

Kintana Object* Migrator User's Guide

Version 5.1

Publication Number: OMUsers-1203A

Kintana Object*Migrator, Version 5.1

This manual, and the accompanying software and other documentation, is protected by U.S. and international copyright laws, and may be used only in accordance with the accompanying license agreement. All rights reserved.

Mercury Interactive, the Mercury Interactive logo, and Object*Migrator are trademarks or registered trademarks of Mercury Interactive Corporation or its wholly owned subsidiary Mercury Interactive (Israel) Ltd. in the United States and/or other countries.

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

Mercury Interactive Corporation

1325 Borregas Avenue

Sunnyvale, CA 94089 USA

Main Telephone: (408) 822-5200

Sales & Information: (800) 837-8911

Customer Support: (877) 837-8457

Fax: (408) 822-5300

Home Page: www.mercuryinteractive.com

Customer Support: support.mercuryinteractive.com

© 1997-2003 Mercury Interactive Corporation. All rights reserved.

Contents

Chapter 1	
Introduction	7
Release Overview	7
Document Overview	8
Intended Audience and Prerequisite Knowledge.....	8
Document Conventions	10
Additional Resources	11
Related Documentation	11
Object* Migrator Documentation	11
Accelerator Documentation	12
Administration Documentation.....	12
Change Management and Configuration Documentation.....	13
Customer Support	14
Chapter 2	
Key Concepts	15
Migration Overview	15
Object Types	16
Migration Capabilities	17
Version Control	18
Object Archive.....	18
Comparison Reports	19
Migration Audit Reports	20
Chapter 3	
Accessing Object*Migrator	21
Installing Object*Migrator.....	21
Obtaining a User Name and Password	21
Opening Oracle Applications.....	22
Chapter 4	
Migrating Objects	23

Running the Object*Migrator Request.....	23
Reviewing the Prerequisites for the Object	23
Running the Report to Migrate the Object.....	24
Viewing the Migration Results.....	28

Chapter 5

Object Type-Specific Migrators and Migration Rules	31
Common Migrator Parameters	32
Concurrent Managers	40
Before Migrating Concurrent Managers	40
Concurrent Managers Migrator Parameters	42
Concurrent Programs.....	43
Before Migrating Concurrent Programs	43
Concurrent Programs Migrator Parameters	44
Descriptive Flexfields	46
Before Migrating Descriptive Flexfields	46
Descriptive Flexfields Migrator Parameters	48
Folders	50
Before Migrating Folders	50
Folders Migrator Parameters	51
FSG Row/Column Sets.....	52
Before Migrating FSG Row/Column Sets.....	53
FSG Row/Column Sets Migrator Parameters	53
Functions	54
Before Migrating Functions	55
Functions Migrator Parameters	56
GUI Menus.....	57
Before Migrating GUI Menus.....	57
GUI Menus Migrator Parameters	58
Help Text	60
Before Migrating Help Text	61
Help Text Migrator Parameters	61
Menus (Character Mode).....	62
Before Migrating Menus	63
Menus Migrator Parameters	63
Messages.....	65
Before Migrating Messages.....	66
Messages Migrator Parameters	66
Named SQL	68
Before Migrating Named SQL.....	68
Named SQL Migrator Parameters.....	68

Printer Definitions	69
Before Migrating Printer Definitions	70
Printer Definitions Migrator Parameters	70
Profile Options	72
Before Migrating Profile Options	72
Profile Options Migrator Parameters	74
QuickCodes (AOL)	76
Before Migrating QuickCode	76
QuickCodes Migrator Parameters	76
Request Groups (Report Groups)	77
Before Migrating Request Groups	78
Request Groups Migrator Parameters.....	78
Request Sets (Report Sets)	80
Before Migrating Request Sets	80
Request Sets Migrator Parameters.....	80
Responsibilities	82
Before Migrating Responsibilities	82
Responsibilities Migrator Parameters.....	82
Users	85
Before Migrating Users	85
Users Migrator Parameters.....	86
Value Sets	89
Before Migrating Value Sets	90
Value Sets Migrator Parameters	91
Zooms	92
Before Migrating Zooms	92
Zooms Migrator Parameters	92
 Chapter 6	
Using Object*Migrator	
Version Control	95
Archiving AOL Object Definitions	95
Saving an Object to the Object Archive	96
Retrieving an Object from the Object Archive.....	97
Running the Object Archive Version Detail Report	98
Purging the Object Archive	99
 Chapter 7	
Comparing Objects	103
Object Comparison Overview	103
Only in Source DB.....	104
Only in Dest DB	104

Differences Exist	104
Running the Object Comparison.....	105
Comparing Objects from Two Oracle Instances	105
Comparing Objects in a Database Instance and the Object Archive	106
Comparing Two Versions of an Object in the Object Archive.....	107
Sample Comparison Report	108
Appendix A	
Exception Messages	111
Appendix B	
AOL Dependent Objects	133
Appendix C	
Using Object*Migrator with	
Kintana Deliver.....	137
Introduction to Kintana Deliver.....	137
Executing Object*Migrator from Kintana Deliver.....	138
Processing the Package in Kintana Deliver	139

Chapter 1 Introduction

Kintana Object*Migrator™ manages Oracle E-Business Suite™ release 10, 11, and 11i Application Object Library (AOL) objects across multiple databases to support mission-critical business applications. Concurrent programs allow users to move objects between instances while maintaining database integrity. Customizations and configurations developed in one instance can be transferred to one or all other instances. Objects can be placed under version control in an archive, providing auditing, rollback, and recovery capabilities. Data can be migrated across different releases of Oracle E-Business Suite, simplifying both upgrades and instance consolidations. With Kintana Object*Migrator, users are able to find out what has changed and what needs to change, thus saving time implementing and maintaining the AOL setups that support critical business processes.

Release Overview

Release 5.1 of Kintana Object*Migrator extends support of Oracle E-Business Suite through release 11.5.9 and the FND.G Minipack. Object*Migrator 5.1 integrates with release 5.0 of the Kintana Product Suite.

For a list of new features in Kintana Object*Migrator 5.1, and information about upgrading to Object*Migrator 5.1, see *Kintana Object*Migrator Upgrade Guide*.



Note

For the most up-to-date information about Object*Migrator 5.1, see the 5.1 Release Notes on the Mercury Interactive Download Center at <http://ITG.MERC-INT.COM/support/download/login.jsp>.

Document Overview

This document provides information for maintaining and configuring Kintana Object*Migrator, and for auditing Object*Migrator executions. It covers the following specific topics:

- *Key Concepts*
- *Accessing Object*Migrator*
- *Migrating Objects*
- *Object Type-Specific Migrators and Migration Rules*
- *Using Object*Migrator Version Control*
- *Comparing Objects*
- *Exception Messages*
- *AOL Dependent Objects*
- *Using Object*Migrator with Kintana Deliver*



Note

The content of this documentation reflects Oracle Release 11i forms. Release 10.7 or 11 forms may vary in appearance.

Intended Audience and Prerequisite Knowledge

This intended audience for this document includes the following:

- Oracle Applications users or developers who maintain AOL configurations across multiple database instances
- Database or applications administrators responsible for setting up the database schema and maintaining one or more Oracle Applications instances
- Database or application administrators responsible for maintaining access and security for Kintana Object*Migrator, or supporting Object*Migrator usage

To use Kintana Object*Migrator, you need to:



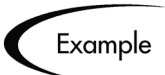
- Understand the key concepts listed and described in *the “Key Concepts”* chapter on page 15

- Be familiar with using Oracle Applications, including submitting concurrent requests
- Understand any AOL objects you will be migrating

Document Conventions

Table 1-1 lists the conventions used in this document.

Table 1-1. Document Conventions

Convention	Description	Example
Button, menu, tabs	Names of interface components that can be clicked (such as buttons, menus, and tabs) are shown in bold.	Apply button
Fields, Windows, Pages	Names of windows, fields, and pages are shown as displayed.	New Request window
Code	Code input and output are shown as displayed.	CauchoConfigFile C:/ITG_Home/conf/ resin.conf
<i>Link</i>	Linked URLs, filenames, and cross references are shown as blue italicized text.	www.merc-int.com
<i>Variable</i>	Variables are shown as italicized text.	<i>ITG_Home</i> /bin directory
Note	Notes contain additional information.	
Caution	Cautions contain important information and instructions. Failure to follow the instructions may result in loss of data.	
Example	Examples contain samples of related procedures.	

Additional Resources

Mercury Interactive provides the following additional resources to help you successfully install, upgrade, and use Kintana Object*Migrator. As indicated below, some of the listed documentation may be of interest to you and accessible to you only if you have purchased additional Mercury Interactive or Kintana products. This section contains the following topics:

- [Related Documentation](#)
- [Customer Support](#)

Related Documentation

The Kintana Library includes additional documents related to the topics discussed in this manual. The following related documentation might be helpful to anyone installing, configuring, and using Kintana Object*Migrator. This section includes the following categories of documentation:

- [Object*Migrator Documentation](#)
- [Accelerator Documentation](#)
- [Administration Documentation](#)
- [Change Management and Configuration Documentation](#)

Object*Migrator Documentation

The following are additional documents in the Kintana Object*Migrator documentation set. You can access this documentation (after you have purchased Object*Migrator) through the Mercury Interactive Download Center at <http://ITG.MERC-INT.COM/support/download/login.jsp>:

<i>Object*Migrator Installation Guide</i>	Provides system requirements and instructions for installing Kintana Object*Migrator.
<i>Object*Migrator Upgrade Guide</i>	Provides system requirements and instructions for upgrading Kintana Object*Migrator from one product release to another.

*Object*Migrator System
Administration Guide*

Provides an architectural overview of Kintana Object*Migrator and instructions for maintaining the product. This document also includes instructions for setting up optional configurations and product features.

Accelerator Documentation

The following manuals document Kintana Accelerators in the Oracle environment. You can access these manuals in the Kintana Library (if you have purchased the relevant products):

- Through the Kintana online help

To access the online help, select **Help > Contents and Index** from the Kintana menu bar.

- Through the Mercury Interactive Download Center:

<http://ITG.MERC-INT.COM/support/download/login.jsp>

*Accelerator for Oracle
Applications*

Documents Kintana Accelerator for Oracle E-Business Suite, which helps to automate change management in the Oracle E-Business Suite environment.

*Accelerator for Oracle
Technology*

Documents Kintana Accelerator for Oracle Technology, which helps to automate change management in application environments built using Oracle tools like SQL*Plus, PL/SQL, Pre-compilers, Oracle Forms, and Oracle Reports.

Administration Documentation

The following manuals might be useful to system, database, and application administrators working with Kintana Object*Migrator. You can access the manuals (if you have purchased the relevant Kintana products):

- Through the Kintana online help

To access the online help, select **Help > Contents and Index** from the Kintana menu bar.

- Through the Mercury Interactive Download Center:

<http://ITG.MERC-INT.COM/support/download/login.jsp>

<i>Kintana Installation</i>	Documents the requirements and procedures for installing products in Kintana Product Suite.
<i>Kintana Upgrade</i>	Documents the requirements and procedures for upgrading a products in Kintana Product Suite. Includes information about new features, upgrade impacts, and upgrade process.
<i>Kintana Migration</i>	Documents how to do instance administration.
<i>Kintana System Administration</i>	Documents how to implement, configure, and maintain Kintana Servers.

Change Management and Configuration Documentation

The following manuals might be useful to system, database, and application administrators working with Kintana Object*Migrator. You can access the documents (if you have purchased the relevant Kintana products):

- Through the Kintana online help

To access the online help, select **Help > Contents and Index** from the Kintana menu bar.

- Through the Mercury Interactive Download Center:

<http://ITG.MERC-INT.COM/support/download/login.jsp>

<i>Processing Packages (Kintana Deliver)</i>	Provides instructions for processing Packages.
--	--

<i>Configuring a Deployment System</i>	Provides instructions for modeling processes in the Kintana Product Suite. Contains process overviews, implementation instructions, and detailed examples.
<i>Using Commands and Tokens in Kintana</i>	Provides reference information about using Commands and Tokens in Kintana Product Suite.
<i>Kintana Security Model</i>	Provides information about the Kintana data security model and instructions for controlling access to different types of entities.

Customer Support

You can access customer support for Kintana products and also additional product information from the Mercury Interactive Support Web site, located at <http://support.mercuryinteractive.com>.

Chapter 2 Key Concepts

This chapter describes the key concepts and definitions related to using Kintana Object*Migrator. Additionally, this chapter provides an overview of how Object*Migrator moves data between instances.

This chapter covers the following topics:

- *Migration Overview*
- *Object Types*
- *Migration Capabilities*
- *Version Control*
- *Comparison Reports*
- *Migration Audit Reports*

Migration Overview

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 2-1 on page 16*. Manually re-keying these configurations would be both time-consuming and error-prone.



Figure 2-1 Migration Overview

Kintana 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.

Object Types

Kintana 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

Kintana 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
- Migration simulation to identify issues with data setups without actually migrating data

Version Control

You can use Kintana 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.

Object Archive

The Object Archive is the version control repository used by Kintana 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 2-2 on page 19](#).

<u>Object*Migrator Executions</u>				
<u>Run #</u>	<u>Object Type</u>	<u>Application</u>	<u>Object</u>	<u>Version Label</u>
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

<u>Status of Object Archive after Object*Migrator executions</u>				
<u>Version Label</u>	<u>Object Type</u>	<u>Application</u>	<u>Object</u>	<u>Rev</u>
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 2-2 Example of Version Label and Revision Number Usage

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 versus the version already present in the destination database. Object*Migrator generates a comparison report showing the differences between the two versions.



Note

Long fields and translated data (for instances running under MLS) are not compared in the comparison reports.



Note

Some child entities are not compared in the comparison reports. Major exceptions are noted in the “*Object Type-Specific Migrators and Migration Rules*” chapter on page 31.

Migration Audit Reports

Kintana 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

Accessing Object*Migrator

This chapter describes how to access the Object*Migrator application installed at your site.

This chapter covers the following topics:

- *Installing Object*Migrator*
- *Obtaining a User Name and Password*
- *Opening Oracle Applications*

Installing Object*Migrator

Before using Kintana 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 *Kintana Object*Migrator Installation Guide* 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 Kintana 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 Application instances:

1. Log on to Oracle Applications.
2. Open the Submit Requests screen.
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 [“Migrating Objects”](#) on page 23.



Note

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

Chapter

4

Migrating Objects

This chapter describes how to run the requests used to migrate objects between Oracle E-Business Suite instances. It includes detailed instructions for executing the report and viewing the migration results.

This chapter covers the following topics:

- [Running the Object*Migrator Request](#)
- [Viewing the Migration Results](#)

Running the Object*Migrator Request

Kintana Object*Migrator migrations are run through the Submit Request screen 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 [“Object Type-Specific Migrators and Migration Rules”](#) on page 31 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 [“AOL Dependent Objects”](#) on page 133.



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 validate that all prerequisites are satisfied to migrate a specific object type, use Kintana 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 screen and the Object*Migrator programs.

2. Open the Submit Requests screen.
 - 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.

The Object*Migrator requests follow the naming convention *Migrate ObjectTypeName*. For example, the object for moving Concurrent Programs is *Migrate Concurrent Programs*.



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

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.

The screenshot shows a 'Parameters' dialog box with the following fields and values:

Source Database	785	VIS5 (11i.5) Database - std
Dest Database	775	VIS3 (11i.3) Database - std
Source Application	ACME Accounts Payable	
Dest Application	ACME Accounts Payable	
Selection Type	1	Specific Object
Specific Conc Program	XXAPINVMT	
New Conc Program Short Name	XXAPINVMT	
New User Program Name		
Conc Program From		
Conc Program To		
Conc Program Like		
Save To Archive	2	No
Version Label		
Version Desc		
Recover From Archive	2	No
From Version Label		
Overwrite if Exists	1	Yes
Partials Allowed	2	No
Report Only	2	No
Compare Only	2	No

At the bottom of the dialog are buttons for OK, Cancel, Clear, and Help.

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 the *“Object Type-Specific Migrators and Migration Rules”* chapter on page 31.



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:

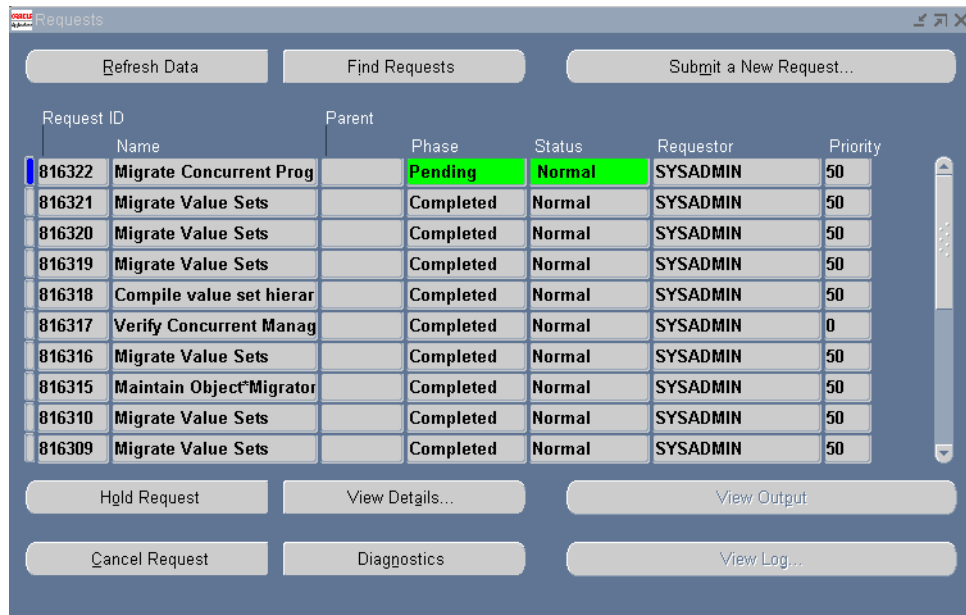
- **Specific Object:**
One object will be selected for migration at this time. The specific object name will need to be entered.
- **Range of Objects:**
A range of objects will be selected for migration at this time. The alphabetic range to use will need to be specified.
- **Wild Card Match**
Multiple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



Request ID	Name	Parent	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 Object*Migrator		Completed	Normal	SYSADMIN	50
816310	Migrate Value Sets		Completed	Normal	SYSADMIN	50
816309	Migrate Value Sets		Completed	Normal	SYSADMIN	50

Viewing the Migration Results

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



Note

When migrating multiple objects (range or wildcard), 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.

- From the menu, select **View > Requests**.

The Find Requests window opens.

Find Requests

My Completed Requests

My Requests In Progress

All My Requests

Specific Requests

Request ID

Name

Date Submitted

Date Completed

Status

Phase

Requestor

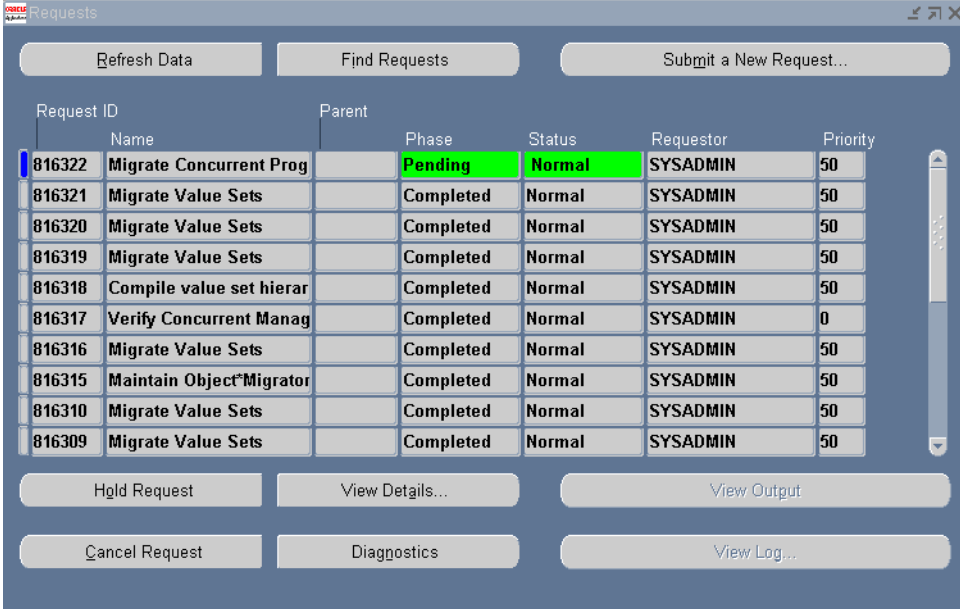
Include Request Set Stages in Query

Order By **Request ID**

Submit a New Request... Clear Find

- Select the criteria to locate the request.
- Click **Find**.

The Requests window opens.



Request ID	Name	Parent	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 Object*Migrator		Completed	Normal	SYSADMIN	50
816310	Migrate Value Sets		Completed	Normal	SYSADMIN	50
816309	Migrate Value Sets		Completed	Normal	SYSADMIN	50

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 shown here.
- **View Log.** Displays the execution log. Errors preventing execution are shown here.

The migration output and logs may contain some error messages indicating specific problems. See *“Exception Messages”* on page 111 for a complete list of error messages and related troubleshooting tips.

Chapter 5

Object Type-Specific Migrators and Migration Rules

Each Oracle AOL object type requires specific information and has unique rules for migrating between instances. Kintana 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 chapter discusses the migration parameters and rules for each supported object type.

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

Table 5-1 lists (in alphabetical order) the object type-specific migrators covered in this chapter along with their associated concurrent program.

Table 5-1. Object Type-Specific Migrators (Sheet 1 of 2)

Object Type	Migrator
<i>Common Migrator Parameters</i>	
<i>Concurrent Managers</i>	Migrate Concurrent Managers
<i>Concurrent Programs</i>	Migrate Concurrent Programs
<i>Descriptive Flexfields</i>	Migrate Descriptive Flexfields
<i>Folders</i>	Migrate Folders
<i>FSG Row/Column Sets</i>	Migrate FSG Row/Col Sets
<i>Functions</i>	Migrate Functions
<i>GUI Menus</i>	Migrate GUI Menus
<i>Help Text</i>	Migrate Help Text
<i>Menus (Character Mode)</i>	Migrate Menus

Table 5-1. Object Type-Specific Migrators (Sheet 2 of 2)

Object Type	Migrator
<i>Messages</i>	Migrate Messages
<i>Named SQL</i>	Migrate Named SQL
<i>Printer Definitions</i>	Migrate Printer Definitions
<i>Profile Options</i>	Migrate Profile Options
<i>QuickCodes (AOL)</i>	Migrate QuickCodes
<i>Request Groups (Report Groups)</i>	Migrate Report Groups
<i>Request Sets (Report Sets)</i>	Migrate Report Sets
<i>Responsibilities</i>	Migrate Responsibilities
<i>Users</i>	Migrate Users
<i>Value Sets</i>	Migrate Value Sets
<i>Zooms</i>	Migration Zooms

Common Migrator Parameters

Table 5-2, “Object*Migrator Common Parameters,” on page 32 lists (in alphabetical order) and defines the parameters that are common to most Object*Migrator object types.

Table 5-2. Object*Migrator Common Parameters

Parameter Name	Required?	Description
Compare Only	Yes	<p>Specifies whether to compare objects between instances: Select one of the following options:</p> <ul style="list-style-type: none"> • 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. <p>For more information, see the “<i>Comparing Objects</i>” chapter on page 103.</p>

Table 5-2. Object*Migrator Common Parameters

Parameter Name	Required?	Description
Dest Application	Yes	<p>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.</p> <p>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.</p>
Dest Database	Yes	<p>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.</p> <p>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.</p> <p>To migrate the object(s) into the Object Archive rather than to another Oracle Applications instance, select Object Archive.</p> <p>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.</p>

Table 5-2. Object*Migrator Common Parameters

Parameter Name	Required?	Description
From Version Label	No	<p>The version label name of the object(s) that will be recovered from the Object Archive. This parameter is used in conjunction with the Recover From Archive parameter. 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.</p> <p>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 107.</p>
New <i>Object_Type</i> Name	No	<p>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.</p> <p>This parameter is used only when Specific Object is selected in the Selection Type field.</p>
New User <i>Object_Type</i> Name/Title	No	<p>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.</p> <p>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.</p>

Table 5-2. Object*Migrator Common Parameters

Parameter Name	Required?	Description
<i>Object_Name</i> From	No	<p>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.</p> <p>Note: The range match is case-sensitive.</p> <p>Note: If objects exist that do not begin with characters, then use from exclamation point (!) to tilde (~) to get all objects.</p> <p>This parameter is used only when Range of Objects is selected in the Selection Type field.</p>
<i>Object_Name</i> Like	No	<p>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.</p> <p>This parameter is used only when Wild Card Match is selected in the Selection Type field.</p>
Overwrite if Exists	Yes	<p>Specifies whether to overwrite object definitions that already exist in the destination databases. Choose one of the following options:</p> <ul style="list-style-type: none"> • 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 in the New <i>Object_Type</i> Name parameter 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 5-2. Object* Migrator Common Parameters

Parameter Name	Required?	Description
Partials Allowed	Yes	<p>Specifies whether migrations can proceed when portions of the migration fail. Choose one of the following options:</p> <ul style="list-style-type: none"> • 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. <p>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 the Partials Allowed parameter to Yes would result in the migration of the concurrent program minus the failing parameter.</p> <ul style="list-style-type: none"> • No: If any of the components of the object fail validation, then the entire object is failed. Additionally, the error report will: 1) Log the failing components as having failed, 2) Log the object as having failed due to a component failure, 3) Log the components non-failing components as having failed due to an object failure. <p>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.</p> <p>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.</p>
Recover From Archive	Yes	<p>Specifies whether to use the Object Archive as the source database. Select one of the following options:</p> <ul style="list-style-type: none"> • 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. <p>This parameter should be set to Yes if Object Archive is selected in the Source Database and should be No if the Source Database is not Object Archive.</p>

Table 5-2. Object* Migrator Common Parameters

Parameter Name	Required?	Description
Report Only	Yes	<p>Specifies whether Object* Migrator should simulate the migration or perform the actual migration. Choose one of the following options:</p> <ul style="list-style-type: none"> • 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. <p>Note: The Report Only parameter must be set to No if the Compare Only parameter is Yes or if the Destination Database is Object Archive.</p>
Save To Archive	Yes	<p>Indicates whether to save an object to the Object Archive, preserving the version for reference or later use. Select one of the following options:</p> <ul style="list-style-type: none"> • 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. <p>Note: This parameter cannot be set to Yes if the Report Only parameter is Yes, or the Compare Only parameter is Yes.</p>

Table 5-2. Object*Migrator Common Parameters

Parameter Name	Required?	Description
Selection Type	Yes	<p>Indicates whether to migrate one or multiple objects. Select one of the following options:</p> <ul style="list-style-type: none"> • Specific Object: One object will be selected for migration at this time. Enter the specific object name in the specific <i>object_name</i> 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 <i>object_name</i> 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 <i>object_name</i> Like field. The wild card parameters are case sensitive. <p>Note: When migrating multiple objects (range or wildcard), 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.</p>
Source Application	Yes	<p>For object types that are owned by specific applications, the application that currently owns the object(s) being migrated (ownership in the source database).</p> <p>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.</p>

Table 5-2. Object*Migrator Common Parameters

Parameter Name	Required?	Description
Source Database	Yes	<p>The database where the object(s) to be migrated currently exists.</p> <p>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.</p> <p>If using an archived version of the object(s) as the source data for the migration, select Object Archive as the source database.</p> <p>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.</p>
Specific <i>Object_Type</i> Name	When Selection Type = Specific Object	<p>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 XXWIPDJPK program.</p> <p>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.</p>
To	No	<p>The end of the object name range to migrate. This field is used in conjunction with the <i>Object_Name</i> 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.</p> <p>Note: If objects exist that do not begin with characters, then use from exclamation point (!) to tilde (~) to get all objects.</p> <p>This parameter is only used when Range of Objects is selected in the Selection Type field.</p>

Table 5-2. Object*Migrator Common Parameters

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 the Save To Archive parameter 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 the Save To Archive parameter is set to Yes . This parameter is also used when comparing two versions of the same object. For more information, see <i>“Comparing Two Versions of an Object in the Object Archive”</i> on page 107.

Concurrent Managers

This migrator migrates the definition of the Concurrent Manager as well as all associated Work Shifts and Specialization Rules.

This section includes the following topics:

- [Before Migrating Concurrent Managers](#)
- [Concurrent Managers Migrator Parameters](#)

Before Migrating Concurrent Managers

Before migrating Concurrent Managers, you need to ensure 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.

Note

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

Note

The Concurrent Manager 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.

Note

The Concurrent Manager Migrator will migrate all workshifts and specialization rules attached to the migrated concurrent manager. It will overwrite the definition of the workshift or specialization rule if it already exists on the destination database.

Note

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 screen for the Migrate Concurrent Manager request.



Note

Only the object-specific parameters are defined here. For more information about the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-3. Concurrent Managers Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see “Object*Migrator Common Parameters” on page 32.
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 5-3. Concurrent Managers Migrator Parameters

Parameter Name	Description / Notes
Conc Manager From	For descriptions of these common parameters, see <i>“Object*Migrator Common Parameters”</i> on page 32.
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 Programs](#)
- [Concurrent Programs Migrator Parameters](#)

Before Migrating Concurrent Programs

Before migrating Concurrent Programs, you need to ensure 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 (if the Mig Incomp Partially parameter is No) should be migrated together. For details about this parameter, see [Table 5-4 on page 44](#).

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 Program 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 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 screen 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 5-2 on page 32](#).

Table 5-4. Concurrent Programs Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see “Object*Migrator Common Parameters” on page 32.
Dest Database	

Table 5-4. Concurrent Programs Migrator Parameters

Parameter Name	Description / Notes
Source Application	The Application to which the Concurrent Program belongs. For a description of this common parameter, see Table 5-2 on page 32 .
Dest Application	For a description of this common parameter, see Table 5-2 on page 32 .
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 5-2 on page 32 . 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 5-2 on page 32 .
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 5-2 on page 32 .
Conc Program From	For a description of this common parameter, see Table 5-2 on page 32 .
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 5-2 on page 32 .

Table 5-4. Concurrent Programs Migrator Parameters

Parameter Name	Description / Notes
Save To Archive	For a description of this common parameter, see Table 5-2 on page 32 .
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. Choose one of the following: <ul style="list-style-type: none"> • 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 5-2 on page 32 .
Compare Only	

Descriptive Flexfields

This section includes the following topics:

- [Before Migrating Descriptive Flexfields](#)
- [Descriptive Flexfields Migrator Parameters](#)

Before Migrating Descriptive Flexfields

Before migrating Descriptive Flexfields, you need to ensure 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 Flexfield 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 Flexfield 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 Flexfield 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 Flexfield program.

- The flexfield in the destination (if it exists) should not be protected. This *is* validated by the Migrate Descriptive Flexfield 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.

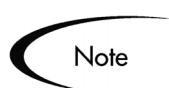
This *is* validated by the Migrate Descriptive Flexfield program.

- 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 screen for the Migrate Descriptive Flexfields request.



Note

Only the object-specific parameters are defined here. For more information about the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-5. Descriptive Flexfields Migrator Parameters

Parameter Name	Description / Notes
Source Database	For a description of this common parameter, see Table 5-2 on page 32 .
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 5-2 on page 32 .

Table 5-5. Descriptive Flexfields Migrator Parameters

Parameter Name	Description / Notes
Desc Flex From	For a description of this common parameter, see Table 5-2 on page 32 .
To	
Desc Flex Like	
Save To Archive	
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Disabled Contexts	<p>Species whether to migrate descriptive flexfields contexts that are disabled in the source. Select one of the following:</p> <ul style="list-style-type: none"> • 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. <p>Caution should be taken when using the No option for Disabled Contexts to ensure 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 the Yes options above.</p>

Table 5-5. Descriptive Flexfields Migrator Parameters

Parameter Name	Description / Notes
Disabled Segments	<p>Specifies whether to migrate disabled segments. Select one of the following:</p> <ul style="list-style-type: none"> • 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. <p>Caution should be taken when using the No option for Disabled Contexts and Disabled Segments to ensure 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 the Yes options above.</p>
Overwrite if Exists	For a description of this common parameter, see Table 5-2 on page 32 .
Report Only	
Compare Only	<p>For a description of this common parameter, see Table 5-2 on page 32.</p> <p>Note: The comparison functionality does not compare column information when comparing objects.</p>

Folders

This section covers the following topics:

- [Before Migrating Folders](#)
- [Folders Migrator Parameters](#)

Before Migrating Folders

Before migrating Folders, you need to ensure 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 screen for the Migrate Folders request.



Note

Only the object-specific parameters are defined here. For more information on the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-6. Folders Migrator Parameters

Parameter Name	Description / Notes
Source Database	For a description of this common parameter, see Table 5-2 on page 32 .
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 Folder 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.
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 Folder 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.

Table 5-6. Folders Migrator Parameters

Parameter Name	Description / Notes
Specific Segment Name	<p>Enter the current folder segment (as named in source database) of the folder(s) to migrate.</p> <p>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.</p> <p>If left blank, the Folder Migrator will not limit by folder segment (just by Specific Folder Name and Specific Owner if they are specified).</p>
Folder Name From	<p>For a description of this common parameter, see Table 5-2 on page 32.</p>
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 screen. Ensure the successful migration of these objects by following the instructions included in this section.

This section covers the following topics:

- [Before Migrating FSG Row/Column Sets](#)
- [FSG Row/Column Sets Migrator Parameters](#)

Before Migrating FSG Row/Column Sets

Before migrating FSG Row/Column Sets, ensure 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 screen for the Migrate FSG Row/Col Sets request.



Note

Only the object-specific parameters are defined here. For more information on one of the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-7. FSG Row/Column Sets Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
Dest Database	
Set Type (required)	Enter/Quickpick the type of FSG Set(s) to migrate: <ul style="list-style-type: none"> • 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 5-2 on page 32 .

Table 5-7. FSG Row/Column Sets Migrator Parameters

Parameter Name	Description / Notes
Specific Row/Column Set	<p>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.</p> <p>The list of values for this parameter does not limit the values by the Set Type parameter.</p> <p>For a description of this common parameter, see Table 5-2 on page 32.</p>
New Row/Column Set	<p>For descriptions of these common parameters, see Table 5-2 on page 32.</p>
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 screen. 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.

This section covers the following topics:

- *Before Migrating Functions*
- *Functions Migrator Parameters*

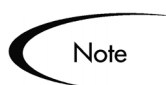
Before Migrating Functions

Before migrating a Function, ensure 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.



Note

If a form referenced by a function does not exist on the destination database, it will be created.

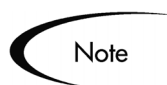
Note: The form file will not be copied to the destination.



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 screen for the Migrate Functions request.



Only the object-specific parameters are defined here. For more information on one of the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-8. Functions Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
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 5-2 on page 32 .
New Function Name	For a description of this common parameter, see Table 5-2 on page 32 .
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 5-8. Functions Migrator Parameters

Parameter Name	Description / Notes
Function Name From	For descriptions of these common parameters, see Table 5-2 on page 32 .
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 screen. 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.

This section covers the following topics:

- [Before Migrating GUI Menus](#)
- [GUI Menus Migrator Parameters](#)

Before Migrating GUI Menus

Before migrating GUI Menus, ensure that a number of prerequisite conditions are met. If the following entities are referenced by the GUI Menu 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 the Single Level Only option is set to 'No')
- If the Single Level Only option 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



Note

If the destination database is 11.5.5 or higher, the migrator will submit the Compile Security concurrent program in the destination instance in order to complete the menu compilations introduced in 11.5.5.



Caution

Care should be taken when migrating entire menu structures to ensure 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 screen for the Migrate GUI Menus request.



Note

Only the object-specific parameters are defined here. For more information on one of the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-9. GUI Menus Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
Dest Database	
Selection Type	

Table 5-9. GUI Menus Migrator Parameters

Parameter Name	Description / Notes
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 5-2 on page 32 .
New Menu Name	For a description of this common parameter, see Table 5-2 on page 32 .
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 5-2 on page 32 .
To	
Menu Name Like	
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. Choose one of the following: <ul style="list-style-type: none"> • Yes: Only migrate the menu entries of the given menu. • No: Migrate the entire Menu sub-tree below the Specific Menu Name. <p>Care should be taken when migrating entire menu structures to ensure that standard menus are not inadvertently overwritten. Standard menus may be referenced from submenus in a custom menu structure.</p>
Save To Archive	For descriptions of these common parameters, see Table 5-2 on page 32 .
Version Label	
Version Desc	
Recover From Archive	
From Version Label	

Table 5-9. GUI Menus Migrator Parameters

Parameter Name	Description / Notes
Overwrite if Exists	For a description of this common parameter, see Table 5-2 on page 32 . 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 = No, as long as the top parent menu did not exist.
Partials Allowed	For descriptions of these common parameters, see Table 5-2 on page 32 .
Report Only	
Compare Only	For a description of this common parameter, see Table 5-2 on page 32 . Note: For better output, it is recommended that comparisons of Menus be run with Single Level Only = Yes.



Migrating a range of parent menus with Single Level = 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 screen. 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.

This section covers the following topics:

- [Before Migrating Help Text](#)
- [Help Text Migrator Parameters](#)

Before Migrating Help Text

Before migrating Help Text, ensure 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.



Note

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 screen for the Migrate Help Text request.



Note

Only the object-specific parameters are defined here. For more information on one of the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-10. Help Text Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
Dest Database	
Source Application	
Dest Application	
Selection Type	

Table 5-10. Help Text Migrator Parameters

Parameter Name	Description / Notes
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 5-2 on page 32 .
New Form Name	For descriptions of these common parameters, see Table 5-2 on page 32 .
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 5-2 on page 32 . 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 screen. 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.

This section covers the following topics:

- [Before Migrating Menus](#)
- [Menus Migrator Parameters](#)

Before Migrating Menus

Before migrating Menus, ensure 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 the Single Level Only option is set to 'No')
- Subroutines reference by the menu (and the menu tree underneath the menu, if the Single Level Only option is set to 'No')
- Macros reference by the menu (and the menu tree underneath the menu, if the Single Level Only option is set to 'No')
- If the Single Level Only option is set to 'Yes', then any Child Menus reference by the Parent Menu must exist in the destination database.



Note

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 screen for the Migrate Menus request.



Note

Only the object-specific parameters are defined here. For more information on one of the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-11. Menus Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
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 5-2 on page 32 .
New Menu Name	For a description of this common parameter, see Table 5-2 on page 32 .
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 5-2 on page 32 .
To	
Menu Name Like	
Single Level Only (Required)	Specifies whether to migrate only the top level of the menu or to include all sub-entries beneath the menu. Choose one of the following: <ul style="list-style-type: none"> • 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 = No) Choose one of the following options: <ul style="list-style-type: none"> • Yes: Migrate and report on Menus owned by standard Oracle Apps. • No: Do not migrate or report on Menus owned by standard Oracle Applications.

Table 5-11. Menus Migrator Parameters

Parameter Name	Description / Notes
Import AOL/Sysadmin Menus	(Relevant only if Single Level Only = No and Import Standard Oracle Menus = Yes) Choose one of the following options: <ul style="list-style-type: none"> • 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 5-2 on page 32 .
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 5-2 on page 32 . Note: For better output, it is recommended that comparisons of Menus be run with Single Level Only = Yes .

Messages

This section discusses migrating Messages, including all of the object-specific parameters on the Submit Request screen. 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.

This section covers the following topics:

- [Before Migrating Messages](#)

- [Messages Migrator Parameters](#)

Before Migrating Messages

Before migrating Messages, ensure 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



Note

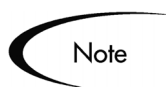
For server side messages only (nothing needs to be done for messages used by Smart Client forms), Oracle Applications requires the execution of the concurrent program “Create message file and report” (Release 10) or “Generate Messages” (Release 11/11i). Due to database limitations with distributed transactions, 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.

This can be done by navigating to the Define Messages character mode form, querying a message from the given application, updating the record without a real change (ex. change a letter in the message and change 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 Oracle Applications System Administration Guide (FNDMDGEN apps/apps 0 y INV usaeng in Release 11/11i).

Messages Migrator Parameters

The following parameters are located on the Submit Request screen for the Migrate Messages request.



Note

Only the object-specific parameters are defined here. For more information on one of the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-12. Messages Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
Dest Database	
Source Application	The Application to which the Message belongs. For a description of this common parameter, see Table 5-2 on page 32 .
Dest Application	For descriptions of these common parameters, see Table 5-2 on page 32 .
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.
Message From	For descriptions of these common parameters, see Table 5-2 on page 32 .
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 screen. 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.

This section covers the following topics:

- [Before Migrating Named SQL](#)
- [Named SQL Migrator Parameters](#)

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 screen for the Migrate Named SQL request.



Note

Only the object-specific parameters are defined here. For more information on one of the common Object* Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-13. Named SQL Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
Dest Database	
Source Application	
Dest Application	
Selection Type	

Table 5-13. Named SQL Migrator Parameters

Parameter Name	Description / Notes
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 5-2 on page 32 .
New Named SQL	For descriptions of these common parameters, see Table 5-2 on page 32 .
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 5-2 on page 32 . 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 screen. 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.



Note

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.

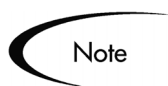
This section covers the following topics:

- [Before Migrating Printer Definitions](#)
- [Printer Definitions Migrator Parameters](#)

Before Migrating Printer Definitions

Before migrating Printer Definitions, ensure 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



Note

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 screen for the Migrate Printer Definitions request.



Note

Only the object-specific parameters are defined here. For more information on one of the common Object* Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-14. Printer Definitions Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
Dest Database	
Selection Type	
Specific Printer Type	<p>Enter the current name (as named in source database) of the printer type to migrate. Object*Migrator will attempt to migrator 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.</p> <p>For a description of this common parameter, see Table 5-2 on page 32.</p>
New Printer Type	For descriptions of these common parameters, see Table 5-2 on page 32 .
Printer Type From	
To	
Printer Type Like	
Import Printers	<ul style="list-style-type: none"> • 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.
Save To Archive	For descriptions of these common parameters, see Table 5-2 on page 32 .
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 screen. 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 ensure 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.

This section covers the following topics:

- [Before Migrating Profile Options](#)
- [Profile Options Migrator Parameters](#)

Before Migrating Profile Options

Before migrating Profile Options, ensure 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 Option 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 userid corresponding to SYSADMIN, the migrator would bring the user id over to the destination as is. It would not get the userid 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 Application 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 the Migrate Profile Option Values parameter 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 ensure 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 screen for the Migrate Profile Options request.



Note

Only the object-specific parameters are defined here. For more information on one of the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-15. Profile Options Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
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 5-2 on page 32 .
New Profile Option Name	For descriptions of this common parameter, see Table 5-2 on page 32 .
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 5-2 on page 32 .

Table 5-15. Profile Options Migrator Parameters

Parameter Name	Description / Notes
Profile Option From	For descriptions of these common parameters, see Table 5-2 on page 32 .
To	
Profile Option Like	
Save To Archive	
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Migrate Profile Option Values (Required)	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • 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)	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • 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)	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • 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 5-2 on page 32 .
Partials Allowed	
Report Only	
Compare Only	

QuickCodes (AOL)

This section discusses migrating QuickCodes, including all of the object-specific parameters on the Submit Request screen. 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.

This section covers the following topics:

- [Before Migrating QuickCode](#)
- [QuickCodes Migrator Parameters](#)

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 screen for the Migrate QuickCodes request.



Note

Only the object-specific parameters are defined here. For more information on one of the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-16. QuickCodes Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
Dest Database	
Source Application	
Dest Application	
Selection Type	

Table 5-16. QuickCodes Migrator Parameters

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 5-2 on page 32 .
New QuickCode Type	For descriptions of these common parameters, see Table 5-2 on page 32 .
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 screen. Ensure the successful migration of these objects by following the instructions included in this section.



Note

In some releases of Oracle E-Business Suite, Request Groups are referred to as Report Groups.

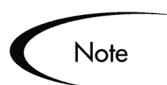
This section covers the following topics:

- [Before Migrating Request Groups](#)
- [Request Groups Migrator Parameters](#)

Before Migrating Request Groups

Before migrating Request Groups, ensure 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

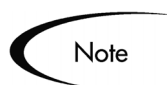


Note

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 screen for the Migrate Report Groups request.



Note

Only the object-specific parameters are defined here. For more information on one of the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-17. Request Groups Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
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 5-2 on page 32 .
New Report Group Name	For descriptions of these common parameters, see Table 5-2 on page 32 .
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 screen. Ensure the successful migration of these objects by following the instructions included in this section.

This section covers the following topics:

- [Before Migrating Request Sets](#)
- [Request Sets Migrator Parameters](#)

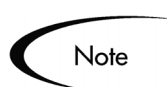
Before Migrating Request Sets

Before migrating Request Sets, ensure 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 screen for the Migrate Report Sets request.



Note

Only the object-specific parameters are defined here. For more information on one of the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-18. Request Sets Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
Dest Database	

Table 5-18. Request Sets Migrator Parameters

Parameter Name	Description / Notes
Source Application	The Application to which the Request Set belongs. For a description of this common parameter, see Table 5-2 on page 32 .
Dest Application	For descriptions of these common parameters, see Table 5-2 on page 32 .
Selection Type	
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 5-2 on page 32 .
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 5-2 on page 32 .
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 so 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 5-2 on page 32 .
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 screen. Ensure the successful migration of these objects by following the instructions included in this section.

This section covers the following topics:

- [Before Migrating Responsibilities](#)
- [Responsibilities Migrator Parameters](#)

Before Migrating Responsibilities

Before migrating Responsibilities, ensure 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 screen for the Migrate Responsibilities request.



Note

Only the object-specific parameters are defined here. For more information on one of the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-19. Responsibilities Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
Dest Database	
Source Application	
Dest Application	
Selection Type	
Specific Responsibility	<p>Enter the current Name (as named in source database) of the Responsibility to migrate. Query this value from the Define Responsibilities form.</p> <p>For a description of this common parameter, see Table 5-2 on page 32.</p>
New Responsibility Name	For descriptions of these common parameters, see Table 5-2 on page 32 .
Responsibility From	
To	
Responsibility Like	
Migrate Enabled Resp Only (required)	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • Yes: Migrate only enabled responsibilities. • No: Migrate both enabled and disabled responsibilities.
Synchronize Workflow Tables	<p>Use this parameter only if the destination is an Oracle 11.5.9 instance. If the instance is <i>not</i> at the 11.5.9 level, this parameter will be ignored.</p> <p>Select one of the following options:</p> <ul style="list-style-type: none"> • 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 concurrent program, which will populate the Workflow local tables for all responsibilities defined in the application.

Table 5-19. Responsibilities Migrator Parameters

Parameter Name	Description / Notes
Migrate Sec Attrib Values	<p>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.</p> <p>If the Migrate Sec Attrib Values parameter 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 note below.</p> <p>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 the Migrate Sec Attrib Values parameter 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.</p> <p>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.</p> <p>If the Migrate Sec Attrib Values parameter 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.</p> <p>It is recommended to set this parameter to No unless the Responsibility for versions in both the source and destination are at version 115.17 or higher, and that the attribute values attached to the responsibilities will be valid in the destination.</p> <p>Note: The table where securing attribute values are stored, <code>ak_resp_security_attr_values</code>, 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.</p>

Table 5-19. Responsibilities Migrator Parameters

Parameter Name	Description / Notes
Save To Archive	For descriptions of these common parameters, see Table 5-2 on page 32 .
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 5-2 on page 32 . 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 screen. Ensure the successful migration of these objects by following the instructions included in this section.

This section covers the following topics:

- [Before Migrating Users](#)
- [Users Migrator Parameters](#)

Before Migrating Users

Before migrating Users, ensure 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.

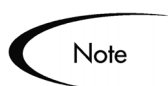


Note

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 screen for the Migrate Users request.



Note

Only the object-specific parameters are defined here. For more information on one of the common Object* Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-20. Users Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
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 5-2 on page 32 .
New User Name	For a description of this common parameter, see Table 5-2 on page 32 . Note: When creating a new user, for security reasons, change the password of the new user once the migration is complete.

Table 5-20. Users Migrator Parameters

Parameter Name	Description / Notes
User Name From	For descriptions of these common parameters, see Table 5-2 on page 32 .
To	
User Name Like	
Create User (Required)	<p>Specifies whether to create a new User in the destination database. Choose one of the following options:</p> <ul style="list-style-type: none"> • Yes: If the User does not exist on the destination database, create the user based on the header information from the source database. <p>Note: If the User already exists, the User Migrator WILL NOT update any of the user header information (password, employee, etc.). 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.</p> <ul style="list-style-type: none"> • No: If the User does not exist on the destination database, set the status of the object to 'Error'.
Active Resp Only (Required)	<p>Determines which responsibilities to migrate. Select one of the following options:</p> <ul style="list-style-type: none"> • 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)	<p>Specifies whether to overwrite responsibilities in the destination database. Select one of the following options:</p> <ul style="list-style-type: none"> • 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.
Disable Non-Migrated Resp (Required)	<p>Specifies whether to disable responsibilities associated with the user in the destination database that were not migrated. Select one of the following options:</p> <ul style="list-style-type: none"> • 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.

Table 5-20. Users Migrator Parameters

Parameter Name	Description / Notes
Migrate Enabled Users Only (Required)	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • Yes: Only migrate enabled users. • No: Migrate both enabled and disabled users.
Migrate Securing Attributes (Required)	<p>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:</p> <ul style="list-style-type: none"> • 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). <p>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.</p> <p>Migration handles managed attribute values in the following ways regardless of the parameter value, if the user:</p> <ul style="list-style-type: none"> • 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. <p>Select one of the following options:</p> <ul style="list-style-type: none"> • Yes: Migrate all securing attribute values for the user. Existing values will be removed or replaced. <p>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.</p> <ul style="list-style-type: none"> • No: Migrate only attributes which Oracle Applications actively manages. Other attributes will neither be migrated from the source nor removed from the destination.

Table 5-20. Users Migrator Parameters

Parameter Name	Description / Notes
Save To Archive	For descriptions of these common parameters, see Table 5-2 on page 32 .
Version Label	
Version Description	
Recover From Archive	
From Version Label	
Overwrite if Exists	
Partials Allowed	
Report Only	
Compare Only	
Synchronize Workflow Tables	<p>Use this parameter only if the destination is an Oracle 11.5.9 instance. If the instance is <i>not</i> at the 11.5.9 level, this parameter will be ignored.</p> <p>Select one of the following options:</p> <ul style="list-style-type: none"> • 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 concurrent 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 screen. Ensure the successful migration of these objects by following the instructions included in this section.

This section covers the following topics:

- [Before Migrating Value Sets](#)
- [Value Sets Migrator Parameters](#)

Before Migrating Value Sets

Before migrating Value Sets, ensure 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.



Caution

This is *not* validated by the Migrate Value Sets program.



Note

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.



Note

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

Caution should be taken when migrating Value Sets to ensure 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.



Note

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 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 screen for the Migrate Value Sets request.



Note

Only the object-specific parameters are defined here. For more information on one of the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-21. Value Sets Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
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 5-2 on page 32 .
New Value Set Name	For descriptions of these common parameters, see Table 5-2 on page 32 .
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 screen. Ensure the successful migration of these objects by following the instructions included in this section.

This section covers the following topics:

- [Before Migrating Zooms](#)
- [Zooms Migrator Parameters](#)

Before Migrating Zooms

Before migrating Zooms, ensure 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)



Note

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 screen for the Migrate Zooms request.



Note

Only the object-specific parameters are defined here. For more information on one of the common Object*Migrator parameters, see [Table 5-2 on page 32](#).

Table 5-22. Zooms Migrator Parameters

Parameter Name	Description / Notes
Source Database	For descriptions of these common parameters, see Table 5-2 on page 32 .
Dest Database	
Source Application	
Dest Application	
Selection Type	
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 5-2 on page 32 .
New Zoom Name	For a description of this common parameter, see Table 5-2 on page 32 .
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 5-2 on page 32 .
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	

Chapter

6

Using Object*Migrator Version Control

Object*Migrator saves data to special archive tables as objects are migrated across databases. You can use this archived data as your source information, thus allowing you to revert your objects back to previous versions. Additionally, you can use Object*Migrator reports to view complete version histories of a specific object or a group of objects.

This chapter covers the following topics:

- *Archiving AOL Object Definitions*
- *Running the Object Archive Version Detail Report*
- *Purging the Object Archive*

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 the “*Comparing Objects*” chapter on page 103.

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 screen, create a new request for an object migration.

The Parameters 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.



Note

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 the *“Object Type-Specific Migrators and Migration Rules”* chapter on page 31.

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 screen, create a new request for an object migration.

The Parameters 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 the “[Object Type-Specific Migrators and Migration Rules](#)” chapter on page 31.
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.

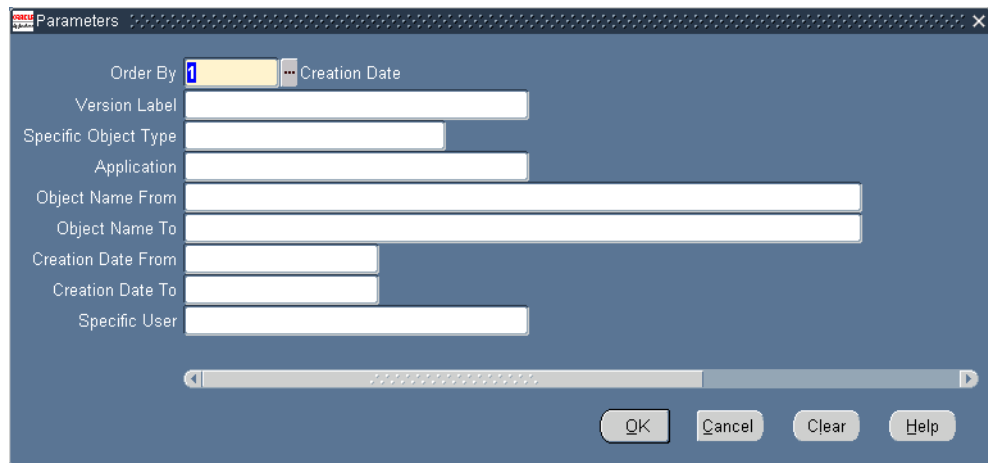
Running the Object Archive Version Detail Report

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 screen.

To run the Object Archive Version Detail Report:

1. Open the Submit Request screen.
2. In the Name field, select **Object Archive Version Detail Report**.

The Parameters window opens for the report.



3. To limit the report results, enter the relevant report parameters (see [Table 6-1 on page 99](#)).
4. Click **OK**.

The report runs according to the specified parameters.

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

[Table 6-1 on page 99](#) lists and defines the Object*Migrator common parameters used with the Object Archive Version Detail report.

Table 6-1. Object*Migrator Common Parameters

Parameter Name	Description
Order By (Required)	Select one of the following: <ul style="list-style-type: none"> • 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, etc.).
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 Version report 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 screen.



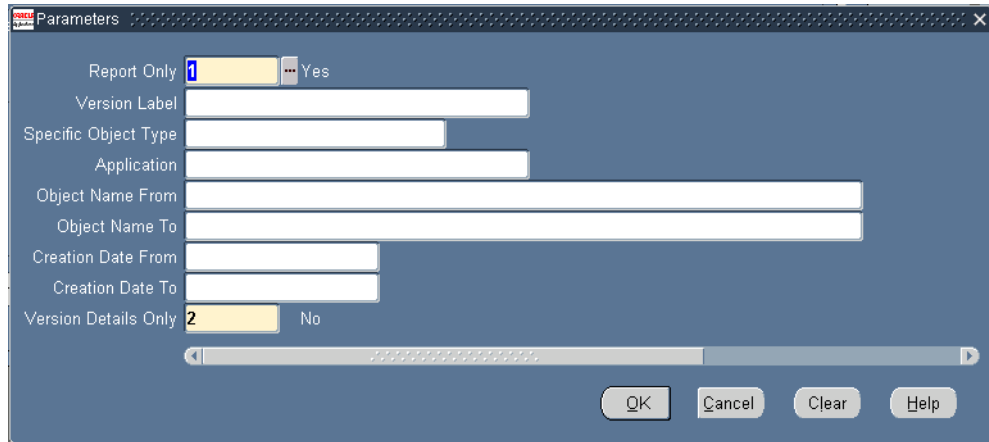
Note

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 screen.
2. In the Name field, select **Purge Object Archive Versions**.

The Parameters window opens for the report.



3. To limit the report results, enter the relevant report parameters (see [Table 6-2 on page 100](#)).
4. Click **OK**.

The report runs according to the specified parameters.

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

[Table 6-2](#) lists and defines the Purge Object Archive Versions parameters.

Table 6-2. Purge Object Archive Versions Parameters

Parameter Name	Description
Report Only (Required)	Choose one of the following options: <ul style="list-style-type: none"> • 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.

Table 6-2. Purge Object Archive Versions Parameters

Parameter Name	Description
Object Type	Limit the data to purge by a specific object type (Concurrent Program, Value Set, etc.).
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)	<p>Choose one of the following options:</p> <ul style="list-style-type: none"> • Yes: Purge the details of the object in the Object Archive but do not purge the version itself. <p>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.</p> <p>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.</p> <ul style="list-style-type: none"> • 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 7 Comparing Objects

Object*Migrator compares objects located in different database instances. For example, objects in a current instance can be compared against an archived versions of objects located in the Object Archive.

This chapter comparing objects using Object*Migrator. It covers the following topics:

- [Object Comparison Overview](#)
- [Running the Object Comparison](#)
- [Sample Comparison Report](#)

Object Comparison Overview

Object comparisons are performed using the same program as migrations. Each Object*Migrator request includes a Compare Only parameter 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 DB](#)
- [Only in Dest DB](#)

- *Differences Exist*

Only in Source DB

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 DB.

Only in Dest DB

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.



Note

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



Some child entities are not compared. Generally, these are noted in the “*Object Type-Specific Migrators and Migration Rules*” chapter on page 31.

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 screen, create a new request for an object migration.

The Parameters 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. In the Compare Only field, select **1 (Yes)**.

With the Compare Only parameter 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 wildcard.

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

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



Note

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.



Note

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 the *“Object Type-Specific Migrators and Migration Rules”* chapter on page 31.

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 screen, create a new request for an object migration.

The Parameters 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. In the Compare Only field, select **1 (Yes)**.
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 the New Object Name parameter, and will instead use the value in the Specific Object Name parameter when retrieving object information for both databases).

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

For a description of each parameter, see the *“Object Type-Specific Migrators and Migration Rules”* chapter on page 31.

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 screen, create a new request for an object migration.

The Parameters 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. In the Compare Only field, select **1 (Yes)**.
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 the New Object Name parameter, and will instead use the value in the Specific Object Name parameter when retrieving object information for both databases).

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

For a description of each parameter, see the *“Object Type-Specific Migrators and Migration Rules”* chapter on page 31.

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:

- o The Maximum Columns field value was removed and the description was changed.
- o A new parameter named New Param was entered.
- o The parameter Org ID was removed.
- o The value set for the parameter Dates To was changed from WIP_SRS_DATES_OP to FND_DATE.

Figure 7-1 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 7-1 Sample Comparison Report Output

Appendix

A

Exception Messages

The following is a list of Kintana 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 Interactive.
- **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 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 Program with Overwrite option 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 option 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 the Partial Allowed option 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 Partial Allowed option to No.

Message: 1203 Error: Report Set Concurrent 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 the Overwrite option 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 Setup 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 option 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 option 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) The Single Level Only parameter 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) The 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 the Overwrite option is set to No.

Action Run Migrate Menus program with Overwrite option 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 option 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 option 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 User Object*Migrator is being run with Create User = No and the User does not exist on the destination database.

Action Run the Migrator with Create User = 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 option 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 Option Object* 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.

Cause The Server referenced by the Profile Option Value does not exist in the destination database.

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 Application 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 the Dest Application parameter 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 option 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 option 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 Interactive Support.

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 the Partial Allowed option 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 option 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 option 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 option 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 option 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 option 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 option 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 option 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 option 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 option 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 = 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 = 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 Type.
- 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 Apps 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 option 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, etc.
- 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 option 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 Managers program with Overwrite option set to Yes.
- Message: 3101** Error: Workshift already exists on destination database.
Cause Specified Workshift already exists in destination database.
Action Run Migrate Concurrent Managers program with Overwrite option 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 Managers program with Overwrite option 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 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 workshift.

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 Managers program with the Partial Allowed option 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 Interactive Support.

Message: 7101 Internal Error: Error getting EXECUTABLE_ID.

Action Contact Mercury Interactive Support.

Message: 7200 Internal Error: Error getting REQUEST_SET_ID.

Action Contact Mercury Interactive Support.

Message: 7201 Internal Error: Error getting REQUEST_SET_PROGRAM_ID.

Action: Contact Mercury Interactive Support.

Message: 7300 Internal Error: Error getting FLEX_SET_VALUE_ID.

Action Contact Mercury Interactive Support.

Message: 7301 Internal Error: Error getting FLEX_VALUE_ID.

Action Contact Mercury Interactive Support.

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: 7400 Internal Error: Error getting APPLICATION_ID.

Action Contact Mercury Interactive Support.

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:

a. The Descriptive Flex Field references a Value Set for Context Override Value Set that does not exist at the destination.

b. The Descriptive Flex Field being migrated has Context Override Value Set properties set and the destination version of Oracle Application is lower than 11.5.5, which does not support context override value set features.

Action Create the Value Set referenced by the Descriptive Flex Field at the destination.

Message: 7500 Internal Error: Error getting REQUEST_GROUP_ID.

Action Contact Mercury Interactive Support.

Message: 7700 Internal Error: Error getting MENU_ID.

Action Contact Mercury Interactive Support.

Message: 7800 Internal Error: Unable to get ZOOM_DEFINITION_ID in destination database.

Action Contact Mercury Interactive Support.

Message: 7801 Internal Error: Unable to get ACTION_SEQUENCE_ID in destination database.

Action Contact Mercury Interactive Support.

Message: 7900 Internal Error: Unable to get RESPONSIBILITY_ID in destination database.

Action Contact Mercury Interactive Support.

Message: 8000 Internal Error: Unable to get USER_ID in destination database.

Action Contact Mercury Interactive Support

Message: 8100 Internal Error: Unable to get PROFILE_OPTION_ID in destination database.

Action Contact Mercury Interactive Support.

Message: 8200 Internal Error: Unable to get PROFILE_OPTION_ID in destination database.

Action Contact Mercury Interactive Support.

Message: 8300 Internal Error: Error getting APPLICATION_ID.

Action Contact Mercury Interactive Support.

Message: 8400 Internal Error: Error getting APPLICATION_ID.

Action Contact Mercury Interactive Support.

Message: 8500 Internal Error: Error getting AXIS_SET_ID.

Action Contact Mercury Interactive Support.

Message: 8502 Internal Error: Error getting APPLICATION_ID.

Action Contact Mercury Interactive Support.

Message: 8502 Internal Error: Error getting EXCEPTION_ID.

Action Contact Mercury Interactive Support.

Message: 8600 Internal Error: Error getting PRINTER_TYPE.

Action Contact Mercury Interactive Support.

Message: 8700 Internal Error: Error getting FOLDER_ID.

Action Contact Mercury Interactive Support.

Message: 8800 Internal Error: Error getting MENU_ID.

Action Contact Mercury Interactive Support.

Message: 8801 Internal Error: Error getting MENU_ID for child menu.

Action Contact Mercury Interactive Support.

Message: 8900 Internal Error: Error getting FUNCTION_ID.

Action Contact Mercury Interactive Support.

Message: 9100 Internal Error: Error getting CONCURRENT_QUEUE_ID.

Action Contact Mercury Interactive Support.

Message: 9101 Internal Error: Error getting CONCURRENT_TIME_PERIOD_ID.

Action Contact Mercury Interactive Support.

Message: 9102 Internal Error: Error getting COMPLEX_RULE_ID.

Action Contact Mercury Interactive Support.

Message: 9103 Internal Error: Error getting COMPLEX_RULE_LINE_ID.

Action Contact Mercury Interactive Support.

Appendix B

AOL Dependent Objects

Table B-1 lists the dependent object for each of the 20 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 B-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 #

Table B-1. AOL Object Migrated By Object* Migrator

No.	Object	Dependencies (Migrated)	Dependencies (Not Migrated)
5	Request Groups (Report Groups)		Concurrent Program Request Set Application #
6	Menu	Sub-menus (optionally migrated)	Application # Forms # Subroutines # Macros #
7	Zoom		Forms # Application #
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	Sub-menus (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 #

Table B-1. AOL Object Migrated By Object*Migrator

No.	Object	Dependencies (Migrated)	Dependencies (Not Migrated)
17	Printer Definitions	Printer Types Printer Drivers Printer Styles	Platforms #
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

C

Using Object*Migrator with Kintana Deliver

Object*Migrator can be run from Kintana Deliver, another product in the Kintana Product Suite. This appendix provides an overview of Kintana Deliver and highlights the integration points between Object*Migrator and Kintana Deliver.

For more information, see *Processing Packages (Kintana Deliver)*.

Note

For information about accessing this manual, see “[Change Management and Configuration Documentation](#)” on page 13. You must have already purchased Kintana Deliver to access this document.

This appendix discusses the following topics:

- [Introduction to Kintana Deliver](#)
- [Executing Object*Migrator from Kintana Deliver](#)
- [Processing the Package in Kintana Deliver](#)

Introduction to Kintana Deliver

Kintana Deliver is designed to automate the deployment of technology solutions. Business system components created or modified by an initiative are grouped into Kintana Deliver 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. Kintana Deliver automatically

deploys the application components necessary for each Package, such as XML content, HTML files, Java programs, Oracle Application 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. Kintana Deliver maintains an audit trail for all activities including Package information, application components altered, approvals obtained and deployments performed.

Executing Object*Migrator from Kintana Deliver

Object*Migrator migrations can be run directly through Kintana Deliver once the Kintana Accelerator for Oracle Applications has been installed and configured. The Accelerator includes a pre-defined Object Type for each type of object migrated using Object*Migrator.

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

In Deliver, 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 Deliver 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 Kintana Deliver:

1. In Kintana Deliver, click the **Add Line** button on the Package window.

The Add Line window opens, displaying a list of Object Types that can be used.

2. In the Object Type field, select the Object Type to be migrated.

The window refreshes to display the parameters required to migrate the specific object. For a description of each parameter, see [Chapter 5 - "Object Type-Specific Migrators and Migration Rules"](#). Both the application name and the specific object name will be validated against the

initial 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 databases are defaulted from the workflow.

3. Enter the parameters to describe the object.
4. Click **OK** to save the Package Line.

Once the Package Line has been entered and saved, the AOL object moves through the Workflow in a similar manner to any other Package Line in Kintana Deliver.

Processing the Package in Kintana Deliver

Once eligible, users can migrate the selected objects to the next environment. When an action step in a Workflow is executed, Kintana Deliver 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.

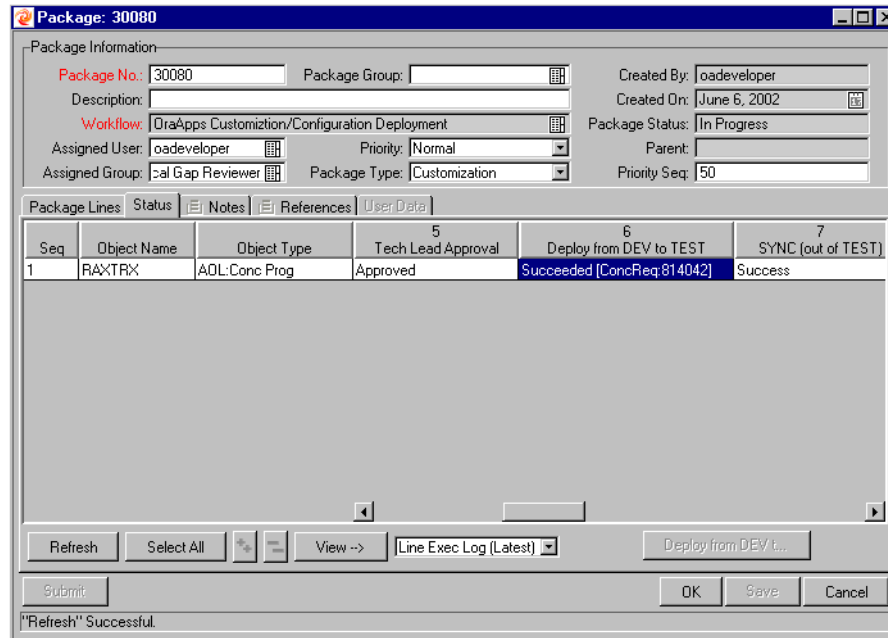


Figure C-1 Processing a Package in Kintana Deliver

When migrating multiple AOL Object request lines, Deliver 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 Deliver. Kintana launches a Web browser session to display the report.

[Kintana Delivero](#) [Package Execution Log History](#)
Execution Log For Batch 30428 - Line 1 [Line Execution Log History](#)
[Batch Execution Log](#)

Object	AOL.Conc Prog - RAXTRX
Package No.	30080
Workflow	OraApps Customization/Configuration Deployment
Workflow Step	6 - Deploy from DEV to TEST
Started	June 6, 2002 05:06:17 PM PDT

KSC OM MIGRATE

Source Command: CLMRMCP1

Successfully submitted: CLMRMCP1
 Concurrent Request ID: 814042

[Kintana Delivero](#)
Concurrent Request Output

Completion Text	Normal completion
Parent Conc. Request ID	

Logfile: [NICKEL - d:\Oxa_Rlli\appl\vis5comm\admin\log\1814042.req](#)
 Outfile: [NICKEL - d:\Oxa_Rlli\appl\vis5comm\admin\out\o814042.out](#)

Figure C-2 Links to Concurrent Request Logs

Index

A

- AOL Dependent Objects
 - list of 133
- AOL Object Definitions
 - archiving 95
- Audience
 - intended 8

C

- Comparison Reports
 - about 19
- Concurrent Managers
 - before migrating 40
 - migrating 40
 - migrator parameters 42
- Concurrent Programs
 - before migrating 43
 - migrating 43
 - migrator parameters 44

D

- Descriptive Flexfields
 - before migrating 46
 - migrating 46
 - migrator parameters 48
- Document
 - about 8

E

- Exception Messages
 - list of 111

F

- Folders
 - before migrating 50
 - migrating 50
 - migrator parameters 51
- FSG Row/Column Sets
 - before migrating 53
 - migrating 52
 - migrator parameters 53
- Functions
 - before migrating 55
 - migrating 54
 - migrator parameters 56

G

- GUI Menus
 - before migrating 57
 - migrating 57
 - migrator parameters 58

H

- Help Text
 - before migrating 61
 - migrating 60
 - migrator parameters 61

K

- Kintana Deliver
 - executing Object*Migrator 138
 - introduction 137
 - processing Package 139
 - using with Object*Migrator 137

M

- Menus
 - before migrating 63
 - migrating 62
 - migrator parameters 63
- Messages
 - before migrating 66
 - migrating 65
 - migrator parameters 66
- Migration
 - capabilities 17
 - overview 15
 - viewing results 28
- Migration Audit Reports
 - about 20
- Migrator Parameters
 - common 32
 - Concurrent Managers 42
 - Concurrent Programs 44
 - Descriptive Flexfields 48
 - Folders 51
 - FSG Row/Column Sets 53
 - Functions 56
 - GUI Menu 58

Help Text 61
 Menus 63
 Messages 66
 Named SQL 68
 Printer Definitions 70
 Profile Options 74
 QuickCodes 76
 Request Groups 78
 Request Sets 80
 Responsibilities 82
 Users 86
 Values Sets 91
 Zooms 92

N

Named SQL
 before migrating 68
 migrating 68
 migrator parameters 68

O

Object
 comparison 103
 prerequisites 23

Object Archive
 about 18
 purging 99
 retrieving an object from 97
 running version detail report 98
 saving an object to 96

Object Comparison
 database instance and the object archive 106
 Differences Exist 104
 from two Oracle instances 105

 in the object archive 107
 Only in Dest DB 104
 Only in Source DB 104
 overview 103
 running 105

Object Types
 about 16

Object*Migrator
 about 7
 executing from Kintana Deliver 138
 installing 21
 running request 23
 using with Kintana Deliver 137

Oracle Applications
 opening 22

P

Password
 obtaining 21

Printer Definitions
 before migrating 70
 migrating 69
 migrator parameters 70

Profile Options
 before migrating 72
 migrating 72
 migrator parameters 74

Q

QuickCode
 before migrating 76

QuickCodes
 migrating 76
 migrator parameters 76

R

Report
 Comparison 108
 Object Archive Version Detail 98
 running to migrate objects 24

Report Groups
 migrating 77

Report Sets
 migrating 80

Reports
 comparison about 19
 migration audit about 20

Request Groups
 before migrating 78
 migrating 77
 migrator parameters 78

Request Sets
 migrating 80
 migrator parameters 80

Responsibilities
 before migrating 82
 migrating 82
 migrator parameters 82

U

User Name
 obtaining 21

Users
 before migrating 85
 migrating 85
 migrator parameters 86

V

Value Sets

- before migrating 90
- migrating 89
- migrator parameters 91

Version Control

- about 18

Z

Zooms

- before migrating 92
- migrating 92
- migrator parameters 92

