NCA. Ensure the successful migration of these objects by following the instructions included in this section.

Before Migrating GUI Menus

Before migrating GUI menus, make sure that a number of prerequisite conditions are met. If the following entities are referenced by the GUI menus in the source database, they must also exist in the destination database prior to the migration.

- Functions referenced by the menu (and the menu tree underneath the menu, if Single Level Only is set to **No**)
- If Single Level Only is set to **Yes,** then any child menus reference by the parent menu must exist in the destination database.
- Menu types referenced by the menu must exist in the destination database. This applies only when menu types are in use in the destination.
- Security Groups

If the destination database is 11.5.5 or higher, the Object Migrator will submit the Compile Security program in the destination instance in order to complete the menu compilations introduced in 11.5.5.

Care should be taken when migrating entire menu structures to make sure that standard menus are not inadvertently overwritten. Standard menus may be referenced from submenus in a custom menu structure.

GUI Menus Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate GUI menus request.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-9. GUI Menus Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Selection Type	
Specific Menu Name	Enter the current menu name (as named in source database) of the menu to migrate. Query this uppercase-only value from the define menus form (using the Smart Client form if the source database is 10.7). For a description of this common parameter, see *Table B-2* on page 167.
New Menu Name	For a description of this common parameter, see *Table B-2* on page 167.
New Menu Title	Enter the new menu title of the menu as it will appear in the destination database after the migration. This gives you the ability to change the title for a GUI menu. If left blank, Object Migrator will use the title from the source database.
Menu Name From	For descriptions of these common parameters, see *Table B-2* on page 167.
To	
Menu Name Like	

Table B-9. GUI Menus Migrator parameters [continued]

Parameter Name	Description/Notes
Single Level Only (required)	Specifies whether to migrate only the top level of the menu, or to include all sub-entries and sub-menus beneath the menu. Select one of the following: • **Yes:** Only migrate the menu entries of the given menu. • **No:** Migrate the entire menu sub-tree below the specific menu name. Care should be taken when migrating entire menu structures to make sure that standard menus are not inadvertently overwritten. Standard menus may be referenced from submenus in a custom menu structure.
Save To Archive	For descriptions of these common parameters, see *Table B-2* on page 167.
Version Label	
Version Desc	
Recover From Archive	
From Version Label	
Overwrite if Exists	For a description of this common parameter, see *Table B-2* on page 167. Note that this parameter governs the parent menu only. If migrating an entire menu tree, submenus that exist will be overwritten without error, even if Overwrite is set to **No,** as long as the top parent menu did not exist.
Partials Allowed	For descriptions of these common parameters, see *Table B-2* on page 167.
Report Only	
Compare Only	For a description of this common parameter, see *Table B-2* on page 167. Note: For better output, it is recommended that comparisons of menus be run with Single Level Only set to **Yes.**

Migrating a range of parent menus with Single Level set to **No** can entail migrating very large amounts of data. It is recommended to migrate parent menus in small batches or individually.

Help Text

This section discusses migrating help text, including all of the object-specific parameters on the Submit Request window. This type of help text, which is for character mode forms, is used in Oracle release 10.7 only. Ensure the successful migration of these objects by following the instructions included in this section.

Before Migrating Help Text

Before migrating help text, make sure that the following prerequisite conditions are met. If the following entities are referenced by the help text in the source database, they must also exist in the destination database prior to the migration:

- The source form and the destination form should be identical in terms of block names and field names.
- The current versions of both forms should be registered in the respective databases.

This type of help text is not supported in Oracle release 11 or 11i; therefore, it cannot be migrated to or from those instances.

Help Text Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Help Text request.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-10. Help Text Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Source Application	
Dest Application	
Selection Type	
Specific Form Name	Enter the current name (as named in source database) of the Form whose help text will be migrated. Query this uppercase only value from the Update Form Information form. This is not the user friendly Title of the Form. For a description of this common parameter, see *Table B-2* on page 167.
New Form Name	For descriptions of these common parameters, see *Table B-2* on page 167.
Form From	
To	
Form Like	
Save To Archive	
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Report Only	
Compare Only	For a description of this common parameter, see *Table B-2* on page 167. Note: Currently, the comparison functionality does not compare the value in the Help Text field when comparing objects.

Menus (Character Mode)

This section discusses migrating character mode menus, including all of the object-specific parameters on the Submit Request window. This type of menu is used in Oracle release 10.7 only. Ensure the successful migration of these objects by following the instructions included in this section.

Before Migrating Menus

Before migrating menus, make sure that a number of prerequisite conditions are met. If the following entities are referenced by the menus in the source database, they must also exist in the destination database prior to the migration.

- Forms referenced by the menu (and the menu tree underneath the menu, if Single Level Only is set to **No**)
- Subroutines reference by the menu (and the menu tree underneath the menu, if Single Level Only is set to **No**)
- Macros reference by the menu (and the menu tree underneath the menu, if Single Level Only is set to **No**)
- If Single Level Only is set to **Yes,** then any child menus reference by the parent menu must exist in the destination database.

Character mode menus are not supported in release 11 or 11i; therefore, they can not be migrated to or from those releases.

Menus Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Menus program.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-11. Menus Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Source Application	
Dest Application	
Selection Type	
Specific Menu Name	Enter the current menu name (as named in source database) of the menu to migrate. Query this uppercase-only value from the Define Menus form (using the character mode forms). For a description of this common parameter, see *Table B-2* on page 167.
New Menu Name	For a description of this common parameter, see *Table B-2* on page 167.
New Menu Title	Enter the new menu title of the menu as it will appear in the destination database after the migration. This gives you the ability to change the title for a menu. If left blank, Object Migrator will use the title from the source database.
Menu Name From	For descriptions of these common parameters, see *Table B-2* on page 167.
To	
Menu Name Like	

Table B-11. Menus Migrator parameters [continued]

Parameter Name	Description/Notes
Single Level Only (Required)	Specifies whether to migrate only the top level of the menu or to include all sub-entries beneath the menu. Select one of the following: • **Yes:** Only migrate the menu entries of the given menu. • **No:** Migrate the entire menu sub-tree below the specific menu name.
Import Standard Oracle Menus	(Use only if Single Level Only is set to **No.**) Select one of the following options: • **Yes:** Migrate and report on menus owned by standard Oracle Applications. • **No:** Do not migrate or report on menus owned by standard Oracle Applications.
Import AOL/Sysadmin Menus	(Relevant only if Single Level Only set to **No** and Import Standard Oracle Menus is set to **Yes.**) Select one of the following options • **Yes:** Migrate and report on menus owned by the system administrator and Application Object Library applications. • **No:** Do not migrate or report on menus owned by the system administrator and Application Object Library applications.
Save To Archive	For descriptions of these common parameters, see *Table B-2* on page 167.
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Partials Allowed	
Report Only	
Compare Only	For a description of this common parameter, see *Table B-2* on page 167. Note: For better output, it is recommended that comparisons of menus be run with Single Level Only set to **Yes.**

Messages

This section discusses migrating messages, including all of the object-specific parameters on the Submit Request window. Ensure the successful migration of these objects by following the instructions included in this section.

This migrator migrates both server-side messages used by character mode forms and client-side messages used by Smart Client/GUI forms.

Before Migrating Messages

Before migrating messages, make sure that a number of prerequisite conditions are met.

If the following entities are referenced by the messages in the source database, they must also exist in the destination database prior to the migration:

- Application
- Category, if the destination supports this information
- Severity, if the destination supports this information

For server-side messages only (nothing needs to be done for messages used by Smart Client forms), Oracle Applications requires the execution of the concurrent program “Create message file and report” (release 10) or “Generate Messages” (release 11/11i). Due to database limitations with distributed transactions, Object Migrator cannot launch this program remotely. As a post migration step for each unique language/application combination successfully migrated, this program needs to be run.

You can run the program by navigating to the Define Messages Character Mode form, querying a message from the given application, updating the record without a real change (for example, change a letter in the message and changing it back), and committing the transaction.

This can also be done by launching the program from the command-line for the destination database. For more information, see the *Oracle Applications System Administration Guide*).

Messages Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Messages request.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-12. Messages Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Source Application	The application to which the Message belongs. For a description of this common parameter, see *Table B-2* on page 167.
Dest Application	For descriptions of these common parameters, see *Table B-2* on page 167.
Selection Type	
Specific Message	Enter the current message name (as named in source database) of the message to migrate. Query this uppercase-only value from the Define Messages form.
New Message Name	Enter a new message name for the message if a name change is desired.

Table B-12. Messages Migrator parameters [continued]

Parameter Name	Description/Notes
Message From	For descriptions of these common parameters, see *Table B-2* on page 167.
To	
Message Name Like	
Save To Archive	
Version Label	
Version Desc	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Partials Allowed	
Report Only	
Compare Only	

Named SQL

This section discusses migrating named SQL, including all of the object-specific parameters on the Submit Request window. Named SQL is used in Oracle Applications release 10.7 only. Ensure the successful migration of these objects by following the instructions included in this section.

Before Migrating Named SQL

Before migrating named SQL, note the following migration restriction:

- Named SQL is not supported in release 11 or 11i; therefore, these objects can not be migrated to or from those releases.

Named SQL Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Named SQL program.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-13. Named SQL Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Source Application	
Dest Application	
Selection Type	
Specific Named SQL	Enter the current name (as named in source database) of the named SQL to migrate. Query this value from the Define Named SQL form. For a description of this common parameter, see *Table B-2* on page 167.

Table B-13. Named SQL Migrator parameters [continued]

Parameter Name	Description/Notes
New Named SQL	For descriptions of these common parameters, see *Table B-2* on page 167.
Named SQL From	
To	
Named SQL Like	
Save To Archive	
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Report Only	
Compare Only	For a description of this common parameter, see *Table B-2* on page 167. Note: Currently, the comparison functionality does not compare the value in the SQL Text field when comparing objects.

Printer Definitions

This section discusses migrating printer definitions, including all of the object-specific parameters on the Submit Request window. Ensure the successful migration of these objects by following the instructions included in this section.

Use this migrator to migrate printer types and all the printers, printer styles, and printer drivers associated with them.

Any printers, printer styles, and printer Drivers that are already assigned to the printer type in the destination database will not be removed, even if they do not exist in the source database.

Before Migrating Printer Definitions

Before migrating printer definitions, make sure that a number of prerequisite conditions are met. If the following entities are referenced by the printer definitions in the source database, they must also exist in the destination database prior to the migration.

- Subroutines or commands
- Platforms

Any new printer styles must have a sequence number that is unique on the destination database.

Printer Definitions Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Printer Definitions program.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-14. Printer Definitions Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Selection Type	
Specific Printer Type	Enter the current name (as named in source database) of the printer type to migrate. Object Migrator will attempt to migrate the printer type, any printer attached to the printer type, any printer style assigned to the printer type, and any printer Driver assigned to the printer type. Query this uppercase-only value from the Define Printer Types form. For a description of this common parameter, see *Table B-2* on page 167.
New Printer Type	For descriptions of these common parameters, see *Table B-2* on page 167.
Printer Type From	
To	
Printer Type Like	
Import Printers	• **Yes:** Along with the printer type itself, migrate any printers attached to the printer type. • **No:** Migrate the printer styles and Drivers attached to the printer type, but not any printers attached to the printer type.

Table B-14. Printer Definitions Migrator parameters [continued]

Parameter Name	Description/Notes
Save To Archive	For descriptions of these common parameters, see *Table B-2* on page 167.
Version Label	
Version Desc	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Partials Allowed	
Report Only	
Compare Only	

Profile Options

This section discusses migrating profile options, including all of the object-specific parameters on the Submit Request window. Ensure the successful migration of these objects by following the instructions included in this section.

Caution should be used when migrating profile option values to make sure that the values remain valid in the destination. When migrating profile option values, values are not re-derived or validated during migration. This could cause problems if the value (for example, the organization id for an inventory org) is not valid or refers to a different entity in the destination database.

Before Migrating Profile Options

Before migrating profile options, make sure that a number of prerequisite conditions are met. If the following entities are referenced by the profile options in the source database, they must also exist in the destination database prior to the migration:

- Users that have a user-level setting for the profile option
- Responsibilities that have a responsibility-level setting for the profile option
- Applications that have a application-level setting for the profile option
- If migrating with Values Only set to **Yes,** the profile option to which the values belong must exist in the destination
- Hierarchy type
- Organizations that have an organization-level setting for the profile option (for organization hierarchy)
- Servers that have a server-level setting for the profile option (for server hierarchies)

Additionally, the following migration rules should be noted:

- Any SQL statements used in profile options being migrated should be valid in the destination database.

  This is not validated by the Migrate Profile Options program.

- When moving profile settings, the Profile Options Migrator does not validate the actual option value. For example, if a profile option was for a user and in the source database the value was a user ID corresponding to SYSADMIN, the migrator would bring the user id over to the destination as is. It would not get the user ID of SYSADMIN in the destination database and would not validate that the user id is valid in the destination.
- When migrating from a release where hierarchy types are not used to an instance that uses hierarchy types, the profile is created using the security hierarchy.
- Object Migrator allows migration of values for profiles using the server hierarchy. Migrating profile option values at the server level may produce Missing Reference errors unless the same servers are shared across Oracle Applications instances, since their server names would differ.

The migrator raises the `oracle.apps.fnd.profile.value.update` Business Event in the destination for each profile for which values have been migrated to or removed from the destination. This occurs only if the destination supports the functionality (11.5.9 or higher) and Migrate Profile Option Values is set to **Yes.**

The following are special considerations relevant to Business Events:

- Raising Business Events can cause real-time processing to occur in the destination, and the migration will not finish until this processing completes. Minimizing real-time processing in subscriptions for the oracle.apps.fnd.profile.value.update event helps make sure efficiency of profile option value migrations.
- You should be sure that custom processing code for subscriptions to the oracle.apps.fnd.profile.value.update event functions correctly when invoked across a database link.
- If Business Event processing encounters an error, it is logged in the destination instance and does not prevent completion of the migration. You should monitor the errors in the destination instance to identify any issues.

Profile Options Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Profile Options program.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-15. Profile Options Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Source Application	
Dest Application	
Selection Type	
Specific Profile Option	Enter the current name (as named in source database) of the profile option to migrate. Query this uppercase only value from the Define Profile Options form. For a description of this common parameter, see *Table B-2* on page 167.
New Profile Option Name	For descriptions of this common parameter, see *Table B-2* on page 167.
New Profile User Name	Enter the new user name of the profile option as it will appear in the destination database after the migration. This gives you the ability to change the user name for a profile option. If left blank, Object Migrator will use the user name from the source database. For a description of this common parameter, see *Table B-2* on page 167.

Table B-15. Profile Options Migrator parameters [continued]

Parameter Name	Description/Notes
Profile Option From	For descriptions of these common parameters, see *Table B-2* on page 167.
To	
Profile Option Like	
Save To Archive	
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Migrate Profile Option Values (Required)	Select one of the following options: • **Yes:** Along with the profile option itself, migrate all the settings for the profile from the source database. Overwrite the settings on the destination database for the same sites, applications, responsibilities, servers, organizations, and users. • **No:** Do not migrate any settings for the profile option from the source database. Do not modify the settings on the destination database.
Migrate Values Only (Required)	Select one of the following options: • **Yes:** Do not migrate the profile option definition. But migrate the profile option settings (at the site, application, responsibility, server, organization, and user levels). • **No:** Do not exclude the profile option definition in the migration.
Remove Non-Migrated Values (Required)	Select one of the following options: • **Yes:** If migrating profile option values, remove any settings that exist in the destination and not the source database. • **No:** Do not remove profile option settings that exist in the destination but not the source.
Overwrite if Exists	For descriptions of these common parameters, see *Table B-2* on page 167.
Partials Allowed	
Report Only	
Compare Only	

QuickCodes (AOL)

This section discusses migrating QuickCodes, including all of the object-specific parameters on the Submit Request window. Ensure the successful migration of these objects by following the instructions included in this section.

QuickCode objects, based on the FND_COMMON_LOOKUPS table, can be viewed or modified using the Define QuickCode form under the application developer responsibility.

Before Migrating QuickCode

Before migrating QuickCodes, note the following migration restriction:

- Quick Codes can not be migrated to or from release 11i to any other release.

QuickCodes Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate QuickCodes request.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-16. QuickCodes Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Source Application	
Dest Application	
Selection Type	

Table B-16. QuickCodes Migrator parameters [continued]

Parameter Name	Description/Notes
Specific QuickCode Type	Enter the current name (as named in source database) of the QuickCode type to migrate. Query this value from the Define QuickCodes form. For a description of this common parameter, see *Table B-2* on page 167.
New QuickCode Type	For descriptions of these common parameters, see *Table B-2* on page 167.
QuickCode Type From	
To	
QuickCode Type Like	
Save To Archive	
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Partials Allowed	
Report Only	
Compare Only	

Request Groups (Report Groups)

This section discusses migrating request groups, including all of the object-specific parameters on the Submit Request window. Ensure the successful migration of these objects by following the instructions included in this section.

In some releases of Oracle E-Business Suite, request groups are referred to as report groups.

Before Migrating Request Groups

Before migrating request groups, make sure that a number of prerequisite conditions are met. If the following entities are referenced by the request groups in the source database, they must also exist in the destination database prior to the migration.

- Concurrent programs
- Request sets
- Applications

If assigned, the request group code referenced by the request group in the source database must be unique within the application in the destination database.

Request Groups Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Report Groups program.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-17. Request Groups Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Source Application	
Dest Application	
Selection Type	
Specific Report Group	Enter the current name (as named in source database) of the request group which to migrate. This value can be queried from the Request Sets form (named Define Report Groups in some releases). This is not the code associated to the request group. For a description of this common parameter, see *Table B-2* on page 167.
New Report Group Name	For descriptions of these common parameters, see *Table B-2* on page 167.
Report Group From	
To	
Report Group Like	
Save To Archive	
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Partials Allowed	
Report Only	
Compare Only	

Request Sets (Report Sets)

This section discusses migrating request sets (also referred to as report sets), including all of the object-specific parameters on the Submit Request window. Ensure the successful migration of these objects by following the instructions included in this section.

Before Migrating Request Sets

Before migrating request sets, make sure that a number of prerequisite conditions are met. If the following entities are referenced by the request sets in the source database, they must also exist in the destination database prior to the migration.

- Concurrent programs
- Concurrent programs parameters (when the request set defines a default value)
- Request set stage function

Request Sets Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Report Sets program.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-18. Request Sets Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Source Application	The application to which the request set belongs. For a description of this common parameter, see *Table B-2* on page 167.
Dest Application	For descriptions of these common parameters, see *Table B-2* on page 167.
Selection Type	

Table B-18. Request Sets Migrator parameters [continued]

Parameter Name	Description/Notes
Specific Report Set	Enter the current name (as named in source database) of the request set to migrate. Query this value from the Request Sets form (named Define Report Sets in some releases). For a description of this common parameter, see *Table B-2* on page 167.
New Report Set Name	Enter a new name for the request set if renaming is desired. For a description of this common parameter, see *Table B-2* on page 167.
New Report Set Code	When migrating to a release 11/11i instance, use this parameter to change the code of the request set as it is migrated. Oracle requires that the request set code is unique in a given instance; if you are using the migrator to create a copy of an existing request set, use this parameter to change the code for the new request set being created.
Report Set From	For descriptions of these common parameters, see *Table B-2* on page 167.
To	
Report Set Like	
Save To Archive	
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Partials Allowed	
Report Only	
Compare Only	

Responsibilities

This section discusses migrating responsibilities, including all of the object-specific parameters on the Submit Request window. Ensure the successful migration of these objects by following the instructions included in this section.

Before Migrating Responsibilities

Before migrating responsibilities, make sure that a number of prerequisite conditions are met. If the following entities are referenced by the responsibilities in the source database, they must also exist in the destination database prior to the migration.

- Data groups and their applications
- Menus
- Forms (applies to release 10.7 character mode responsibilities only)
- Request groups
- Security groups
- Inventory orgs (applies only if both source and destination are multi-org enabled)
- Functions
- Functions or menus referenced by the responsibility as exclusions

Character mode responsibilities cannot be migrated to any release later than 10.7.

Responsibilities Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Responsibilities program.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-19. Responsibilities Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Source Application	
Dest Application	
Selection Type	
Specific Responsibility	Enter the current name (as named in source database) of the responsibility to migrate. Query this value from the Define Responsibilities form. For a description of this common parameter, see *Table B-2* on page 167.
New Responsibility Name	For descriptions of these common parameters, see *Table B-2* on page 167.
Responsibility From	
To	
Responsibility Like	
Migrate Enabled Resp Only (required)	Select one of the following options: • **Yes:** Migrate only enabled responsibilities. • **No:** Migrate both enabled and disabled responsibilities.

Table B-19. Responsibilities Migrator parameters [continued]

Parameter Name	Description/Notes
Synchronize Workflow Tables	Use this parameter only if the destination is an Oracle 11.5.9 instance. If the instance is not at the 11.5.9 level, this parameter will be ignored. Select one of the following options: • **Yes:** After the migration is complete, populate the workflow local tables automatically for all responsibilities being migrated. • **No:** Do not populate the workflow local tables. This option is recommended for mass migrations. In such cases, you can synchronize using the Synchronize WF Local Tables program, which will populate the workflow local tables for all responsibilities defined in the application.

Table B-19. Responsibilities Migrator parameters [continued]

Parameter Name	Description/Notes
Migrate Sec Attrib Values	In some patch levels of Oracle E-Business Suite release 11i, users can define specific values for securing attributes as part of the responsibility definitions. This parameter governs whether securing attribute values associated with the responsibility in the source will be migrated with the responsibility to the destination. If Migrate Sec Attrib Values is set to **Yes,** security attribute values will be migrated from the source to the destination. Existing securing attribute values in the destination will be removed and replaced. This setting should be used with caution, as noted lates in these notes. Securing attribute values are not validated or re-derived during migration. The vast majority of Securing Attributes values must be entered without any validation in the Responsibilities form. Migrating with Migrate Sec Attrib Values set to **Yes** could thus cause problems if the value is not valid (such as the organization ID for an inventory org) or refers to a different entity in the destination. In addition, Oracle implements the data structures for securing attributes before fully implementing the logic. Migrating securing attribute values from an instance where they are defined, to an instance where the table structures are in place but the logic is not, will define values that cannot be removed in the destination. If Migrate Sec Attrib Values is set to **No,** securing attribute values will not be migrated from the source to the destination. Existing securing attribute values in the destination will remain in place, as long as the responsibility definition still includes the particular Securing Attribute. It is recommended to set this parameter to **No** unless the responsibility for versions in both the source and destination are at version 11.5.17 or higher, and that the attribute values attached to the responsibilities will be valid in the destination. Note: The table where securing attribute values are stored, ak_resp_security_attr_values, is currently delivered by Oracle without indexes defined. If you plan to define securing attributes values in responsibilities, even if you do not plan to migrate them, consider defining indexes on this table to improve migration performance.

Table B-19. Responsibilities Migrator parameters [continued]

Parameter Name	Description/Notes
Save To Archive	For descriptions of these common parameters, see *Table B-2* on page 167.
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Partials Allowed	
Report Only	
Compare Only	For a description of this common parameter, see *Table B-2* on page 167. Note: Currently, the comparison functionality does not compare excluded items or securing attributes when comparing objects.

Users

This section discusses migrating users, including all of the object-specific parameters on the Submit Request window. Ensure the successful migration of these objects by following the instructions included in this section.

Before Migrating Users

Before migrating users, make sure that a number of prerequisite conditions are met. If the following entities are referenced by the users in the source database, they must also exist in the destination database prior to the migration.

- Employees referenced by any new users being migrated
- Responsibilities
- Customers and Suppliers referenced by any new users
- Security Groups
- Securing Attributes

Additionally, user passwords should be reset at the destination instance for the user to be functional.

If customer information is associated with the user, the definition of that information between environments must be consistent. In the 11i release, this definition changed and migrations from the newer definition to the old definition are not allowed.

Users Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Users program.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-20. Users Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Selection Type	
Specific User Name	Enter the current name (as named in source database) of the user which to migrate. Query this value from the Define Users form. For a description of this common parameter, see *Table B-2* on page 167.
New User Name	For a description of this common parameter, see *Table B-2* on page 167. Note: When creating a new user, for security reasons, change the password of the new user once the migration is complete.
User Name From	For descriptions of these common parameters, see *Table B-2* on page 167.
To	
User Name Like	

Table B-20. Users Migrator parameters [continued]

Parameter Name	Description/Notes
Create User (Required)	Specifies whether to create a new user in the destination database. Select one of the following options: • **Yes:** If the user does not exist on the destination database, create the user based on the header information from the source database. Note: If the user already exists, the Users Migrator will not update any of the user header information (password, employee, and so forth). It will only update the user responsibilities and securing attribute information. When creating a new user, for security reasons, change the password of the new user once the migration is complete. • **No:** If the user does not exist on the destination database, set the status of the object to Error.
Active Resp Only (Required)	Determines which responsibilities to migrate. Select one of the following options: • **Yes:** Migrate only the user responsibilities that either do not have an end date or have an end date in the future. • **No:** Migrate all user responsibilities.
Overwrite Resp End Date (Required)	Specifies whether to overwrite responsibilities in the destination database. Select one of the following options: • **Yes:** If the user and the user responsibility already exists on the destination database and the user responsibility has an end date, overwrite the end date with the end date from the source database. • **No:** Do not overwrite the end date if the user responsibility exists on the destination database and has an end date.

Table B-20. Users Migrator parameters [continued]

Parameter Name	Description/Notes
Disable Non-Migrated Resp (Required)	Specifies whether to disable responsibilities associated with the user in the destination database that were not migrated. Select one of the following options: • **Yes:** If the user responsibility exists on the destination database but not on the source database, then set the end date to the current date. • **No:** Do not modify user responsibilities that exist on the destination database but not on the source database.
Migrate Enabled Users Only (Required)	Select one of the following options: • **Yes:** Only migrate enabled users. • **No:** Migrate both enabled and disabled users.

Table B-20. Users Migrator parameters [continued]

Parameter Name	Description/Notes
Migrate Securing Attributes (Required)	Typically, securing attributes values associated with a user are entered without validation in Oracle Applications. However, the following securing attributes are actively managed, as their values are related to other values on the user header record: • ICX_HR_PERSON_ID (related to the Employee associated with the user) • TO_PERSON_ID (also related to the Employee) • ICX_CUSTOMER_CONTACT_ID (related to the Customer) • ICX_SUPPLIER_CONTACT_ID (related to the Supplier). These values are created when the user is created, and are maintained as the information on the user record changes. These values can be manually removed or amended using the Define Users form. Migration handles managed attribute values in the following ways regardless of the parameter value, if the user: • Already exists in the destination, the managed values from the destination are retained. • Does not already exist, the attributes are migrated to the destination and updated with the appropriate values from the user record. • Is being migrated from a release which does not manage the attributes, the attributes will be automatically created and defaulted. Select one of the following options: • **Yes:** Migrate all securing attribute values for the user. Existing values will be removed or replaced. Note: When migrating Securing Attributes, non-managed values are not re-derived or re-validated during migration. This could cause problems if the value (for example, the organization ID for an inventory org) is not valid or refers to a different entity in the destination. • **No:** Migrate only attributes which Oracle Applications actively manages. Other attributes will neither be migrated from the source nor removed from the destination.

Table B-20. Users Migrator parameters [continued]

Parameter Name	Description/Notes
Save To Archive	For descriptions of these common parameters, see *Table B-2* on page 167.
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Partials Allowed	
Report Only	
Compare Only	
Synchronize Workflow Tables	Use this parameter only if the destination is an Oracle 11.5.9 instance. If the instance is not at the 11.5.9 level, this parameter will be ignored. Select one of the following options: • **Yes:** After the migration is complete, populate the workflow local tables automatically for all users being migrated. • **No:** Do not populate the workflow local tables. This option is recommended for mass migrations. In such cases, you can synchronize using the Synchronize WF Local Tables program, which will populate the workflow local tables for all users defined in the instance.

Value Sets

This section discusses migrating value sets, including all of the object-specific parameters on the Submit Request window. Ensure the successful migration of these objects by following the instructions included in this section.

Before Migrating Value Sets

Before migrating value sets, make sure that a number of prerequisite conditions are met. If the following entities are referenced by the value sets in the source database, they must also exist in the destination database prior to the migration.

- Parent value sets
- Validation types
- Security groups

SQL statements used in value sets being migrated should be valid in the destination database.

This is not validated by the Migrate Value Sets program.

Value sets defined with hierarchical security can be migrated only to other instances that support hierarchical security. Otherwise, the migration will error for the value set.

When migrating value sets with hierarchical security, the Compile Value Set Hierarchies program will be submitted in the destination to compile the migrated information. This applies only when the destination instance supports hierarchical security.

Caution should be taken when migrating value sets to make sure that the transaction data relating to the value set in the destination remains valid. The migration replaces the definition in the destination, which can remove existing value set values.

When defining Table-validated value sets, you can specify additional ID and meaning columns. For each column defined, you can specify a data type and size value. As of 11.5.8, the inclusion of size and type information for any specified columns is required. The Object Migrator enforces that this information is specified when migrating to any release 11i instance, since the resulting functional error occurs in all 11i instances.

Value Sets Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Value Sets request.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-21. Value Sets Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Selection Type	
Specific Value Set	Enter the current name (as named in source database) of the value set which to migrate. Query this value from the Define Value Sets form. For a description of this common parameter, see *Table B-2* on page 167.
New Value Set Name	For descriptions of these common parameters, see *Table B-2* on page 167.
Value Set From	
To	
Value Set Like	
Save To Archive	
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Report Only	
Compare Only	

Zooms

This section discusses migrating zooms, including all of the object-specific parameters on the Submit Request window. Ensure the successful migration of these objects by following the instructions included in this section.

Before Migrating Zooms

Before migrating zooms, make sure that a number of prerequisite conditions are met. If the following entities are referenced by the zooms in the source database, they must also exist in the destination database prior to the migration.

- Forms (and their appropriate blocks and fields)

Zooms are not supported in release 11 or 11i; therefore, they can not be migrated to or from those releases.

Zooms Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Zooms program.

Only the object-specific parameters are defined here. For more information on one of the common Object Migrator parameters, see *Table B-2* on page 167.

Table B-22. Zooms Migrator parameters

Parameter Name	Description/Notes
Source Database	For descriptions of these common parameters, see *Table B-2* on page 167.
Dest Database	
Source Application	
Dest Application	
Selection Type	

Table B-22. Zooms Migrator parameters [continued]

Parameter Name	Description/Notes
Specific Zoom Name	Enter the current name (as named in source database) of the zoom which to migrate. Query this value from the Define Zooms form. For a description of this common parameter, see *Table B-2* on page 167.
New Zoom Name	For a description of this common parameter, see *Table B-2* on page 167.
New Zoom Title	Enter the new title of the zoom as it will appear in the destination database after the migration. This gives you the ability to change the title for a zoom. If left blank, Object Migrator will use the title from the source database.
Zoom Name From	For descriptions of these common parameters, see *Table B-2* on page 167.
To	
Zoom Name Like	
Save To Archive	
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Partials Allowed	
Report Only	
Compare Only	For a description of this common parameter, see *Table B-2* on page 167. Note: Currently, the comparison functionality does not compare the value in the Zoom Step field when comparing objects.

Appendix C
Exception Messages

The following is a list of Object Migrator exception messages. Object Migrator exception messages fall into the following categories:

- **Internal Error Messages.** Require action and are generally caused by some system problem and should be passed directly to the system administrator for follow-up with Mercury.
- **Error Messages.** Require action and are generally caused by some form of invalid data and can often be resolved by the user.
- **Warning Messages.** Do not require action but often provide useful information or a reminder to the user.

Message: 1000	Error: Object already exists in version label
Cause	The Object being saved to the object archive already exists in the specified version label.
Action	Either save the Object to another version label or use the Purge Object Archive Versions Detail Report program to remove the object from the version label.
Message: 1100	Error: Concurrent program already exists
Cause	Specified new concurrent program name already exists in the destination environment.

Action	Run the Migrate Concurrent Programs program with Overwrite set to **Yes.**
Message: 1101	Error: Executable type does not exist in destination
Cause	The executable program referenced by the concurrent program is a type of executable not supported in the destination.
Action	Do not migrate this program to the destination until the destination supports the executable type.
Message: 1102	Error: Another concurrent program exists with the same user program name.
Cause	A different concurrent program name with the same user program name as the object being migrated exists in the destination environment.
Action	Change the user program name of the object being migrated.
Message: 1103	Error: Concurrent request class does not exist on destination database.
Cause	Concurrent request class associated with object being migrated does not exist on destination database.
Action	Create request class in destination database.
Message: 1104	Error: Incompatible program does not exist on destination database.
Cause	The concurrent program in migration references an incompatible program name that does not exist in the destination database.
Action	Migrate incompatible concurrent program first or remove the reference to the incompatible program from the concurrent program in source database.
Message: 1105	Error: Value set does not exist on destination database.
Cause	The specified value set referenced by the object in migration must exist on the destination database prior to migration.
Action	Migrate all value sets reference by the concurrent program, prior to migrating the concurrent program.
Message: 1106	Error: Default profile option does not exist on destination database.
Cause	The specified default profile option referenced by the object in migration must exist on the destination database prior to migration.
Action	Setup default profile options referenced by the concurrent program, prior to migrating the concurrent program.

Message: 1200	Error: Report set already exists
Cause	Specified new report set name already exists in destination database
Action	Run Migrate Report Sets program with Overwrite set to **Yes.**
Message: 1201	Error: Concurrent program does not exist on destination database
Cause	The concurrent program referenced by the object in migration does not exist on the destination database.
Action	Migrate the referenced concurrent program prior to migrating the report set.
Message: 1202	Error: Concurrent program parameter does not exist on destination database.
Cause	One of the parameters used by the concurrent program referenced by the report set in migration does not exist on the destination database. Migrating a concurrent program with Partials Allowed set to **Yes** can allow a concurrent program to come over without some of its parameters.
Action	Migrate the referenced concurrent program prior to migrating the report set. Set Partials Allowed should be set to **No.**
Message: 1203	Error: Report Set program does not exist on destination database.
Cause	The report set header is tied to a concurrent program that does not exist in the destination database.
Action	Migrate the referenced concurrent program prior to migrating the report set.
Message: 1204	Error: Value set already exists.
Cause	Specified New value set name already exists in the destination database.
Action	Run the Migrate Value Sets program with Overwrite set to **Yes.**
Message: 1301	Error: Parent value set should co-migrate or exist in the destination.
Cause	In order to migrate a value set that has a parent value set, the parent value set must exist in the destination or be part of the same migration.
Action	Migrate the parent value set first, or at the same time as the current value set.
Message: 1303	Error: Value set with hierarchical security is not supported in the destination hence cannot be migrated.

Cause	The value set being migrated is set up to use hierarchical security, but the destination does not support hierarchical security configurations.
Action	Do not migrate this value set to the destination until the destination supports hierarchical security, or modify the value set to use a security configuration that is supported in the destination.
Message: 1401	Error: Table application does not exist in the destination.
Cause	The application short name which owns the table associated with the descriptive flexfield in migration must exist in the destination database prior to migrating the descriptive flexfield.
Action	Set up application short name in the destination database prior to migrating the descriptive flexfield.
Message: 1402	Error: Desc. Flex Table Does Not Exist or Registered under different App. in Dest.
Cause	The Table owning the descriptive flexfield in migration does not exist in the destination database or is not registered to the same application as in the source database.
Action	Register table under correct application in the destination database.
Message: 1403	Error: Desc. Flex Title is already used by an existing Flex in Destination.
Cause	The destination instance already has a different Descriptive Flex with the same title as the Descriptive Flex in migration. Different Descriptive Flex refers to a Descriptive Flex where the name is different then the new descriptive flex name of the object in migration.
Action	Either change the Title of the Different Descriptive Flex in the destination environment or change the Title of the Descriptive Flex in Source.
Message: 1404	Error: Column used for this Flex is already registered to a different Flex in dest instance.
Cause	A single column may only be registered to one descriptive flexfield. The descriptive flex in migration is registered to a column that is already registered to a different descriptive flex in the destination instance.
Action	Use a different column for either the flexfield which references it in either the source database or the destination database.
Message: 1405	Error: Value set Does Not Exist in the Dest instance.
Cause	The descriptive flexfield in migration references a value set that does not exist in the destination database.

Action	Migrate the referenced value set prior to migrating the descriptive flexfield.
Message: 1406	Error: Profile option used in Flex Does Not Exist or is disabled in Destination.
Cause	The descriptive flexfield in migration references a profile option that does not exist in the destination database.
Action	Set up the referenced profile option prior to migrating the descriptive flexfield.
Message: 1407	Error: Protected Flag in Destination environment is set to **Yes.**
Cause	The descriptive flexfield in migration has been secured against overwrites in the destination instance.
Action	Set the Protected Flag to **No** for that flexfield in the destination database prior to migration.
Message: 1411	Error: Column not registered in destination environment.
Cause	The column referenced by the descriptive flexfield is not registered in the destination.
Action	Register the column in the destination or remove it from the descriptive flexfield definition.
Message: 1500	Error: Report group already exists.
Cause	Specified new report group name already exists in destination database.
Action	Run Migrate Report Groups program with Overwrite set to **Yes.**
Message: 1501	Error: Concurrent Program / Report does not exist on destination database.
Cause	The concurrent program or report referenced by the object in migration does not exist in the destination database.
Action	Migrate the referenced concurrent program or report prior to migrating the report group.
Message: 1502	Error: Report set does not exist on destination database.
Cause	The report set referenced by the object in migration does not exist on the destination database.
Action	Migrate the referenced report set prior to migrating the report group.
Message: 1503	Error: Application does not exist on destination database.

Cause	The application which owns the report set in migration does not exist on the destination database.
Action	Set up application short name in the destination database prior to migrating the descriptive flexfield. Or migrate the object to a different destination application which does exist in the destination database.
Message: 1700	Error: Parent menu already exists.
Cause	Specific menu name already exists in the destination instance.
Action	Run Migrate Menus program with Overwrite set to **Yes.**
Message: 1701	Error: Form does not exist on destination database.
Cause	Form referenced by menu in migration does not exist in the destination instance or is registered under a different application.
Action	If the form does not exist then create it and register under the same application as the in the source database. If the form exists, then it must be registered under the correct owning application.
Message: 1702	Error: Child menu does not exist on destination database.
Cause	Menu in migration references a child menu which does not exist in the destination instance. This can happen under two conditions: 1) Single Level Only was set to **Yes.** The system will not attempt to migrate the child menu from the source instance under this condition, therefore the child menu must exist in the destination instance. 2) Import Standard Oracle Menus was set to **No.** The menu in migration references a child menu which is a standard Oracle menu which has been removed in the destination instance. The program considers a standard Oracle menu any menu which is owned by an Oracle Application.
Action	Three options: 1) Create the missing menu in the destination instance and rerun the migration s before. 2) Rerun the migration program with Single Level Only set to **No.** Use this option with caution. 3) Rerun the migration program with Import Standard Oracle Menus set to **Yes.** Use this option with caution.
Message: 1703	Error: Subroutine does not exist on destination database.
Cause	The Subroutine referenced by the menu in migration does not exist in the destination instance.
Action	Create the Subroutine in the destination instance and rerun the migration.
Message: 1704	Error: Macro does not exist on destination database.

Cause	The Macro referenced by the menu in migration does not exist in the destination instance.
Action	Create the Macro in the destination instance and retry the migration.
Message: 1705	Error: Referenced Form does not exist on destination database.
Cause:	The Form referenced by the menu in migration does not exist in the destination instance.
Action	Create the Form in the destination instance and retry the migration.
Message: 1706	Error: Another menu exists with the same menu title.
Cause	The name of the menu in migration already exists in the destination instance and Overwrite is set to **No.**
Action	Run Migrate Menus program with Overwrite set to **Yes.**
Message: 1800	Error: Zoom definition already exists.
Cause	Specified new zoom name already exists in destination database.
Action	Run Migrate Zooms program with Overwrite set to **Yes.**
Message: 1801	Error: Zoom destination application does not exist in destination database.
Cause	The destination application referenced by the zoom in migration does not exist in the destination database.
Action	Set up the application short name of the referenced application in the destination database. Register the Destination form to this application in the destination instance.
Message: 1802	Error: Zoom Destination form does not exist in destination database.
Cause	The Destination form referenced by the zoom in migration does not exist for the owning application in the destination instance.
Action	Create the form in the destination instance if it does not already exist. Register the form to the owning application in the destination instance. Note: The Form must have the same owning application in the source and destination instances.
Message: 1803	Error: Zoom source application does not exist in destination database.

Cause	The source application referenced by the zoom in migration does not exist in the destination database.
Action	Set up the application short name of the referenced application in the destination instance. Register the Source form to this application in the destination instance.
Message: 1804	Error: Zoom Source form does not exist in destination database.
Cause	The Source form referenced by the zoom in migration does not exist for the owning application in the destination instance.
Action	Create the form in the destination instance if it does not already exist. Register the form to the owning application in the destination instance. Note: The form must have the same owning application in the source and destination instances.
Message: 1805	Error: Zoom source zone does not exist in destination database.
Cause	The source zone referenced by the zoom in migration does not exist on the Zoom Source form in the destination instance.
	Action
Message: 1806	Error: Zoom Source field does not exist in destination database.
Cause	The Source field referenced by the zoom in migration does not exist on the Zoom Source form and zone in the destination instance.
Action	Synchronize the Source form in the two instances. At a minimum, the Zoom Source form, zone, and field for the zoom in migration, must exist in the destination instance.
Message: 1807	Error: Another zoom exists with the same zoom title.
Cause	Zoom title must be unique within a database and application. The destination database and application already has a different zoom defined with the same Title as the zoom in migration.
Action	Rename the title of either the zoom being migrated, or the zoom with the same title in the destination database and application.
Message: 1900	Error: Responsibility already exists.
Cause	Specified new responsibility name already exists in destination database.
Action	Run Migrate Responsibilities program with Overwrite set to **Yes.**

Message: 1901	Error: Data group does not exist in destination database.
Cause	The data group referenced by the responsibility in migration does not exist in the destination database.
Action	Set up the referenced data in the destination database.
Message: 1902	Error: Form does not exist in destination database.
Cause	The first Form referenced by the responsibility in migration does not exist for the owning application in the destination instance.
Action	Create the form in the destination instance if it does not already exist. Register the form to the owning application in the destination instance. Note: The form must have the same owning application in the source and destination instances.
Message: 1903	Error: Menu does not exist in destination database.
Cause	The top menu referenced by the responsibility in migration does not exist in the destination database.
Action	Create the top menu in the destination database.
Message: 1904	Error: Report group does not exist in destination database.
Cause	The report group referenced by the responsibility in migration does not exist in the destination instance.
Action	Create the report group in the destination database.
Message: 1905	Error: Data group application does not exist in destination database.
Cause	The data group application referenced by the responsibility does not exist in the destination database.
Action	Add the reference application to the data group in the destination database.
Message: 2000	Error: User does not exist on the destination database.
Cause	The Users Migrator is being run with Create User set to **No** and the user does not exist on the destination database.
Action	Run the migrator with Create User set to **Yes** or manually create the user on the destination database before rerunning the migration.
Message: 2001	Error: Employee does not exist on the destination database.

Cause	The employee referenced by the new user being migrated does not exist in the destination database.
Action	Enter the employee on the destination database before rerunning the migration.
Message: 2002	Error: Responsibility does not exist on the destination database.
Cause	User references a responsibility that does not exist on the destination database.
Action	Create the responsibility on the destination database before rerunning the migration.
Message: 2100	Error: Profile option already exists.
Cause	Specified new profile option name already exists in destination database.
Action	Run Migrate Profile Options program with Overwrite set to **Yes.**
Message: 2101	Error: Another profile exists with the same user profile name.
Cause	User profile name must be unique within a database. The destination database has a different profile option defined with the same user name as the profile option in migration.
Action	Rename the user name of either the profile option being migrated, or the profile option with the same user name in the destination database.
Message: 2102	Error: Application does not exist in destination database.
Cause	The application referenced by the profile option value does not exist in the destination database.
Action	Set up the referenced application in the destination database.
Message: 2103	Error: Responsibility does not exist in destination database.
Cause	The responsibility referenced by the profile option value does not exist for the owning application in the destination instance.
Action	Create the responsibility in the destination instance.
Message: 2104	Error: User does not exist in destination database.
Cause	The user referenced by the profile option value does not exist in the destination database.
Action	Create the user in the destination database.

Message: 2105	Error: Invalid Profile Level.
Cause	The profile option value references a profile value not recognized by the Profile Options Migrator.
Action	Change the level to one of the following: site, application, responsibility, server (if supported), organization (if supported), or user.
Message: 2107	Error: Security profiles cannot be migrated to instances that do not support this hierarchy.
Cause	The destination does not support hierarchy types, but the profile being migrated requires support for the security hierarchy type.
Action	Do not migrate this profile option to this instance.
Message: 2108	Error: Organization profiles cannot be migrated to instances that do not support this hierarchy.
Cause	The destination does not support hierarchy types, but the profile being migrated requires support for the organization hierarchy type.
Action	Do not migrate this profile option to this instance.
Message: 2109	Error: Server does not exist on destination database. The server referenced by the profile option value does not exist in the destination database.
Cause	Object Migrator allows migration of values for profiles using the server hierarchy. Migrating profile option values at the server level may produce Missing Reference errors unless the same servers are shared across Oracle Applications instances, since their server names would differ.
Action	Define the server in the destination database prior to migrating profile option values or do not migrate values for this Profile.
Message: 2110	Error: Organization does not exist on destination database.
Cause	The organization (operating unit) referenced by the profile option value does not exist in the destination database.
Action	Define the organization in the destination database prior to migrating profile option values.
Message: 2111	Error: Another profile exists with the same profile name.

Cause	Profile option names must be unique within a database. The destination database has a profile option belonging to a different application that uses the same profile option name.
Action	If the information in the destination is correct for the profile option being migrated, resubmit the migration, setting Dest Application to the correct value. Otherwise, change the name of the profile being migrated.
Message: 2200	Error: Help text already exists.
Cause	Help text already exists for the specified form in destination database.
Action	Run Migrate Help Text program with Overwrite set to **Yes.**
Message: 2201	Error: Application does not exist in destination database.
Cause	The application referenced by the help text does not exist in the destination database.
Action	Set up the referenced application in the destination database.
Message: 2202	Error: Form does not exist in destination database.
Cause	The Form referenced by help text does not exist for the owning application in the destination instance.
Action	Register the Form in the destination instance.
Message: 2203	Error: Block does not exist in destination database.
Cause	The block name referenced by the help text does not exist for the owning form in the destination database.
Action	Verify that the form registered in the destination database is the same as the form registered in the source database.
Message: 2204	Error: Field does not exist in destination database.
Cause	The field name referenced by the help text does not exist for the owning form and block in the destination database.
Action	Verify that the form registered in the destination database is the same as the form registered in the source database.
Message: 2300	Error: QuickCode type already exists.
Cause	Specified new QuickCode type already exists in destination database.
Action	Run Migrate QuickCodes program with Overwrite set to **Yes.**

Message: 2301	Error: Customization level does not exist on destination database.
Cause	The customization level specified in the QuickCode type does not exist in the destination database.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 2302	Error: Using application does not exist on destination database.
Cause	The QuickCode type is being used by an application that does not exist in the destination database.
Action	Create the application in the destination database or run the Migrate QuickCodes program with Partials Allowed set to **Yes.**
Message: 2400	Error: Named SQL already exists.
Cause	Specified new named SQL already exists in destination database.
Action	Run Migrate Named SQL program with Overwrite set to **Yes.**
Message: 2500	Error: Row/Col Set already exists.
Cause	Specified New Row/Col Set already exists in destination database.
Action	Run Migrate FSG Row/Column Set program with Overwrite set to **Yes.**
Message: 2501	Error: Flexfield structure does not exist on destination database.
Cause	Row/Col Set references a flexfield structure that does not exist in destination database.
Action	Create the flexfield structure in destination database and rerun the migrator program.
Message: 2502	Error: Standard Axes does not exist on destination database.
Cause	Row/Col Set references a Standard Axes that does not exist in destination database.
Action	Create the Axes in destination database and rerun the migrator program.
Message: 2600	Error: Printer type already exists on destination database.
Cause	Specified printer type already exists in destination database.
Action	Run Migrate Printer Definitions program with Overwrite set to **Yes.**

Message: 2601	Error: Printer already exists on destination database.
Cause	Specified printer already exists in destination database.
Action	Run Migrate Printer Definitions program with Overwrite set to **Yes.**
Message: 2602	Error: Printer Information already exists on destination database.
Cause	Specified printer information (assignment) already exists in destination database.
Action	Run Migrate Printer Definitions program with Overwrite set to **Yes.**
Message: 2603	Error: Printer style already exists on destination database.
Cause	Specified printer style already exists in destination database.
Action	Run Migrate Printer Definitions program with Overwrite set to **Yes.**
Message: 2604	Error: Printer Driver already exists on destination database.
Cause	Specified printer Driver already exists in destination database.
Action	Run Migrate Printer Definitions program with Overwrite set to **Yes.**
Message: 2605	Error: Another printer driver exists with the same user name.
Cause	User names must be unique across printer drivers. A driver with a different code already exists and has the same name as the driver being validated.
Action	Change the user name of the existing printer driver or the new driver.
Message: 2606	Error: Platform is not defined in destination database.
Cause	Printer Driver refers to a platform not defined on the destination database.
Action	Add the platform to the destination database using the Define Special QuickCodes form.
Message: 2607	Error: Printer Subroutine does not exist on destination database.
Cause	Row/Col Set references a Set of Books that does not exist in destination database.
Action	Create the Set of Books in destination database and rerun the migrator program.

Message: 2608	Error: Another printer style exists with the same user name.
Cause	User names must be unique across printer styles. A style with a different code already exists and has the same name as the style being validated.
Action	Change the user name of the existing printer style or the new style.
Message: 2609	Error: Another printer style exists with the same sequence.
Cause	Sequences must be unique across printer styles. A style with a different code already exists and has the same sequence as the style being validated.
Action	Change the sequence of the existing printer style or the new style.
Message: 2700	Error: Folder already exists on destination database.
Cause	Specified Folder already exists in destination database.
Action	Run Migrate Folders program with Overwrite set to **Yes.**
Message: 2701	Error: Responsibility does not exist on destination database.
Cause	Folder references a responsibility that does not exist in destination database.
Action	Create the responsibility in destination database and rerun the migrator program.
Message: 2702	Error: User does not exist on destination database.
Cause	Folder references a user that does not exist in destination database.
Action	Create the user in destination database and rerun the migrator program.
Message: 2703	Error: Owner does not exist on destination database.
Cause	Folder is owned by an applications user that does not exist in destination database.
Action	Create the user in destination database and rerun the migrator program.
Message: 2704	Error: Another folder exists with the same owner and segment.
Cause	Another folder exists on the destination database with the same owner and the same referenced segment.
Action	Delete the other folder and rerun the migrator program.

Message: 2800	Error: Parent menu already exists on destination database.
Cause	Menu already exists in destination database.
Action	Run Migrate GUI Menus program with Overwrite set to **Yes.**
Message: 2801	Error: Menu does not exist on destination database.
Cause	Parent menu references a menu that does not exist in destination database.
Action	Create the menu in destination database and rerun the migrator program or run the migrator program with Single Level set to **No.**
Message: 2802	Error: Function does not exist on destination database.
Cause	Menu references a form function that does not exist in destination database.
Action	Create the function in destination database and rerun the migrator program.
Message: 2803	Error: Child menu does not exist on destination database.
Cause	Parent menu references a menu that does not exist in destination database.
Action	Create the menu in destination database and rerun the migrator program or run the migrator program with Single Level set to **No.**
Message: 2804	Error: Another menu exists with the same menu title.
Cause	Menu title must be unique across menus. A menu with a different name exists that has the same menu title.
Action	Change the menu title of the existing menu or the menu being validated.
Message: 2805	Error: Menu type does not exist in the destination database where menu type can be: **Home Page, Data Security, Standard, HTML Tab,** or **Unknown.**
Cause	The given type menu is not defined in the destination database.
Action	You are probably migrating the menu from a lower version of Oracle Applications to a higher version. Make sure the particular type is available in the destination database.
Message: 2900	Error: Function already exists on destination database.
Cause	Specified function already exists in destination database.
Action	Run Migrate Functions program with Overwrite set to **Yes.**

Message: 2901	Error: Form application does not exist on destination database.
Cause	Function references a form that references an application that does not exist in destination database.
Action	Create the application in destination database and rerun the migrator program.
Message: 2902	Error: Form does not exist on destination database.
Cause	Function references a form that does not exist in destination database and there is not enough information to create the form.
Action	Create the form in destination database and rerun the migrator program.
Message: 2903	Error: Another function exists with the same user name.
Cause	User name must be unique across functions. A function with a different name exists that has the same user name.
Action	Change the user name of the existing function or the function being validated.
Message: 2904	Error: Object does not exist at the destination.
Cause	Function references an object that does not exist in the destination. The object could be any one of RESOURCES, ORGANIZATION, PROJECTS, TASKS, Contract, Program, and so forth.
Action	Create all the missing objects referenced by the function at the destination.
Message: 2905	Error: Region application does not exist at the destination.
Cause	Function references a region and the region does not exist or is with a different application at the destination.
Action	Create all the regions referenced by the function under the same application as in source.
Message: 2906	Error: Region code with application does not exist at the destination.
Cause	Function references a region through region code that does not exist at the destination.
Action	Create all the regions with correct region code referenced by the function at the destination. AKLOAD utility may be used for migrating region details from source to the destination.

Message: 2907	Error: Maintenance Mode Support value <*Value*> is not valid in the destination.
Cause	The destination requires Maintenance Mode Support information on functions, but does not support the Maintenance Mode Support value specified on the function in the source instance.
Action	Change the Maintenance Mode Support value so that it is supported in the destination, or get the destination to the same patch level before migrating the function.
Message: 2908	Error: Context dependence value <*Value*> is not valid in the destination.
Cause	The destination requires context dependence information on functions, but does not support the context dependence support value specified on the function in the source instance.
Action	Change the context dependence value so that it is supported in the destination, or get the destination to the same patch level before migrating the function.
Message: 2909	Error: Type value <*Value*> is not valid in the destination.
Cause	The destination validates type information on functions, but does not support the type value specified on the function in the source instance.
Action	Change the type value so that it is supported in the destination, or get the destination to the same patch level before migrating the function. Validate that the value is valid in the source.
Message: 3000	Error: Message already exists on destination database.
Cause	Specified Message already exists in destination database.
Action	Run Migrate Messages program with Overwrite set to **Yes.**
Message: 3001	Error: Application does not exist on destination database.
Cause	Message is for an application that does not exist in destination database.
Action	Create the application in destination database and rerun the migrator program.
Message: 3002	Error: Language code does not exist on destination database.
Cause	Message references a language code that does not exist in destination database.

Action	Enter the appropriate language code in FND_LANGUAGES in destination database and rerun the migrator program.
Message: 3003	Error: Message category does not exist in the destination lookup.
Cause	Message being migrated has a category that is missing in the destination instance.
Action	Update the message category in the source, and then try migrating again.
Message: 3004	Error: Message severity does not exist in the destination lookup.
Cause	Message being migrated has a severity that is missing in the destination instance.
Action	Update the message severity in the source, and then try migrating again.
Message: 3100	Error: Concurrent Manager already exists on destination database.
Cause	Specified Concurrent Manager already exists in destination database.
Action	Run Migrate Concurrent Manager Definitions program with Overwrite set to **Yes.**
Message: 3101	Error: Workshift already exists on destination database.
Cause	Specified work shift already exists in destination database.
Action	Run Migrate Concurrent Manager Definitions program with Overwrite set to **Yes.**
Message: 3102	Error: Combined Specialization Rule already exists on destination database.
Cause	Specified Specialization Rule already exists in destination database.
Action	Run Migrate Concurrent Manager Definitions program with Overwrite set to **Yes.**
Message: 3103	Error: Program library does not exist on destination database.
Cause	Concurrent Manager references a program library that does not exist in destination database.
Action	If a custom program Concurrent Manager executable was created, then install it in the destination database. If the missing library is a standard library, then contact Oracle Support to see why it is missing from the destination database.

Message: 3104	Error: Concurrent Manager is active in the destination database.
Cause	The Concurrent Manager Definitions Migrator will not migrate an existing Concurrent Manager Definition if the manager is currently active in the destination database.
Action	Deactivate the Concurrent Manager on the destination database or make sure it does not have a current work shift.
Message: 3105	Error: Combined Rule / Oracle Id / Concurrent Program / Request Type / User does not exist on the destination database.
Cause	A specialization rule references an AOL object that does not exist on the destination database.
Action	Create the appropriate AOL object in the destination database or run the Migrate Concurrent Manager Definitions program with Partials Allowed set to **Yes** (in which case the specialization rule will not get migrated).
Message: 4100	Warning: Printer does not exist on destination database.
Action	Create printer on destination database and assign it to appropriate concurrent program.
Message: 4101	Warning: Printer style does not exist on destination database.
Action	Create printer style on destination database and assign it to appropriate concurrent program.
Message: 7100	Internal Error: Error getting CONCURRENT_PROGRAM_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 7101	Internal Error: Error getting EXECUTABLE_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 7200	Internal Error: Error getting REQUEST_SET_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 7201	Internal Error: Error getting REQUEST_SET_PROGRAM_ID.
Action:	Contact Mercury Support. (http://support.mercury.com)

Message: 7207	Error: Another Request Set has the same translation for the User Request Set Name at the Destination.
Action:	Check the data.
Message: 7208	Error: Another Request Set has the same translation for the User Request Set Name at the Destination.
Action:	Check the data.
Message: 7300	Internal Error: Error getting FLEX_SET_VALUE_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 7301	Internal Error: Error getting FLEX_VALUE_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 7311	Error: Null value found for type or size of ID column in the value set definition.
Cause	Table-validated value set being migrated has an ID column specified, but the related type or size information is missing.
Action	Update the value set in the source instance by adding the related type or size information for the ID column, and then try migrating again.
Message: 7312	Error: Null value found for type or size of meaning column in the value set definition.
Cause	Table-validated value set being migrated has a meaning column specified, but the related type or size information is missing.
Action	Update the value set in the source instance by adding the related type or size information for the meaning column, and then try migrating again.
Message: 7313	Error: Flexfield Structure for the new value does not exist in destination.
Cause	When copying value set with value attributes, if the new value set name does not exist in DFF Flexfield Segment Values an error is generated. User must add an entry into the DFF Flexfield Segment Values for the new value set before copying a value set with value attributes into a new value set.
Message: 7400	Internal Error: Error getting APPLICATION_ID.
Action	Contact Mercury Support. (http://support.mercury.com)

Message: 7406	Error: Context override value set does not exist at the destination or the destination database does not have support for context override value set.
Cause	This error happens under one of the following circumstances: • The descriptive flexfield references a value set for context override value set that does not exist at the destination. • The descriptive flexfield being migrated has context override value set properties set and the destination version of Oracle Applications is lower than 11.5.5, which does not support context override value set features.
Action	Create the value set referenced by the descriptive flexfield at the destination.
Message: 7500	Internal Error: Error getting REQUEST_GROUP_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 7700	Internal Error: Error getting MENU_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 7800	Internal Error: Unable to get ZOOM_DEFINTION_ID in destination database.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 7801	Internal Error: Unable to get ACTION_SEQUENCE _ID in destination database.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 7900	Internal Error: Unable to get RESPONSIBILITY _ID in destination database.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 8000	Internal Error: Unable to get USER_ID in destination database.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 8100	Internal Error: Unable to get PROFILE_OPTION_ID in destination database.
Action	Contact Mercury Support. (http://support.mercury.com)

Message: 8200	Internal Error: Unable to get PROFILE_OPTION_ID in destination database.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 8300	Internal Error: Error getting APPLICATION_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 8400	Internal Error: Error getting APPLICATION_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 8500	Internal Error: Error getting AXIS_SET_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 8502	Internal Error: Error getting APPLICATION_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 8502	Internal Error: Error getting EXCEPTION_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 8600	Internal Error: Error getting PRINTER_TYPE.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 8700	Internal Error: Error getting FOLDER_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 8800	Internal Error: Error getting MENU_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 8801	Internal Error: Error getting MENU_ID for child menu.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 8900	Internal Error: Error getting FUNCTION_ID.
Action	Contact Mercury Support. (http://support.mercury.com)

Message: 9100	Internal Error: Error getting CONCURRENT_QUEUE_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 9101	Internal Error: Error getting CONCURRENT_TIME_PERIOD_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 9102	Internal Error: Error getting COMPLEX_RULE_ID.
Action	Contact Mercury Support. (http://support.mercury.com)
Message: 9103	Internal Error: Error getting COMPLEX_RULE_LINE_ID.
Action	Contact Mercury Support. (http://support.mercury.com)

Index

M

N

O

P

Q

R

U

V

Z