

**Oracle® Financial Services Data Migration Scripts:
OFSA 4.5 to OFSAA**

User Guide

Release 5.1

Part No. E18045-01

July 2010

Oracle Financial Services Data Migration Scripts: OFSA 4.5 to OFSAA User Guide, Release 5.1

Part No. E18045-01

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

Primary Author: Jilna Surag

Contributing Author: Jamie Davis

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

Contents

Send Us Your Comments

Preface

1 Introduction

About the Guide.....	1-1
Overview of Migration Tool	1-1
Terminology.....	1-2

2 Preparing a Migration Plan

Overview of Preparing a Migration Plan.....	2-1
Become Familiar with New Release.....	2-1
Upgrade Path.....	2-1
Upgrade Method.....	2-2
Prepare a Backup Strategy.....	2-2
Prepare a Testing Plan.....	2-2
Test Your Migration Plan on a Test Environment.....	2-3

3 Pre-Migration Activities

Steps in Pre-Migration.....	3-1
-----------------------------	-----

4 Migrating the Database

Steps in Migrating the Database.....	4-1
--------------------------------------	-----

5 Post Migration Activities	
Activities after migration.....	5-1
6 Troubleshooting and Logging	
Log File and Status File.....	6-1
A Migration Errors – Trouble Shooting	
Trouble Shooting the Errors.....	A-1
B Table Mapping	
Mapping of OFSA 4.5 tables to OFSAA tables.....	B-1
Users and Folders.....	B-1
Dimension Members and Attributes.....	B-2
Common Tables.....	B-4
Hierarchies and Filters.....	B-6
Common for FTP/ALM.....	B-8
Transfer Pricing/ FTP.....	B-9
Risk Manager/ ALM.....	B-10
Profitability/OFSPM.....	B-14
Ledger, Instrument Tables, Transaction Tables, Lookup Tables.....	B-15
C Objects Not Migrated	
Objects Comments.....	C-1
D Issues	
Release Notes.....	D-1

Send Us Your Comments

Oracle Financial Services Data Migration Scripts: OFSA 4.5 to OFSAA User Guide, Release 5.1 Part No. E18045-01

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document. Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Oracle E-Business Suite Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: appsdoc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

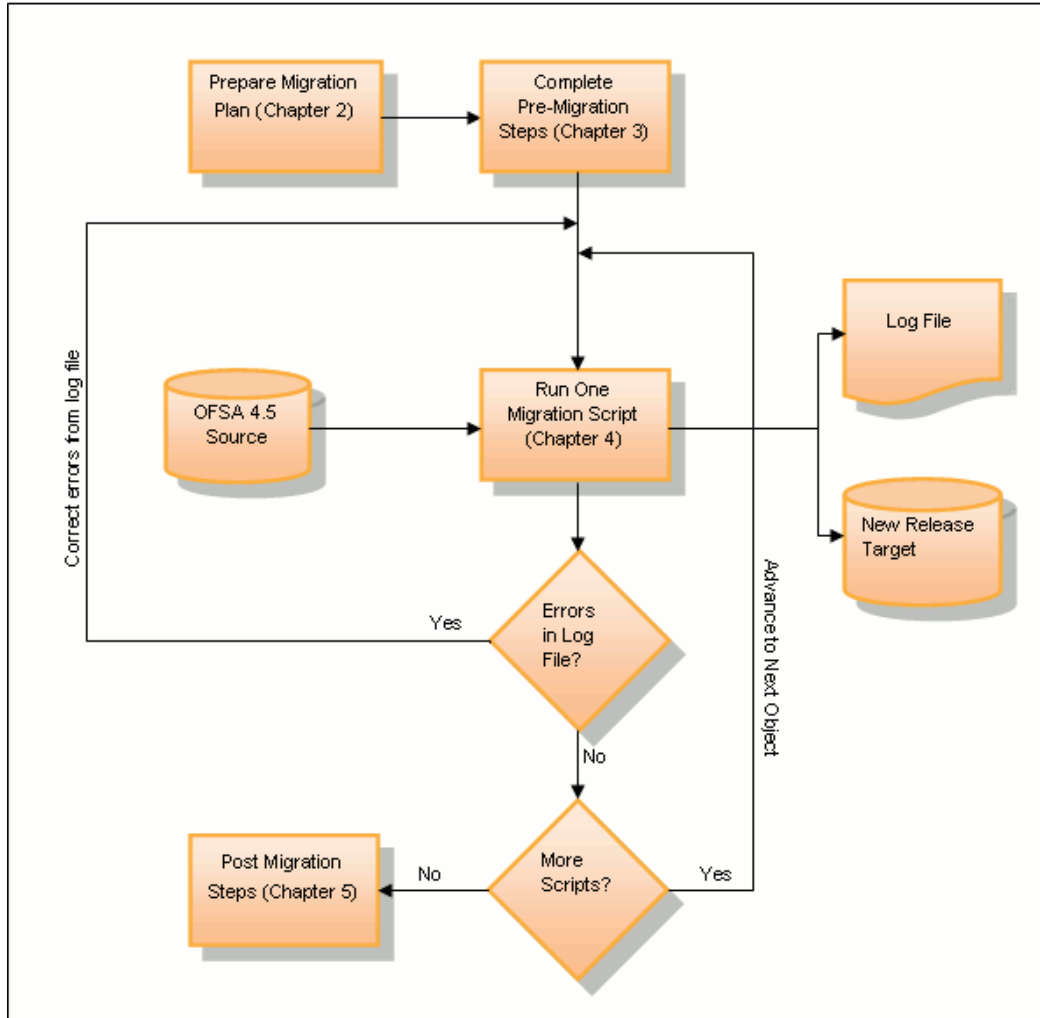
Introduction

About the Guide

This document describes how to use the Oracle Financial Services Migration Tool. It also provides detailed information about the migration process, source and target data models, and acceptance testing of the target system.

Overview of Migration Tool

The Migration Tool is designed to allow customers to easily carry forward as much data as possible to the OFSAAI. Following is the general approach a customer will follow during the migration.



Implementation Note

As with any major release, there are functional differences that result in difficulty or the inability to migrate certain data. In these cases, this guide will try to recommend the best approach to the customer in order to either manually migrate or re-implement using the new application.

Terminology

The name of the Oracle Financial Services Migration Tool has been shortened in this guide to Migration Tool.

The following operating systems have been shortened in this guide to Windows, where appropriate:

- Microsoft Windows NT 4.0

- Microsoft Windows 2000
- Microsoft Windows XP

The name for a number of Unix-based operating systems, including Red Hat Enterprise Linux and Solaris, has been shortened in this guide to UNIX, where appropriate.

Application Abbreviations:

OFSA: Oracle Financial Services Analytical Applications

OFSAI: Oracle Financial Services Analytical Applications Infrastructure

ALM: Asset | Liability Management

FTP: Funds Transfer Pricing

OFSPM: Profitability Management

General Definitions:

InfoDom: An Information Domain is the area of analysis. It forms the backbone for all the analysis inside Infrastructure. It is the knowledge domain where information is stored, which consists of specific data models, Technical and Business definition of data to help in processing data for analysis. An application install may contain one or more Information Domains.

Dimension: Leaf, Processing Leaf, or Processing Dimension. Refers to a leaf in OFSA 4.5 OFSA_CATALOG_OF_LEAVES. There are two types of dimensions Key and Simple. A key dimension can be of type Ledger Only (i.e. financial_elem_id) or Both Instrument and Ledger (i.e. gl_account_id). A simple dimension is also known as a User-Defined Code (i.e. accrual_basis_cd).

Member: Leaf Value or Dimension Value. Members may be of leaf level or node level.

Hierarchy: Refers to Tree Rollup in OFSA 4.5

Segment: Refers to Folder in OFSA 4.5 with similar functionality. Basically a storage mechanism that can be used to restrict access to objects.

Database Schemas:

CONFIG: Stores all administrative and security metadata related to users. An application install can have only one CONFIG schema.

ATOMIC: Stores all the metadata related to FTP, ALM, OFSPM, Dimensions, and Rates within an InfoDom. An application install can have one or more ATOMIC schema.

Preparing a Migration Plan

Overview of Preparing a Migration Plan

Before you execute any migration scripts, the following steps should be taken:

- Become familiar with the features and general usage of the new release
- Determine the upgrade path to the new release
- Choose an upgrade method
- Prepare a backup strategy
- Develop a testing plan
- Test your migration plan on a test database

Become Familiar with New Release

To execute acceptance tests, users involved in the migration process will need to be familiar with the architecture, tools, and user interface of the new release. For more information, see *OFSAA User Guides*. In particular, users will want to become skilled at accessing similar objects such as business rules, dimensions, users, folders, and hierarchies in both source and target installations.

Upgrade Path

For the initial version (Oracle Financial Services Analytical Applications release 5.1 and Infrastructure release 7.1), the following upgrade path is supported:

Current Release	Upgrade Path
OFSA 4.5.39+ Server / Database	Direct upgrade is supported using the Financial Services 4.5 Migration Tool and this guide.

Upgrade Method

For the initial version (Oracle Financial Services Analytical Applications release 5.1 and Infrastructure release 7.1), the only method supported is a manual script-driven process using the Migration Tool described in this guide.

Prepare a Backup Strategy

The ultimate success of your migration depends heavily on the design and execution of an appropriate backup strategy. To develop a backup strategy, consider the following questions:

- How long can the production database remain inoperable before business consequences become intolerable?
- What backup strategy should be used to meet your availability requirements?
- Are backups archived in a safe, offsite location?
- How quickly can backups be restored (including backups in offsite storage)?
- Have recovery procedures been tested successfully?

Prepare a Testing Plan

You need a series of carefully designed tests to validate all stages of the upgrade process. Executed rigorously and completed successfully, these tests ensure that the process of upgrading the production database is well understood, predictable, and successful. Perform as much testing as possible before upgrading the production database. Do not underestimate the importance of a test program.

The testing plan must include the following types of tests:

- **Minimal Testing:** Entails moving all or part of an application from the current database to the new database and running the application without enabling any new features. Minimal testing is a very limited type of testing that may not reveal potential issues that may appear in a "real-world" production environment.

However, minimal testing will immediately reveal any application startup or invocation problems.

- **Functional Testing:** Includes a set of tests in which new and existing functionality of the system are tested after the upgrade. Functional testing includes all database, networking, and business/application components. The objective of functional testing is to verify that each component in common between the source and target systems behaves properly, as well as confirming that new features are working.
- **Performance Testing:** Compares the performance of various business functions on both the source and target systems. Unlike volume or load stress testing, base performance testing is meant to provide a real-world comparison of common usage without the setup and long run times. Successful completion of base performance testing should be considered a prerequisite to volume or load stress testing.

Test Your Migration Plan on a Test Environment

Create a test environment that will not interfere with the current production environment. Practice migration of the application metadata using the test environment. The best test, if possible, is performed on an exact copy of the database to be migrated, rather than on a downsized copy or test data. This will familiarize you with the migration process and minimize unexpected issues.

Pre-Migration Activities

Steps in Pre-Migration

Prior to executing the Migration Tool scripts for specific object types, read the latest release notes to identify any infrastructure and application issues that may impact migration, then follow these steps:

1. **Understand What's New in this Release**

1. **Schema Structure**

OFSAAI install has at a minimum two application schema; *CONFIG* and *ATOMIC*. *CONFIG* schema contains metadata like users, segments, groups, etc. that are related to security. All other objects and metadata related to rules and processes are present in the *ATOMIC* schema. Since all procedures are executed in *ATOMIC* schema, the *ATOMIC* schema owner should have proper grants for tables in *CONFIG* schema.

2. **Dimension Management: OFSA 4.5 Vs. OFSAA**

The OFSA 4.5 processing leaves are known as dimensions in OFSAA. The *leaf_num_id* for leaves in 4.5 is known as *dimension_id* for dimensions in OFSAA. The value of *leaf_num_id* should be retained in OFSAA as *dimension_id* in all the tables where it has been referenced. The list of tables where this synch operation has to be carried out is provided below in **Step 6: Synchronize OFSAAI Data Model with OFSA 4.5**.

In OFSA 4.5 the leaf members and their attributes are stored the same table whereas in OFSAA, display codes, language specific names, and attributes for the members are stored in separate tables.

DIM_<DIMENSION NAME>_B - Display Code, Leaf / Node Indicator, Enable / Disable

DIM_<DIMENSION NAME>_TL - Language specific Name, Description

DIM_<DIMENSION NAME>_ATTR - Attribute value

DIM_<DIMENSION NAME>_HIER - Hierarchy Parent / Child association

Metadata information about dimensions:

REV_DIMENSIONS_B - Display Code, Column Name, Tables, Key / Simple, ...

REV_DIMENSIONS_TL- Language specific Name, Description

Metadata information about dimension attributes:

REV_DIM_ATTRIBUTES_B - Label, Data Type, Default value, ...

REV_DIM_ATTRIBUTES_TL - Language specific Name, Description

Due to business requirement, management of dimensions must follow a dual format approach. OFSAA retains the way leaves were stored in OFSA 4.5.

OFSA_LEAF_DESC	OFSA_NODE_DESC	OFSA_LEVEL_DESC
OFSA_DETAIL_ELEM_B	OFSA_DETAIL_LEAVES	OFSA_DETAIL_ORG_UNIT
OFSA_DETAIL_OTHER_COA	OFSA_IDT_ROLLUP	

3. New seeded dimension: "Product"

OFSAA introduces a new seeded dimension "Product" with dimension_id (leaf_num_id) = 4. Since there are dependencies on the dimension_id (e.g., for business rules), if you have any user-defined dimensions in OFSA 4.5, you will need to retain the dimension_id of the dimension while adding it in the target instance. There is a potential conflict if you have already defined a dimension called "Product" or leaf_num_id of 4 is assigned to a different user-defined dimension.

We suggest the following actions when appropriate for these three cases:

1. OFSA 4.5 has a dimension "Product" with leaf_num_id = 4.

Solution: No action required.

2. OFSA 4.5 has a dimension "Product" with leaf_num_id <> 4 (e.g. 9).

Solution: Update the dimension_id of the "Product" in the new instance to reflect the leaf_num_id of the "Product" in 4.5.3 .

```

update rev_dimensions_b
set dimension_id = <Product leaf_num_id>
where dimension_id = 4;

```

```

update rev_dimensions_tl
set dimension_id = <Product leaf_num_id>
where dimension_id = 4;

```

```

update rev_dim_attributes_b
set dimension_id = <Product leaf_num_id>
where dimension_id = 4;

```

```

update rev_dim_attributes_tl
set dimension_id = <Product leaf_num_id>
where dimension_id = 4;

```

3. OFSA 4.5 has a different user-defined dimension (i.e. channel_id) with leaf_num_id = 4.

Solution: Update the dimension_id of the "Product" in the new instance using a value of MAX (leaf_num_id in OFSA 4.5) +1.

```

update rev_dimensions_b
set dimension_id = (select max(leaf_num_id)
                   from ofsa_catalog_of_leaves@<dblink>)
where dimension_id = 4;

```

```

update rev_dimensions_tl
set dimension_id = (select max(leaf_num_id)
                   from ofsa_catalog_of_leaves@<dblink>)
where dimension_id = 4;

```

```

update rev_dim_attributes_b
set dimension_id = (select max(leaf_num_id)
                   from ofsa_catalog_of_leaves@<dblink>)
where dimension_id = 4;

```

```

update rev_dim_attributes_tl
set dimension_id = (select max(leaf_num_id)
                   from ofsa_catalog_of_leaves@<dblink>)
where dimension_id = 4;

```

Please ensure you retain the leaf_num_id for other user-defined dimensions and carry them forward to OFSAA. You can use the following query to find the dimension_id (leaf_num_id) of all the key dimensions.

In Source:

```
select * from ofsa_catalog_of_leaves;
```

In Target:

```
select * from rev_dimensions_b where simple_dimension_flag =
'N' ;
```

4. **[Optional] Cleanup:**

Hierarchies (Tree Rollups) and Hierarchy (Tree) Filters

Since the dimension member table in OFSAA includes leaf and node members, the migration converts the OFSA 4.5 tree rollup nodes to new unique node

dimension members. Therefore, you may want to delete obsolete tree rollups from your 4.5 instance prior to migration. This will help to avoid "clutter" of unnecessary node members in the dimension member pool of the target database.

If any nodes exist in tree filters which are not in the underlying tree rollup, the tree rollup will be noted in the status log and the tree filter will not be migrated.

Result Tables

Since result tables may contain a tremendous amount of data which may be updated when you re-run a process, you may wish to save migration processing time by removing result data prior to migration. This can include:

1. FTP and ALM detail cash flow audit records (OFSA_PROCESS_CASH_FLOWS). This data is migrated when you run the FTP or ALM rules script.

2. ALM:

ALM result data is migrated when you run the script for ALM result data ("migrate_result_tables"). If you don't want to migrate any of the OFSA 4.5 ALM results, you can simply skip that script.

Note: Some ALM process rules may have a selective reprocessing flag on, which impacts the set of data it will overwrite when you rerun the process.

OFSA_EXCHANGE_RATES_AUDIT

OFSA_INTEREST_RATES_AUDIT

RES_DTL%

CONS_DTL%

OFSA_RESULT_MASTER

OFSA_CONSOLIDATED_MASTER

OFSA_RESULT_SCENARIO

OFSA_IDT_RESULT_HEADER

OFSA_RESULT_BUCKET

OFSA_TM_STOCH_VAR

OFSA_TM_STOCH_TOT_VAR

OFSA_TM_STOCH_MKT_VAL

Users

Username cannot exceed 20 characters in length. Run the following SQL to

determine if any usernames exceed this limit.

```
SELECT DISTINCT
  username
,CASE WHEN LENGTH(username) > 20
      THEN 'FAIL: UserName cannot exceed length of 20'
      ELSE 'OK'
      END user_status
FROM ofsa_users;
```

```
SELECT DISTINCT
  access_name
,CASE WHEN LENGTH(access_name) > 20
      THEN 'FAIL: UserName cannot exceed length of 20'
      ELSE 'OK'
      END user_status
FROM ofsa_catalog_of_ids;
```

```
SELECT DISTINCT
  chg_login
,CASE WHEN LENGTH(chg_login) > 20
      THEN 'FAIL: UserName cannot exceed length of 20'
      ELSE 'OK'
      END user_status
FROM ofsa_catalog_of_ids;
```

Folders

Folder names cannot exceed 10 chars in length. Run the following SQL to determine if any folders exceed this limit.

```
SELECT DISTINCT
  folder_name
,CASE WHEN LENGTH(folder_name) > 10
      THEN 'FAIL: Folder Name cannot exceed length of 10'
      ELSE 'OK'
      END folder_status
FROM ofsa_id_folders;
```

```
SELECT
  folder_name
,CASE WHEN LENGTH(folder_name) > 10
      THEN 'FAIL: Folder Name cannot exceed length of 10'
      ELSE 'OK'
      END folder_status
FROM ofsa_catalog_of_ids;
```

Object Names and Descriptions

The following special characters are not allowed in names or descriptions in the new release:

1. &
2. @
3. ~

4. '
5. +

To accommodate these restrictions, if you use any of these special characters in names or descriptions, you can remove or replace them prior to migration. Otherwise the migration scripts will use the following logic:

- Names and Descriptions: Replace special character with underscore.
- Names checked for uniqueness. In the unlikely case a replacement makes a name non-unique, it will prefix the name with a sequential number.

Use the following SQL to determine if any names or descriptions contain special characters.

```

SELECT
  sys_id_num
, id_desc_short
, id_desc_long
FROM (
  SELECT
    sys_id_num
  , id_desc_short
  , id_desc_long
  , CASE WHEN INSTR(id_desc_short, '&')>0 OR
    INSTR(id_desc_short, '@')>0
    OR INSTR(id_desc_short, '~')>0 OR
    INSTR(id_desc_short, '''')>0
    OR INSTR(id_desc_short, '+')>0
    THEN '1'
    ELSE '0' END short_status
  , CASE WHEN INSTR(id_desc_long, '&')>0 OR
    INSTR(id_desc_long, '@')>0
    OR INSTR(id_desc_long, '~')>0 OR
    INSTR(id_desc_long, '''')>0
    OR INSTR(id_desc_long, '+')>0
    THEN '1'
    ELSE '0' END long_status
  FROM ofsa_catalog_of_ids)
WHERE short_status = 1 or long_status = 1;

```

2. Backup Procedure

At a minimum take a backup of CONFIG and ATOMIC schema. Make sure the folder has necessary permission to create dump and log files.

You can use the below command to create a backup –

```
exp <schema name>/<password>@<database sid> file=<export filename>
log=<log filename> full=N
```

3. Set aside enough space

1. Target Database

Make sure the OFSAA target database has sufficient space to execute the migration tool. The target database size should be double the size of the source OFSA 4.5 database or at least two times the size all product tables (All Rule + Common + Dimension Management) + 1 times size of all the fact tables (Instrument, Transaction, Ledger, and User-defined Lookup tables) + Results tables. Your database administrator can assist with this task.

2. RAM

The migration scripts require a minimum of 800MB RAM.

4. Rate Manager Validation in OFSA 4.5

OFSAA 4.5 has a feature to validate all the exchange rates – this should be executed for all the exchange rates defined in Rate Manager.

Validate all currency conversion rates defined in OFSA 4.5 in order to avoid any loss of data. Only currency rates with exchange_rate_status_cd of 1 can be used by processing engine. If the currency validation is not performed prior to migration, then the currency rates with exchange_rate_status_cd of 0-Not Yet Validated, 2-Invalid, and 3-Processing, will continue to be in the same state in OFSAA.

In OFSA 4.5, log in to Rate Manager; go to Tools ->Run ->Exchange Rate Validation.

5. Create reference IRC

This step applies to customers with ALM or FTP installed and where rate index rules have been defined. This is due to the design change in OFSAA where the valuation curve code becomes a selection in the rate index rule instead of the FTP and ALM process rules.

Within OFSA 4.5 Rate Manager, assign a reference IRC to each active currency.

In OFSA 4.5, you may also want to create additional rate indexes and select them in the process in such a way that a rate index is always used in combination with the same valuation curve code as you have selected in the OFSA 4.5 process rule.

6. Synchronize OFSAAI Data Model with OFSA 4.5

If you have registered any user-defined columns, tables and indexes in OFSA 4.5, you will need to create them in the OFSAA instance prior to migration. The column names in the OFSAA instance should match the names in the OFSA 4.5 instance, but the column order does not have to match. There may be additional columns in the OFSAA instance due to new requirements for certain table classifications.

For more information regarding the steps for adding user-defined dimensions, columns, tables, and so on, through ERWin to create logical and physical objects for user-defined objects, and various utilities for registering and validating the additions, see *OFSAA Infrastructure User Guide release 7.1*. This will help to ensure that the proper data model is in place and all seeded information and metadata updates are captured before starting migration.

Synchronize Dimension Metadata:

Please set the `dimension_id` in the tables listed below to match the `leaf_num_id` of leaves stored in `OFSA_CATALOG_OF_LEAVES`:

- `REV_DIMENSIONS_B`
- `REV_DIMENSIONS_TL`
- `REV_DIM_ATTRIBUTES_B`
- `REV_DIM_ATTRIBUTES_TL`

Note: It is important to retain the `leaf_num_id` used in OFSA 4.5 `OFSA_CATALOG_OF_LEAVES`, as this number is used in other dependent tables (e.g. `OFSA_CATALOG_OF_LEAVES` and its OFSAA target table).

7. Set up required GRANT permission

1. Grants to be created in CONFIG schema

The migration is executed from the `ATOMIC` schema. Therefore, it is essential that the `ATOMIC` schema be given select, insert, and update privileges to the tables in the `CONFIG` schema. For migration, these privileges should be assigned to the tables `CSSMS_GROUP_MAST`, `CSSMS_USR_PROFILE`, `CSSMS_USR_GROUP_MAP`, `CSSMS_SEGMENT_MAST`, `DSNMASTER`, `DB_MASTER`.

Log into `CONFIG` schema and execute the following commands.

```
GRANT SELECT, INSERT, UPDATE on CSSMS_GROUP_MAST to <ATOMIC SCHEMA>;
```

```
GRANT SELECT, INSERT, UPDATE on CSSMS_USR_PROFILE to <ATOMIC SCHEMA>;
```

```
GRANT SELECT, INSERT, UPDATE on CSSMS_USR_GROUP_MAP to <ATOMIC SCHEMA>;
```

```
GRANT SELECT, INSERT, UPDATE on CSSMS_SEGMENT_MAST to <ATOMIC SCHEMA>;
```

```
GRANT SELECT, INSERT, UPDATE on DSNMASTER to <ATOMIC SCHEMA>;
```

```
GRANT SELECT, INSERT, UPDATE on DB_MASTER to <ATOMIC SCHEMA>;
```

2. Grants to be created to ATOMIC schema

For logging, a directory has to be created on the server (E.g., `/home/<user>/log`) and it has to be mapped using the create directory command.

Grant create directory permission to ATOMIC schema:

```
GRANT CREATE ANY DIRECTORY to <ATOMIC SCHEMA>;
```

Create Directory 'ERRORDIR' using the following command:

```
CREATE OR REPLACE DIRECTORY ERRORDIR as '<Absolute Path of Log Directory>';
```

E.g., in Windows

```
CREATE OR REPLACE DIRECTORY ERRORDIR as 'C:\Migration\Log';
```

E.g., in Linux

```
CREATE OR REPLACE DIRECTORY ERRORDIR as '/home/<user>/log';
```

Grant exec privilege on UTL_FILE to ATOMIC schema

```
GRANT EXECUTE on sys.utl_file to <ATOMIC SCHEMA>;
```

Make sure the log directory has write privileges for others if you created the directory as a non-DBA OS group user. Proper file write permission should be granted for the folder in which the log files will be written.

Log into ATOMIC schema and run the below commands to test privileges.

```
declare
  outFile utl_file.file_type;
begin
  outFile := utl_file.fopen('ERRORDIR', 'Test.txt', 'W');
  utl_file.put_line(outFile, 'Testing the Output in linux');
  utl_file.fclose(outFile);
end;
/
```

Please check that the file "Test.txt" is generated successfully in the log directory you set in Step 7(b). If the file is not created, please review/repeat the step.

8. Create DB Link

Log into target database via SQL Plus and create a database link from target database (OFSAA) to the source database (OFSAA 4.5). You will need the following information about the source database: Oracle SID, OFSA Owner, and OFSA Owner Password to create this link. Use the following commands at the SQL prompt of target database to grant the permission to create database link and create the database link by SYS or user with equivalent privileges:

To Grant permission to create DB Link

```
GRANT CREATE DATABASE LINK TO <ATOMIC SCHEMA>;
CREATE DATABASE LINK <DB_LINK>
  CONNECT TO <OFSAA OWNER>
  IDENTIFIED BY <OFSAA OWNER PWD>
  USING <SOURCE SID>;
```

The <DB_LINK> will be passed as parameter in all the migration scripts (see also: Step 11, where you will need the database link name so you can update this parameter within the migration script). Only a user with DBA role can create this link. Please contact your DBA or Oracle support for further help.

9. Set up your shell environment on Target Database server

```
export ORACLE_HOME=$myhome
```

```
export ORACLE_SID=$mysid
```

\$ORACLE_HOME/bin/oraenv (assuming this prompts you for the information regarding your new instance, to update TNSNAMES)

10. Copy Migration Packages

- Create a new directory "Migration" under home directory (or the user directory) and copy the following scripts in target database machine in text mode:
 1. Migration_Stagging_Tables.sql
 2. fsi_migds.sql
 3. fsi_migdb.sql
 4. fsi_migups.sql
 5. fsi_migupb.sql
 6. fsi_utils.sql
 7. fsi_utilb.sql
- Edit Migration_Stagging_Tables.sql: For all occurrences of "G_DBLINK" within the file, replace with database link name, which you created in step 8.

11. Create Staging Tables

Log into sqlplus, and execute the following at the prompt:

```
SQL> spool migrationStaggingTables.log  
SQL> @Migration_Stagging_Tables.sql  
SQL> spool off
```

Check the contents of **migrationStaggingTables.log** in the current directory for any errors and **correct before continuing**.

Log into **ATOMIC** schema and execute the following scripts in the order given below to create the packages and procedures necessary for migration:

```
SQL> @fsi_utils.sql  
SQL> @fsi_migds.sql  
SQL> @fsi_migups.sql  
SQL> @fsi_migdb.sql  
SQL> @fsi_migupb.sql  
SQL> @fsi_utilb.sql
```

12. NLS_CHARACTERSET Warning

If your NLS_CHARACTERSET on Target and Source instances are different, please be cautious of possible character conversion when you try to copy some tables from Source to Target. In order to avoid character conversion, use the DDL table creation statement from the Source instance in creating the table in the Target instance and make sure the table definition remain the same.

13. Pre-Migration: Important Checklist

Before running the migration scripts, please confirm the following have been successfully completed:

1. Data model upload for the new release is completed in the target database
2. DB link from target to source was created with privilege granted to ATOMIC schema.
3. All the migration scripts, provided in Step 11 above, were executed successfully in target database.
4. Tables related to user-defined dimensions have been created in the target database in the new format DIM_<DIMENSION NAME>_B, DIM_<DIMENSION NAME>_TL, DIM_<DIMENSION NAME>_ATTR, and DIM_<DIMENSION NAME>_HIER.
5. Metadata about the user defined dimensions exists in REV_DIMENSIONS_B and REV_DIMENSION_TL; these tables are populated when you run the "Add Dimension" utility. This metadata is crucial for migration of user defined dimensions.
6. User-Defined instrument and transaction tables have been created in the target database and include the new product_id dimension, defined as number(14). The dimension product_id should already exist as a seeded column in the seeded instrument tables.
7. Target database has sufficient space to execute the migration. Target database size must be approximately 2 times the source 4.5 database size; see details earlier in this document. Please contact your DBA to find the tablespace and database size in source as well as target database.
8. A physical directory in the server has been mapped to logical directory using the create directory command of Step 7(b).
9. Sufficient RAM is available before running the scripts. (Minimum 800MB).
10. For the rate index rules in ALM and FTP stochastic processing: Confirm that a reference currency has been assigned in OFSA 4.5 to each active currency there.

Migrating the Database

Steps in Migrating the Database

At this point, you are prepared to migrate the data from the original OFSA 4.5 source to the OFSAA target database. All the scripts below are assumed to be run from a UNIX shell with the same environment as defined in the Chapter 3. Some of the pre-migration steps discussed earlier in this document are highlighted again here for your reference. For more information on specific objects migrated within each step, see Table Mapping, page B-1.

1. Folders and Users

1. Prerequisite

Ensure that all the CMMS_* tables are available and verify that the ATOMIC user has insert privilege on them.

2. How to execute the procedure

Log into target DB *ATOMIC* schema. Run the following script to migrate users and folders:

```
SQL> set serveroutput on
SQL> exec fsi_util.migrate_user_folder_tbls (<CONFIG
SCHEMA>, <G_DBLINK>);
```

where<CONFIG SCHEMA>is the name of CONFIG schema and<G_DBLINK>is the database link from target to source.

Check the contents of **status.txt** and **migrateUserFolderData.log** in the log directory for any errors and correct before continuing.

2. Dimensions - Seeded and User Defined:

1. Prerequisite

Check that the metadata info for all dimensions in

OFSA_CATALOG_OF_LEAVES of source database is seeded in REV_DIMENSION_B and REV_DIMENSION_TL. This metadata info is captured when you run the "Add Dimension" utility. If metadata for any dimension is missing in REV_DIMENSION_B or REV_DIMENSION_TL table, that dimension will not be migrated.

2. How to execute the procedure

Log into target DB ATOMIC schema. Run the following scripts to migrate dimension data, such as dimension members and attributes:

```
SQL> set serveroutput on
SQL> exec fsi_util.migrate_dimension (<G_DBLINK>);
```

where <G_DBLINK> is the database link from target to source.

Check the contents of **status.txt** and **migrateDimensionData.log** in the current directory for any errors and correct before continuing.

3. Common Tables

1. Prerequisite

Dimension data must be migrated before you execute this procedure.

2. How to execute the procedure

Log into target DB ATOMIC schema. Run the following scripts to migrate common tables, such as interest rate codes, fiscal year info, etc.:

```
SQL> set serveroutput on
SQL> exec fsi_util.migrate_common_tbls (<G_DBLINK>)
```

where <G_DBLINK> is the database link from target to source.

Check the contents of **status.txt** and **migrateCommonData.log** in the current directory for any errors and correct before continuing.

4. Hierarchies

1. Prerequisite

Dimension data and common tables migration must be completed before you execute this procedure. Confirm that all the DIM_<DIMENSION NAME>_HIER tables exist for all seeded and user defined processing dimensions. Confirm that the metadata information for all the dimensions in your OFSA 4.5 OFSA_CATALOG_OF_LEAVES table is available in the REV_DIMENSION_B and REV_DIMENSION_TL tables and that the leaf_num_id of each dimension is preserved.

2. How to execute the procedure

Log into target DB ATOMIC schema. Run the following script to migrate

hierarchies:

```
SQL> set serveroutput on
SQL> exec fsi_util.migrate_hierarchy(<G_DBLINK>)
```

where <G_DBLINK> is the database link from target to source.

Check the contents of **status.txt** and **migrateHierarchy.log** in the current directory for any errors and correct before continuing.

5. Business Rules and Results

1. Prerequisite

Log into target DB *ATOMIC* schema. This procedure should be executed only after Steps 1 to 4 (listed above) are executed successfully.

2. How to execute the procedure

There are separate migration scripts for each set of application-specific objects, categorized according to the associated OFSAA application. Objects shared by more than one application are handled internally in each procedure individually, but in such a way that they are migrated only once irrespective of which application script is run first.

Run the following scripts in the order given below to migrate various business rules:

- Funds Transfer Pricing Rule Migration:

```
SQL> set serveroutput on
SQL> exec fsi_util.migrate_tpRule_tbls (<G_DBLINK>)
```

where <G_DBLINK> is the database link from target to source.

Check the contents of **status.txt** and **migrateTPRule.log** in the current directory for any errors and correct before continuing.

- Asset | Liability Management Rule Migration

```
SQL> set serveroutput on
SQL> exec fsi_util.migrate_almRule_tbls (<G_DBLINK>)
```

where <G_DBLINK> is the database link from target to source.

Check the contents of **status.txt** and **migrateALMRule.log** in the current directory for any errors and correct before continuing.

Note: Rules shared by FTP and ALM

The FTP and ALM rule migration scripts contain logic to ensure that rules shared by the two applications are only migrated once. For example, if you run the FTP rule migration it will migrate your prepayment rules; the ALM

rule script will then recognize that these rules have already been migrated and will not attempt to migrate them again.

- Asset | Liability Management Results Tables Migration

Prerequisite:

Run ALM rule related tables migration (see above).

Execute the procedure:

```
SQL> set serveroutput on
SQL> exec fsi_util.migrate_result_tables (G_DBLINK)
```

where <G_DBLINK> is the database link from target to source.

Check the contents of **status.txt** and **migrateResultTablesData.log** in the current directory for any errors and correct before continuing.

- Profitability Management Rule Related Tables Migration.

```
SQL> set serveroutput on
SQL> exec fsi_util.migrate_pftRule_tbls(<G_DBLINK>)
```

where <G_DBLINK> is the database link from target to source.

Check the contents of **status.txt** and **migratePFTRule.log** in the current directory for any errors and correct before continuing.

6. Fact Tables: Ledger, Instrument, Transaction, and Lookup tables

1. Prerequisite

This procedure can be executed independently to migrate ledger, instrument, transaction, or user-defined lookup tables, one at a time. As discussed earlier, ensure that all the tables exist in the target instance and that they include the `product_id` dimension. If the OFSA 4.5 source table includes a `product_id` dimension, then the value of that leaf (dimension member) will be migrated, otherwise the migration scripts will automatically populate this column with "-1".

The instrument or transaction table can have a new name in the target instance. However, as discussed earlier, the target structure should be similar: At minimum, it should include all column names defined in the OFSA 4.5 source table. The migration procedure expects the new target table name as a parameter. If you want to retain the same name (recommended), then the first two parameters will have the same value.

Important: If you specify a target table name different than the source table name, any objects which depend on the original

table name will be affected. This could impact (but is not limited to) objects such as your target data element filters, expressions, ALM or FTP process rules, etc., where OFSA 4.5 had a specific table selection.

2. How to execute the procedure

Run the following script to migrate ledger, instrument, transaction, or user-defined lookup tables:

```
SQL> set serveroutput on
SQL> exec fsi_util.migrate_user_acc_tables(<SRC TABLE>,<TRG
TABLE>, <G_DBLINK>) ;
```

where <SRC TABLE> is the OFSA 4.5 source table name, <TRG TABLE> is the OFSAA target table name, and <G_DBLINK> is the database link between target and source.

Check the contents of **status.txt** and **migrateInstruments.log** in the current directory for any errors and correct before continuing.

7. Generic Script to Migrate Any Table

1. Prerequisite

This module migrates any generic table which doesn't contain any processing leaf. The structure of the table in the target instance must match the structure of the OFSA 4.5 table you are migrating.

2. How to Execute the Procedure

Run the following script to migrate other user-defined tables:

```
SQL> set serveroutput on
SQL> exec fsi_util.migrate_any_table(<SRC TABLE>,<G_DBLINK >);
```

where <SRC TABLE> is the OFSA 4.5 source table name, and <G_DBLINK> is the database link between target and source.

Check the contents of **status.txt** and **migrateGenericData.log** in the current directory for any errors and correct before continuing.

Post Migration Activities

Activities after migration

- **Passwords**

Passwords are not migrated from OFSA 4.5. The user will be prompted for a new password upon initial login.

- **User mapping: Groups and Roles**

In the OFSAAI security design, users are mapped to groups, and groups are mapped to roles. Migrated users need to be mapped to one or more user groups, which in turn must be mapped to specific segments and roles for any of the installed applications, so users will have access to appropriate rules and other objects in the application.

- **Preferences (Global and Application-Specific)**

Application administrators should set the global preferences and application preferences (for each installed application) through the UI. Administrators can define settings at the level for 'All Users', and designate whether each setting is editable by individual users.

Some of the application preferences options will vary by application, such as auto-balancing leaves for ALM, as of date, etc.

After the administrator has defined global and application preferences, each individual user should review his own preferences settings. For more information on preferences, see *OFSAAI User Guide*.

- **FTP processes**

1. Processes with both transfer pricing and option cost selections

In OFSA 4.5, the transfer pricing process rule allows users to combine standard transfer pricing and stochastic / option cost selections. In OFSAA, these are split

into two separate process rule types. For process rules which have both transfer pricing and option cost selections, only the transfer pricing selections will be migrated. If you wish to produce option cost results based on the combined OFSA 4.5 rule, you can build new process rules through the FTP UI.

2. Processes with propagation selection

Propagation settings are not migrated due to the design change between OFSA 4.5 and OFSAA. In OFSA 4.5, propagation selections were available within the process rule. In OFSAA, propagation patterns are globally defined in a separate UI. If you process FTP using the rate propagation feature, please check and update the propagation patterns UI in the OFSAA.

3. Processes with only option cost selections (no transfer pricing selections)

(Also see "Pre-Migration" Step 5.)

In OFSA 4.5 the valuation curve selection is in the process rule. In OFSAA, the valuation curve selection is in the stochastic rate indexing rule, so the migration scripts cannot migrate the OFSA 4.5 valuation curve selection. After migration, you may want to change the valuation curve selection associated with the functional currency in your OFSAA stochastic rate indexing rules, so the selection there is in synch with the valuation curve selection in your OFSA 4.5 process.

4. Processes with ledger migration selection

In OFSA 4.5, the process reads the selection in the active configuration. In OFSAA FTP , accrual basis is a selection in the process rule. The migration scripts default this to product accrual basis. This may not be the same selection you ran with in OFSA 4.5 from the configuration. Please check applicable process rules and select the accrual basis accordingly.

- **Rate Index Rules in FTP and ALM**

See notes under FTP and ALM process rules, regarding valuation curve selection, and see "Pre-Migration" step 5.

- **ALM: Active Time Bucket rule**

The migration scripts create a new time bucket rule for every configuration which has time buckets defined. Since only one time bucket rule can be active at a given time, the migration scripts determine which configuration (with time buckets defined) has the most recent update, and use that set of time buckets as the active time bucket rule.

You may want to activate a different time bucket rule prior to launching an ALM process.

- **ALM processes**

1. Transfer Pricing Flag

The transfer pricing flag will be turned off for all deterministic processes. If any of your OFSA 4.5 processes have the transfer pricing flag set on, you can navigate to the process in the OFSAA UI, set the switch on, and select an appropriate transfer pricing rule.

2. Stochastic Processes: Valuation Curve selection

(See also: pre-migration step 5.)

In OFSA 4.5 the valuation curve selection is in the process rule. In OFSAA, the valuation curve selection is in the stochastic rate indexing rule, so the migration scripts cannot migrate the OFSA 4.5 valuation curve selection. After migration, you may want to change the valuation curve selection associated with the reporting currency in your OFSAA stochastic rate indexing rules, so the selection there is in synch with the valuation curve selection in your OFSA 4.5 process.

- **OFSPM: Allocations that incorporate parentheses:**

The migration scripts do not migrate allocations that incorporate parentheses, as the new release does not support parentheses. If you have any such allocations, you can rebuild them in OFSAA, which supports nested expressions (equivalent to OFSA 4.5 formula id) such that an allocation can "filter on" an expression. OFSAA allocation rules also support expressions in their drivers.

- **User-Defined Views**

The migration scripts do not migrate custom views. To create a single table view in OFSAA:

1. Create logical/physical in ERWin data with same name/structure as view.
2. Register this table by import data model and running the validation routine.
3. Drop table and create view with same name in the schema directly.

Troubleshooting and Logging

Log File and Status File

Each procedure produces a log and status file. These files are overwritten each time you run the procedure. The log file contains the list of tables and number of rows migrated for these tables. Most procedures contain two parts: Staging and Target Migration; the log captures the details for both. (For instrument, transaction, and ledger tables, there are no staging tables.) The log is produced in the LOG DIRECTORY created in pre-migration step 7b. The status file is created by the spool command when executing SQL. The status file gives the status of the procedure executed. If there is an error, the error along with possible solution is provided. If the procedure does not complete successfully, you must correct the error before proceeding.

Procedure names and the associated log file names are:

Procedure Description	Procedure Description	Log File Generated
Users And Folders	migrate_user_folder_tbls	migrateUserFolderData.log
Dimensions	migrate_dimension	migrateDimensionData.log
Common Tables	migrate_common_tbls	migrateCommonData.log
Hierarchies	Migrate_hierarchy	migrateHierarchy.log
Funds Transfer Pricing Rules	migrate_tpRule_tbls	migrateTPRule.log
ALM Rules	migrate_almRule_tbls	migrateALMRule.log
ALM results	migrate_result_tables	migrateResultTablesData.log

OFSPM Rules	migrate_pftRule_tbls	migratePFTRule.log
Ledger, Instrument, and Transaction Tables, and User-Defined Lookup tables	migrate_user_Acc_tables	migrateUserAccData.log
Other user-defined tables	migrate_any_table	migrateGenericData.log

Sample Migration log file

```

Time of Procedure Execution : 26-NOV-2009 06:11:32
INSERTING INTO TABLE FSI_USERS_T
***** Inserted 61 Rows Successfully in FSI_USERS_T
INSERTING INTO TABLE FSI_USER_GROUPS_T
***** Inserted 30 Rows Successfully in FSI_USER_GROUPS_T

***** Starting Migration From Staging to Target *****
INSERTING INTO TABLE CSSMS_USR_PROFILE
***** Inserted 61 Rows Successfully in CSSMS_USR_PROFILE
INSERTING INTO TABLE CSSMS_GROUP_MAST
***** Inserted 30 Rows Successfully in CSSMS_GROUP_MAST

```

Sample Status.txt when there is no Error

```

Last Procedure Executed :- MIGRATE_ALMRULE_TBL
Time of Procedure Execution : 11-DEC-2009 05:12:35
Procedure MIGRATE_ALMRULE_TBL Completed Successfully
*** NO Errors found ***

```

Sample Status.txt when there is an Error

```

Last Procedure Executed :- MIGRATE_COMMON_TBL
Time of Procedure Execution : 11-DEC-2009 12:12:19
Procedure MIGRATE_COMMON_TBL Executed with ERRORS
*** Errors Details ***

Table Name :- 'LEDGER_STAT'
Oracle Error Message :- ORA-00904: "PRODUCT_ID": invalid identifier

Suggested Solution (Please Contact OFFS Support):
=====
Please check the structure of the Table Name :- 'LEDGER_STAT'. A column
(See Error Message) is missing.

```

Migration Errors – Trouble Shooting

Trouble Shooting the Errors

This section gives basic troubleshooting for the errors that might be faced during the migration process. If the error is not listed here or you would like further details regarding the solution, please contact your System Administrator or Oracle Support.

You can query FSI_ERROR_LOG table to see the error message. This table is created by the migration scripts and stores error messages after every procedure is run. When you execute each procedure, it automatically truncates this table. You may also manually delete the contents.

Error Description	Solution	Occurrence	Severity
ORA-02019: connection description for remote database not found	<i>Check whether the Database Link passed as a parameter is Created see Pre-Migration Activities, page 3-1</i>	Low	Low
ORA-00942: table or view does not exist	<i>Check whether the table is present or not</i>	Medium	Low
ORA-00904: "<COLUMN NAME>": invalid identifier E.g., ORA-00904: "SYS_ID_NUM": invalid identifier	<i>Check the table name in FSI_ERROR_LOG. Check whether the column name captured in the ERROR is present in the table. In the example, if Table Name points to ABC then ensure Table ABC contains "SYS_ID_NUM" column.</i>	High	Low

ORA-01722: invalid number	<i>Check the datatype of the Table column. A Number Data type was expected by procedure whereas Varchar2/Char was passed to it.</i>	Medium	Low
ORA-00001: unique constraint (<PRIMARY KEY NAME>) violated For example : ORA-00001: unique constraint (FSI_DB_INFO_PK) violated	<i>A duplicate row exists and cannot be inserted into a table. This should not occur. Please check with your DBA or contact Oracle Helpdesk</i>	Low	High
ORA-06502: PL/SQL: numeric or value error: number precision too large	<i>Table column has size smaller than expected. Please check the table name and column mapping.</i>	Low	Low
ORA-12899: value too large for column	<i>Check the NLS_CHARACTERSET of Target and Source. Typical OFSA 4.5 instance is set to US7ASCII, whereby OFSAAI instance is set to UTF8. When NLS_CHARACTERSET differs character byte lengths can be different.</i>	Low	High

Table Mapping

Mapping of OFSA 4.5 tables to OFSAA tables

This section describes the mapping of OFSA 4.5 tables to the OFSAA tables, categorized by associated application, functional object, or commonality across applications.

See also: Pre-Migration Activities, page 3-1 and Post Migration Activities, page 5-1.

Users and Folders

Users:

The underlying paradigm for user, user group, and folders has changed in OFSAA, but the user experience should be similar to OFSA 4.5. In OFSAA, security management functions are mapped to roles. Roles and segments are mapped to user groups, and users are mapped to user groups.

Passwords will not be migrated; users will be prompted for a new password during initial login.

Folders:

All the existing folders will be migrated to OFSAA format tables. The rules will be migrated into the same folders as they were in OFSA 4.5.

For more information, see Post-Migration activities, page 5-1.

Object Category	Source	Target
Users	OFSA_USER_GROUP_ASSIGNMENT	CSSMS_USR_GROUP_MAP

Object Category	Source	Target
	OFSA_USERS	CSSMS_USR_PROFILE
	OFSA_USER_GROUPS	CSSMS_GROUP_MAST
Preferences	OFSA_IDT_CONFIGURE	REV_APP_USER_PREFERENCES
Folders	OFSA_ID_FOLDERS	CSSMS_SEGMENT_MAST

Dimension Members and Attributes

See also: Hierarchies, page B-6.

In OFSAA, processing dimension data is held in dual format: It is stored in legacy OFSA 4.5 format and the new OFSAA format, to support engines and UIs which will ultimately run off the newer format in a future release, enabling greater flexibility in dimension management. A reverse population utility manages member data from the new format to legacy-format tables within the target database.

Corresponding to each dimension, in the OFSAA format, there are four tables:

DIM_<DIMENSION NAME>_B - Display Code, Leaf / Node Indicator, Enable / Disable

DIM_<DIMENSION NAME>_TL - Language specific Name, Description

DIM_<DIMENSION NAME>_ATTR - Attribute values

DIM_<DIMENSION NAME>_HIER - Hierarchy parent / child association

The OFSA 4.5 dimension members are migrated to their corresponding DIM_%_B and %_TL tables. Also note that in OFSA 4.5, OFSA_DETAIL_OTHER_COA holds user-defined processing dimensions as well as members of the seeded General Ledger dimension type. Therefore, there could be multiple dimension in the OFSA 4.5 table; their members would be stored in separate OFSAA tables, based on the dimension.

The OFSA 4.5 columns (attributes) of a given dimension are transposed to rows, and are stored in the corresponding DIM_<DIMENSION NAME>_ATTR table.

In OFSA 4.5, the root and nodes in a hierarchy are stored separately from the leaf-member pool. In OFSAA, nodes are stored in the same tables as leaf members. Therefore, during migration to OFSAA, we populate the nodes used in hierarchies as members in the corresponding DIM_<DIMENSION NAME>_B and DIM_<DIMENSION NAME>_TL table, identifying the nodes with LEAF_ONLY_FLAG='N'. The nodes are given a unique sequence number. To ensure a unique display name, if the script encounters a duplicate name, it will add the dimension id as a prefix to the name and description.

OFSAA also stores metadata about dimensions in REV_DIMENSIONS_B and REV_DIMENSIONS_TL tables. The attribute metadata is stored in REV_DIM_ATTRIBUTES_B and REV_DIM_ATTRIBUTES_TL table. For seeded dimensions, these tables are populated during installation.

Note: See Pre-Migration Activities, page 3-1.

User must create the OFSAA dimension tables for user defined dimensions, run the "Add Dimension" utility to seed metadata in REV_DIMENSIONS_B and REV_DIMENSIONS_TL tables, and synch the dimension_id with the OFSA_CATALOG_LEAVES leaf_num_id before executing the dimension migration.

'V' type leaves are not supported in OFSA 4.5 and will not be migrated to OFSAA.

Object Category	Source	Target
Dimension Members and Attributes	OFSA_LEAF_DESC	DIM_<DIMENSION NAME>_B
		DIM_<DIMENSION NAME>_TL
		OFSA_LEAF_DESC
	OFSA_NODE_DESC	DIM_<DIMENSION NAME>_B
		DIM_<DIMENSION NAME>_TL
		OFSA_NODE_DESC
	OFSA_DETAIL_ELEM	DIM_FINANCIAL_ELEMENTS_B
		DIM_FINANCIAL_ELEMENTS_TL
		DIM_FINANCIAL_ELEMENTS_ATTTR
	OFSA_DETAIL_LEAVES	OFSA_DETAIL_ELEM_B
		DIM_COMMON_COA_B
		DIM_COMMON_COA_TL

Object Category	Source	Target
		DIM_COMMON_COA_ATTR
		OFSA_DETAIL_LEAVES
	OFSA_DETAIL_ORG_UNIT	DIM_ORG_UNIT_B
		DIM_ORG_UNIT_TL
		DIM_ORG_UNIT_ATTR
		OFSA_DETAIL_ORG_UNIT
	OFSA_DETAIL_OTHER_COA	DIM_GENERAL_LEDGER_B
		DIM_GENERAL_LEDGER_TL
		DIM_GENERAL_LEDGER_ATTR
		DIM_<DIMENSION NAME>_B
		DIM_<DIMENSION NAME>_TL
		OFSA_DETAIL_OTHER_COA

Common Tables

The Common Tables procedure migrates several tables which are common to all products; examples include database info, preferences, application info, rate management data like interest rates, currencies, etc. Other common tables such as hierarchies, instruments, transactions, etc., are handled through other migration steps.

Object Category	Source	Target
Various Metadata	OFSA_DB_INFO	FSI_DB_INFO
	OFSA_APPLICATIONS	FSI_APPLICATIONS

Object Category	Source	Target
	OFSA_DATA_IDENTITY	FSI_DATA_IDENTITY
	OFSA_FISCAL_YEAR_INFO	FSI_FISCAL_YEAR_INFO
	OFSA_CATALOG_OF_IDS	FSI_M_OBJECT_DEFINITION_B FSI_M_OBJECT_DEFINITION_TL
	OFSA_LOOKUP	FSI_ENGINE_LOOKUP
Rate Management	OFSA_CURRENCIES	FSI_CURRENCIES
	OFSA_EXCHANGE_RATE_HIST	FSI_EXCHANGE_RATE_HIST
	OFSA_EXCHANGE_RATE_HIST_AUDIT	FSI_EXCHANGE_RATE_HIST_AUDIT
	OFSA_EXCHNG_RATE_DIRECT_ACCESS	FSI_EXCHNG_RATE_DIRECT_ACCESS
	OFSA_EXCHNG_RATE_CONV_FORMULA	FSI_EXCHNG_RATE_CONV_FORMULA
	OFSA_FIXED_CURRENCIES_AUDIT	FSI_FIXED_CURRENCIES_AUDIT
	OFSA_FIXED_CURRENCIES	FSI_FIXED_CURRENCIES
	OFSA_IRCS	FSI_IRCS
	OFSA_IRC_RATE_TERMS	FSI_IRC_RATE_TERMS
	OFSA_IRC_RATE_HIST	FSI_IRC_RATE_HIST
	OFSA_IRC_TS_PARAM_HIST	FSI_IRC_TS_PARAM_HIST
Tuning Options	Currently not migrated...	

Object Category	Source	Target
Expressions (Formulas)	OFSA_IDT_FORMULA	REV_EXPRESSION_DETAILS
Preferences	OFSA_IDT_CONFIGURE	REV_APP_USER_PREFERENCES

Hierarchies and Filters

Hierarchies

See also: Dimensions and Attributes, page B-2.

In OFSA 4.5, hierarchies for the various dimensions are stored in a single table, OFSA_IDT_ROLLUP. In OFSAA, for each dimension, there is a separate hierarchy table: DIM_<DIMENSION NAME>_HIER. The OFSA 4.5 rows are moved to their corresponding DIM_<DIMENSION NAME>_HIER table based on the dimension_id.

In OFSAA, a parent child relationship is created in the hierarchical model. As discussed earlier, the root and all nodes in OFSA 4.5 are stored as members in the corresponding OFSAA dimension member table; a flag identifies those members as a node type rather than a leaf.

Migration also creates a default hierarchy for each processing dimension. It uses a two level format, i.e. a root node and all the members of the dimension as its children. The hierarchy rule is named DEFAULT_HIER:<DIMENSION ID>, where the dimension name is based on leaf_field in OFSA_CATALOG_OF_LEAVES. In the event the hierarchy name exceeds 30 characters, the 3 trailing characters "_ID" will be excluded from the rule name, and the prefix will be shortened to "DE. The default hierarchies will be placed in the "ALL" folder. A new unique hierarchy_id will be generated based on Max(sys_id_num) +1 from OFSA_CATALOG_OF_IDS table.

As mentioned in **Chapter 3: Pre-Migration Activities**, before executing the migration script, please ensure that:

- all the tables related to user defined dimensions are in place.
 - DIM_<DIMENSION NAME>_B - Display Code, Leaf / Node Indicator, Enable / Disable
 - DIM_<DIMENSION NAME>_TL - Language specific Name, Description
 - DIM_<DIMENSION NAME>_ATTR - Attribute values
 - DIM_<DIMENSION NAME>_HIER - Hierarchy parent / child association

- Metadata for the user defined dimensions exists in REV_DIMENSIONS_B and REV_DIMENSIONS_TL tables, with dimension_id the same as the OFSA_CATALOG_OF_IDS leaf_num_id.

Filters

The OFSA 4.5 data, group, and tree (hierarchy) filter types are now available under a single rule type (Filters) in OFSAA.

Migration scripts update the hierarchy filter table in OFSAA with the latest node details and levels.

Based on level restrictions from OFSA 4.5, the maximum levels in a hierarchy is 15 including Root, lower level nodes, and the leaf members.

For data element filters, the migration scripts do not check the validity of the table and column names. Further, the OFSAA data element UI performs more validations than the OFSA 4.5 UI. The migration script will migrate OFSA 4.5 data filters "as is" and if there is any invalid data, the OFSAA UI will prevent you from saving cases which violate these rules:

Ranges:

1. The "To" value should be \geq the "From" value
2. The ranges should not overlap.

Method = Formula:

Return type of formula should be the same as source column data type. Criteria should be complete; there may be rare cases in OFSA 4.5 where a column has been selected, but no criteria have been defined, so the filter will produce invalid SQL.

Tree (hierarchy) filters: If any (non-leaf-level) nodes exist in the tree filters which are not in the underlying rollup (hierarchy), the rollup will be noted in the status log and the rollup will not be migrated.

Object Category	Source	Target
Hierarchies	OFSA_IDT_ROLLUP	DIM_<DIMENSION NAME>
		OFSA_IDT_ROLLUP
	OFSA_LEVEL_DESC	REV_HIERARCHY_LEVELS
		OFSA_LEVEL_DESC

Object Category	Source	Target
Filters	OFSA_IDT_DATA_FILTER	REV_DATA_FILTER
	OFSA_IDT_GRP_FILTER	REV_GROUP_FILTER
	OFSA_IDT_VIEW_FILTER	REV_HIERARCHY_FILTER

Common for FTP/ALM

The FTP and ALM rule migration scripts contain logic to ensure that rules shared by the two applications are only migrated once. Examples include prepayments, prepayment tables, payments and repricing patterns, etc.

For example, if you run the FTP rule migration it will migrate your prepayment rules; the ALM rule script will then recognize that these rules have already been migrated and will not attempt to migrate them again.

Target requires a hierarchy selection; migration scripts will create and populate with default hierarchy for the appropriate dimension. Target currency will default to functional currency from OFSA_DB_INFO.

The valuation curve is now a selection in the rate index rule, instead of the OFSA 4.5 design where it was selected in the process rule. Each active currency should have at least one associated interest rate code. The migration scripts assign the minimum IRC associated with each active currency as the valuation curve for that currency.

Application administrators should review the rate index rules and confirm selections for the valuation curve(s) in each rule during the post migration analysis.

Object Category	Source	Target
Prepayment rules	OFSA_IDT_PREPAYMENT	FSI_M_PPMT
	OFSA_PP_ORGDATE_ASSUMP	FSI_M_PPMT_CALC_PARAMS
Prepayment Tables (Prepayment Model Rules)	OFSA_IDT_PREPAY_TBL	FSI_M_PPMT_MODEL
	OFSA_PP_TBL_HYPERCUBE	FSI_PPMT_MODEL_HYPERCUBE
	OFSA_PP_HYPERCUBE_MAP	FSI_PPMT_MODEL_HYPERCUBE_MAP

Object Category	Source	Target
Detailed cash flow audit records	OFSA_PROCESS_CASH_FLOWS	FSI_O_PROCESS_CASH_FLOWS
Payment Patterns	OFSA_PAYMENT_PATTERN	FSI_PAYMENT_PATTERN
	OFSA_PAYMENT_PATTERN_EVENT	FSI_PAYMENT_PATTERN_EVENT
Rate Index Rules	OFSA_IDT_RATE_INDEX	FSI_M_RATE_INDEX
Repricing Patterns	OFSA_REPRICE_PATTERN	FSI_REPRICE_PATTERN
	OFSA_REPRICE_PATTERN_EVENT	FSI_REPRICE_PATTERN_EVENT
		FSI_REPRICE_PATTERN_EVENT_DTL
Payment Schedule	OFSA_PAYMENT_SCHEDULE	FSI_D_PAYMENT_SCHEDULE

Transfer Pricing/ FTP

Transfer pricing rules requires a hierarchy selection; migration will create and populate with default hierarchy for the appropriate dimension. Target currency will default to functional currency from OFSA_DB_INFO. There are a few new features for which the migration scripts populate a default value equivalent to OFSA 4.5 functionality.

Transfer pricing process rules migrate a few new features with default values equivalent to OFSA 4.5 functionality.

For rules using:

- Weighted Term method - Cashflow Type = Discounted Cash Flow.
- Duration method - Output Duration to Instrument switch = off.
- Single Rate Discounting = Coupon Rate.

Transfer pricing process rules migrate a few new features with default values equivalent to OFSA 4.5 functionality.

For processes with:

- Ledger migration
- Charge/Credit Method: Use Ledger Level
 - Migration Dimension: Use the transfer pricing product. This can be determined by looking at OFSA_CATALOG_OF_IDS, which stores the leaf identifier for the process rule.
 - Accrual Basis: Use product accrual basis. Depends on selection in OFSA 4.5 configuration, this may not emulate your OFSA 4.5 results.

For more information regarding handling of OFSA 4.5 process rules which have both transfer pricing and option cost selections, and processes which have propagation selections, and option-cost-only processes (handling of valuation curve), see remarks in Post Migration Activities, page 5-1.

- If Detail Cash Flow flag is on, number of records is set to 5, and record selection code "Set to First" (i.e. first 5).
- Deterministic processes: Interpolation method = Linear.

Object Category	Source	Target
Transfer Pricing Rules	OFSA_IDT_TRANSFER_PRICE	FSI_M_TRANSFER_PRICE_RULE
	OFSA_TP_REDEMPTION_CURVE_DTL	FSI_M_TP_REDEMPT_CURVE_DTL
	OFSA_TP_UNPRICED_ACCT_DTL	FSI_M_TP_UNPRICED_ACCT_DTL
TP Process Rules	OFSA_IDT_TP_PROCESS	FSI_M_TP_PROCESS_RULE
		FSI_TP_SELCTD_COLUMNS
	OFSA_TP_STOCH_ASSUMPTION	FSI_TP_STOCH_ASSUMPTION
	OFSA_TP_PROC_TABLES	FSI_TP_PROC_TABLES

Risk Manager/ ALM

In ALM, many of the tables' data have been transposed from rows to columns; e.g. Auto

Balancing, Leaf (Product) Characteristics, and Transaction Strategies.

Some target requires a hierarchy selection; migration scripts will create and populate with default hierarchy for the appropriate dimension. Target currency will default to functional currency from OFSA_DB_INFO.

For rules with rate-volume relationships, rate dependency pattern rules are generated based on unique combination of forecast balance sys_id_num, currency, and product. The new rate dependency is saved as an embedded rule selection within the migrated forecast balance rule.

An OFSA 4.5 process rule can be either deterministic (scenario based) or stochastic. In OFSAA, these are separate rule types, further differentiated by whether they are static or dynamic.

Depending on selections in an OFSA 4.5 rule, any of the following would qualify a rule as dynamic:

1. It has forecast balance, maturity strategy (a.k.a. maturity mix), and pricing margins. (Note: if it has one, it should have all 3, otherwise it's bad data – an unlikely case)
2. It has a transaction strategy
3. It has formula leaves (formula results). This option is only available in scenario based (deterministic) processes.

The transfer pricing flag will be set off for all deterministic processes. For more information regarding any OFSA 4.5 processes which have the transfer pricing flag set on, see Post-Migration Activity, page 5-1.

Object Category	Source	Target
Configuration, Modeling Buckets, Gap Buckets	OFSA_IDT_CONFIGURE	FSI_TIME_BUCKET_MASTER
	OFSA_CNFG_DATE_BUCKET	FSI_INCOME_SIMULATION_BUCKETS
	OFSA_IDT_DGAP	FSI_M_LR_IRR_BUCKETS
	OFSA_DGAP_AUX	FSI_LR_IRR_BUCKETS_AUX
Discount Rates (Discount Methods)	OFSA_IDT_DISCOUNT_RATE	FSI_M_DISCOUNT_RATE
Forecast Balances	OFSA_IDT_FORECAST_BAL	FSI_M_FORECAST_BAL

Object Category	Source	Target
		FSI_RATE_DEPENDENCY_RELATION
	OFSA_FBAL_DIMENSIONS	FSI_FBAL_DIMENSIONS
	OFSA_FBAL_ROLL_INTO	FSI_FBAL_ROLL_INTO
	OFSA_FBAL_VOLUMES	FSI_FBAL_VOLUMES
		FSI_RATE_TIERS
Forecast Rates	OFSA_IDT_FORECAST_RATES	FSI_M_FORECAST_RATES
	OFSA_FCAST_XRATES	FSI_FCAST_XRATES
	OFSA_FCAST_IRCS	FSI_FCAST_IRCS
	OFSA_FCAST_XRATE_STRCT_CHG	FSI_FCAST_XRATE_STRCT_CHG
	OFSA_FCAST_XRATE_DIRECT_INPUT	FSI_FCAST_XRATE_DIRECT_INPUT
	OFSA_FCAST_IRC_CHG_BASE_VAL	FSI_FCAST_IRC_CHG_BASE_VAL
	OFSA_FCAST_IRC_DIRECT_INPUT	FSI_FCAST_IRC_DIRECT_INPUT
	OFSA_FCAST_IRC_STRCT_CHG_BKT	FSI_FCAST_IRC_STRCT_CHG_BKT
	OFSA_FCAST_IRC_STRCT_CHG_VAL	FSI_FCAST_IRC_STRCT_CHG_VAL
	OFSA_FCAST_RATES_BUCKETS	FSI_FCAST_RATES_BUCKETS
	OFSA_FCAST_RATES_SCENARIOS	FSI_FCAST_RATES_SCENARIOS

Object Category	Source	Target
Formula Leaves (Formula Results)	OFSA_IDT_TM_FORMULA	FSI_M_FORMULA_RESULTS
	OFSA_TM_FORMULA_AUX	FSI_M_FORMULA_AUX
	FSA_TM_FORMULA_COEF	FSI_M_FORMULA_COEF
Characteristics (Product Characteristics)	OFSA_IDT_LEAF_CHARACTERISTICS	FSI_M_PROD_CHARACTERISTICS
Maturity Strategies(Maturity Mix)	OFSA_IDT_MATURITY	FSI_M_MATURITY
	OFSA_MATURITY_AUXILIARY	FSI_MATURITY_AUXILIARY
Pricing Margins	OFSA_IDT_PRICING_MARGIN	FSI_M_PRICING_MARGIN
Process Rules	OFSA_IDT_TM_PROCESS	FSI_M_ALM_PROCESS
	OFSA_TM_SCENARIO_ASSUM P	FSI_ALM_DETERMINISTIC_PROCE SS
	OFSA_TM_NEW_BUSINESS	FSI_ALM_NEW_BUSINESS
	OFSA_TM_PROC_INSTRMNTS	FSI_ALM_PROC_TABLES
		FSI_ALM_PROCESS_PRODUCT_LIS T
	OFSA_TM_FIN_ELEM_SET	FSI_TM_FIN_ELEM_SET
	OFSA_TM_STOCH_ASSUMP	FSI_ALM_STOCHASTIC_PROCESS
	OFSA_ID_TYPE_FLAGS	FSI_ID_TYPE_FLAGS
Result tables including audit tables	OFSA_IDT_RESULT_HEADER	FSI_O_RESULT_HEADER

Object Category	Source	Target
	OFSA_RESULT_BUCKET	FSI_O_RESULT_BUCKET
	OFSA_RESULT_SCENARIO	FSI_O_RESULT_SCENARIO
	OFSA_RESULT_MASTER	FSI_O_RESULT_MASTER
	OFSA_CONSOLIDATED_MASTER	FSI_O_CONSOLIDATED_MASTER
	OFSA_TM_STOCH_VAR	FSI_O_STOCH_VAR
	OFSA_TM_STOCH_TOT_VAR	FSI_O_STOCH_TOT_VAR
	OFSA_TM_STOCH_MKT_VAL	FSI_O_STOCH_MKT_VAL
	OFSA_EAR_LEAF_AVG	FSI_O_EAR_LEAF_AVG
	OFSA_EAR_LEAF_DTL	FSI_O_EAR_LEAF_DTL
	OFSA_EAR_TOTAL_AVG	FSI_O_EAR_TOTAL_AVG
	OFSA_EAR_TOTAL_DTL	FSI_O_EAR_TOTAL_DTL
	OFSA_INTEREST_RATES_AUDIT	FSI_INTEREST_RATES_AUDIT
	OFSA_EXCHANGE_RATES_AUDIT	FSI_EXCHANGE_RATES_AUDIT
Transaction Strategies	OFSA_IDT_TRANS_STRATEGIES	FSI_M_TRANS_STRATEGIES FSI_M_TRANS_MASTER
	OFSA_FIN_ELEM_SET_DTL	FSI_FIN_ELEM_SET_DTL

Profitability/OFSPM

The migration of allocation rules from OFSA 4.5 to OFSAA has seen a complete structural change, in terms of data storage logic. Primarily the concept of a multiple page allocation rule is eliminated in this migration. Migration will translate multi-page

rules from OFSA 4.5 into multiple individual rules in OFSAA. It will name each page based on the OFSA 4.5 allocation rule name with a sequential number (i.e. Fixed_Cost_1, Fixed_Cost_2). It will not be possible to split out the data that has been previously generated by multi-page rules into multiple identity codes. The association of sys_id_num to identity_code will only be truly correct going forward for the allocation's first page. Rules with parentheses will not be migrated. Rules with more than one source, operator, and driver will not be migrated.

Object Category	Source	Target
Allocation rules	OFSA_IDT_ALLOC	FSI_M_ALLOCATION_RULE
	OFSA_ALLOC_LEAVES	FSI_M_ALLOC_DETAILS
		FSI_M_ALLOC_LEAF_SELECTION
Lookup Driver Table rules	OFSA_IDT_LOOKUP_TABLE	FSI_M_LOOKUP_TABLE
	OFSA_LOOKUP_TABLE_KEY_MAPPING	FSI_M_LOOKUP_TABLE_KEY_MAPPING
Static Driver Table rules	OFSA_TABLE_DESC	FSI_M_ALLOC_TABLE_ID
	OFSA_IDT_TABLE	FSI_M_ALLOC_TABLE_DETAILS
	OFSA_MAP_TABLES	FSI_MAP_TABLES

Ledger, Instrument Tables, Transaction Tables, Lookup Tables

You can individually migrate Ledger, as well as seeded or user-defined instrument, transaction, and user-defined lookup tables, as mentioned earlier with the script "migrate_user_Acc_tables".

When you migrate the table, the script allows you to specify the name of the existing source and target table. The target table must contain a minimum of all the source columns using the same column names as in source, and the new product_id dimension where applicable to the table type. As mentioned earlier, the product_id column will be seeded with -1, unless the OFSA 4.5 source table already has a product_id column (in which case the values will be directly migrated). Depending on table classification, the target table may have additional new columns.

The following default values are used:

PRODUCT_ID: If product_id is a new dimension in the target instance (not present in

OFSA 4.5 source) then it will receive a default value of -1 in OFSAA.

BALANCE_TYPE_CD: LEDGER_STAT balance_type_cd will receive a default value of 500:"Legacy Migration".

Object Category	Source	Target
Ledger	LEDGER_STAT	LEDGER_STAT
Instrument	<INSTRUMENT NAME>	FSI_D_<INSTRUMENT NAME>
Transaction	<INSTRUMENT NAME>_TRNS	FSI_D_<INSTRUMENT NAME>_TXNS
Lookup	<LOOKUP NAME>	<LOOKUP NAME>

Objects Not Migrated

Objects Comments

4.5 Profitability Functional Area	Comments
OFSPM: Allocations (some cases)	<ol style="list-style-type: none">1. Rules with more than one Source, Operator, and Driver will not be migrated.2. Migration scripts will translate multi-page rules from OFSA 4.5 into multiple individual rules in OFSAA. It will name them based on the OFSA 4.5 allocation name, numbered sequentially. It will not be possible to split out the data that has been previously generated by multi-page rules into multiple identity codes. The association of sys_id_num to identity_code will only be truly correct going forward for the allocation's first page.3. Filters: OFSAA will restrict to appropriate columns; if any non-compliant definitions encountered in OFSA 4.54. OFSAA does not support rules with multiple macros and/or multiple constants

4.5 Balance and Control Functional Area	Comments
Correction Rules	Correction rules are not migrated.
Correction Processing	You can create Cash Flow Edit rules in ALM, and Cash Flow Edit rules are scheduled for the next release of FTP.
4.5 Risk Manager Functional Area	Comments
Leaf Characteristics, Transaction Strategies (TP fields)	TP IRC and TP Margin are no longer defined in ALM but user can select a TP Rule with the ALM process.
4.5 Transfer Pricing Online Functional Area	Comments
Transfer Pricing Online (5.1: Pricing Management)	Pricing Management selections should be freshly implemented in 5.1.
4.5 Common Code Functional Area	Comments
Configuration (subset of data)	<ol style="list-style-type: none"> <li data-bbox="873 1190 1365 1451">1. In the new applications, Global and Application Preferences replace some of the options from the 4.5 Configuration rules. Preferences are not rule-based; so there is no direct mapping from 4.5. For more information regarding setup of Preferences, see Post Migration Activities, page 5-1. <li data-bbox="873 1486 1365 1680">2. For ALM, the time and gap bucket definitions from the 4.5 Configuration ID's will be migrated to ALM Time Bucket Rules, named based on the source Configuration ID. Only 1 may be active at a given time.
Error reporting	OFSA_PROCESS_ERRORS is not migrated.

4.5 Common Code Functional Area	Comments
Leaves, Nodes (some special cases)	<ol style="list-style-type: none"> 1. 'V' type leaves are not supported in OFSA 4.5 and will not be migrated to OFSAA. 2. Member names must be unique; script will concatenate text to non-unique member names to ensure the name is unique. This logic applies to leaf-level members, and node-level members from 4.5 hierarchies.
Transformation IDs	Not migrated.
Transformed Hierarchies, Ledger, and ALM results.	Not migrated.
Tree Filters (special cases)	If any (non-leaf-level) nodes exist in a tree filter which are not in the underlying rollup hierarchy, the rollup will be noted in the status log and the rollup will not be migrated
User-Defined Codes	The current migration does not migrate OFSA 4.5 user-defined codes. They can be stored as simple dimensions in OFSAA.
User-Defined Views	See Post Migration Activities, page 5-1.
User passwords	Passwords will not be migrated; users will be prompted for a new password during initial login.
Other Functional Areas	Comments
4.5 Seeded Rules	Rules in the reserved range (designated in OFSA 4.5 OFSA_CATALOG_OF_IDS where the sys_id_num is between 79999 and 100000) will be excluded from migration.

4.5 Tuning Option	Comments
Process Tuning	Tuning options can be manually entered via the UI.

D

Issues

Release Notes

For more information on latest issues and resolutions, see *Oracle Financial Services 4.5 Migration Tool v7.1 Release Notes*.