

opentext™

OpenText™ Project and Portfolio Management (PPM)

Software version: 12.2.4

Object Migrator Guide and Reference

Go to Help Center online

<https://admhelp.microfocus.com/ppm/>



Document release date: October 2023

Send Us Feedback



Let us know how we can improve your experience with the Object Migrator Guide and Reference.

Send your email to: admdocteam@opentext.com

Legal Notices

© Copyright 2025 Open Text.

The only warranties for products and services of Open Text and its affiliates and licensors ("Open Text") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Disclaimer

Certain versions of software accessible here may contain branding from Hewlett-Packard Company (now HP Inc.) and Hewlett Packard Enterprise Company. This software was acquired on September 1, 2017 by Micro Focus and is now offered by OpenText, a separately owned and operated company. Any reference to the HP and Hewlett Packard Enterprise/HPE marks is historical in nature, and the HP and Hewlett Packard Enterprise/HPE marks are the property of their respective owners.

Contents

Getting Started with Object Migrator	7
Getting Started Overview	7
New Features Overview	8
Architecture Overview	8
Example Object Migrator Environment	10
Object Migrator Concurrent Programs	10
Deploying Objects Across Database Instances	13
Migration Capabilities	14
Version Control Overview	14
Reports Overview	15
Comparison Reports	16
Migration Audit Reports	16
Installation and Upgrade Requirements	17
Considerations for Version 12.2.4	17
Upgrade Requirements	18
File System and Tablespace Requirements	18
Platform Requirements	19
Instance Requirements	19
Distributed Database Option - Recommended	20
Application Naming Requirements	20
Version Control Considerations	20
Sizing Requirements for Object Archive Tables	21
Obtaining a User Name and Password	21
Installing and Configuring Object Migrator	22
Pre-Installation Setup	22
Registering Object Migrator as a custom application	22
Creating Tablespaces and Rollbacks	26
Granting more privileges to the CLM user	27
Installing Object Migrator	27
Configuring Your System to Work with Object Migrator	31
Creating and Registering Database Links	31
Adding Object Migrator Concurrent Programs	34
Running the Maintain Object Migrator Views program	35
Validating Installation	36
Registering Custom Applications	37
Preparing Patch File System	38

KUpgrading Object Migrator	39
Optional Configurations for Object Migrator	43
Setting Up a Separate Responsibility for Migrations	43
Setting Up an Unrestricted Migrator	47
Using Object Migrator Without the Distributed Database Option	53
Migrating and Comparing Objects	55
Running the Object Migrator Request	55
Viewing the Migration Results	59
Auditing Migration Executions	62
Comparing Objects	63
Running the Object Comparison	64
Sample Comparison Report	66
Maintaining Object Migrator	68
Maintaining Database Links	68
Maintaining Value Sets	68
Maintaining Object Migrator Views	69
Using Object Migrator Custom Views	70
Switching to Non-Validated Mode	71
open_links Database Parameter	72
Managing Custom Applications	73
Purging the Object Migrator Interface Tables	74
Purging the Object Migrator Archive Details	76
Recompiling the Descriptive Flexfields	77
Maintaining the Data Model	77
Setting Up Object Migrator Security	77
Managing Database Instances	78
Adding a New Database Instance	78
Removing a Database Instance	79
Using Version Control	80
Archiving AOL Object Definitions	80
Saving an Object to the Object Archive	80
Retrieving an Object from the Object Archive	81
Running the Object Archive Version Detail Report	82
Purging the Object Archive	84
Using Object Migrator with PPM Deployment Management	87
Using Object Migrator with PPM Deployment Management Overview	87

Running Object Migrator from PPM Deployment Management	88
Process Packages in PPM Deployment Management	90
 Appendix 2: AOL-Dependent Objects	 91
 Appendix 2: Migration Rules and Parameters	 95
Common Migrator Parameters	95
Concurrent Managers	105
Before Migrating Concurrent Managers	105
Concurrent Managers Migrator Parameters	106
Concurrent Programs	107
Before Migrating Concurrent Programs	107
Concurrent Programs Migrator Parameters	108
Descriptive Flexfields	109
Before Migrating Descriptive Flexfields	109
Descriptive Flexfields Migrator Parameters	110
Folders	112
Before Migrating Folders	112
Folders Migrator Parameters	112
FSG Row/Column Sets	113
Before Migrating FSG Row/Column Sets	113
FSG Row/Column Sets Migrator Parameters	113
Functions	114
Before Migrating Functions	114
Functions Migrator Parameters	115
GUI Menus	116
Before Migrating GUI Menus	116
GUI Menus Migrator Parameters	116
Messages	118
Before Migrating Messages	118
Messages Migrator Parameters	118
Printer Definitions	119
Before Migrating Printer Definitions	119
Printer Definitions Migrator Parameters	119
Profile Options	120
Before Migrating Profile Options	120
Profile Options Migrator Parameters	122
QuickCodes (AOL)	123
Before Migrating QuickCode	123
QuickCodes Migrator Parameters	123
Request Groups (Report Groups)	124

Before Migrating Request Groups	124
Request Groups Migrator Parameters	124
Request Sets (Report Sets)	125
Before Migrating Request Sets	125
Request Sets Migrator Parameters	125
Responsibilities	126
Before Migrating Responsibilities	126
Responsibilities Migrator Parameters	126
Users	129
Before Migrating Users	129
Users Migrator Parameters	129
Value Sets	133
Before Migrating Value Sets	133
Value Sets Migrator Parameters	134
Appendix 2: Exception Messages	135

Getting Started with Object Migrator

- ["Getting Started Overview" below](#)
- ["New Features Overview" on the next page](#)
- ["Architecture Overview" on the next page](#)
- ["Object Migrator Concurrent Programs" on page 10](#)
- ["Deploying Objects Across Database Instances" on page 13](#)
- ["Migration Capabilities" on page 14](#)
- ["Version Control Overview" on page 14](#)
- ["Reports Overview" on page 15](#)

Getting Started Overview

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

Every Oracle E-Business Suite customer (regardless of whether they are currently implementing, upgrading, or maintaining Oracle E-Business Suite) has the challenge of configuring AOL objects to meet its needs. As each object is configured, the object must be migrated to the multiple environments (Development, QA, Staging, Production) the customer uses to manage its E-Business Suite implementation.

Object Migrator can help you:

- Automate setup and deployment

Each AOL setup deployment does not require the re-keying of the setup information—a complex, repetitive, and error prone manual task requiring hours of database administrator, system administrator, and developer time.

- Run diagnostic reports to verify proper migration and identify any errors

Identifying if a migration was successful requires manual verification, as does fixing an improper migration. With Object Migrator, comparison reports list the differences between objects in different environments, and validations minimize errors.

- Rollback improperly migrated objects

If an improper object is still migrated, Object Migrator's version control functionality enables the change to be automatically rolled back.

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

New Features Overview

Object Migrator version 12.2.4 includes the following enhancement:

Resolves the issue that arises when attempting to submit a concurrent request during the migration of GUI menus, concurrent programs, value sets, or concurrent managers. The problem lies in the insufficient capacity of the error message parameter, preventing the error message from adequately storing the error details.

Architecture Overview

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

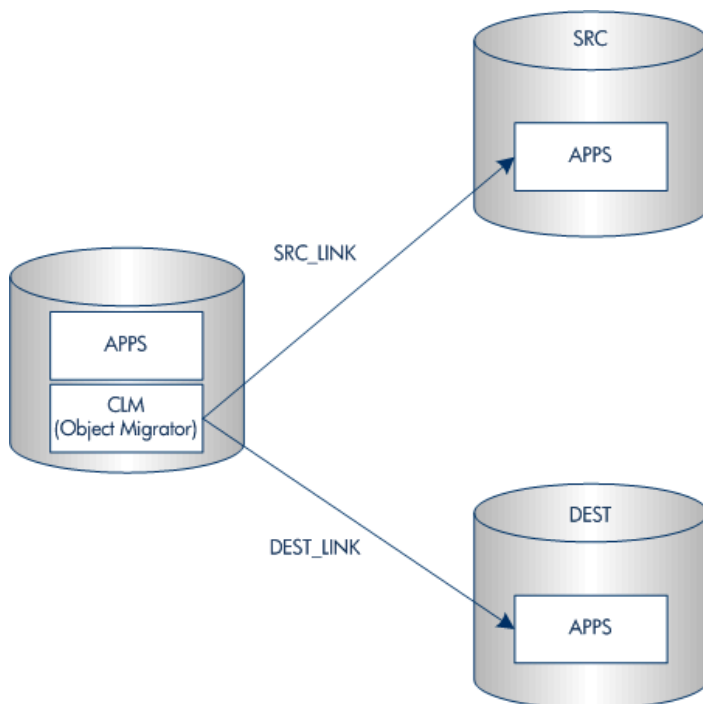
Object Migrator has its own database schema. This schema contains the Object Migrator database objects and communicates with the `APPS` schema. These are required in order to migrate Application Object Library (AOL) setup data, as well as retain archived versions of objects and capture migration audit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

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

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

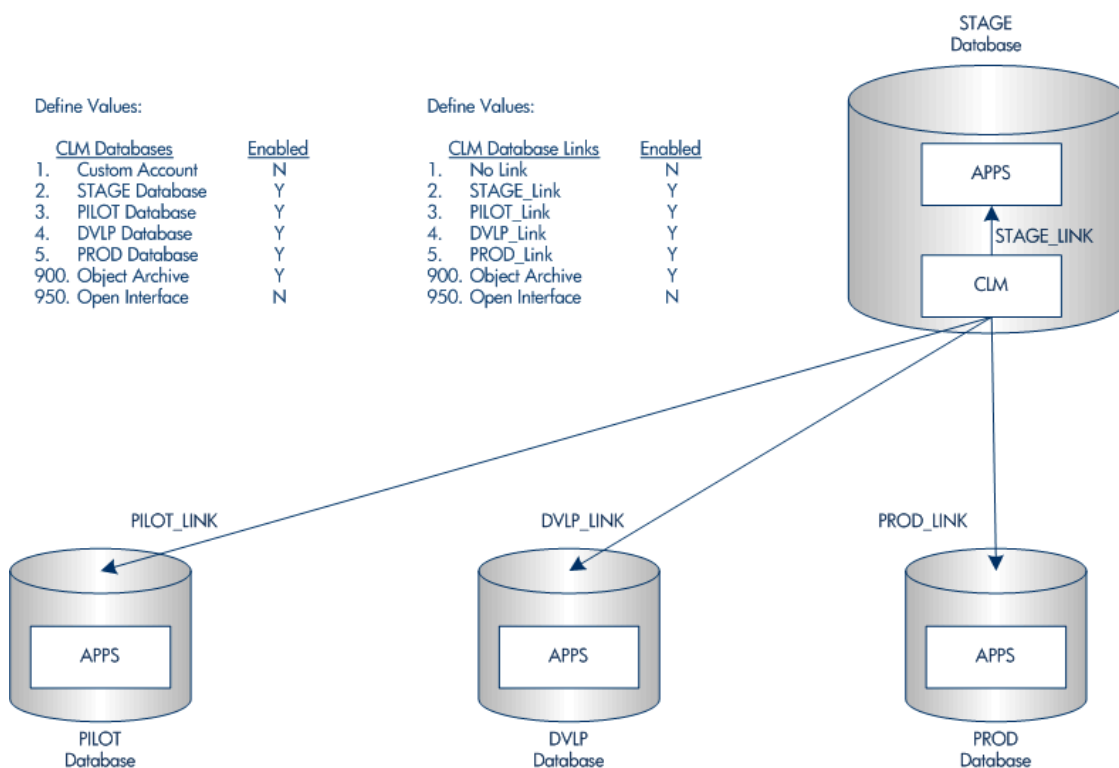
The following shows the relationship between the Object Migrator schema, `APPS`, source, and destination databases.

Figure 1-1. Object Migrator architecture



Example Object Migrator Environment

The following Object Migrator environment has four database instances named STAGE, DVLP, PILOT, and PROD. STAGE is a small staging database that only has Application Object Library installed and is never refreshed.



Object Migrator Concurrent Programs

Object Migrator installs several report files that are run as concurrent programs from the Submit Requests form. Oracle reports files have rdf file extensions. There is one report file for each of the Object Migrator modules. Some Object Migrator modules are not applicable to all Oracle E-Business Suite releases.

Each object type that is supported for migration has a separate concurrent program. There are also separate concurrent programs for administration tasks. Each of the concurrent programs included with Object Migrator is included in the following table.

Table 1-1. Object Migrator Concurrent Program and Report Names

Concurrent Program Name	Report Name	Notes
Object Migrator Audit Report	CLMRMAD1.rdf	
Maintain Object Migrator Views	CLMRDVW1.rdf	
Migrate Concurrent Managers	CLMRMCM1.rdf	<p>When migrating to an Oracle E-Business Suite Release 12 destination, the Concurrent Programs migrator submits the following concurrent requests:</p> <ul style="list-style-type: none"> • Build Concurrent Request Queue View • Verify <p>Oracle has added FAILOVER_MAX and FAILOVER_MIN columns. Those columns and any data in them is migrated with this migrator.</p>
Migrate Concurrent Programs	CLMRMCP1.rdf	Supports migrations of multiple organization category functionality.
Migrate Descriptive Flexfields	CLMRMDF1.rdf	Supports the synchronize reference field functionality.
Migrate Folders	CLMRMFD1.rdf	Supports behavior mode functionality.
Migrate FSG Row/Col Sets	CLMRMFS1.rdf	Supports migration from Release 11 to Release 11i, but not E-Business Suite Release 12.
Migrate Functions	CLMRMFN1.rdf	IREP columns are maintained by the Integration Repository. IREP columns are not moved by this migrator because they are necessary to the proper function of your database applications.
Migrate GUI Menus	CLMRMG1.rdf	
Migrate Messages	CLMRMMS1.rdf	

Table 1-1. Object Migrator Concurrent Program and Report Names, continued

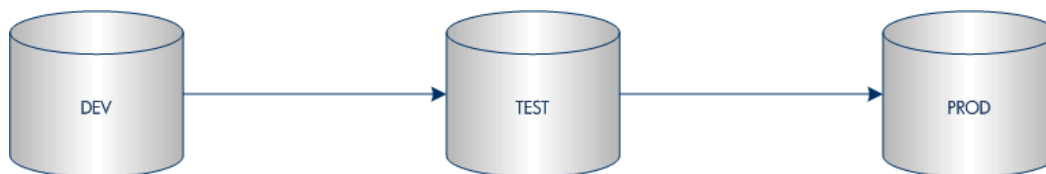
Concurrent Program Name	Report Name	Notes
Migrate Printer Definitions	CLMRMPT1.rdf	
Migrate Profile Options	CLMRMPO1.rdf	Supports the hierarchy type access level for the server responsibility.
Migrate QuickCodes	CLMRMCL1.rdf	Supports a new column named ASSIGN_LEAF_ONLY.
Migrate Report Groups	CLMRMRG1.rdf	
Migrate Report Sets	CLMRMRS1.rdf	
Migrate Responsibilities	CLMRMRP1.rdf	Supports migration of the mobile applications responsibilities.
Migrate Users	CLMRMUS1.rdf	
Migrate Value Sets	CLMRMVS1.rdf	
Migration Comparison Report	CLMRMMC1.rdf	
Object Archive Version Detail Report	CLMRMVR1.rdf	
Purge Object Archive Versions	CLMRDPV1.rdf	
Purge Object Migrator Interface Tables	CLMRDPG1.rdf	

Deploying Objects Across Database Instances

Individual applications in the Oracle E-Business Suite rely on the configuration of common objects (AOL objects) within their database instances. When customizing or configuring Oracle E-Business Suite applications, it is often necessary to deploy these objects from one instance to another.

For example, you may have one instance for customizing your Oracle E-Business Suite (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 1-2. Migration overview" below](#). Manually re-keying these configurations would be both time-consuming and error prone.

Figure 1-2. Migration overview



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.

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). You can migrate or compare objects. For each program, users specify parameters detailing information about how to run the program.

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.

Migration Capabilities

Object Migrator supports the following capabilities 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 E-Business Suite
- Migration simulation to identify issues with data setups without actually migrating data (can run a migration simulation as a “dry run” before an actual migration)

Version Control Overview

You can use Object Migrator to save data to 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. You can use Object Migrator reports to view complete version histories of a specific object or a group of objects.

The object archive is the version control repository used by 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 version label is defined when the Object Migrator program is run. 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
- Gives the new object a revision number that tracks the number of times a particular object has been archived

A particular object 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.

Reports Overview

The following reports are available in Object Migrator:

- Comparison reports
- Migration audit reports

Table 1-2. Reports used in system maintenance and administration

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

Comparison Reports

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

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

Some child entities are not compared in the comparison reports. Major exceptions are noted in "[Migration Rules and Parameters](#)" on page 95.

Migration Audit Reports

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.

Installation and Upgrade Requirements

- ["Considerations for Version 12.2.4" below](#)
- ["File System and Tablespace Requirements" on the next page](#)
- ["Platform Requirements" on page 19](#)
- ["Instance Requirements" on page 19](#)
- ["Distributed Database Option - Recommended" on page 20](#)
- ["Application Naming Requirements" on page 20](#)
- ["Version Control Considerations" on page 20](#)
- ["Obtaining a User Name and Password" on page 21](#)

Considerations for Version 12.2.4

Consider the following:

- Do not customize any forms related to Object Migrator including Concurrent Programs Definitions and Submit Concurrent Request Parameters. Customizations to these forms are not supported.
- Do not convert Object Migrator custom schema (CLM) using the Oracle Applications Tablespace Model (OATM). When you migrate entities using OATM, use the OATM utility and exclude the Object Migrator custom schema.
- The 12.2.4 directory contains scripts and code for your Oracle E-Business Suite Release 12.2 installation.

For example, when you extract the 12.2.4 Object Migrator installation bundle, a directory named `CLM_12_2_4/install/12.2.4` is created. This directory contains scripts for the installation of the Object Migrator product for Oracle E-Business Suite Release 12.2.

- Oracle Applications 10.x, including Migrate Zooms, Migrate Help Text, Migrate Menus, and the Migrate Named SQL migrators, is no longer supported.

In Oracle E-Business Suite 12.0.0 and later releases, the Migrate FSG Row/Col Sets migrator is not supported. For more information on specific migrators, see ["Object Migrator Concurrent Programs" on page 10](#).

Upgrade Requirements

This section lists all requirements before upgrading to Object Migrator 12.2.4.

- The PPM Extension for Oracle E-Business Suite and PPM Center applications should be at version 9.40 or later.

Earlier versions may work. For details, contact support.

- Oracle E-Business Suite releases.

Release	Release Version and Patch Details
11	All
11i	11.5.1-11.5.10.2 and the FND.H Mini Pack
12	12.0.0-12.0.4

- Use a custom SQL account to run the Object Migrator. It works better for separating Object Migrator objects and standard Oracle E-Business Suite application objects.
- Version 7.5 of Object Migrator backs up the data for archived users during the upgrade process. Make sure there is adequate database space before beginning the upgrade.

File System and Tablespace Requirements

Object Migrator requires a minimum amount of disk space to be successfully installed or upgraded.

Object Migrator requires a minimum amount of tablespace in order to function properly.

For details of the requirements, *System Requirements and Compatibility Matrix*.

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

Platform Requirements

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

Installation of Object Migrator utilizes a shell script to drive the installation. The location in which Object Migrator is being installed must support Bourne shell executions.

Instance Requirements

Object Migrator runs as concurrent programs using Oracle E-Business Suite. The instance hosting Object Migrator must be able to run a concurrent program.

Object Migrator requires that Oracle E-Business Suite have the Applications Object Library (AOL) installed.

Object Migrator:

- Resides as a custom application within the Oracle E-Business Suite instance
- Uses database tables to store object information

Object Migrator should be installed in an instance that will not be refreshed. For example, install Object Migrator on a production instance or an instance dedicated to hosting Object Migrator.

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

Distributed Database Option - Recommended

Object Migrator uses database links to access all databases. The Distributed Database Option is required to insert, delete, or update data on a remote database. It is strongly recommended that this option be active on all the databases being used as sources and destinations. For information on how to install Object Migrator without the Distributed Database Option, see ["Using Object Migrator Without the Distributed Database Option" on page 53](#).

Note: If using Object Migrator with Deployment Management, the Distributed Database Option is required.

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

Application Naming Requirements

For fresh installation, ensure that there are no applications with the short name `CLM` registered in Oracle E-Business Suite. In addition, ensure that no applications with the name Object Migrator are registered in Oracle E-Business Suite.

For upgrade, ensure that the current version of Object Migrator is 12.2 or later, and that the short name of Object Migrator should be `CLM` in Oracle E-Business Suite, and the SQL account used to run Object Migrator should also be `CLM`.

Version Control Considerations

Object Migrator uses archive tables to store version information for AOL objects. These tables reside on the instance where Object Migrator is installed.

We recommend using version control functionality on only one installation. If Object Migrator is installed on multiple databases and uses the version control functionality on all the installations, multiple AOL object repositories will exist.

Sizing Requirements for Object Archive Tables

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

If periodic storage of versions of entire groups of objects is planned rather than only objects that change, then more tablespace will be required than the recommended 30 megabytes.

We recommend starting with a 30 megabyte allocation and then adding tablespace as needed.

Obtaining a User Name and Password

Each user who is going to run Object Migrator must have an Oracle E-Business Suite 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.

Installing and Configuring Object Migrator

- ["Pre-Installation Setup" below](#)
- ["Installing Object Migrator" on page 27](#)
- ["Configuring Your System to Work with Object Migrator" on page 31](#)
- ["Preparing Patch File System" on page 38](#)

Pre-Installation Setup

Before beginning the Object Migrator installation, you need to complete the instructions in this section.

Registering Object Migrator as a custom application

As part of setting up custom application, use the Oracle AD Splicer utility (adsplce) to register Object Migrator as a custom application of Oracle E-Business Suite Release 12.2. This ensures that Object Migrator is set up for online patching.

For more information about EBS 12.2, refer to [Developing and Deploying Customizations in Oracle E-Business Suite Release 12.2](#) (Doc ID 1577661.1) at Oracle Support website.

Note:

- Do not use the Applications window to register applications in EBS 12.2.
- When installing or upgrading to EBS Release 12.2, run adsplce only after you apply the 12.2.3 or later Release Update Pack (12.2.2 is also a secure code level for adsplce, but we only certified 12.2.3). Running adsplce before your instance is at the 12.2.3 code levels may cause file synchronization issues.
- Run adsplce on the Run edition file system and ensure that the EBS instance is not in an online patching cycle.

Follow these steps to register Object Migrator as a custom application using adsplice:

1. Download Patch 3636980 "Support Diagnostics (IZU) patch for AD Splice" from the [My Oracle Support](#) website.
2. Manually copy the three .txt files from the **3636980\izu\admin** directory to your own temporary directory.
3. Rename izuprod.txt to clmprod.txt, and izuterr.txt to clmterr.txt.
4. In newprods.txt, clmprod.txt, and clmterr.txt, change all lowercase izu to clm and all uppercase IZU to CLM (keep the case sensitivity).
5. Change all references to prodid 278 to your own unique number. It is recommended you pick a number above 50,000 that is not already in use.
6. In clmterr.txt, change Oracle_Support_Diagnostic_Tools to Object_Migrator. Make sure the application short name CLM is not duplicate with any existing application short names and no existing application is with the name Object Migrator.
7. Copy clmprod.txt, clmterr.txt, and newprods.txt to the \$APPL_TOP/admin directory.
8. Change to the \$APPL_TOP/admin directory.
9. Run adsplice.
10. After you run adsplice, verify the creation of the customer application in the database by doing the following:
 - a. Connect to the Run Edition and log in to SQL*Plus as APPS user, for example:

```
sqlplus APPS/APPS
```
 - b. Run the following SQL query to confirm that the fnd_application table has one row:

```
select * from fnd_application where  
application_short_name = 'CLM';
```
 - c. Run the following SQL query to confirm that the application name is Object Migrator:

```
select application_name from fnd_application_tl where
```

```
language='US' and application_id in (select  
application_id from fnd_application where  
application_short_name = 'CLM')
```

- d. Run the following SQL query to check the product installations table has one row for your custom product. This statement should return one row only:

```
select * from fnd_product_installations where  
APPLICATION_ID in (select application_id from  
fnd_application where application_short_name = 'CLM')
```

- e. Run the following SQL query to check the database user. This statement should return one row only:

```
select * from dba_users where username = 'CLM';
```

- f. You should also be able to log in to the database as user `CLM` with password `CLM`.

```
sqlplus CLM/CLM
```

11. To ensure the new environment file is picked up, log out then log back in again as user `applmgr`. Do the following:

- a. When connecting to the Run Edition file system, run the following command to confirm `$CLM_TOP` is set correctly:

```
echo $CLM_TOP
```

For example, the value of `$CLM_TOP` is

`/u01/oracle/VIS/fs1/EBSapps/appl/clm/12.0.0`, in which `fs1` is the name of the current Run Edition file system.

- b. Run the following command to check the file system is created correctly:

```
ls $CLM_TOP
```

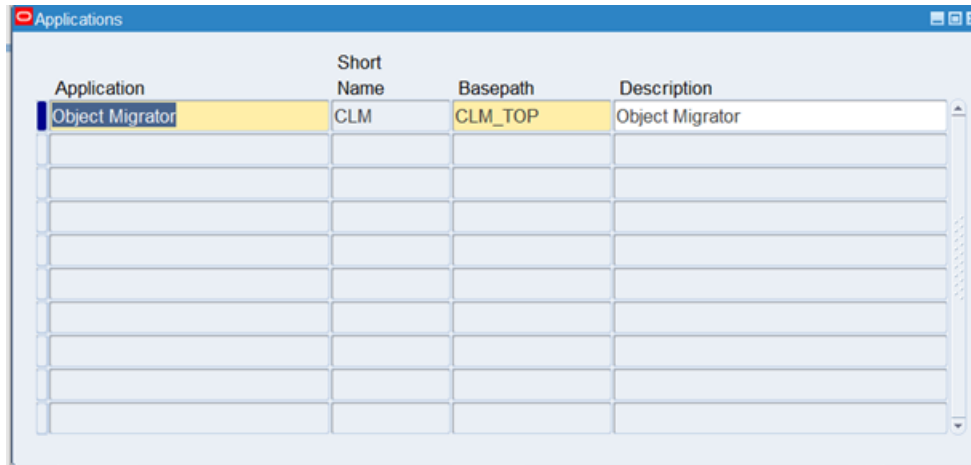
This command should return the following directory list:

```
admin log mesg out sql
```

You can find the CLM application from Oracle EBS by doing the following:

1. Log in to the Oracle E-Business Suite for this database instance.
2. Through the system administrator responsibility, navigate to the registered application. For example, **System Administrator** > **Application** > **Register**.
3. From the menu, select **View** > **Find**, and search for the application named `Object Migrator`.

The system returns the following result.



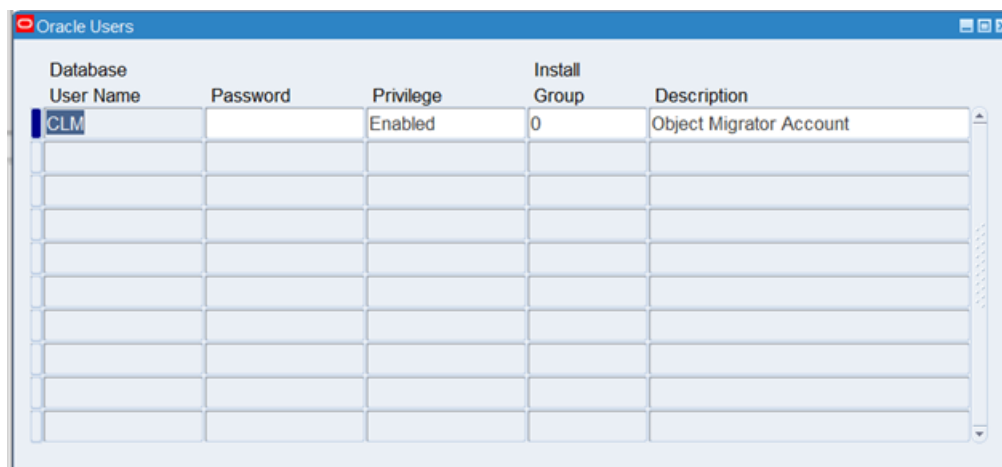
The screenshot shows a window titled 'Applications' with a table containing the following data:

Application	Short Name	Basepath	Description
Object Migrator	CLM	CLM_TOP	Object Migrator

You can find the CLM user from Oracle EBS by doing the following:

- a. Log in to the Oracle E-Business Suite for this database instance.
- b. Through the system administrator responsibility, navigate to the registered user. For example, **System Administrator** > **Security** > **Oracle** > **Register**.
- c. From the menu, select **View** > **Find**, and search for the database user named `CLM`.

The system returns the following result.



For more information about creating custom applications, refer to section 1 to section 3 of [Creating a Custom Application in Oracle E-Business Suite Release 12.2](#) (Doc ID 1577707.1) at the Oracle Support website.

Creating Tablespaces and Rollbacks

We recommend that Object Migrator have separate tablespaces. If Object Migrator will reside in a production Oracle E-Business Suite instance, then it must have separate tablespaces.

To create tablespaces and rollbacks:

1. Create the following tablespaces, or ensure the existing tablespace is increased to accommodate the extra demand.

Tablespace	Recommended Size	Function
CLM_DATA	60 MB	Saves interface and archive data
CLM_INDEX	30 MB	Saves index data

See the size requirements for the tablespaces in ["File System and Tablespace Requirements" on page 18](#).

You can name the tablespace anything you want.

2. Add at least one rollback segment per additional tablespace, to support Object Migrator migration transactions.

Create them in a separate tablespace reserved for rollback segments and use the OPTIMAL size constraint to make sure that the rollback segments automatically deallocate space as it becomes free.

For example:

```
CREATE PUBLIC ROLLBACK SEGMENT RB1
    TABLESPACE RB_TS
    STORAGE (OPTIMAL 60 M);
```

Granting more privileges to the CLM user

To grant more privileges to the CLM user, run the following SQL query with the sysdba account in the Oracle EBS database:

```
GRANT CREATE VIEW TO CLM;
GRANT CREATE PROCEDURE TO CLM;
GRANT CREATE SYNONYM TO CLM;
-- Suppose the tablespace names are CLM_DATA and CLM_INDEX
ALTER USER CLM quota unlimited on CLM_DATA;
ALTER USER CLM quota unlimited on CLM_INDEX;
```

Installing Object Migrator

All object names created by the install scripts begin with CLM. If an application with a name of CLM already exists, please contact support.

Install Object Migrator in the Run Edition of the Oracle EBS instance that hosts Object Migrator and ensure the instance is not in an online patching cycle. Suppose the root directory of your Oracle E-Business Suite application-tier installation is /u01/oracle/VIS, you can run the following command to source the Run Edition:

```
source /u01/oracle/VIS/EBSapps.env run
```

For details about how to connect to the Run Edition, refer to [Developing and Deploying Customizations in Oracle E-Business Suite Release 12.2](#) (Doc ID 1577661.1) at Oracle Support website.

To install Object Migrator:

1. Ensure you have completed the steps in ["Pre-Installation Setup" on page 22](#).
2. Connect to the Run Edition of the EBS instance that hosts Object Migrator.
3. Place the bundle file `CLM_1224.zip` into the folder where the Object Migrator installation is to be staged. For example, you can do the following:
 - a. Create a directory under `$CLM_TOP` by running the following command:

```
mkdir $CLM_TOP/install
```

- b. Upload `CLM_1224.zip` to the `$CLM_TOP/install` directory.
4. Extract the Object Migrator bundle by running the following command:
For example for UNIX, uncompress and unarchive the files using the following commands:

```
unzip CLM_1224.zip
```

This creates an installation folder, `CLM_12_2_4`, with subfolders of:

- `install/12.2.4` containing all the installation scripts to install Object Migrator
 - `srw` containing all of the Object Migrator executables
5. Navigate to the `install` folder.

For example:

```
cd $CLM_TOP/install/CLM_12_2_4/install/12.2.4
```

6. Run the `CLM_INSTALL` script.

The installation script will start and use the Bourne shell. The script validates the information typed at each prompt. The install process can be stopped by typing a period (.) at any of the prompts.

The script runs all the Object Migrator installation scripts to:

- Create database objects used by Object Migrator
- Create value sets and register concurrent programs used to run migrators, system maintenance, and administration reports
- Create a limited number of grants and synonyms to the APPS account to facilitate request submission

- Define incompatibilities between different Object Migrator programs
- Grant privilege of Object Migrator-owned database objects to the Oracle EBS APPS account
- Create synonyms for Object Migrator-owned database objects under the Oracle EBS APPS account

- a. Open a command window.
- b. Run the following command.

```
chmod ugo+x CLMMAIN_DYN
```

- c. Run the script. For example:

```
sh CLM_INSTALL
```

- d. Type values for the following prompts.

Prompt	Description
Have you registered the Object Migrator user using adsplice ? (Y/N) :	Confirm (Y or N) whether the Object Migrator SQL account has been created using adsplice.
Enter username of Object Migrator sql account (. to exit):	The user name for the Object Migrator SQL account that was created and registered using adsplice in Oracle E-Business Suite.
Enter the password for \$clm_acct sql account (. to exit):	The password for this Object Migrator SQL account.
Please reenter for verification:	The password for this Object Migrator SQL account.
Please enter the connect string (ex. PROD, DEV, etc.) for the Object Migrator database.	Specifies the database on which Object Migrator is being installed. <ul style="list-style-type: none">◦ When using SQL*Net 1.0, type the entire two-task database connect string. (For example, T:dev_ machine:MIGRATE)◦ When using SQL*Net 2.0 or later, type the database identifier (usually ORACLE_SID).

Prompt	Description
Enter the tablespace for the Object Migrator tables (. to exit):	All Object Migrator tables will go into the specified tablespace. The SQL account must have permissions to write to this tablespace; for example, <code>CLM_DATA</code> .
Enter the tablespace for the Object Migrator indexes (. to exit):	All Object Migrator indexes will go into the specified tablespace. The SQL account must have permissions to write to this tablespace; for example, <code>CLM_INDEX</code> .
Enter username of APPS user - usually APPS (. to exit):	The user name for the Oracle E-Business Suite APPS account.
Enter the password for \$apps_acct user (. to exit):	The password for this APPS account.
Please reenter for verification:	The password for this APPS account.

7. The script outputs log messages as standard output on your screen and saves the output to a log file named `clm_log.out` in the current folder.
8. Move the `rdf` files from the `srw` folder of the Object Migrator bundle to their permanent location under `$CLM_TOP` by following these steps:
 - a. Source the Run Edition and make sure you are in the Run Edition file system.
 - b. Create the report folder under `$CLM_TOP` by running the following command:

```
mkdir -p $CLM_TOP/reports/US
```
 - c. Copy the report files by running the following command:

```
cp $CLM_TOP/install/CLM_12_2_4/srw/* $CLM_TOP/reports/US
```
 - d. List all files under the `$CLM_TOP/reports/US` directory by running the following command:

```
ll $CLM_TOP/reports/US
```

You will see 22 files with the extension `.rdf`.
9. (Optional) The installation files are not needed for continued use of Object Migrator, but it is recommended that you save them.

Object Migrator must be configured before it can be used to migrate data. For these instructions, see ["Configuring Your System to Work with Object Migrator" below](#).

Configuring Your System to Work with Object Migrator

After installing Object Migrator, you need to configure Oracle E-Business Suite system to work with it. Configuration consists of the following:

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

The example given is a simplified case that assumes the system administrator will be able to access all Object Migrator functions. See ["Setting Up Object Migrator Security" on page 77](#) for more information about security options.

Creating and Registering Database Links

1. To create database links on CLM schema, use either of the following two methods:
 - With TNS name
 - a. Add an entry in the `tnsnames.ora` file for all the databases that are used in your Object Migrator implementation. Make sure that you update the `tnsnames.ora` file on each separate database server.

Note: The database configurations (for example, `tnsnames.ora`) must support connection using the database link.

- b. Create database links to all valid source and destination databases using the APPS account.

Perform this step in the instance containing Object Migrator using the SQL account under which Object Migrator was installed. For example:

```
SQL> create database link PROD_LINK
2 connect to APPS identified by APPS/APPS
3 using 'PROD' ;
```

- Without TNS name

Create database links to all valid source and destination databases using the APPS account.

The following is an example:

```
CREATE DATABASE LINK PROD_LINK
CONNECT TO APPS
IDENTIFIED BY APPS
USING '(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
(HOST=16.186.74.75)(PORT=1521))
(CONNECT_DATA=(SID=VIS)))';
```

2. Create the same database links on the Oracle EBS APPS schema.

Note: If you are using a version earlier than Object Migrator 12.2, you need to create database links on only the CLM schema. However, on Object Migrator release 12.2 and later, you must create the same database links on both the CLM schema and the Oracle EBS APPS schema.

If you update database links on the CLM schema, you must update the corresponding database links on the EBS APPS schema.

3. Test each link after creating it by executing the following in SQL*Plus:

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

4. Add each database in your Object Migrator environment to the CLM_ DATABASES value set, including the database where Object Migrator is installed.

- a. Select the **System Administrator** responsibility.
- b. Navigate to the Define Value Set Values window. For example, **Application > Validation > Values**.
- c. Type CLM_DATABASES in the **Name** field.
- d. Click **Find**.
- e. Query all the records in the **Values, Effective** tab. To bring up the following:
 - **Object Migrator account** initially set to **Disabled**
 - **Object archive** for version control
 - **Open Interface** initially set as **Disabled**
- f. Add additional records in the **Values, Effective** tab, one per database instance. The values typed in the **Value** and **Translated Value** fields need to be integers with no leading zeroes.
- g. Use the **Description** to identify the database instance to users.
- h. Clear the Enabled option for the Value **1** if it is not already disabled.

The screenshot shows the 'Segment Values' window with the 'Value Set' tab selected. The 'Name' field is 'CLM_DATABASES'. Below the tabs, there is a table with the following data:

Value	Translated Value	Description	Enabled	From	To
1	1	Object Migrator Acct	<input type="checkbox"/>		
50	50	PPMQAVM78_EBS121_LINK	<input checked="" type="checkbox"/>		
60	60	PPMPCOE05_EBS122_LINK	<input checked="" type="checkbox"/>		
70	70	PPMQAVM36_EBS122_LINK	<input checked="" type="checkbox"/>		
900	900	Object Archive	<input checked="" type="checkbox"/>		
950	950	Open Interface	<input type="checkbox"/>		

Buttons at the bottom: Define Child Ranges, Move Child Ranges, View Hierarchies.

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

- Select the **System Administrator** responsibility.
- Navigate to the Define Value Set Values window. For example, **Application > Validation > Values**.
- Type CLM_DB_LINKS in the **Name** field.
- Click **Find**.
- Query all the records in the **Values, Effective** tab.
- Add and enable all the database links that you created.

Make sure that the **Value** column matches with the corresponding **Value** column in the CLM_DATABASES value set.

The screenshot shows the 'Segment Values' window with the 'Value Set' tab selected. The 'Name' field is 'CLM_DB_LINKS' and the 'Database Links' field is empty. The 'Values (CLM_DB_LINKS)' section is expanded, showing the 'Values, Effective' tab. The table below lists the values and their descriptions, with checkboxes for 'Enabled'.

Value	Translated Value	Description	Enabled	From	To
1	1	No Link	<input type="checkbox"/>		
50	50	PPMQAVM78_EBS121_LINK	<input checked="" type="checkbox"/>		
60	60	PPMPCOE05_EBS122_LINK	<input checked="" type="checkbox"/>		
70	70	PPMQAVM36_EBS122_LINK	<input checked="" type="checkbox"/>		
900	900	Object Archive	<input checked="" type="checkbox"/>		
950	950	Open Interface	<input type="checkbox"/>		

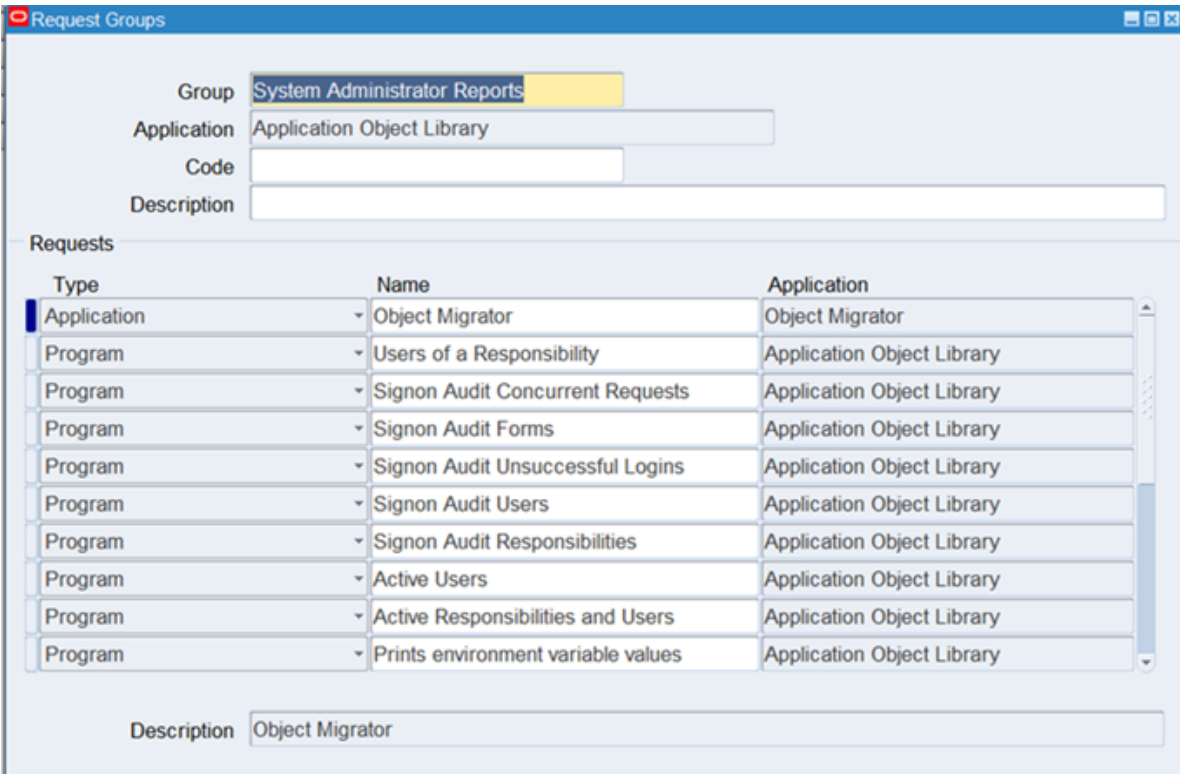
Adding Object Migrator Concurrent Programs

Object Migrator concurrent programs can be added individually or as a group to a responsibility. Depending on the security constraints for your company, you may want to create a dedicated Object Migrator responsibility. These instructions assume that programs are added to an existing responsibility.

To add the Object Migrator application or individual module concurrent programs to the appropriate request security groups:

1. Navigate to the Define Request Security Group window. For example, **Security > Responsibility > Request**.
2. Query a request group called **System Administrator Reports**.
3. On the **Requests** tab insert a record by clicking **Add Row** from the toolbar.
4. Select **Application** in the **Type** field.
5. Select **Object Migrator** as the application. Or, add individual programs.

For a list of individual programs, see ["Object Migrator Concurrent Programs" on page 10](#).



The screenshot shows the 'Request Groups' window with the 'System Administrator Reports' group selected. The 'Requests' tab is active, displaying a table of requests. The first row is selected, showing 'Application' as the Type, 'Object Migrator' as the Name, and 'Object Migrator' as the Application.

Type	Name	Application
Application	Object Migrator	Object Migrator
Program	Users of a Responsibility	Application Object Library
Program	Signon Audit Concurrent Requests	Application Object Library
Program	Signon Audit Forms	Application Object Library
Program	Signon Audit Unsuccessful Logins	Application Object Library
Program	Signon Audit Users	Application Object Library
Program	Signon Audit Responsibilities	Application Object Library
Program	Active Users	Application Object Library
Program	Active Responsibilities and Users	Application Object Library
Program	Prints environment variable values	Application Object Library

6. For optional setups, see ["Optional Configurations for Object Migrator" on page 43](#).

Running the Maintain Object Migrator Views program

The Maintain Object Migrator Views program defines the views used by some Object Migrator parameters when submitting Object Migrator migrations. It keeps

the Object Migrator views current with the number and statuses of your databases.

We recommend scheduling this program to run once a day.

For details regarding the support of these views, see ["Maintaining Object Migrator Views" on page 69](#).

To run the Maintain Object Migrator Views program:

1. If you have more than four databases in your Object Migrator environment, increase the value for the open_links database parameter to at least 20 and bounce the database. For more information, see ["open_links Database Parameter" on page 72](#).
2. Open the Submit Requests form. For example, **Other > Requests > Run**.
3. Query for and launch the **Maintain Object Migrator Views** program, using the following values.

Parameter Name	Value
Type	Request
Name	Maintain Object Migrator Views
Report Only	No . Re-create the Object Migrator views based on the current enabled values in the CLM_DATABASES and CLM_DB_LINKS value sets and the current statuses of your remote databases.

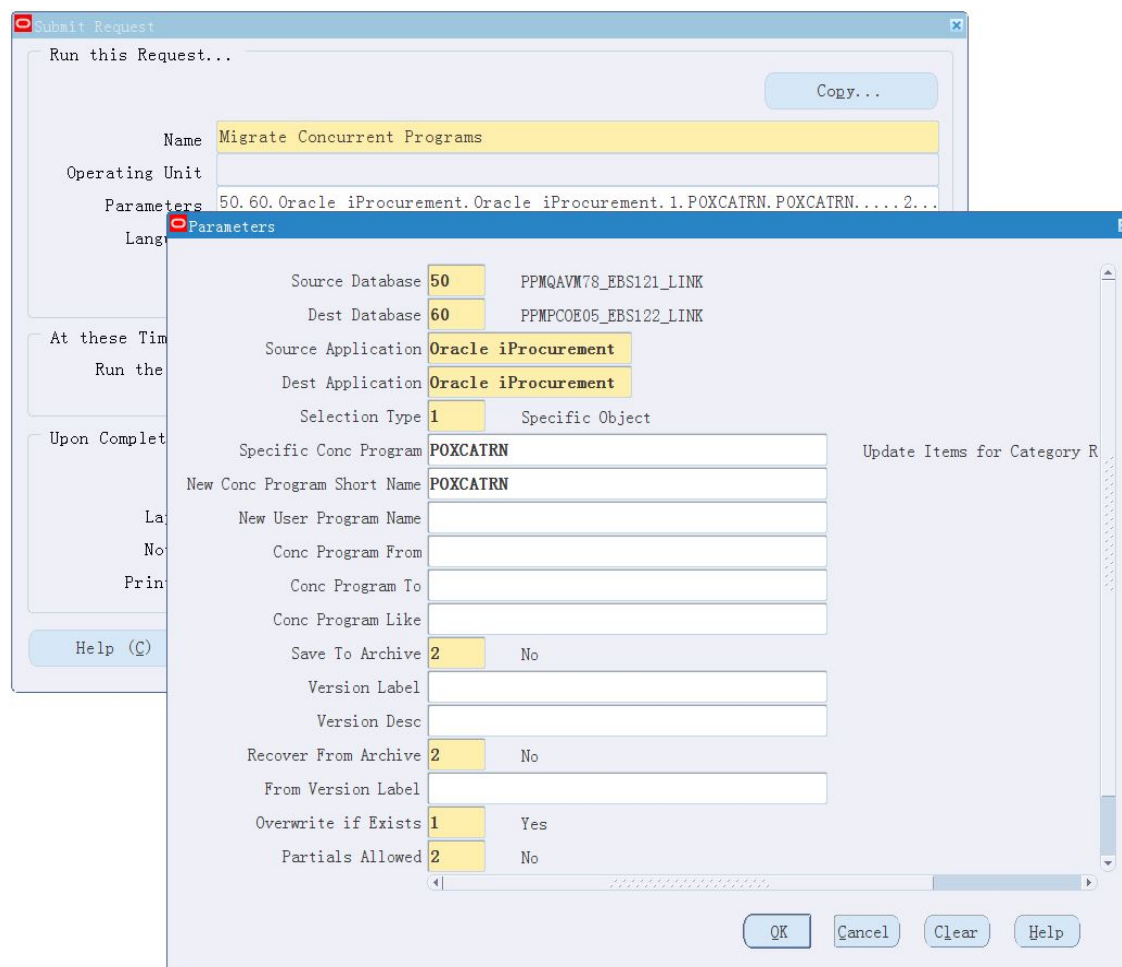
4. Make sure to view the report output of this program. The output lists all the enabled databases for use with Object Migrator and any errors encountered while trying to connect to these databases.

For information regarding the support of these views, see ["Maintaining Object Migrator Views" on page 69](#).

Validating Installation

This step is optional, but recommended.

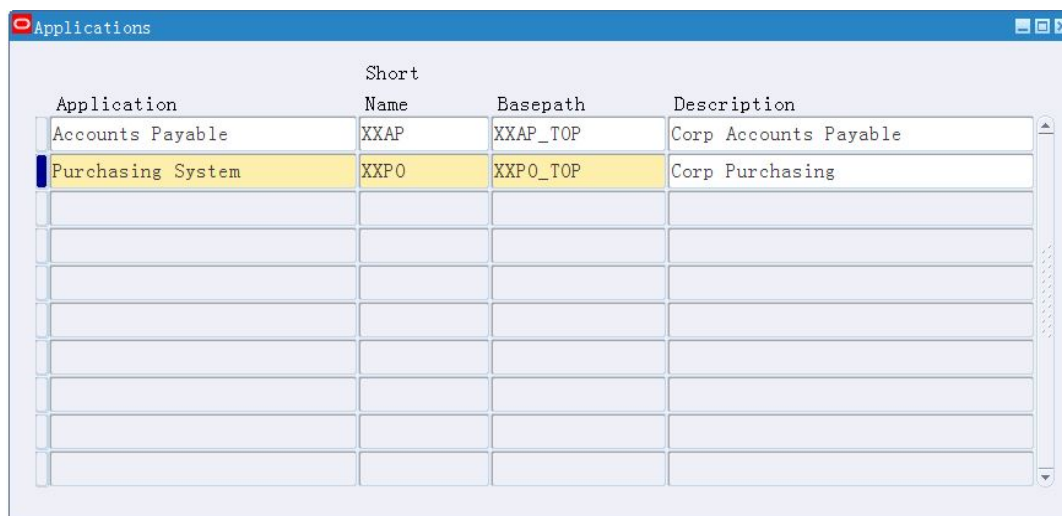
To validate your installation, launch one of the Object Migrator programs with **Report Only** set to **Yes**.



Registering Custom Applications

If you have custom applications for which you want to migrate data, you must register them within Oracle Applications.

1. In the Object Migrator instance, define any custom applications for which data will be migrated. For more information, see ["Managing Custom Applications" on page 73](#).



Application	Short Name	Basepath	Description
Accounts Payable	XXAP	XXAP_TOP	Corp Accounts Payable
Purchasing System	XXPO	XXPO_TOP	Corp Purchasing

2. See ["Optional Configurations for Object Migrator" on page 43](#) for additional optional configurations.

Preparing Patch File System

To prepare a patch file system, follow these steps:

1. Connect to the Run Edition of the EBS instance that hosts Object Migrator, and start an online patching cycle by running the following command:

```
adop phase=prepare
```

After the preparation phase is completed, the same folder structure for CLM is created on the Patch Edition file system.

For example, source the Patch Edition file system, make sure that you are on the Patch Edition file system, and then run the following command:

```
ls $CLM_TOP
```

This command returns the following directory list:

```
admin log mesg out sql
```

2. Copy the `rdf` files to the Patch Edition file system by follow these steps:
 - a. Source the Patch Edition file system and make sure that you are on the Patch Edition file system.

In that way, `$CLM_TOP` is pointing to the top directory of CLM custom application in the patch file system; for example,

```
/u01/oracle/VIS/fs2/EBSapps/appl/clm/12.0.0.
```

- b. Create the new directory `reports/US` under `$CLM_TOP` using the following command:

```
mkdir -p $CLM_TOP/reports/US
```

- c. Copy the `rdf` files from the Run Edition file system to the Patch Edition file system. For example, you can use the following command:

```
cp  
/u01/oracle/VIS/fs1/EBSapps/appl/clm/12.0.0/reports/US  
/*  
/u01/oracle/VIS/fs2/EBSapps/appl/clm/12.0.0/reports/US
```

3. Connect to the Run Edition of the EBS instance that hosts Object Migrator, and complete the patching cycle by running the following command:

```
adop phase=finalize,cutover,cleanup
```

KUpgrading Object Migrator

Upgrade Object Migrator in the run edition of the Oracle EBS instance that hosts Object Migrator and ensure the instance is not in an online patching cycle.

Suppose the root directory of your Oracle E- Business Suite application-tier installation is `/u01/oracle/VIS`, you can run the following command to source the run edition:

```
source /u01/oracle/VIS/EBSapps.env run
```

For details about how to connect to the run edition, refer to [Developing and Deploying Customizations in Oracle E-Business Suite Release 12.2](#) (Doc ID 1577661.1) at the Oracle Support website.

To upgrade Object Migrator:

Note: When you are upgrading Object Migrator, you can keep EBS server up, so that no extra actions need to be taken after the upgrade.

1. Connect to the run edition of the EBS instance that hosts Object Migrator. Make sure the instance is not in an online patching cycle.
2. Place the bundle file `CLM_UPG1224.zip` into the top folder of Object Migrator. For example, you can do the following:

- Create a directory under `$CLM_TOP` by running the following command:

```
mkdir $CLM_TOP/upgrade
```

- Upload `CLM_UPG1224.zip` to the `$CLM_TOP/upgrade` directory.

3. Extract the Object Migrator bundle.

For example for UNIX, extract the files using the following command:

```
unzip CLM_UPG1224.zip
```

This process creates the `$CLM_TOP/upgrade/CLM_12_2_4` directory containing the upgrade files. The subfolders in the directory include `upgrade/12.2.4`, containing all the scripts needed to upgrade Object Migrator.

4. Navigate to the script folder.

For example: `cd $CLM_TOP/upgrade/CLM_12_2_4/12.2.4`

5. Run the `CLM_UPGRADE` script.

The upgrade script starts and uses the Bourne shell. The script validates the information typed in each prompt. The upgrade process can be stopped by typing a period (.) in any prompts.

The script also implements the following:

- Remove HP or HPE from the development company of Object Migrator.
- Incorporate the code change of the defect QCCR1L56472 into Object Migrator.

To run the `CLM_UPGRADE` script:

- a. Open a command window.
- b. Run the script.

For example: `sh ./CLM_UPGRADE`

- c. Type values for the following prompts.

Prompt	Description
Have you registered the Object Migrator user using adsplice ? (Y/N) :	Confirm (YorN) whether the Object Migrator SQL account has been created using adsplice.
Enter user name of Object Migrator sql account (. to exit):	The user name for the Object Migrator SQL account that was created and registered using adsplice in Oracle E- Business Suite.
Enter the password for \$clm_acct sql account (. to exit):	The password for this Object Migrator SQL account.
Please reenter for verification:	The password for this Object Migrator SQL account.
Please enter the connect string (ex. PROD, DEV, etc.) for the Object Migrator database.	Specifies the database on which Object Migrator is being installed. <ul style="list-style-type: none"> ◦ When using SQL*Net 1.0, type the entire two-task database connect string. (For example, T:dev_ machine:MIGRATE) . ◦ When using SQL*Net 2.0 or later, type the database identifier (usually ORACLE_SID).
Enter the tablespace for the Object Migrator tables (. to exit)	All Object Migrator tables goes into the specified tablespace. The SQL account must have permissions to write to this tablespace. For example, CLM_DATA.
Enter the tablespace for the Object Migrator indexes (. to exit)	All Object Migrator indexes goes into the specified tablespace. The SQL account must have permissions to write to this tablespace. For example, CLM_INDEX.
Enter user name of APPS user - usually APPS (. to exit):	The user name for the Oracle E-Business Suite APPS account.
Enter the password for \$apps_acct user (. to exit):	The password for this APPS account.
Please reenter for verification:	The password for this APPS account.

6. The script generates log messages as standard output on your screen and saves the output to a log file named `clm_log.out` in the current folder.
7. (Optional) The upgrade files are not needed for continued use of Object Migrator, but it is recommended that you save them.

Optional Configurations for Object Migrator

- ["Setting Up a Separate Responsibility for Migrations" below](#)
- ["Setting Up an Unrestricted Migrator" on page 47](#)
- ["Using Object Migrator Without the Distributed Database Option" on page 53](#)

Setting Up a Separate Responsibility for Migrations

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

To set up a separate responsibility for migrations:

1. Sign on to the Oracle E-Business Suite on the instance in which the Object Migrator is installed.
2. Select the **System Administrator** responsibility.
3. Navigate to the Define Menu form. For example, **Application > Menu**.
4. Create a new menu called **CLM_MIGRATE**.

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

The screenshot shows the 'Menus' form with the following fields and values:

- Menu: CLM_MIGRATE
- User Menu Name: CLM_MIGRATE
- Menu Type: (empty)
- Description: Object Migrator Main Menu
- Icon: (empty)

A 'View Tree...' button is located to the right of the 'Menu' field.

Seq	Prompt	Submenu	Function	Description	Grant
1	Migrate		Run Reports	Migrate Objects	<input checked="" type="checkbox"/>
2	View Reports		View All Concurrent Requ	View Migration	<input checked="" type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>

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

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

The screenshot shows the 'Menus' form with the following fields and values:

- Menu: CLM_MAIN
- User Menu Name: CLM_MAIN
- Menu Type: (empty)
- Description: Object Migrator Main Menu

A 'View Tree...' button is located to the right of the 'Menu' field.

Seq	Prompt	Submenu	Function	Description	Grant
1	Navigate	CLM_MIGRATE		CLM_MIGRATE	<input checked="" type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>

6. Navigate to the Define Request Group form. For example, **Security > Responsibility > Request**.
7. Create a new request group called CLM_TOOLS.

Add the Object Migrator application to the request group. Or add individual programs if the group should include limited functionality.

The screenshot shows a web application window titled "Request Groups". It contains several input fields and a table. The "Group" field is set to "CLM_TOOLS". The "Application" field is set to "Object Migrator". The "Code" field is empty. The "Description" field is set to "Object Migrator". Below these fields is a section titled "Requests" which contains a table with three columns: "Type", "Name", and "Application". The first row of the table is highlighted in yellow and contains the values "Application", "Object Migrator", and "Object Migrator". There are several empty rows below the first row. At the bottom of the form, there is a "Description" field set to "Object Migrator".

Type	Name	Application
Application	Object Migrator	Object Migrator

8. Navigate to the Define Responsibility form. For example, **Security > Responsibility > Define**.

9. Define a new responsibility called `Object Migrator`.

Add the request group and menu previously defined. And it is possible to include a standard menu and define explicit menu exclusions.

Responsibilities

Responsibility Name: Object Migrator
Application: Object Migrator
Responsibility Key: OBJECT_MIGRATOR
Description: Object Migrator Responsibility

Effective Dates
From: 01-JUN-2021
To:

Available From
☒ Oracle Applications
☐ Oracle Self Service Web Applications
☐ Oracle Mobile Applications

Menu: CLM_MAIN
Web Host Name:
Web Agent Name:

Data Group
Name: Standard
Application: System Administration

Request Group
Name: CLM_TOOLS
Application: Object Migrator

Menu Exclusions | Excluded Items | Securing Attributes

Type	Name	Description
Function		

10. Navigate to the Define Concurrent Programs window. For example, **Concurrent > Program > Define**.
 - Modify default values of the Object Migrator concurrent programs, as desired.
 - If there are Object Migrator programs not in use, disable them to prevent them from appearing in select lists.
 - If different groups of users should have different access to a program, copy the program, and modify it appropriately.

See ["Setting Up an Unrestricted Migrator" on the next page](#) for a sample of these types of changes.
11. Navigate to the Define Application User window. For example, **Security > User > Define**.
12. Assign the custom responsibility defined in [step 9](#) to the users who would be doing the migrations.

The screenshot shows the 'Users' window with the following fields and values:

- User Name: SYSADMIN
- Password: (empty)
- Description: System Administrator
- Status: Active
- Password Expiration: Days (empty), Accesses (empty), None (selected)
- Person: (empty)
- Customer: SYSADMIN SYSADMIN
- Supplier: (empty)
- E-Mail: (empty)
- Fax: (empty)
- Effective Dates: From 01-JAN-1951, To (empty)

The 'Direct Responsibilities' tab is selected, showing a table of responsibilities:

Responsibility	Application	Description	Security Group	Effective Dates From	Effective Dates To
iMeeting Guest user responsi	iMeeting (Obsolete)		Standard	14-DEC-2001	
iMeeting End User responsib	iMeeting (Obsolete)		Standard	14-DEC-2001	
iMeeting System Configure r	iMeeting (Obsolete)		Standard	14-DEC-2001	
iMeeting System Monitor resp	iMeeting (Obsolete)		Standard	14-DEC-2001	
Object Migrator	Object Migrator		Standard	01-JUN-2021	

Setting Up an Unrestricted Migrator

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

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

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

1. Navigate to the value set window.
2. Create a new value set called **CLM_DATABASES_UNRES** by copying the existing value set **CLM_DATABASES**.

The 'Value Sets' window displays the configuration for the 'CLM_DATABASES_UNRES' value set. The 'Value Set Name' is 'CLM_DATABASES_UNRES' and the 'Description' is 'Unrestricted Database Names'. The 'List Type' is 'List of Values' and the 'Security Type' is 'No Security'. Under 'Format Validation', the 'Format Type' is 'Char', 'Maximum Size' is '30', and 'Precision' is empty. There are checkboxes for 'Numbers Only (0-9)', 'Uppercase Only (A-Z)', and 'Right-justify and Zero-fill Numbers (0001)', all of which are unchecked. 'Min Value' and 'Max Value' fields are empty. Under 'Value Validation', the 'Validation Type' is 'Independent'. An 'Edit Information' button is located at the bottom right.

- Remove the databases that are considered critical from the **CLM_DATABASES_UNRES** value set. The values in this new value set must match the values in the original value set for each database specified.

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

The 'Segment Values' window shows the 'Values' tab for the 'CLM_DATABASES_UNRES' value set. The 'Name' is 'CLM_DATABASES_UNRES' and the 'Description' is 'Unrestricted Database Names'. Below, the 'Values (CLM_DATABASES_UNRES)' section shows a table with columns: Value, Translated Value, Description, Enabled, From, and To. The first two rows are highlighted in blue.

Value	Translated Value	Description	Enabled	From	To
50	50	PPMQAVM78_EBS121_LINK	✓		
60	60	PPMPCOE05_EBS122_LINK	✓		
			✗		
			✗		
			✗		
			✗		
			✗		

Buttons at the bottom: 'Define Child Ranges', 'Move Child Ranges', and 'View Hierarchies'.

4. Copy all the existing Object Migrator concurrent programs to create identical concurrent programs for the unrestricted migrator.
 - a. Navigate to the Define Concurrent Programs window. For example, **Concurrent > Program > Define**.
 - b. Query Object Migrator in the **Program** field.
 - c. Use the Down Arrow key to review all the concurrent programs that belong to Object Migrator.

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

The screenshot shows the 'Concurrent Programs' window with the following details:

- Program:** Migrate Value Sets (Enabled)
- Short Name:** CLMRMVS1
- Application:** Object Migrator
- Description:** Migrate Value Sets
- Executable:**
 - Name:** CLMRMVS1
 - Method:** Oracle Reports
 - Options:** (empty)
 - Priority:** (empty)
- Request:**
 - Type:** Request
 - Incrementor:** CLMRMVS1
 - MLS Function:** Migrate Value Sets
 - Operating Unit Mode:** Normal
 - ☒ Use in SRS
 - ☐ Allow Disabled Values
 - ☐ Run Alone
 - ☐ Restart on System Failure
 - ☐ Enable Trace
 - ☐ NLS Compliant
 - ☐ Recalculate Default Parameters
- Output:**
 - Format:** Text
 - ☒ Save (G)
 - ☒ Print
 - Columns:** 132
 - Rows:** 45
 - Style:** Landscape
 - ☐ Style Required
 - Printer:** Printer
- Business Events:**
 - ☐ Request Submitted (Y)
 - ☐ Request Running
 - ☐ Post Processing Ended
 - ☐ Request On Hold
 - ☐ Program Completed
 - ☐ Request Completed (Z)
 - ☐ Request Resumed
 - ☐ Post Processing Started
- Buttons:** Copy to..., Session Control, Incompatibilities, Parameters (G)

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

- e. If your Oracle E-Business Suite release does not support Copy To functionality, copy the program using the Concurrent Programs Migrator.
 - i. Navigate to the Submit Requests window. For example, **Other > Requests > Run**.
 - ii. Launch the Concurrent Programs Migrator as shown in the following window to copy CLMRMVS1 and create a new program called CLMRMVS1_UNRES.
5. Assign the new value set to the new concurrent programs.
 - a. Navigate to the Define Concurrent Programs window. For example, **Concurrent > Program > Define**.
 - b. Query for the new concurrent program, CLMRMVS1_UNRES.
 - c. Click **Parameters**.
 - d. Change the value set associated with Destination Database from **CLM_DATABASES** to **CLM_DATABASES_UNRES**.

Concurrent Programs Migrator window showing the 'Concurrent Program Parameters' dialog for the 'Migrate Value Sets' program.

Program: Migrate Value Sets (Enabled)

Application: Object Migrator

Conflicts Domain: [Empty] Security Group: [Empty]

Seq	Parameter	Description	Enabled
5	Object Type	Object Type	<input checked="" type="checkbox"/>
10	Source Database	Source Database	<input checked="" type="checkbox"/>
15	Source DB Value	Source DB Value	<input checked="" type="checkbox"/>
20	Destination Database	Dest Database	<input checked="" type="checkbox"/>

Validation:

Value Set: CLM_OBJECT_TYPE_SHORT Description: Object Types

Default Type: SQL Statement Default Value: Select 'Value Set' from sys.d

☒ Required ☐ Enable Security Range: [Empty]

Display:

Display Size: 20 Description Size: 50

Concatenated Description Size: 25 Prompt: Object Type

Token: P_OBJECT_TYPE

6. Save your work.

7. Create a custom menu. See ["Setting Up a Separate Responsibility for Migrations" on page 43](#).
8. Create a request group for the unrestricted migrator.
 - a. Navigate to the Define Request Groups window. For example, **Security > Responsibility > Request**.
 - b. Create a new request group called **CLM_TOOLS_UNRESTRICTED**.
 - c. Associate all the unrestricted migrator concurrent programs to this request group.

Request Groups

Group: CLM_TOOLS_UNRESTRICTED

Application: Object Migrator

Code: CLM_TOOLS_UNRESTRICTED

Description: CLM_TOOLS_UNRESTRICTED

Requests

Type	Name	Application
Program	Migrate Value Sets	Object Migrator

Description: Migrate Value Sets

9. Create a new responsibility for the unrestricted migrator.
 - a. Navigate to the Define Responsibility window. For example, **Security > Responsibility > Define**.
 - b. Create a new responsibility called **Object Migrator - Unrestricted**.
 - c. Associate the new request group CLM_TOOLS_UNRESTRICTED to this responsibility

as shown in the following window.

Responsibilities

Responsibility Name: Object Migrator - Unrestricted
Application: Object Migrator
Responsibility Key: OBJECT_MIGRATOR_UNRESTRICT
Description: Unrestricted Object Migrator

Effective Dates: From 01-JUN-2021 To

Available From

- ☒ Oracle Applications
- ☐ Oracle Self Service Web Applications
- ☐ Oracle Mobile Applications

Menu: CLM_MAIN
Web Host Name:
Web Agent Name:

Data Group

Name: Standard
Application: Object Migrator

Request Group

Name: CLM_TOOLS_UNRESTRICTED
Application: Object Migrator

Menu Exclusions | Excluded Items | Securing Attributes

Type	Name	Description
Function		

10. Assign the new responsibility to users.
 - a. Navigate to the Define Application User window. For example, **Security > Use > Define**.
 - b. Search for the user to whom the responsibility will be assigned.
 - c. Add the **Object Migrator - Unrestricted** responsibility to the user.

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

Users

User Name: DEVELOPER
Password:
Description: Customization developer
Status: Active

Person:
Customer:
Supplier:
E-Mail:
Fax:

Effective Dates
From: 01-JUN-2021
To:

Direct Responsibilities | Indirect Responsibilities | Securing Attributes

Responsibility	Application	Description	Security Group	Effective Dates
				From To
Object Migrator - Unrestricted	Object Migrator		Standard	01-JUN-2021

Using Object Migrator Without the Distributed Database Option

We do not recommend using Object Migrator without the Distributed Database option.

Note: When Object Migrator is used with Deployment Management, the Distributed Database Option is required.

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

To work around this restriction, install Object Migrator on each of the instances that can potentially be a Destination Database. Then, whenever an object is to be moved to a given database, log on to the database and run Object Migrator there,

setting the Destination Database to be the current database. The Source Database can be a remote database because Object Migrator queries data from only the source database and Oracle allows queries across database links even without the Distributed Database Option.

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

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

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

- Install Object Migrator in a custom account, and instead of creating a link to the APPS account, grant all the appropriate objects from APPS to the custom account. Then, enable value 1 in CLM_DATABASES and rename it to **Current Database**. Contact support for a sample script to execute these grants.

Migrating and Comparing Objects

- ["Running the Object Migrator Request" below](#)
- ["Viewing the Migration Results" on page 59](#)
- ["Auditing Migration Executions" on page 62](#)
- ["Comparing Objects" on page 63](#)

Running the Object Migrator Request

Object Migrator migrations are run through the Submit Request window in Oracle E-Business Suite. By default, Object Migrator defines incompatibilities between migration programs and archive or interface purge programs. This protects the integrity of the migrations.

When 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.

Note: The following procedure reflects Oracle E-Business Suite Release 11i. Navigation paths may differ depending on your Oracle version.

To run the migration program:

1. Each Oracle AOL object type requires specific information and has unique rules for migrating between instances. See [" Migration Rules and Parameters" on page 95](#) for the prerequisites and rules related to specific object types.
2. 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 91](#).

3. Log on to Oracle E-Business Suite.

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

4. Open the Submit Requests window. In some cases, the form name may be Run Requests.

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.

5. In the **Name** field, query the migration program for the object you want to migrate.

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 displayed.

Submit Request

Run this Request...

Name: Migrate Concurrent Programs

Operating Unit:

Parameters:

Language: American English

Copy...

Language Setti...

Debug Options

At these Times...

Run the Job: As Soon as Possible

Schedule...

Upon Completion...

☒ Save all Output Files ☐ Burst Output

Layout:

Notify:

Print to: noprint

Options...

Delivery Opts

Help (C)

Submit

Cancel

The Parameters window opens for the selected object. Parameters vary according to the object type that you are migrating.

Parameters

Source Database 50 PPMQAVM78_EBS121_LINK

Dest Database 60 PPMPCOE05_EBS122_LINK

Source Application Human Resources

Dest Application Human Resources

Selection Type 1 Specific Object

Specific Conc Program ADSPEDKPOST ADS Danish Postal Code Upload

New Conc Program Short Name ADSPEDKPOST

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

OK Cancel Clear Help

6. Specify the parameters for the migration.

- Specify the Source Database and Dest Database for each migration.
- Specify any additional parameters required by the object type.

For a detailed description common and object-specific parameters, see ["AOL-Dependent Objects" on page 91](#).

It is possible to migrate one or multiple objects at a time, by changing the **Selection Type** field on the Parameter window. Options are:

- **Specific Object.** One object is selected for migration at this time. The specific object name needs to be specified.
- **Range of Objects.** A range of objects is selected for migration at this time. The alphabetic range to use needs to be specified.

- **Wild Card Match.** Multiple objects are migrated with object names matching a specified string. The string to match, including the percent (%) sign, needs to be specified.

7. Click **OK**.

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

Submit Request

Run this Request...

Copy...

Name: Migrate Concurrent Programs

Operating Unit:

Parameters: 50. 60. Human Resources. Human Resources. 1. ADSPEDKPOST. ADSPEDKPOST. 2.

Language: American English

Language Setti... Debug Options

At these Times...

Run the Job: As Soon as Possible Schedule...

Upon Completion...

☒ Save all Output Files ☐ Burst Output

Layout: Options...

Notify: Delivery Opts

Print to: noprint

Help (Q) Submit Cancel

8. Click **Submit**.

The migration will proceed based on the selected scheduling options.

9. Either submit another request or view the results of the migration in the **Requests** window.

Viewing the Migration Results

You can view the results of the migration using the standard Oracle E-Business Suite request result window. After a migration is complete, the Request window is displayed containing the details and logs related to the migration.

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

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

1. Navigate to the View Requests window. For example, **Requests > View**.

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

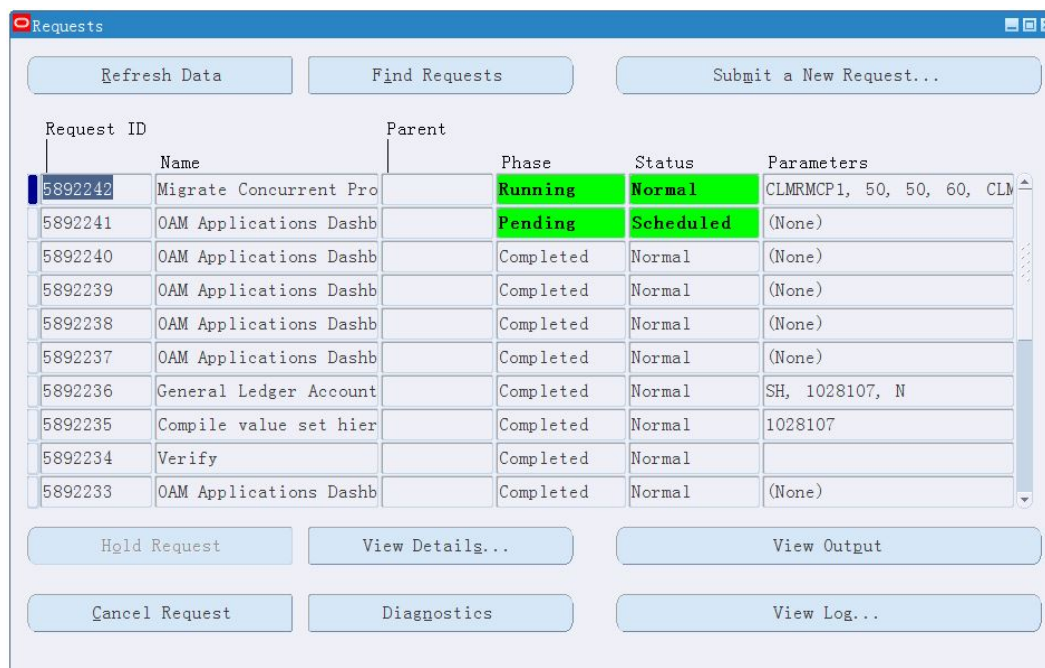
Order By **Request ID**

Select the Number of Days to View:

Submit a New Request... Clear Find

2. Select the criteria to locate the request. For example, **All My Requests**.
3. Click **Find**.

The Requests window opens.



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

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

The migration output and logs may contain some error messages indicating specific problems. See ["Exception Messages" on page 135](#) for a complete list of error messages and troubleshooting tips.

Auditing Migration Executions

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

To run Object Migrator Audit Report:

1. Open the Submit Requests window. For example, **Other > Requests > Run**.

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

2. Specify values for the following parameters.

Parameter Name	Value
Type	Request.
Name	Object Migrator Audit Report.
Order By	Select one of the following options: <ul style="list-style-type: none">• Run Date. Order the Audit Report by the Run Date of the migrator program.• Object Type. Order the Audit Report by the object type that was migrated. Within a specific object type, order the report by the program Run Date.
Run Date From	Type the starting Run Date of the programs to include in the audit report.
To	Type the ending Run Date of the programs to include in the audit report.
Specific Object Type	Type a specific object type to limit the audit report to a specific Object Migrator object type.

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

Comparing Objects

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

When Compare Only is set to **Yes**, Object Migrator extracts the object information into temporary tables. It then compares the detailed attributes of the objects and lists the differences in the output report. The report categorizes the differences as follows.

Category	Description
Only in Source Database	The object or object detail exists only in the source database. There is nothing in the destination database with the corresponding object name or object detail name.
Only in Dest Database	The object or object detail exists only in the destination database. There is nothing in the source database with the corresponding object name or object detail name.
Differences Exist	Objects or object details with this status either have differences with their specific attributes or have details where some sort of differences exist.

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, when comparing a range of concurrent programs on one database against the range on another database, if a concurrent program exists on the source database and not the destination database, it does not report on the parameters belonging to concurrent programs.

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

Some child entities are not compared, see "[Migration Rules and Parameters](#)" on [page 95](#).

Running the Object Comparison

To run the object comparison:

1. Log on to Oracle E-Business Suite.

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

2. Open the Submit Requests window.
 - a. From the menu, select **View > Request**.
 - b. Click **Submit a New Request**.
 - c. Select the **Single Request** radio button.
 - d. Click **OK**.

The Submit Request window opens.

3. In the **Name** field, select any object.

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

The Parameters window opens for the selected object. Parameters vary according to your selection.

4. To compare objects, use the following table as guidance to specify parameters as necessary to achieve the result set you want.

Parameter Name	Compare Objects in Two Databases	Compare Objects in a Database and the Object Archive	Compare Versions of an Object in the Object Archive
Source Database	The first instance containing the object to compare	Object Archive	Object Archive
Destination Database	The second instance containing the object to compare	The name of the database containing the object to compare	Object Archive
Recover From Archive		Yes	Yes
Save to Archive			No
Version Label			The version of the first object to be compared
From Version Label		Select the appropriate label	The second version of the object to be compared

5. In the **Compare Only** field, select **1 (Yes)** for Compare Only.
6. Specify any other parameters required to define the object or objects to compare. For a description of each parameter, see "[Migration Rules and Parameters](#)" on page 95.

To compare by individual objects, specify the object to compare. Object Migrator will ignore the value in New Object Name and instead use the value of Specific Object Name when retrieving information.

To compare by a range of objects, specify the range using Object From and Object To. If you specify a range, objects that do not exist on one database will be shown versus another database, and the differences for the objects that do exist on both databases.

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

7. Click **OK**.
8. After the objects are compared, the results are displayed in the output report.

Sample Comparison Report

The following parameters were used to construct the example:

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

The following 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

Object Migrator Guide and Reference

Migrating and Comparing Objects

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

Maintaining Object Migrator

- ["Maintaining Database Links" below](#)
- ["Maintaining Value Sets" below](#)
- ["Maintaining Object Migrator Views" on the next page](#)
- ["Using Object Migrator Custom Views" on page 70](#)
- ["Managing Custom Applications" on page 73](#)
- ["Purging the Object Migrator Interface Tables" on page 74](#)
- ["Purging the Object Migrator Archive Details" on page 76](#)
- ["Recompiling the Descriptive Flexfields" on page 77](#)
- ["Maintaining the Data Model" on page 77](#)
- ["Setting Up Object Migrator Security" on page 77](#)
- ["Managing Database Instances" on page 78](#)

Maintaining Database Links

If you change passwords you must keep Object Migrator database links updated.

To maintain database links:

1. Drop the appropriate database link.
2. Recreating it using the new password.
3. Run the **Maintain Object Migrator Views** program.

Maintaining Value Sets

If you can change databases and database links you must keep the CLM_ DATABASES and CLM_DB_LINKS value sets updated.

For more information, see ["Managing Database Instances" on page 78](#).

Maintaining Object Migrator Views

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

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

For more details on the Object Migrator views, see ["Using Object Migrator Custom Views" on the next page](#).

Run this program:

- Periodically on a regular schedule
- After any changes are made to the `CLM_DB_LINKS` or `CLM_DATABASES` value sets
- When an Object Migrator source or destination becomes available or unavailable

Note: Some objects are not applicable to all releases of Oracle E-Business Suite. If a particular object is not applicable to a given Oracle instance, the instance will be omitted from the view definition.

To run the Maintain Object Migrator Views program:

1. Open the Submit Requests form. For example, **Other > Requests > Run**.
2. Query for and launch the **Maintain Object Migrator Views** program.
This prompts the user for report parameters, allowing the user to run the program in Report Only mode.
3. Specify values for the following parameters.

Parameter Name	Value
Type	Request
Name	Maintain Object Migrator Views
Report Only	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • Yes. Do not modify the Object Migrator views. Just report on their current definition. • No. Re-create the Object Migrator views based on the current enabled values in the CLM_DATABASES and CLM_DB_LINKS value sets and the current statuses of your remote databases.

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

Using Object Migrator Custom Views

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

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

For example, suppose CLM_DATABASES has the following enabled values:

Value	Meaning
2	DEV
3	QA
4	PROD
900	Object Archive

CLM_DB_LINKS has the enabled values:

Value	Meaning
2	DEV_LINK
3	A_LINK
4	PROD_LINK
900	Object Archive

The view for the Users Migrator would be:

```

SELECT      2 db_value, user_name object_name
FROM        fnd_user@DEV_LINK
UNION
SELECT      3 db_value, user_name object_name
FROM        fnd_user@QA_LINK
UNION
SELECT      4 db_value, user_name object_name
FROM        fnd_user@PROD_LINK
UNION

```

<select clause to query the Object Archive tables against the local database>

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

Performance to retrieve the list of values for the first time can be affected by the number of database links that need to be traversed.

Run the ["Maintaining Object Migrator Views" on page 69](#) procedure often to ensure that your database links are valid. Database unions with invalid database links can cause the entire view to become invalid. This results in an error when the Object Migrator program is submitted.

To keep the Object Migrator views current with the current database status, schedule the Maintain Object Migrator Views program to run periodically. We suggest running it daily or weekly.

Switching to Non-Validated Mode

If significant problems are encountered with the Object Migrator views, switch to the non-validated object name mode using `CLMSMBS9.sql`. To revert back to validated parameters with a list of values, use `CLMSMBS8.sql`.

To switch to the non-validated object name mode:

1. Using SQL*Plus, log on to the Object Migrator database.
2. Navigate to the installation directory.
3. Run the `CLMSMBS9.sql` script against the Object Migrator SQL account.
4. When you are prompted for `inst_value_sets`, type Y.

For example:

```
sqlplus <OM_acct>/<OM_password>  
SQL> @CLMSMBS9.sql  
Enter value for inst_value_sets: Y  
Enter value for inst_value_sets: Y  
SQL> exit;
```

After you run this script, the object name parameters will have no validation or list of values.

To revert back to validated parameters with a list of values:

1. Using SQL*Plus, log on to the Object Migrator database.
2. Navigate to the installation directory.
3. Run the `CLMSMBS8.sql` script against the Object Migrator SQL account.
4. When you are prompted for `inst_value_sets`, type Y.

For example:

```
sqlplus <OM_acct>/<OM_password>  
SQL> @CLMSMBS8.sql  
Enter value for inst_value_sets: Y  
Enter value for inst_value_sets: Y  
SQL> exit;
```

open_links Database Parameter

The `open_links` database parameter controls the number of different databases that can be accessed through database links in a given session. It does not control how many times a specific link can be used in that session.

The default value for this parameter is 4. The suggested value is 20.

If you expect more than four databases defined in `CLM_DATABASES`, the value for this parameter needs to be increased in the Object Migrator instance.

To change the `open_links` parameter:

1. Stop the database.
2. Modify the init.ora file by adding (or modifying) the line:

```
open_links=20
```

3. Restart the database.

Managing Custom Applications

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

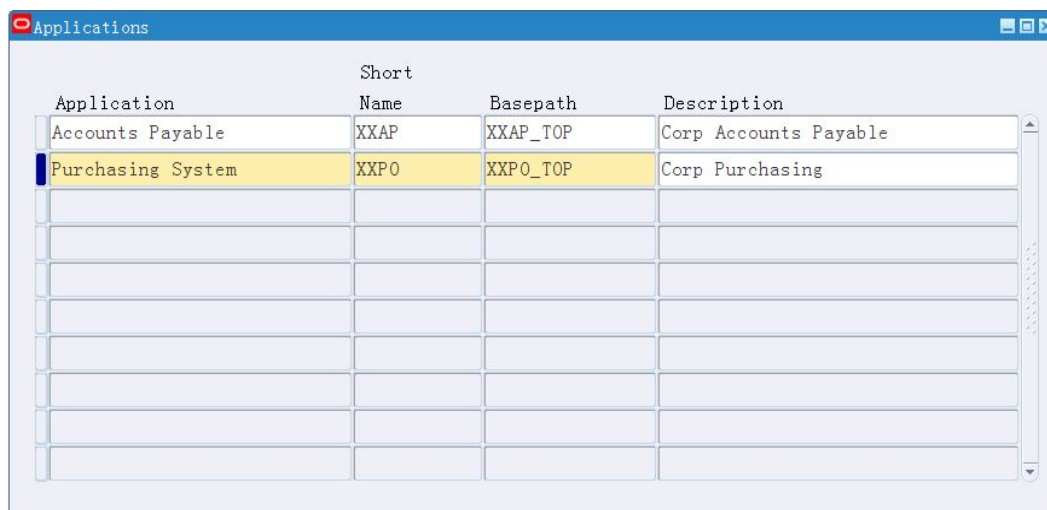
Register each custom application for which data will be migrated. Otherwise, Object Migrator will be unable to migrate data for the application.

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

To register each application where data will be migrated to or from:

1. Start Oracle E-Business Suite from the Object Migrator instance.
2. Navigate to the register form. For example, **Application > Register**.

The following is an example of registering custom applications.



Application	Short Name	Basepath	Description
Accounts Payable	XXAP	XXAP_TOP	Corp Accounts Payable
Purchasing System	XXPO	XXPO_TOP	Corp Purchasing

Purging the Object Migrator Interface Tables

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

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

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

To remove any fragmentation created through the insertion and deletion of records in the temporary tables, run this program at least once a month with Truncate Table set to Yes and all migrators selected.

To protect data integrity during migrations, the Object Migrator installation program automatically defines incompatibilities such that migrations cannot run while interface tables are being purged.

To run the Purge Object Migrator Interface Tables program:

1. Open the Submit Requests window. For example, **Other > Requests > Run**.

Use this form to launch the Purge Object Migrator Interface Tables program.

This will prompt the user for report parameters to restrict the tables to purge.

2. Specify the following parameters.

Parameter Name	Value
Type	Request
Name	Purge Object Migrator Interface Tables

Parameter Name	Value
Truncate Tables	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • Yes. Purge the tables by truncating them. This removes all the records from the table and removes and table fragmentation. Note that records cannot be selectively deleted based on age if this parameter is set to Yes. Unless there are reasons to retain records, set this parameter to Yes. • No. Purge the tables by deleting rows from the table.
Days Old	Type the least number of days old the records need to be in order to be eligible for deletion. This parameter is only relevant if Truncate Tables is set to No .
Conc Prog Migrator Tables	Indicate if the tables used by the Concurrent Programs Migrator will be purged.
Rep Set Migrator Tables	Indicate if the tables used by the Report Sets Migrator will be purged.
Value Set Migrator Tables	Indicate if the tables used by the Value Sets Migrator will be purged.
Desc Flex Migrator Tables	Indicate if the tables used by the Descriptive Flexfields Migrator will be purged.
Request Group Migrator Tables	Indicate if the tables used by the Request Groups Migrator will be purged.
Menu Migrator Tables	Indicate if the tables used by the Menus Migrator will be purged.
Responsibility Migrator Tables	Indicate if the tables used by the Responsibilities Migrator will be purged.
User Migrator Tables	Indicate if the tables used by the Users Migrator will be purged.
Profile Options Migrator Tables	Indicate if the tables used by the Profile Options Migrator will be purged.
FSG Row/Column Set Migrator Tables	Indicate if the tables used by the FSG Row/ Column Sets Migrator will be purged.

Parameter Name	Value
QuickCode Migrator Tables	Indicate if the tables used by the QuickCodes Migrator will be purged.
GUI Menu Migrator Tables	Indicate if the tables used by the GUI Menus Migrator will be purged.
Function Migrator Tables	Indicate if the tables used by the Functions Migrator will be purged.
Printer Migrator Tables	Indicate if the tables used by the Printer Definitions Migrator will be purged.
Folder Migrator Tables	Indicate if the tables used by the Folders Migrator will be purged.
Message Migrator Tables	Indicate if the tables used by the Messages Migrator will be purged.
Concurrent Manager Migrator Tables	Indicate if the tables used by the Concurrent Manager Definitions Migrator will be purged.

Purging the Object Migrator Archive Details

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

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

Use the Purge Object Archive Version program to purge entire archives (by Archive label) or subsets of an archive, selecting by object type or object name. The Object Migrator installation program automatically defines incompatibilities such that migrations cannot run while interface tables are being purged. This is to protect data integrity during migrations. For additional details on running this program, see ["Purging the Object Archive" on page 84](#).

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

Recompiling the Descriptive Flexfields

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

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

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

Maintaining the Data Model

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

Setting Up Object Migrator Security

Object Migrator uses standard Oracle E-Business Suite security functionality to control access to Object Migrator programs. ["Configuring Your System to Work with Object Migrator" on page 31](#) provides some default security setup information. Beyond this default security setup, you may want to consider additional internal security for Object Migrator. Your security tasks might include the following:

- **Request groups.** Set up a request group for each grouping of programs if there are different groups of users who should access different sets of programs. System administrators might have access to all programs, however business analysts and developers have access only to migrator and archive programs.

- **Responsibilities.** For each grouping of programs, set up a custom responsibility for the user group that will access the group of programs. See ["Optional Configurations for Object Migrator" on page 43](#) for detailed examples.
- **Users.** The Oracle E-Business Suite instance where Object Migrator is installed needs to have a logon defined for each user who will submit Object Migrator requests (this is optional if integrating with Deployment Management). Create the user and assign the appropriate responsibility that includes Object Migrator programs.

Managing Database Instances

- ["Adding a New Database Instance" below](#)
- ["Removing a Database Instance" on the next page](#)

Adding a New Database Instance

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

To register a new database instance:

1. On the Object Migrator instance, define a database link from the Object Migrator SQL account to the APPS account of the new database instance.
For examples, see ["Configuring Your System to Work with Object Migrator" on page 31](#).
2. Ensure that your `tnsnames.ora` file includes all the databases in your Object Migrator environment.
3. Ensure that you have properly updated database links.
4. Add a value to the `CLM_DATABASES` value set for the new database instance.
For examples, see ["Configuring Your System to Work with Object Migrator" on page 31](#).
5. Add a value to the `CLM_DB_LINKS` value set for the new database link. Use the exact database link name for the description.

Use the same number in the **Value** field that the database is assigned in the CLM_DATABASES value set. For examples, see ["Configuring Your System to Work with Object Migrator" on page 31](#).

6. Run the Maintain Object Migrator Views Concurrent program to:
 - Re-create all the object views used by Object Migrator for inclusion of the new database.
 - Perform a validation check for all enabled database names and links.

Removing a Database Instance

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

To remove a database instance:

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

Using Version Control

- ["Archiving AOL Object Definitions" below](#)
- ["Running the Object Archive Version Detail Report" on page 82](#)
- ["Purging the Object Archive" on page 84](#)

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.

For information on comparing archived objects to objects in other instances, see ["Comparing Objects" on page 63](#).

Saving an Object to the Object Archive

To save an Object Migrator supported objects to the object archive:

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

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

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

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

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

If you specify a new version label, Object Migrator will create the label for you.

If you select an existing version label, the object will be archived with other objects with the same label.

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

4. In the **Version Desc** field, type 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. Specify any other parameters required to define the migration. See "[Migration Rules and Parameters](#)" on page 95.
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

To restore an archived object definition to an instance, retrieve objects that were previously archived using the following procedure.

To retrieve an Object from the object archive:

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

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

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

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

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

4. Specify any other parameters required to define the migration. For a description of the parameters, see "[Migration Rules and Parameters](#)" on [page 95](#).
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.

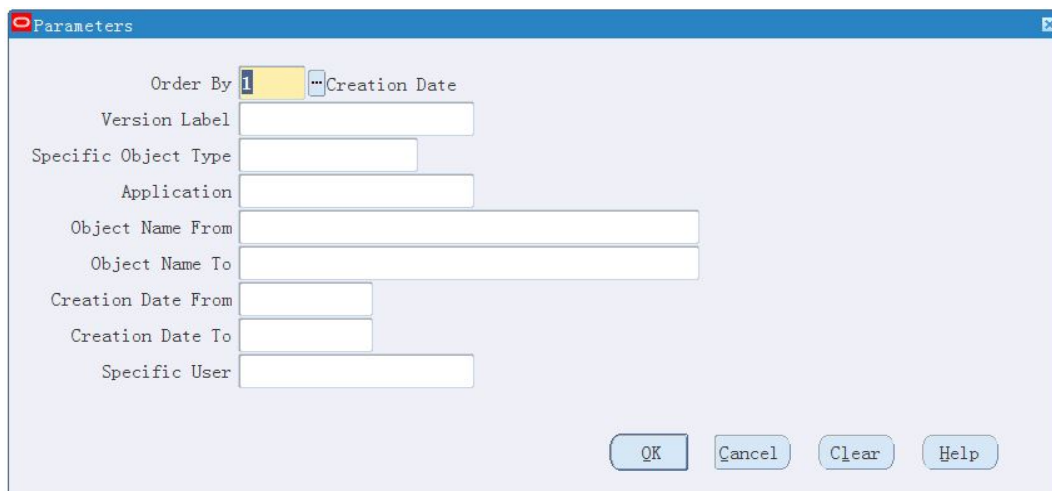
Running the Object Archive Version Detail Report

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 E-Business Suite Submit Requests window.

To run the Object Archive Version Detail Report:

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

The Parameter window opens for the request.



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

- Order By**: A dropdown menu showing '1' and a button to select 'Creation Date'.
- Version Label**: A text input field.
- Specific Object Type**: A text input field.
- Application**: A text input field.
- Object Name From**: A text input field.
- Object Name To**: A text input field.
- Creation Date From**: A text input field.
- Creation Date To**: A text input field.
- Specific User**: A text input field.
- Buttons**: OK, Cancel, Clear, and Help.

3. To limit the report results, type the relevant report parameters.

The following table lists and defines the Object Migrator common parameters used with Object Archive Version Detail Report.

Parameter Name (*Required)	Description
Order By *	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, and so forth).
Application	Limit the report by a specific application.
Object Name From	Type the starting object name to include in the report.
Object Name To	Type 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.

4. Click **OK**.

The report runs according to the specified parameters.

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

Purging the Object Archive

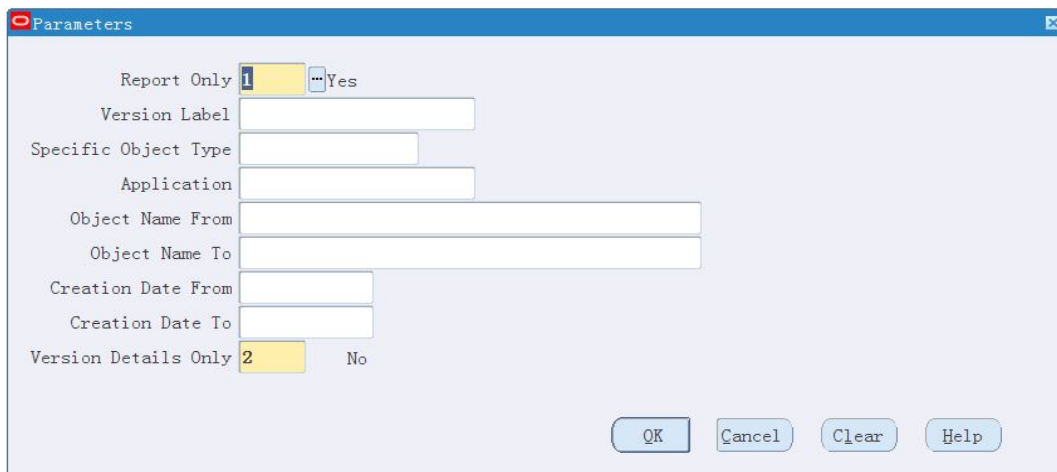
The Purge Object Archive Versions program purges information from the AOL object archive. Use the purge functionality to remove old versions of objects that will never be used or to remove object versions that were incorrectly saved to the object archive. Run this report from the Oracle E-Business Suite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

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

To purge the object archive:

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

The Parameter window opens for the request.



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

- Report Only:** A dropdown menu showing '1' with a 'Yes' button next to it.
- Version Label:** A text input field.
- Specific Object Type:** A text input field.
- Application:** A text input field.
- Object Name From:** A text input field.
- Object Name To:** A text input field.
- Creation Date From:** A text input field.
- Creation Date To:** A text input field.
- Version Details Only:** A dropdown menu showing '2' with a 'No' button next to it.
- Buttons:** OK, Cancel, Clear, and Help.

3. To limit the report results, type the relevant report parameters.

Parameter Name (*Required)	Description
Order By *	<p>Select one of the following:</p> <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.
Object Type	Limit the data to purge by a specific object type (concurrent program, value set, and so forth).
Application	Limit the data to purge to a specific application.
Object Name From	Type the starting object name to include in the purge. This parameter is case sensitive.
Object Name To	Type 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 *	<p>Select 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. 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. After the details have been purged, you will not be able to ever use this object version as a source for migrations or as a version to run comparisons against. • No. Remove the object version as well as the details for the object. Use this setting only if you want to remove all record of the object versions from the object archive.

4. Click **OK**.

The report runs according to the specified parameters.

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

Using Object Migrator with PPM Deployment Management

- ["Using Object Migrator with PPM Deployment Management Overview" below](#)
- ["Running Object Migrator from PPM Deployment Management" on the next page](#)
- ["Process Packages in PPM Deployment Management" on page 90](#)

Using Object Migrator with PPM Deployment Management Overview

Object Migrator can be run from PPM Deployment Management.

PPM Deployment Management is designed to automate the deployment of technology solutions. Business system components created or modified by an initiative are grouped into Deployment Management packages. These packages have workflows associated with them that automate the process of moving each package through required steps, which typically involve system build, testing, quality assurance, staging, and final deployment. Deployment Management automatically deploys the application components necessary for each package.

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. Deployment Management maintains an audit trail for all activities including package information, application components altered, approvals obtained and deployments performed.

For more information, see the following documentation:

- *Deployment Management User Guide*
- *System Requirements and Compatibility Matrix*

Running Object Migrator from PPM Deployment Management

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

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

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

To create a package to migrate AOL objects in Deployment Management:

1. Open the PPM Workbench. See the *Getting Started* guide for details.
2. From the shortcut bar, select **Change Mgmt > Packages**.
3. In the Package Workbench, click **New Package** to create a new package.

The Package window opens.

4. Complete the fields in the **Package Information** section.
5. Click the **New Line** button on the Package window.

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

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

The Add Line window refreshes to display the parameters required to migrate the specific object. For a description of each parameter, see "[Migration Rules and Parameters](#)" on page 95. Both the application name and the specific object

name are validated against the source environment in the workflow. The version label and version description are used only if the workflow is configured to use the object archive. The source and destination environments are designated by the workflow.

Edit Line

Object Type Information

Object Type: AOL:Conc Prog

Sequence: 1 Application Code: None

Parameters

Source Application: CLM

Selection Type: Specific Object

Conc Program: CLMRMCM1

New User Program Name: CLMRMCM1_BAK

Conc Program From:

Conc Program To:

Version Label:

Version Desc:

Overwrite if Exists: ☒ Yes ☐ No

Partials Allowed: ☐ Yes ☒ No

Clear OK Apply Cancel

AOL:Conc Prog' parameters loaded.

7. Click **OK** to save the package line.
8. Specify the parameters to describe the object.

After the package line has been typed and the package submitted, the AOL object moves through the workflow like any other package line in Deployment Management.

Process Packages in PPM Deployment Management

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

Package: 30070

Package Information

Package No.: 30070 Package Group: Created By: Admin User
 Description: Created On: August 22, 2017
 Workflow: OM_DEV78_QA05_PROD36 Package Status: In Progress
 Assigned User: Priority: Low Parent:
 Assigned Group: Package Type: Customization Priority Seq: 50
 Percent Complete: 0

Package Lines Status Notes References UserData

Seq	Object Name	Object Type	1 Approve (From DEV to QA)	2 Migrate from DEV to QA	3 Retry? (From DEV to QA)	4 Approve (from QA to PROD)
1	CLMRMCM1	AOL: Conc Prog	Approved	Succeeded [ConcReq:5932125]		Eligible

Refresh Select All All View --> Line Exec Log (Latest) Retry? (From QA to PROD)

Submit OK Save Cancel

Ready

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

For more information, see [Deployment Management](#).

Appendix 2: AOL-Dependent Objects

"Table A-1. AOL object migrated by Object Migrator, continued" on page 94 lists the dependent object for each of the AOL objects migrated by Object Migrator. # represents items that Object Migrator will not migrate.

Table A-1. AOL object migrated by Object Migrator

Object	Dependencies (Migrated)	Dependencies (Not Migrated)
Concurrent Programs	Executables	Value Sets Application # Security Groups # Resource Consumer Groups # Concurrent Request Types # Request Sets Executable Types # Profile Options
Concurrent Manager	Work Shifts Specialization Rules	Application # User Oracle User # Conc Program Request Type # Program Library # Security Group #
Value Sets	Value Sets (parent)	Security Group #

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

Object	Dependencies (Migrated)	Dependencies (Not Migrated)
Descriptive Flex Fields		Value Sets Profiles Application # Tables # Security Groups #
Request Groups (Report Groups)		Concurrent Program Request Set Application #
Menu	Submenus (optionally migrated)	Application # Forms # Subroutines # Macros #
Responsibility		Request Group Menu (Char/GUI) Forms (Character only)# Data Groups # Application # Form Functions (for exclusions) Inventory Org # Security Group # Attributes #

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

Object	Dependencies (Migrated)	Dependencies (Not Migrated)
Users		Responsibilities Employees # Customers # Suppliers # Security Groups # Security Attributes #
GUI Menu	Submenus (optionally migrated)	Functions Security Groups #
Folders		Users Responsibilities
FSG Row/Col Sets (For Oracle E-Business Suite Release 11i only)		Set of Books # Standard Axes #
Functions	Form Definition	Application (form) # Objects # Region Applications# Region Codes #
Messages		Application #
Printer Definitions	Printer Types Printer Drivers Printer Styles	Platforms #
Profile Options		Users Responsibilities Application#

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

Object	Dependencies (Migrated)	Dependencies (Not Migrated)
Quick Codes		Application #
Request Sets (Report Sets)		Application # Concurrent Program Request Set Stage Function #

Appendix 2: Migration Rules and Parameters

- ["Common Migrator Parameters" below](#)
- ["Concurrent Managers" on page 105](#)
- ["Concurrent Programs" on page 107](#)
- ["Descriptive Flexfields" on page 109](#)
- ["Folders" on page 112](#)
- ["FSG Row/Column Sets" on page 113](#)
- ["Functions" on page 114](#)
- ["GUI Menus" on page 116](#)
- ["Messages" on page 118](#)
- ["Printer Definitions" on page 119](#)
- ["Profile Options" on page 120](#)
- ["QuickCodes \(AOL\)" on page 123](#)
- ["Request Groups \(Report Groups\)" on page 124](#)
- ["Request Sets \(Report Sets\)" on page 125](#)
- ["Responsibilities" on page 126](#)
- ["Users" on page 129](#)
- ["Value Sets" on page 133](#)

Common Migrator Parameters

["Table B-1. Object Migrator common parameters, continued" on page 104](#) lists and defines the parameters, in alphabetical order, that are common to most Object Migrator object types.

Table B-1. Object Migrator common parameters

Parameter Name	Required?	Description
Compare Only	Yes	<p>Specifies whether to compare objects between instances. Options are:</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>The comparison functionality does not compare column information, extended items, or securing attributes. For more information, see "Comparing Objects" on page 63.</p>
Dest Application	Yes	<p>Short name of the application in the destination database to own the object after migration. This defaults to the application chosen as the source application. The list of applications comes from the database where Object Migrator is installed. If the desired application does not appear in the list, contact your system administrator.</p>

Table B-1. Object Migrator common parameters, continued

Parameter Name	Required?	Description
Dest Database	Yes	<p>The database to which objects are migrated. For object types that allow renaming, this can be the same as the source database, but is typically not the same. If the desired destination database is not an option on the list, contact your system administrator to have the database name added to the appropriate value set. To migrate the objects into the object archive rather than to another Oracle E-Business Suite instance, select Object Archive.</p> <p>In general, the source and destination databases do not need to be on the same Oracle E-Business Suite release because 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>
From Version Label	No	<p>The version label name of the objects that will be recovered from the object archive. This parameter is used with Recover From Archive. When trying to recover a specific object, select from all the valid version labels for that object. When recovering a range of objects, select from all version labels that have at least one object of the object type (such as concurrent program and value set) to recover.</p> <p>This parameter is also used when comparing two versions of the same object.</p> <p>For more information, see "Running the Object Comparison" on page 64.</p>

Table B-1. Object Migrator common parameters, continued

Parameter Name	Required?	Description
New <Object_Type> Name	No	<p>The name to give the object in the destination database after migration.</p> <p>Default value is the object name from the source database.</p> <p>Accept the default or modify the value. This parameter is used only when Specific Object is selected in the Selection Type field.</p>
New User <Object_Type> Name/Title	No	<p>Some object types require that the user names and titles be unique within a given application. You can change the user name or 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>
<Object_Name> From	No	<p>The beginning of the object name range to migrate. This field is used 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>The range match is case sensitive.</p> <p>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>

Table B-1. Object Migrator common parameters, continued

Parameter Name	Required?	Description
<Object_Name> Like	No	<p>Type 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. Options are:</p> <ul style="list-style-type: none"> • Yes. If the objects being migrated already exists in the destination database and the destination application, overwrite the objects definition with the new definition. When migrating and renaming a specific object, use the value of <Object_Type> Name as the object name to check against. • No. If the objects being migrated already exists in the destination database and destination application, then the object will not be overwritten. The migration will fail and indicate the reason in the output report.

Table B-1. Object Migrator common parameters, continued

Parameter Name	Required?	Description
Partials Allowed	Yes	<p>Specifies whether migrations can proceed when portions of the migration fail. Options are:</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. • No. If any of the components of the object fail validation, then the entire object is failed and the error report will: <ul style="list-style-type: none"> • Log the failing components as having failed • Log the object as having failed due to a component failure • Log the components non-failing components as having failed due to an object failure. <p>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. Options are:</p> <p>Yes. Use the object archive as the data source for the migration.</p> <p>No. Do not use the object archive as the data source for the object migration.</p> <p>This parameter should be set to Yes if Object Archive is selected in the Source Database and should be set to No if the Source Database is not Object Archive.</p>

Table B-1. Object Migrator common parameters, continued

Parameter Name	Required?	Description
Report Only	Yes	<p>Specifies whether Object Migrator should simulate the migration or perform the actual migration. Options are:</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>Report Only must be set to No if Compare Only is set to Yes or if the Destination Database is set to 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. Options are:</p> <ul style="list-style-type: none"> • Yes. Put the objects 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 E-Business Suite instance, or specify the object archive as the destination database. • No. Do not store the objects being migrated into the object archive. <p>This parameter cannot be set to Yes if Report Only or Compare Only is set to Yes.</p>

Table B-1. Object Migrator common parameters, continued

Parameter Name	Required?	Description
Selection Type	Yes	<p>Indicates whether to migrate one or multiple objects. Options are:</p> <ul style="list-style-type: none"> • Specific Object. One object will be selected for migration. Type 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. Specify the alphanumeric range to use in the <i><Object_Name></i> From and To fields. 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>When migrating multiple objects (range or wild card), the migration fails if no objects can be migrated successfully, but succeeds if even one object is migrated. Always review the request output carefully.</p>
Source Application	Yes	<p>Short name of the application that owns the object in the source database that you want to migrate. For example, GL or OM.</p> <p>The valid list of applications comes from the database where Object Migrator is installed. If the desired application does not appear on the list, contact your system administrator.</p>

Table B-1. Object Migrator common parameters, continued

Parameter Name	Required?	Description
Source Database	Yes	<p>The database where the objects to be migrated currently exist.</p> <p>If the desired source database is not an option on the list, contact your system administrator to have the database name added to the appropriate value set.</p> <p>If using an archived version of the objects as the source data for the migration, select Object Archive as the source database.</p> <p>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 <Object Type> [Name]	When Selection Type = Specific Object	<p>The name of the object to migrate, typically the short name or code. This value is used (with the application value, if it is defined) to uniquely identify the object to migrate. For example, XXWIPDJPCCK.</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 type a value in this field.</p>

Table B-1. Object Migrator common parameters, continued

Parameter Name	Required?	Description
To	No	<p>The end of the object name range to migrate. This field is used with the <i><Object_Name></i> From field. Object Migrator performs an alphabetic search using this parameter so the value does not have to be exact.</p> <p>For example, to migrate all objects for a given application, migrate from A to zzz.</p> <p>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>
Version Description	Version Description	<p>This field should be typed only if Save To Archive is set to Yes.</p> <p>The description to attach to the object versions as they are stored in the object archive. This value is also the description for the version label if this is a new version label.</p> <p>If this field is left blank and the version label already exists, the Object Migrator uses the description of the version label as the description of the object versions.</p>
Version Label	When Save to Archive = Yes	<p>A version label is required if Save To Archive is set to Yes. The version label name for objects saved into the object archive. If a new version label is specified, Object Migrator creates the label. If an existing version label is specified, Object Migrator adds the objects to the existing label. This parameter is also used when comparing two versions of the same object. For more information, see "Running the Object Comparison" on page 64.</p>

Concurrent Managers

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

When migrating to an Oracle E-Business Suite Release 12 destination, the Concurrent Programs migrator submits the following concurrent requests:

- Build Concurrent Request Queue View
- Verify

Before Migrating Concurrent Managers

Before migrating Concurrent Managers, you need to ensure the following tasks are complete:

1. If the following entities are referenced by the Concurrent Managers in the source database, ensure that they 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
2. Any programs referenced by a migrated complex rule must include both an application and program entry if the destination is Release 11 or later.
3. Ensure the concurrent manager on the destination database is deactivated.

The Concurrent Manager Definitions Migrator migrates only new concurrent managers or managers that are not active on the destination database.

4. Ensure you have a backup of all work shift and specialization rules from the destination database.

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

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

Concurrent Managers Migrator Parameters

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

Only the object-specific parameters are defined here. For more information about the common Object Migrator parameters, see ["Table B-2. Concurrent Manager Definitions Migrator parameters" below](#).

Table B-2. Concurrent Manager Definitions Migrator parameters

Parameter Name	Description
Specific Conc Manager Name	Type the source Concurrent Manager name of the Concurrent Manager to migrate. Query this uppercase-only value from the Define Concurrent Manager form.
New Conc Manager Name	Type a new name for the Concurrent Manager, if desired. Otherwise, the name from the source instance is used.
New Description	Optionally, type a new description for the Concurrent Manager.

Concurrent Programs

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

Before Migrating Concurrent Programs

Before migrating concurrent programs, you need to:

1. Ensure that if the following entities are referenced by the concurrent program in the source database, that they 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
2. If there are any incompatible programs defined in the source instance, then those concurrent programs should either be present in the destination instance or, when Mig Incomp Partially is set to **No**, should be migrated together. For details about this parameter, see ["Table B-3. Concurrent Programs Migrator parameters , continued" on page 109](#).
3. Some patch levels of the Oracle E-Business Suite Release 11i require each incompatibility defined for a concurrent program to be qualified by an incompatibility type, that determines the scope within which incompatibilities are evaluated by Concurrent Managers.

Incompatibilities can be evaluated within a specific conflict domain, or globally across all conflict domains. The Concurrent Programs Migrator migrates the incompatibility type if the destination supports incompatibility types. If

incompatibility types are used in the destination but not in the source, the Object Migrator defaults the incompatibility type for the program incompatibility based on the same logic Oracle uses during upgrades.

Concurrent Programs Migrator Parameters

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

Only the object-specific parameters are defined here. For more information about the common Object Migrator parameters, see ["Common Migrator Parameters" on page 95](#).

Table B-3. Concurrent Programs Migrator parameters

Parameter Name (*Required)	Description
Specific Conc Program	Type the short name 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 "Common Migrator Parameters" on page 95 . For example, FNDSCARU is the short name for the Active Users report.
New Conc Program Short Name	Accept the default or type a new concurrent program name. For a description of this common parameter, see "Common Migrator Parameters" on page 95 .
New User Program Name	Accept the default value that is derived from the source database or type a new value. The value is given to the object on the destination database. If left blank, Object Migrator uses the user name from the source database. For a description of this common parameter, see "Common Migrator Parameters" on page 95 .

Table B-3. Concurrent Programs Migrator parameters , continued

Parameter Name (*Required)	Description
Conc Program Like	Type a search string to match by concurrent program short name. For a description of this common parameter, see <Object Name> Like in "Common Migrator Parameters" on page 95 .
Mig Incomp Partially * (Parameter may not be displayed)	Specifies the treatment of incompatibilities during migration. Select 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.

Descriptive Flexfields

Before Migrating Descriptive Flexfields

Before migrating descriptive flexfields, you need to:

1. Ensure that if the following entities are referenced by the descriptive flexfields in the source database, that they exist in the destination database prior to the migration:
 - Profile options
 - Value sets
 - Security groups
2. Ensure that the following is true:
 - Owning Table of the descriptive flexfield in the source database exists, and is registered to the same application in the destination database.

- The application owning the table that owns the flexfield in the source database must exist in the destination database with the same application short name.
- The form that uses the flexfield in the source should be the same as the form that uses the flexfield in the destination. This is not validated.
- Table structures that own the descriptive flexfield being migrated should be the same in the source and destination databases. This is not validated.
- Any SQL statements used in descriptive flexfields being migrated should be valid in the destination database. This is not validated.
- 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.
- The flexfield in the destination (if it exists) should not be protected. This is validated.
- 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.
- Be sure that existing transaction data relating to the flexfield in the destination remains valid. The migration replaces the definition in the destination, which can remove or change the value set for existing segments and contexts.

Descriptive Flexfields Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Descriptive Flexfields program.

Only the object-specific parameters are defined here. For more information about the common Object Migrator parameters, see ["Table B-1. Object Migrator common parameters, continued" on page 104.](#)

Table B-4. Descriptive Flexfields Migrator parameters

Parameter Name	Description
Specific Desc Flex	<p>Type the name 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.</p> <p>For a description of this common parameter, see "Table B-1. Object Migrator common parameters, continued" on page 104.</p>
Disabled Contexts	<p>Specifies 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, are migrated to the destination. • No. Flexfield contexts which are disabled in the source are not migrated to the destination. <p>Select No to prevent contexts and segments still in development from being migrated with the remainder of the flexfield. If the context or segment exists in the destination, it will be removed from the flexfield definition.</p> <p>When using No for disabled contexts and disabled segments, make sure that the transaction data relating to the flexfield in the destination remains valid.</p>
Disabled Segments	<p>Specifies whether to migrate disabled segments. Select one of the following:</p> <p>Yes. All segments from the source, including disabled segments, are migrated to the destination.</p> <p>No. Flexfield segments which are disabled in the source are not migrated to the destination.</p> <p>If the context or segment exists in the destination it is removed, even if it is enabled.</p> <p>If the intent is to disable use of the flexfield context or segment in the destination, it should be disabled in the source and migrated with Yes.</p>

Folders

Before Migrating Folders

If the following entities are referenced by the folders in the source database, they must also exist in the destination database prior to the migration:

- Owner (application user)
- Segment (form blockname)
- Responsibilities or users referenced by a default folder setting

Folders Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Folders program.

Only the object-specific parameters are defined here. For more information on the common Object Migrator parameters, see ["Table B-1. Object Migrator common parameters, continued" on page 104.](#)

Table B-5. Folders Migrator parameters

Parameter Name	Description
Specific Folder Name	Type the name of the folders to migrate. If left blank, the Folders Migrator will not limit by folder name (just by specific owner and specific segment if they are specified).
Specific Owner	Type the owner of the folders to migrate. The list of values for this parameter is not validated against the specific folder parameter. It is only validated against the application users defined in the source database. If left blank, the Folders Migrator will not limit by folder owner (just by specific folder name and specific segment if they are specified).

Table B-5. Folders Migrator parameters, continued

Parameter Name	Description
New Owner Name	Optionally, type a new owner name.
Specific Segment Name	<p>Type the folder segment of the folders 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 Folders Migrator will not limit by folder segment (just by specific folder name and specific owner if they are specified).</p>

FSG Row/Column Sets

The FSG row/column sets migrator is for Oracle E-Business Suite Release 11i only.

Before Migrating FSG Row/Column Sets

Before migrating FSG row/column sets, make sure that if the following entities are referenced by the Row Set or Column Set in the source database, they also exist in the destination database prior to the migration:

- Flexfield structures
- Standard axes
- Set of books

FSG Row/Column Sets Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate FSG Row/Col Sets request.

Only the object-specific parameters are defined here. For more information on the common Object Migrator parameters, see ["Table B-1. Object Migrator common parameters, continued" on page 104.](#)

Table B-6. FSG Row/Column Sets Migrator parameters

Parameter Name	Description
Set Type (required)	Type the type of FSG Sets to migrate: <ul style="list-style-type: none">• Column to migrate FSG Column Sets• Row to migrate FSG Row Sets
Specific Row/Column Set	Type the name of the FSG Row Set or FSG Column Set to migrate. Query this value from the Define Row Set or Define Column Set forms. The list of values for this parameter does not limit the values by Set Type. For a description of this common parameter, see "Table B-1. Object Migrator common parameters, continued" on page 104.

Functions

These are objects that define functions for Oracle E-Business Suite Release 11 and later.

Before Migrating Functions

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 Oracle E-Business Suite 11.5.5 or later, as object functionality was added in Oracle E-Business Suite 11.5.5.
- Region applications referenced by a function must exist on the destination database if the destination is Oracle E-Business Suite 11.5.5 or later.
- Region codes referenced by a function must exist on the destination database if the destination is Oracle E-Business Suite 11.5.5 or later.
- The context dependence value, maintenance mode support value, and type value referenced by a function must exist on the destination database if the

destination includes this information. This is validated by the migrate Functions Migrator.

When migrating from instances that do not include maintenance mode support, the maintenance mode support value will default to **None**.

When migrating from instances that do not include the **Context Dependence** field, the context dependence value will default to responsibility.

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

Note: In some patch levels of Oracle E-Business Suite Release 11i, user function names do not need to be unique. The Functions Migrator enforces the appropriate logic based on the rules in place in the destination.

Functions Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Functions program.

Only the object-specific parameters are defined here. For more information on the common Object Migrator parameters, see ["Table B-1. Object Migrator common parameters, continued" on page 104](#).

Parameter Name	Description
Specific Function	Type the name of the function to migrate. Query this uppercase-only value from the Define Form Functions form (using the Smart Client application). For a description of this common parameter, see "Table B-1. Object Migrator common parameters, continued" on page 104 .
New User Function Name	Type 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 uses the user name from the source database.

GUI Menus

Before Migrating GUI Menus

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

- Functions referenced by the menu (and the menu tree underneath the menu, if Single Level Only is set to No)
- If Single Level Only is set to Yes, then any child menus reference by the parent menu must exist in the destination database.
- Menu types referenced by the menu must exist in the destination database. This applies only when menu types are in use in the destination.
- Security Groups

If the destination database is Oracle E-Business Suite Release 11.5.5 or later, the Object Migrator will submit the Compile Security program in the destination instance in order to complete the menu compilations introduced in Oracle E-Business Suite 11.5.5.

Note: When migrating entire menu structures, make sure that standard menus are not overwritten. Standard menus may be referenced from submenus in a custom menu structure.

GUI Menus Migrator Parameters

Migrating a range of parent menus with Single Level set to No can entail migrating very large amounts of data. It is recommended to migrate parent menus in small batches or individually.

The following parameters are located on the Submit Request window for the Migrate GUI menus request.

Only the object-specific parameters are defined here. For more information on the common Object Migrator parameters, see ["Table B-1. Object Migrator common parameters, continued" on page 104.](#)

Table B-8. GUI Menus Migrator parameters

Parameter Name	Description
Specific Menu Name	Type the menu name of the menu to migrate. Query this uppercase-only value from the define menus form. For a description of this common parameter, see "Table B-1. Object Migrator common parameters, continued" on page 104.
New Menu Title	Accept the default or type a new title for the menu after migration. If left blank, Object Migrator uses the title from the source database.
Single Level Only (required)	Select one of the following: <ul style="list-style-type: none"> • Yes. Only migrate the top level menu entries of the given menu. • No. Migrate the entire menu and sub-tree below the specific menu name. When migrating entire menu structures, make sure that standard menus are not overwritten. Standard menus may be referenced from submenus in a custom menu structure.
Overwrite if Exists	For a description of this common parameter, see "Table B-1. Object Migrator common parameters, continued" on page 104. Note that this parameter governs the parent menu only. If migrating an entire menu tree, submenus that exist will be overwritten without error, even if Overwrite is set to No, as long as the top parent menu did not exist.
Compare Only	For a description of this common parameter, see "Table B-1. Object Migrator common parameters, continued" on page 104. For better output, it is recommended that comparisons of menus be run with Single Level Only set to Yes.

Messages

Before Migrating Messages

If the following entities are referenced by the messages in the source database, they must also exist in the destination database prior to the migration:

- Application
- Category, if the destination supports this information
- Severity, if the destination supports this information

For server-side messages only, Oracle E-Business Suite requires the execution of `Generate Messages`. Due to database limitations with distributed transactions, Object Migrator cannot launch this program remotely. As a post migration step for each unique language and application combination successfully migrated, this program needs to be run.

You can run the program by navigating to the Define Messages Character Mode form, querying a message from the given application, updating the record without a real change, and committing the transaction. For example, change a letter in the message and change it back.

This can also be done by launching the program from the command-line for the destination database. For more information, see the *Oracle E-Business Suite System Administration Guide*.

Messages Migrator Parameters

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

Only the object-specific parameters are defined here. For more information on the common Object Migrator parameters, see ["Table B-1. Object Migrator common parameters, continued" on page 104](#).

Table B-9. Messages Migrator parameters

Parameter Name	Description
Specific Message	Type the message name of the message to migrate. Query this uppercase-only value from the Define Messages form.

Printer Definitions

Use this migrator to migrate printer types and all the printers, printer styles, and printer drivers associated with them.

Any printers, printer styles, and printer drivers that are already assigned to the printer type in the destination database will not be removed, even if they do not exist in the source database.

Before Migrating Printer Definitions

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

- Subroutines or commands
- Platforms

Any new printer styles must have a sequence number that is unique on the destination database.

Printer Definitions Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Printer Definitions program.

Only the object-specific parameters are defined here. For more information on the common Object Migrator parameters, see ["Table B-1. Object Migrator common parameters, continued" on page 104.](#)

Table B-10. Printer Definitions Migrator parameters

Parameter Name	Description
Specific Printer Type	Type the name of the printer type to migrate. Object Migrator attempts to migrate the printer type, any printer attached to the printer type, any printer style assigned to the printer type, and any printer Driver assigned to the printer type. Query this uppercase-only value from the Define Printer Types form. For a description of this common parameter, see "Table B-1. Object Migrator common parameters, continued" on page 104.
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, but not any printers attached to the printer type.

Profile Options

Caution should be used when migrating profile option values to make sure that the values remain valid in the destination. When migrating profile option values, values are not re-derived or validated during migration. This could cause problems if the value is not valid or refers to a different entity in the destination database. For example, the organization id for an inventory org.

Before Migrating Profile Options

If the following entities are referenced by the profile options in the source database, they must also exist in the destination database prior to the migration:

- Users that have a user-level setting for the profile option
- Responsibilities that have a responsibility-level setting for the profile option
- Applications that have a application-level setting for the profile option
- If migrating with Values Only set to **Yes**, the profile option to which the values belong must exist in the destination
- Hierarchy type

- Organizations that have an organization-level setting for the profile option (for organization hierarchy)
- Servers that have a server-level setting for the profile option (for server hierarchies)

Additionally, the following migration rules should be noted:

- Any SQL statements used in profile options being migrated should be valid in the destination database.

This is not validated by the Migrate Profile Options program.

- When moving profile settings, the Profile Options Migrator does not validate the actual option value. For example, if a profile option was for a user and in the source database the value was a user ID corresponding to `SYSADMIN`, the migrator would bring the user id over to the destination as is. It would not get the user ID of `SYSADMIN` in the destination database and would not validate that the user id is valid in the destination.
- When migrating from a release where hierarchy types are not used to an instance that uses hierarchy types, the profile is created using the security hierarchy.
- Object Migrator allows migration of values for profiles using the server hierarchy. Migrating profile option values at the server level may produce Missing Reference errors unless the same servers are shared across Oracle E-Business Suite instances, because their server names would differ.

The migrator raises the `oracle.apps.fnd.profile.value.update` Business Event in the destination for each profile where values have been migrated to or removed from the destination. This occurs only if the destination supports the functionality (11.5.9 or later) and Migrate Profile Option Values is set to Yes.

The following are special considerations relevant to Business Events:

- Raising Business Events can cause real-time processing to occur in the destination, and the migration will not finish until this processing completes. Minimizing real-time processing in subscriptions for the `oracle.apps.fnd.profile.value.update` event helps make sure efficiency of profile

option value migrations.

- You should be sure that custom processing code for subscriptions to the `oracle.apps.fnd.profile.value.update` event functions correctly when invoked across a database link.
- If Business Event processing encounters an error, it is logged in the destination instance and does not prevent completion of the migration. You should monitor the errors in the destination instance to identify any issues.

Profile Options Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Profile Options program.

Only the object-specific parameters are defined here. For more information on the common Object Migrator parameters, see ["Table B-1. Object Migrator common parameters, continued" on page 104](#).

Table B-11. Profile Options Migrator parameters

Parameter Name (*Required)	Description
Specific Profile Option	Type the name of the profile option to migrate. Query this uppercase only value from the Define Profile Options form.
New Profile User Name	Accept the default value or type a new value for the profile user name. If left blank, Object Migrator uses the user name from the source database.
Migrate Profile Option Values *	Options are: <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.

Table B-11. Profile Options Migrator parameters, continued

Parameter Name (*Required)	Description
Migrate Values Only *	Options are: <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 *	Options are: <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.

QuickCodes (AOL)

QuickCode objects, based on the FND_COMMON_LOOKUPS table, can be viewed or modified using the Define QuickCode form under the application developer responsibility.

Before Migrating QuickCode

Quick Codes can not be migrated to or from Release 11i to any other release.

QuickCodes Migrator Parameters

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

Only the object-specific parameters are defined here. For more information on the common Object Migrator parameters, see ["Table B-1. Object Migrator common parameters, continued" on page 104.](#)

Table B-12. QuickCodes Migrator parameters

Parameter Name	Description
Specific QuickCode Type	Type the name of the QuickCode type to migrate. Query this value from the Define QuickCodes form.

Request Groups (Report Groups)

In some releases of Oracle E-Business Suite, request groups are referred to as report groups.

Before Migrating Request Groups

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

- Concurrent programs
- Request sets
- Applications

If assigned, the request group code referenced by the request group in the source database must be unique within the application in the destination database.

Request Groups Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Report Groups program.

Only the object-specific parameters are defined here. For more information on the common Object Migrator parameters, see ["Table B-1. Object Migrator common parameters, continued" on page 104.](#)

Table B-13. Request Groups Migrator parameters

Parameter Name	Description
Specific Report Group	Type the name of the request group which to migrate. This value can be queried from the Request Sets form (Define Report Groups in some releases). This is not the code associated to the request group.

Request Sets (Report Sets)

Before Migrating Request Sets

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

- Concurrent programs
- Concurrent programs parameters (when the request set defines a default value)
- Request set stage function

Request Sets Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Report Sets program.

Only the object-specific parameters are defined here. For more information on the common Object Migrator parameters, see ["Table B-1. Object Migrator common parameters, continued" on page 104.](#)

Table B-14. Request Sets Migrator parameters

Parameter Name	Description
Specific Report Set	Type the name of the request set to migrate. Query this value from the Request Sets form (named Define Report Sets in some releases).

Table B-14. Request Sets Migrator parameters, continued

Parameter Name	Description
New Report Set Code	When migrating to an Oracle E-Business Suite Release 11, 11i, or 12 instance, use this parameter to change the code of the request set as it is migrated. Oracle requires that the request set code is unique in a given instance. If you are using the migrator to create a copy of an existing request set, use this parameter to change the code for the new request set being created.

Responsibilities

Before Migrating Responsibilities

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
- 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

Responsibilities Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Responsibilities program.

Only the object-specific parameters are defined here. For more information on the common Object Migrator parameters, see ["Table B-1. Object Migrator common parameters, continued" on page 104.](#)

Table B-15. Responsibilities Migrator parameters

Parameter Name	Description
Specific Responsibility	Type the name of the responsibility to migrate. Query this value from the Define Responsibilities form.
Migrate Enabled Resp Only (required)	Options are: <ul style="list-style-type: none">• Yes. Migrate only enabled responsibilities.• No. Migrate both enabled and disabled responsibilities.
Synchronize Workflow Tables	Use this parameter only if the destination is an Oracle 11.5.9 instance. If the instance is not at the 11.5.9 level, this parameter will be ignored. Options are: <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 program, which will populate the workflow local tables for all responsibilities defined in the application.

Table B-15. Responsibilities Migrator parameters, continued

Parameter Name	Description
Migrate Sec Attrib Values	<p>For some patch levels of Oracle E-Business Suite Release 11i, users can define values for securing attributes as part of the responsibility.</p> <p>We recommend using No, unless the:</p> <ul style="list-style-type: none"> Responsibility for versions in both the source and destination are 11.5.17 or later Attribute values attached to the responsibilities will be valid in the destination <p>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> <ul style="list-style-type: none"> No. Security attribute values are not migrated from the source to the destination. Existing securing attribute values in the destination remain in place, as long as the responsibility definition still includes the particular Securing Attribute. Yes. Migrate all security attribute values for the responsibility. Existing values are removed and replaced. <p>Securing attribute values are not validated or re-derived during migration and are often typed without any validation in the Responsibilities form. This could 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>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</p>

Users

Before Migrating Users

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

User passwords should be reset at the destination instance for the user to be functional.

If customer information is associated with the user, the definition of that information between environments must be consistent. In the Oracle E-Business Suite 11i release, this definition changed and migrations from the newer definition to the old definition are not allowed.

Users Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Users program.

Only the object-specific parameters are defined here. For more information on the common Object Migrator parameters, see ["Table B-1. Object Migrator common parameters, continued" on page 104](#).

Table B-16. Users Migrator parameters

Parameter Name (*Required)	Description
Specific User Name	Type the name of the user which to migrate. Query this value from the Define Users form.

Table B-16. Users Migrator parameters, continued

Parameter Name (*Required)	Description
New User Name	When creating a new user, for security reasons, change the password of the new user after the migration is complete.
Create User *	<p>Specifies whether to create a new user in the destination database. Options are:</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. • No. If the user does not exist on the destination database, set the status of the object to Error. <p>If the user already exists, the Users Migrator will not update any of the user header information. It only updates the user responsibilities and securing attribute information. When creating a new user, for security reasons, change the password of the new user after the migration is complete.</p>
Active Resp Only *	<p>Determines which responsibilities to migrate. Options are:</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 *	<p>Specifies whether to overwrite responsibilities in the destination database. Options are:</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 *	<p>Specifies whether to disable responsibilities associated with the user in the destination database that were not migrated. Options are:</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 date. • No. Do not modify user responsibilities that exist on the destination database but not on the source database.

Table B-16. Users Migrator parameters, continued

Parameter Name (*Required)	Description
Migrate Enabled Users Only *	Options are: <ul style="list-style-type: none">• Yes. Only migrate enabled users.• No. Migrate both enabled and disabled users.

Table B-16. Users Migrator parameters, continued

Parameter Name (*Required)	Description
Migrate Securing Attributes *	<p>For some patch levels of Oracle E-Business Suite Release 11i, users can define values for securing attributes as part of the user definition.</p> <p>Securing attributes values associated with a user are typed without validation in Oracle E-Business Suite. The following securing attributes are actively managed, because 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 are automatically created and defaulted. <p>Options are:</p> <ul style="list-style-type: none"> • No. Migrate only attributes which Oracle E-Business Suite actively manages. Other attributes will neither be migrated from the source nor removed from the destination. • Yes. Migrate all securing attribute values for the user. Existing values will be removed or replaced. <p>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</p>

Table B-16. Users Migrator parameters, continued

Parameter Name (*Required)	Description
	different entity in the destination.
Synchronize Workflow Tables	<p>Use this parameter only if the destination is an Oracle 11.5.9 instance. Otherwise it is ignored. Options are:</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 program, which will populate the workflow local tables for all users defined in the instance.

Value Sets

Before Migrating Value Sets

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

- Parent value sets
- Validation types
- Security groups

SQL statements used in value sets being migrated should be valid in the destination database. This is not validated by the Migrate Value Sets program.

Value sets defined with hierarchical security can be migrated only to other instances that support hierarchical security. Otherwise, the migration errors for the value set.

When migrating value sets with hierarchical security, the Compile Value Set Hierarchies program is submitted in the destination to compile the migrated information. This applies only when the destination instance supports hierarchical security.

Make sure that the transaction data relating to the value set in the destination remains valid. The migration replaces the definition in the destination, which can remove existing value set values.

When defining Table-validated value sets, you can specify additional ID and meaning columns. For each column defined, you can specify a datatype and size value. As of Release 11.5.8, the inclusion of size and type information for any specified columns is required. The Object Migrator enforces that this information is specified when migrating to any Release 11i instance, because the resulting functional error occurs in all Release 11i instances.

Value Sets Migrator Parameters

The following parameters are located on the Submit Request window for the Migrate Value Sets request.

Only the object-specific parameters are defined here. For more information on the common Object Migrator parameters, see ["Table B-1. Object Migrator common parameters, continued" on page 104](#).

Table B-17. Value Sets Migrator parameters

Parameter Name	Description
Specific Value Set	Type the name of the value set which to migrate. Query this value from the Define Value Sets form.

Appendix 2: Exception Messages

The following is a list of Object Migrator exception messages that includes 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 the support.
- **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	Details
Message: 1000	<ul style="list-style-type: none">• Error: Object already exists in version label• Cause: The Object being saved to the object archive already exists in the specified version label.• Action: Either save the Object to another version label or use the Purge Object Archive Versions Detail Report program to remove the object from the version label.
Message: 1100	<ul style="list-style-type: none">• Error: Concurrent program already exists• Cause: Specified new concurrent program name already exists in the destination environment.• Action: Run the Migrate Concurrent Programs program with Overwrite set to Yes.
Message: 1101	<ul style="list-style-type: none">• 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	Details
Message: 1102	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	Details
Message: 1200	<ul style="list-style-type: none">• Error: Report set already exists• Cause: Specified new report set name already exists in destination database• Action: Run Migrate Report Sets program with Overwrite set to Yes.
Message: 1201	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• Error: Concurrent program parameter does not exist on destination database.• Cause: One of the parameters used by the concurrent program referenced by the report set in migration does not exist on the destination database. Migrating a concurrent program with Partials Allowed set to Yes can allow a concurrent program to come over without some of its parameters.• Action: Migrate the referenced concurrent program prior to migrating the report set. Set Partials Allowed should be set to No.
Message: 1203	<ul style="list-style-type: none">• Error: Report Set program does not exist on destination database.• Cause: The report set header is tied to a concurrent program that does not exist in the destination database.• Action: Migrate the referenced concurrent program prior to migrating the report set.
Message: 1204	<ul style="list-style-type: none">• Error: Value set already exists.• Cause: Specified New value set name already exists in the destination database.• Action: Run the Migrate Value Sets program with Overwrite set to Yes.

Message	Details
Message: 1301	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• 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	Details
Message: 1403	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Error: Profile option used in Flex Does Not Exist or is disabled in Destination. • Cause: The descriptive flexfield in migration references a profile option that does not exist in the destination database. • Action: Set up the referenced profile option prior to migrating the descriptive flexfield.
Message: 1407	<ul style="list-style-type: none"> • 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	Details
Message: 1411	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Error: Report group already exists. • Cause: Specified new report group name already exists in destination database. • Action: Run Migrate Report Groups program with Overwrite set to Yes.
Message: 1501	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Error: Parent menu already exists. • Cause: Specific menu name already exists in the destination instance. • Action: Run Migrate Menus program with Overwrite set to Yes.

Message	Details
Message: 1701	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Error: Child menu does not exist on destination database. • Cause: Menu in migration references a child menu which does not exist in the destination instance. This can happen under two conditions: 1) Single Level Only was set to Yes. The system will not attempt to migrate the child menu from the source instance under this condition, therefore the child menu must exist in the destination instance. 2) Import Standard Oracle Menus was set to No. The menu in migration references a child menu which is a standard Oracle menu which has been removed in the destination instance. The program considers a standard Oracle menu any menu which is owned by an Oracle Application. • Action: Three options: 1) Create the missing menu in the destination instance and rerun the migration s before. 2) Rerun the migration program with Single Level Only set to No. Use this option with caution. 3) Rerun the migration program with Import Standard Oracle Menus set to Yes. Use this option with caution.
Message: 1703	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	Details
Message: 1705	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Error: Another menu exists with the same menu title. • Cause: The name of the menu in migration already exists in the destination instance and Overwrite is set to No. • Action: Run Migrate Menus program with Overwrite set to Yes.
Message: 1900	<ul style="list-style-type: none"> • Error: Responsibility already exists. • Cause: Specified new responsibility name already exists in destination database. • Action: Run Migrate Responsibilities program with Overwrite set to Yes.
Message: 1901	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	Details
Message: 1904	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Error: User does not exist on the destination database. • Cause: The Users Migrator is being run with Create User set to No and the user does not exist on the destination database. • Action: Run the migrator with Create User set to Yes or manually create the user on the destination database before rerunning the migration.
Message: 2001	<ul style="list-style-type: none"> • 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: Type the employee on the destination database before rerunning the migration.
Message: 2002	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Error: Profile option already exists. • Cause: Specified new profile option name already exists in destination database. • Action: Run Migrate Profile Options program with Overwrite set to Yes.

Message	Details
Message: 2101	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Error: Invalid Profile Level. • Cause: The profile option value references a profile value not recognized by the Profile Options Migrator. • Action: Change the level to one of the following: site, application, responsibility, server (if supported), organization (if supported), or user.
Message: 2107	<ul style="list-style-type: none"> • 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	Details
Message: 2108	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• Error: Server does not exist on destination database. The server referenced by the profile option value does not exist in the destination database.• Cause: Object Migrator allows migration of values for profiles using the server hierarchy. Migrating profile option values at the server level may produce Missing Reference errors unless the same servers are shared across Oracle E-Business Suite instances, because 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	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• Error: Another profile exists with the same profile name.• Cause: Profile option names must be unique within a database. The destination database has a profile option belonging to a different application that uses the same profile option name.• Action: If the information in the destination is correct for the profile option being migrated, resubmit the migration, setting Dest Application to the correct value. Otherwise, change the name of the profile being migrated.

Message	Details
Message: 2300	<ul style="list-style-type: none"> • Error: QuickCode type already exists. • Cause: Specified new QuickCode type already exists in destination database. • Action: Run Migrate QuickCodes program with Overwrite set to Yes.
Message: 2301	<ul style="list-style-type: none"> • Error: Customization level does not exist on destination database. • Cause: Contact support. • Action: The customization level specified in the QuickCode type does not exist in the destination database.
Message: 2302	<ul style="list-style-type: none"> • 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 Partial Allowed set to Yes.
Message: 2500	<ul style="list-style-type: none"> • Error: Row/Col Set already exists. • Cause: Specified New Row/Col Set already exists in destination database. • Action: Run Migrate FSG Row/Column Set program with Overwrite set to Yes.
Message: 2501	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	Details
Message: 2600	<ul style="list-style-type: none"> • Error: Printer type already exists on destination database. • Cause: Specified printer type already exists in destination database. • Action: Run Migrate Printer Definitions program with Overwrite set to Yes.
Message: 2601	<ul style="list-style-type: none"> • Error: Printer already exists on destination database. • Cause: Specified printer already exists in destination database. • Action: Run Migrate Printer Definitions program with Overwrite set to Yes.
Message: 2602	<ul style="list-style-type: none"> • Error: Printer Information already exists on destination database. • Cause: Specified printer information (assignment) already exists in destination database. • Action: Run Migrate Printer Definitions program with Overwrite set to Yes.
Message: 2603	<ul style="list-style-type: none"> • Error: Printer style already exists on destination database. • Cause: Specified printer style already exists in destination database. • Action: Run Migrate Printer Definitions program with Overwrite set to Yes.
Message: 2604	<ul style="list-style-type: none"> • Error: Printer Driver already exists on destination database. • Cause: Specified printer Driver already exists in destination database. • Action: Run Migrate Printer Definitions program with Overwrite set to Yes.
Message: 2605	<ul style="list-style-type: none"> • 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	Details
Message: 2606	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Error:Folder already exists on destination database. • Cause: Specified Folder already exists in destination database. • Action: Run Migrate Folders program with Overwrite set to Yes.
Message: 2701	<ul style="list-style-type: none"> • 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	Details
Message: 2702	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Error: Parent menu already exists on destination database. • Cause: Menu already exists in destination database. • Action: Run Migrate GUI Menus program with Overwrite set to Yes.
Message: 2801	<ul style="list-style-type: none"> • Error: Menu does not exist on destination database. • Cause: Parent menu references a menu that does not exist in destination database. • Action: Create the menu in destination database and rerun the migrator program or run the migrator program with Single Level set to No.
Message: 2802	<ul style="list-style-type: none"> • 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	Details
Message: 2803	<ul style="list-style-type: none"> • Error: Child menu does not exist on destination database. • Cause: Parent menu references a menu that does not exist in destination database. • Action: Create the menu in destination database and rerun the migrator program or run the migrator program with Single Level set to No.
Message: 2804	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Error: Menu type does not exist in the destination database where menu type can be: Home Page, Data Security, Standard, HTML Tab, or Unknown. • Cause: The given type menu is not defined in the destination database. • Action: You are probably migrating the menu from a earlier version of Oracle E-Business Suite to a later version. Make sure the particular type is available in the destination database.
Message: 2900	<ul style="list-style-type: none"> • Error: Function already exists on destination database. • Cause: Specified function already exists in destination database. • Action: Run Migrate Functions program with Overwrite set to Yes.
Message: 2901	<ul style="list-style-type: none"> • 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	Details
Message: 2902	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Error: Object does not exist at the destination. • Cause: Function references an object that does not exist in the destination. The object could be any one of RESOURCES, ORGANIZATION, PROJECTS, TASKS, Contract, Program, and so forth. • Action: Create all the missing objects referenced by the function at the destination.
Message: 2905	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	Details
Message: 2907	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Error: Message already exists on destination database. • Cause: Specified Message already exists in destination database. • Action: Run Migrate Messages program with Overwrite set to Yes.
Message: 3001	<ul style="list-style-type: none"> • 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	Details
Message: 3002	<ul style="list-style-type: none"> • Error: Language code does not exist on destination database. • Cause: Message references a language code that does not exist in destination database. • Action: Type the appropriate language code in FND_LANGUAGES in destination database and rerun the migrator program.
Message: 3000	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Error: Concurrent Manager already exists on destination database. • Cause: Specified Concurrent Manager already exists in destination database. • Action: Run Migrate Concurrent Manager Definitions program with Overwrite set to Yes.
Message: 3101	<ul style="list-style-type: none"> • Error: Workshift already exists on destination database. • Cause: Specified work shift already exists in destination database. • Action: Run Migrate Concurrent Manager Definitions program with Overwrite set to Yes.
Message: 3102	<ul style="list-style-type: none"> • Error: Combined Specialization Rule already exists on destination database. • Cause: Specified Specialization Rule already exists in destination database. • Action: Run Migrate Concurrent Manager Definitions program with Overwrite set to Yes.

Message	Details
Message: 3103	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Error: Concurrent Manager is active in the destination database. • Cause: The Concurrent Manager Definitions Migrator will not migrate an existing Concurrent Manager Definition if the manager is currently active in the destination database. • Action: Deactivate the Concurrent Manager on the destination database or make sure it does not have a current work shift.
Message: 3105	<ul style="list-style-type: none"> • Error: Combined Rule / Oracle Id / Concurrent Program / Request Type / User does not exist on the destination database. • Cause: A specialization rule references an AOL object that does not exist on the destination database. • Action: Create the appropriate AOL object in the destination database or run the Migrate Concurrent Manager Definitions program with Partial Allowed set to Yes (in which case the specialization rule will not get migrated).
Message: 4100	<ul style="list-style-type: none"> • Warning: Printer does not exist on destination database. • Action: Create printer on destination database and assign it to appropriate concurrent program.
Message: 4101	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Internal Error: Error getting CONCURRENT_PROGRAM_ID. • Action: Contact support.

Message	Details
Message: 7101	<ul style="list-style-type: none"> • Internal Error: Error getting EXECUTABLE_ID. • Action: Contact support.
Message: 7200	<ul style="list-style-type: none"> • Internal Error: Error getting REQUEST_SET_ID. • Action: Contact support.
Message: 7201	<ul style="list-style-type: none"> • Internal Error: Error getting REQUEST_SET_PROGRAM_ID. • Action: Contact support.
Message: 7207	<ul style="list-style-type: none"> • Error: Another Request Set has the same translation for the User Request Set Name at the Destination. • Action: Check the data.
Message: 7208	<ul style="list-style-type: none"> • Error: Another Request Set has the same translation for the User Request Set Name at the Destination. • Action: Check the data.
Message: 7300	<ul style="list-style-type: none"> • Internal Error: Error getting FLEX_SET_VALUE_ID. • Action: Contact support.
Message: 7301	<ul style="list-style-type: none"> • Internal Error: Error getting FLEX_VALUE_ID. • Action: Contact support.
Message: 7311	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	Details
Message: 7313	<ul style="list-style-type: none"> • Error: Flexfield Structure for the new value does not exist in destination. • Cause: When copying value set with value attributes, if the new value set name does not exist in DFF Flexfield Segment Values an error is generated. User must add an entry into the DFF Flexfield Segment Values for the new value set before copying a value set with value attributes into a new value set.
Message: 7400	<ul style="list-style-type: none"> • Internal Error: Error getting APPLICATION_ID. • Action: Contact support.
Message: 7406	<ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • The descriptive flexfield references a value set for context override value set that does not exist at the destination. • The descriptive flexfield being migrated has context override value set properties set and the destination version of Oracle E-Business Suite is earlier than 11.5.5, which does not support context override value set features. • Action: Create the value set referenced by the descriptive flexfield at the destination.
Message: 7500	<ul style="list-style-type: none"> • Internal Error: Error getting REQUEST_GROUP_ID. • Action: Contact support.
Message: 7700	<ul style="list-style-type: none"> • Internal Error: Error getting MENU_ID. • Action: Contact support.
Message: 7801	<ul style="list-style-type: none"> • Internal Error: Unable to get ACTION_SEQUENCE_ID in destination database. • Action: Contact support.
Message: 7900	<ul style="list-style-type: none"> • Internal Error: Unable to get RESPONSIBILITY_ID in destination database. • Action: Contact support.

Message	Details
Message: 8000	<ul style="list-style-type: none"> • Internal Error: Unable to get USER_ID in destination database. • Action: Contact support.
Message: 8100	<ul style="list-style-type: none"> • Internal Error: Unable to get PROFILE_OPTION_ID in destination database. • Action: Contact support.
Message: 8200	<ul style="list-style-type: none"> • Internal Error: Unable to get PROFILE_OPTION_ID in destination database. • Action: Contact support.
Message: 8300	<ul style="list-style-type: none"> • Internal Error: Error getting APPLICATION_ID. • Action: Contact support.
Message: 8400	<ul style="list-style-type: none"> • Internal Error: Error getting APPLICATION_ID. • Action: Contact support.
Message: 8500	<ul style="list-style-type: none"> • Internal Error: Error getting AXIS_SET_ID. • Action: Contact support.
Message: 8502	<ul style="list-style-type: none"> • Internal Error: Error getting APPLICATION_ID. • Action: Contact support.
Message: 8502	<ul style="list-style-type: none"> • Internal Error: Error getting EXCEPTION_ID. • Action: Contact support.
Message: 8600	<ul style="list-style-type: none"> • Internal Error: Error getting PRINTER_TYPE. • Action: Contact support.
Message: 8700	<ul style="list-style-type: none"> • Internal Error: Error getting FOLDER_ID. • Action: Contact support.
Message: 8800	<ul style="list-style-type: none"> • Internal Error: Error getting MENU_ID. • Action: Contact support.
Message: 8801	<ul style="list-style-type: none"> • Internal Error: Error getting MENU_ID for child menu. • Action: Contact support.
Message: 8900	<ul style="list-style-type: none"> • Internal Error: Error getting FUNCTION_ID. • Action: Contact support.

Message	Details
Message: 9100	<ul style="list-style-type: none">• Internal Error: Error getting CONCURRENT_QUEUE_ID.• Action: Contact support.
Message: 9101	<ul style="list-style-type: none">• Internal Error: Error getting CONCURRENT_TIME_PERIOD_ID.• Action: Contact support.
Message: 9102	<ul style="list-style-type: none">• Internal Error: Error getting COMPLEX_RULE_ID.• Action: Contact support.
Message: 9102	<ul style="list-style-type: none">• Internal Error: Error getting COMPLEX_RULE_LINE_ID.• Action: Contact support.