

Oracle Financial Services
Data Foundation
User Guide 8.0.6.0.0

August 2018

ORACLE®
Financial Services

Oracle Financial Services Data Foundation User Guide, Release 8.0.6.0.0

Copyright © 2018 Oracle and/or its affiliates. All rights reserved.

Primary Author: Shwetha Bharmoji Rao

Contributor: Raghu Kodanda

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

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 is software or related documentation that 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 END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government. This software or hardware 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 that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications. This software or hardware 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.

Table of Contents

	Preface.....	<i>i</i>
	<i>Intended Audience.....</i>	<i>i</i>
	<i>Documentation Accessibility.....</i>	<i>i</i>
	<i>Access to Oracle Support.....</i>	<i>i</i>
	<i>Related Information Sources.....</i>	<i>i</i>
CHAPTER 1	Introduction to OFSDF	1
	<i>Overview</i>	<i>1</i>
	<i>Components of OFSDF.....</i>	<i>1</i>
	<i>Relationship to Oracle Financial Services Analytical Applications.....</i>	<i>2</i>
	<i>OFSDF Prerequisite Components and Tools.....</i>	<i>3</i>
CHAPTER 2	Understanding OFSDF	5
	<i>Background.....</i>	<i>5</i>
	<i>OFSDF Architecture</i>	<i>5</i>
	<i>Differences from Traditional Warehouse Architecture.....</i>	<i>7</i>
	<i>Staging Data Model</i>	<i>8</i>
	<i>Overview/Design.....</i>	<i>8</i>
	<i>Details of the Staging Data Model</i>	<i>10</i>
	<i>Business Data:</i>	<i>11</i>
	<i>Product Processors</i>	<i>12</i>
	<i>Reference/Master Data.....</i>	<i>14</i>
	<i>Reporting Data Model</i>	<i>14</i>
	<i>Reporting Area Details</i>	<i>15</i>
	<i>OFSDF Logical Data Model</i>	<i>18</i>
	<i>Relationship to the OFSDF Physical Model.....</i>	<i>18</i>
	<i>OFSDF LDM Content Details</i>	<i>20</i>
	<i>Mapping from Logical Data Model to OFSDF Staging Area.....</i>	<i>22</i>
CHAPTER 3	Dimension Loading Process	23
	<i>Dimension Table Population.....</i>	<i>23</i>
	<i>Overview of Seeded Dimensions.....</i>	<i>23</i>
	<i>Overview of SCD Process</i>	<i>23</i>
	<i>Prerequisites</i>	<i>25</i>
	<i>Tables Used by the SCD Component</i>	<i>25</i>

<i>Guidelines for Configuring User Defined (Custom) SCDs</i>	28
<i>Executing the SCD Component</i>	28
<i>Checking the Execution Status</i>	29
<i>LOAD DIM TABLES THROUGH SCD</i>	30
<i>LOAD KEY DIMENSIONS USING DRMLoader AND HIERARCHY FLATTENING</i>	30
<i>Loading Data from STG_INTF Tables to DIM_INTF Tables</i>	30
<i>Executing The Loading Procedure using Batch Framework</i>	31
<i>Executing The Hierarchy Flattening Procedure using Batch Framework</i>	31
<i>Executing The SCD for Loading Key Dimensions using Batch Framework</i>	31
<i>Improving SCD Performance</i>	32

CHAPTER 4 *Reclassification of Standard Dimensions*..... 35

<i>Overview of Reclassification of Standard Dimensions</i>	35
<i>Overview of Reclassification of Standard Dimensions Population</i>	35
<i>Dimension Data Expectations through SCD</i>	36
<i>Overview of Mapper for Reclassification of Standard Dimensions</i>	36
<i>Maintenance of Mapper for Reclassification of Standard Dimensions</i>	37
<i>Prerequisites for Mapper Maintenance</i>	39
<i>Possible Mapping Combinations</i>	40
<i>Loading Mapper Maintenance through Backend</i>	43
<i>Usage of Mapper Tables in Data Flow and Reports</i>	44

CHAPTER 5 *Legal Entity Consolidation* 45

<i>Introduction</i>	45
<i>Consolidation Procedures</i>	45
<i>Types of Consolidation</i>	45
<i>Consolidation Activities</i>	46
<i>Data Flow</i>	46

CHAPTER 6 *Executing Run through Run Management* 47

<i>Introduction</i>	47
<i>Summary and Details Page</i>	47
<i>Navigation within the Summary Page</i>	47
<i>Search Section</i>	47
<i>List of Runs Section</i>	48
<i>List of Runs Summary Grid</i>	48
<i>Navigation within Run Default Parameters Window</i>	49
<i>Run Details Section</i>	49
<i>Run Execution Parameters Section</i>	49

	<i>Navigation within Run Execution Parameters Window</i>	51
	Run Details Section	51
	Run Execution Parameters Section	51
	<i>Navigation within Run Execution Summary Page</i>	52
	Run Execution Summary Section	53
	Run Execution Details Section	53
	Run Execution Grid	54
CHAPTER 7	<i>Run Execution from Command Line</i>	55
CHAPTER 8	<i>Account Dimension Population</i>	59
	<i>Overview of Account Dimension</i>	59
	<i>Overview of Account Dimension Population</i>	59
	<i>Executing the Account Dimension SCD</i>	61
	<i>Handle Multiple GAAP Codes for Same Account Number for the Same MIS Date in SCD</i> .	61
	<i>Handle Multiple Load Runs for Same Account Number for the Same MIS Date in SCD</i>	62
CHAPTER 9	<i>Customer Dimension Population</i>	63
	<i>Customer Dimension Loading Overview</i>	63
	<i>Customer Dimension Population using Stage Party Master Entity</i>	63
	<i>Prerequisites</i>	63
	<i>Customer Dimension Population using Stage Customer Master Entity</i>	63
	<i>Prerequisites</i>	63
	<i>Execution of Customer Dimension Population Batch</i>	63
	<i>Error Messages</i>	64
	<i>Error Messages</i>	64
CHAPTER 10	<i>Loading Multiple Load Runs in OFSAA</i>	65
	<i>Overview</i>	65
	<i>Objective</i>	65
	<i>Design Details</i>	66
	<i>Implementation</i>	66
	<i>Loading OFSAA Staging Tables</i>	68
	<i>Complete Snapshot Load Example</i>	68
	<i>Incremental Snapshot Load Example</i>	71
	<i>Post Stage Load Process</i>	72
	<i>Register Load Run Details</i>	72
	<i>Updating Load as Completed</i>	73
	<i>Loading data into OFSAA Results Tables from Staging Tables</i>	73
	<i>Complete Snapshot Load Scenario</i>	74

	<i>Incremental Load Scenario</i>	76
CHAPTER 11	Exchange Rates Population	77
	<i>Overview of Exchange Rates Table</i>	77
	<i>Overview of Exchange Rates Population</i>	77
	<i>Handling Alternate Currency</i>	77
	<i>Multiple Execution of T2T in Run Chart</i>	78
	<i>Execution of Currency Exchange Rates Population T2T</i>	78
	<i>Execution through Batch</i>	78
	<i>Execution through Run Management</i>	78
	<i>Currency Execution Rates - Batch Execution</i>	78
CHAPTER 12	Account Summary Population	81
	<i>Overview of Account Summary Tables</i>	81
	<i>Overview of Account Summary Population</i>	81
	<i>Executing the Account Summary Population T2T</i>	83
	<i>Executing through Batch</i>	83
	<i>Execution through Run Management</i>	83
	<i>Fact Common Account Summary - Batch Execution</i>	83
	<i>Fact Common Account Summary - Run Execution</i>	84
	<i>Checking the Execution Status</i>	85
	<i>Account Summary T2Ts</i>	85
CHAPTER 13	Customer Summary Population	87
	<i>Overview of Common Customer Summary Tables</i>	87
	<i>Prerequisites</i>	87
	<i>Executing the Customer Summary Population T2T</i>	88
	<i>Execution through Batch</i>	88
	<i>Execution through Run Management</i>	88
	<i>Error Messages</i>	88
CHAPTER 14	Fact Transaction Summary	89
	<i>Table to Table</i>	89
	<i>Executing the Fact Transaction Summary</i>	90
	<i>Execution through Batch</i>	90
	<i>Execution through Run Management</i>	91
CHAPTER 15	Loan Account Summary Population	93

<i>Overview of Loan Account Summary Tables</i>	93
<i>Overview of Loan Account Summary Population</i>	93
<i>Executing the Loan Account Summary Population T2T</i>	93
<i>Execution through Batch</i>	93
<i>Execution through Run Management</i>	94
<i>Error Messages</i>	94
<i>Checking the Execution Status</i>	94
<i>Loan Account Summary T2Ts</i>	94

CHAPTER 16 *Deposit Borrowings Summary Population* 95

<i>Overview of Deposit Borrowings Summary Tables</i>	95
<i>Overview of Deposit Borrowings Summary Population</i>	95
<i>Executing the Deposit Borrowings Summary Population T2T</i>	95
<i>Execution through Batch</i>	95
<i>Execution through Run Management</i>	96
<i>Error Messages</i>	96
<i>Checking the Execution Status</i>	96
<i>Deposit Borrowings Summary T2Ts</i>	96

CHAPTER 17 *Cards Summary Population* 97

<i>Overview of Cards Summary Tables</i>	97
<i>Overview of Cards Summary Population</i>	97
<i>Executing the Cards Summary Population T2T</i>	97
<i>Executing through Batch</i>	97
<i>Execution through Run Management</i>	98
<i>Error Messages</i>	98
<i>Checking the Execution Status</i>	98
<i>Cards Summary T2Ts</i>	98

CHAPTER 18 *Mitigants Results Population* 99

<i>Overview of Mitigants Results Tables</i>	99
<i>Overview of Mitigants Results Population</i>	99
<i>Executing the Populating Mitigants Results T2T</i>	99
<i>Executing through Batch</i>	99
<i>Execution through Run Management</i>	100
<i>Error Messages</i>	100
<i>Checking the Execution Status</i>	100
<i>Mitigants Results T2Ts</i>	100

CHAPTER 19	Fiduciary Services Investment Summary Population	101
	<i>Overview of Fiduciary Services Investment Summary Table</i>	101
	<i>Overview of Fiduciary Services Investment Summary Population</i>	101
	<i>Executing the Fiduciary Services Investment Summary Population T2T</i>	102
	<i>Executing through Batch Framework</i>	102
	<i>Execution through Run Management</i>	102
	<i>Error Messages</i>	102
	<i>Checking the Execution Status</i>	102
	<i>Fiduciary Services Investment Summary T2Ts</i>	103
CHAPTER 20	Populating Party Attributes	105
	<i>Overview of Party Attributes Results Tables</i>	105
	<i>Overview of Party Attributes Results Population</i>	105
	<i>Executing the Populating Party Attributes T2T</i>	106
	<i>Executing through Batch Framework</i>	106
	<i>Execution through Run Management</i>	106
	<i>Error Messages</i>	106
	<i>Checking the Execution Status</i>	106
	<i>Party Attributes Results T2Ts</i>	107
CHAPTER 21	Placed Collateral Population	109
	<i>Overview of Placed Collateral Population Tables</i>	109
	<i>Overview of Placed Collateral Population</i>	109
	<i>Executing the Placed Collateral Population T2Ts</i>	109
	<i>Executing through Batch Framework</i>	109
	<i>Execution through Run Management</i>	110
	<i>Error Messages</i>	110
	<i>Checking the Execution Status</i>	110
	<i>Placed Collateral Population Results T2Ts</i>	110
CHAPTER 22	Credit Score Model and Probability of Default Model	111
	<i>Overview of Credit Score Model and Probability of Default Model</i>	111
	<i>Overview of Credit Score Model and Probability of Default Model Population</i> ...	112
	<i>Staging Data Expectation for Credit Score Model</i>	112
	<i>Overview of Mapper for Credit Score Model to Regulatory Credit Score Model</i> .	112
	<i>Maintenance of Mapper for Credit Score Model to Regulatory Credit Score</i>	

<i>Model</i>	112
<i>Prerequisites for Mapper Maintenance</i>	113
<i>Possible Mapping Combinations</i>	114
<i>Loading Mapper Maintenance through Backend</i>	117
<i>Executing the Credit Score Model and Probability of Default Model T2Ts</i>	117
<i>Execution through Run Management</i>	117
<i>Error Messages</i>	118
<i>Checking the Execution Status</i>	118
<i>Credit Score Model and Probability of Default Model Population T2Ts</i>	118

CHAPTER 23 *GL to Management Reporting* 119

<i>Overview of GL to Management Reporting</i>	119
<i>Overview of GL to Management Reporting Population</i>	119
<i>Overview of Mapper for GL to Management Reporting</i>	119
<i>Maintenance of Mapper for GL to Management Reporting</i>	119
<i>Prerequisites for Mapper Maintenance</i>	121
<i>Possible Mapping Combinations</i>	121
<i>Loading Mapper Maintenance through Backend</i>	130
<i>Executing the GL to Management Reporting T2Ts</i>	130
<i>Execution through Run Management</i>	130
<i>Error Messages</i>	130
<i>Checking the Execution Status</i>	130
<i>GL to Management Reporting T2Ts</i>	131

CHAPTER 24 *Other Miscellaneous Population* 133

<i>Overview of Other Results Tables</i>	133
<i>Overview of Other Results Population</i>	134
<i>Executing the Various Other T2T Batches</i>	136
<i>Executing T2Ts through Batch Framework</i>	136
<i>Executing SCDs through Seeded Batches</i>	136
<i>Execution through Run Management</i>	136
<i>Error Messages</i>	137
<i>Checking the Execution Status</i>	137
<i>Various Other Results T2Ts</i>	138

CHAPTER 25 *Basel Processing to FSDF Results Integration* 139

<i>Overview of Basel Processing to FSDF Results Integration Tables</i>	139
<i>Overview of Basel Processing to FSDF Results Integration</i>	140

	<i>Executing the BASEL Processing to FSDF Results Integration T2Ts</i>	141
	<i>Checking the Execution Status</i>	142
	<i>BASEL Processing to FSDF Results Integration Results T2Ts</i>	143
CHAPTER 26	<i>LLFP Processing to FSDF Results Integration</i>	145
	<i>Overview of LLFP Processing to FSDF Results Integration Tables</i>	145
	<i>Overview of LLFP Processing to FSDF Results Integration</i>	145
	<i>Executing the LLFP Processing to FSDF Results Integration T2Ts</i>	145
	<i>Checking the Execution Status</i>	147
	<i>LLFP Processing to FSDF Results Integration Results T2Ts</i>	147
CHAPTER 27	<i>LRM Processing to FSDF Results Integration</i>	149
	<i>Overview of LRM Processing to FSDF Results Integration Tables</i>	149
	<i>Overview of LRM Processing to FSDF Results Integration</i>	149
	<i>Executing the LRM Processing to FSDF Results Integration T2Ts</i>	149
	<i>Checking the Execution Status</i>	151
	<i>LRM Processing to FSDF Results Integration Results T2Ts</i>	151
CHAPTER 28	<i>Metadata Browser</i>	153
	<i>Overview</i>	153
	<i>Object View</i>	153
	<i>Metadata Publish</i>	154
	<i>Metadata Object to Application Map</i>	154
CHAPTER 29	<i>Time Dimension Population</i>	155
	<i>Overview of Time Dimension Population</i>	155
	<i>Prerequisites</i>	155
	<i>Tables Used by the Time Dimension Population Transformation</i>	156
	<i>Executing the Time Dimension Population Transformation</i>	156
	<i>Checking the Execution Status</i>	157
CHAPTER 30	<i>Recommendation for Backdated Run</i>	159
	<i>Overview of Backdated Run Execution</i>	159
	<i>Required Changes</i>	159
	<i>Recommendations</i>	159
	<i>Backdated Run Execution using Latest Record Indicator (LRI) batch</i>	160

CHAPTER 31	Using OFSDF	165
	<i>Delivery Mechanism</i>	165
	<i>Installing OFSDF</i>	165
	<i>OFSDF Supporting Documentation</i>	165
	<i>Data Dictionary</i>	166
	<i>Download Specifications</i>	166
	<i>Extending OFSDF Physical Data Model</i>	166
	<i>Customization Process Guidelines</i>	167
	<i>Staging Area Design Guidelines</i>	167
	<i>Results Area Design Guidelines</i>	168
	<i>Upgrading Data Model</i>	169
CHAPTER 32	Data Quality Rules Execution	171
	<i>Data Quality Framework</i>	171
	<i>Data Quality Summary</i>	171
	<i>Create Data Quality Rule</i>	172
	<i>View Data Quality Rule</i>	175
	<i>Modify Data Quality Rule</i>	175
	<i>Copy Data Quality Rule</i>	176
	<i>Approve/Reject Data Quality Rule</i>	176
	<i>Delete Data Quality Rule</i>	177
	<i>Data Quality Groups Summary</i>	177
	<i>Create Data Quality Group</i>	177
	<i>Execute Data Quality Group</i>	179
	<i>View Data Quality Group</i>	179
	<i>Modify Data Quality Group</i>	179
	<i>Copy Data Quality Group</i>	180
	<i>View Data Quality Group Summary Log</i>	180
	<i>Delete Data Quality Group</i>	180
	<i>Data Quality Rules For Staging Tables</i>	181
	<i>Data Quality Groups for Staging Tables</i>	181
	<i>DQ Group Execution</i>	181
	<i>Batch Execution of DQ Rules</i>	185
CHAPTER 33	Compare Data Model Reports	187
	<i>Creating Data Model Report from ERwin</i>	187
	<i>Extracting Data Model Report from ERwin</i>	192
	<i>Comparing Data Model Reports</i>	194
APPENDIX A	Naming Conventions Used in OFSDF Data Model 197	

	<i>Naming Conventions Used in OFSDF LDM and PDM</i>	197
	<i>OFSDF Logical Data Model Naming Conventions</i>	197
	<i>LDM Logical View Mode Entity Naming Conventions</i>	197
	<i>LDM Physical View Mode Table Naming Conventions</i>	198
	<i>PDM Logical View Mode Naming Conventions</i>	199
	<i>PDM Physical View Mode Naming Conventions</i>	200
	<i>Domains (PDM and LDM)</i>	202
APPENDIX B	Standard Data Expectations	205
APPENDIX C	How to Define a Batch	207
	<i>Batch Definition</i>	207
APPENDIX D	Template to Generate Data Dictionary and Download Specification for ERwin 9.x	209
	<i>Prerequisite</i>	209

Preface

This chapter discusses the following topics:

- Intended Audience
- Documentation Accessibility
- Access to Oracle Support
- Related Information Sources

Intended Audience

Welcome to Release 8.0.6.0.0 of the Oracle Financial Services Data Foundation User Guide.

This document is the user guide and reference guide for the Oracle Financial Services Data Foundation (OFSDF) release 8.0.6.0.0, and is intended for System Administrator and all users who are instrumental in configuring and administering OFSDF with Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) platform.

See Related Information Sources for more Oracle product information.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Information Sources

Refer to the following documentation on [OHC Documentation Library](#):

- Technical Whitepaper on Data Model Document Generation ([OHC](#))
- Oracle Financial Services Data Foundation Application Pack Installation Guide 8.0.6.0.0 ([OHC](#))
- OFSAAI Installation Guide Release 8.0.6.0.0 ([OHC](#))

This chapter introduces you to Oracle Financial Services Data Foundation (OFSDF), its components, its relationship with Oracle Financial Services Analytical Applications Infrastructure (OFSAAI), and the key prerequisites for running OFSDF.

This chapter includes the following topics:

- Overview
- Components of OFSDF
- Relationship to Oracle Financial Services Analytical Applications
- OFSDF Prerequisite Components and Tools

Overview

The Oracle Financial Services Data Foundation (OFSDF) is an analytical data warehouse platform for the Financial Services industry. OFSDF combines an industry data model for Financial Services along with a set of management and infrastructure tools that allows Financial Services Institutions to develop, deploy, and operate analytical solutions spanning key functional areas in Financial Services, including:

1. Enterprise Risk Management
2. Enterprise Performance Management
3. Customer Insight
4. Financial Crime and Compliance Management

OFSDF is a comprehensive data management platform that helps institutions to manage the analytical data life cycle from sourcing to reporting and business intelligence/BI using a unified, consistent platform and toolset..

Components of OFSDF

OFSDF consists of the following components, which are described in additional detail in the following sections.:

Table 1. Components

Criteria	Description
Financial Services Logical Data Model (FS LDM)	<p>A reference model of the Financial Services domain spanning Banking and Capital Markets. This model captures the business processes of a typical Financial Services institution in detail. These primarily include core banking business activities such as Retail Banking, Corporate/Merchant Banking, Wealth Management, Trading, and Securities Services.</p> <p>This is a logical data model, which means that it cannot be readily used. The structures modeled in the OFSDF LDM provide an abstract, graphical model of the Financial Services domain using Entity-Relationship modeling. It can be thought of as a detailed blueprint for organizing data within a Financial Services institution. It provides a reference guide for institutions to understand the salient data related to a specific business process.</p> <p>To actually use the OFSDF in practice, customers deploy the OFSDF Analytical Warehouse Model, which is derived from the LDM (blueprint).</p>
Financial Services Analytical Warehouse Data Model	<p>A physical data model that supports data sourcing and reporting related to key analytical use cases in the Financial Services industry.</p> <p>The Warehouse Model is a physical data model, unlike the OFSDF LDM, meaning that it is readily deployable, and consists of database object definitions, and additional supporting scripts.</p> <p>It is organized into two distinct sets of tables based on purpose: The Staging model, which facilitates data sourcing from the bank's internal operational systems such as Lending systems, Trading Systems, Collateral Management systems, and Master Data Management systems.</p> <p>The Reporting model which facilitates the storage of outputs from analytical tools, applications, and engines in a manner that is conducive to BI reporting.</p> <p>The warehouse model is typically deployed into production via a set of management tools called the Oracle Financial Services Analytical Application Infrastructure (OFSAAI). The AAI is a separate product, and is a prerequisite for OFSDF (see Infrastructure).</p>
Supporting scripts	<p>As part of the OFSDF package, there are additionally a number of scripts provided for basic operations such as internal data movement between the staging and reporting areas.</p>

Relationship to Oracle Financial Services Analytical Applications

The OFSDF is very closely related to the Oracle Financial Services Analytical Applications (OFSAA) in the following ways:

1. Data Model

- **OFSDF Staging Model** provides the complete data sourcing foundation for OFSAA applications. All application-specific input data requirements are captured as part of the staging data model. 'OFSDF Staging Model' is a combination of all staging models supplied with each OFSAA application.
- **OFSDF Reporting model** provides the complete reporting data model common to all the OFSAA Business Intelligence (BI) applications. This includes a single set of conformed dimensions as well as unified fact tables used for cross-functional reporting. For more information, see OFSDF Reporting model. OFSDF reporting model is the superset of all the BI-application specific reporting models.
- **Logical Data Model (LDM):** The OFSDF Logical Data Model is independent of OFSAA analytical applications, and more aligned to the underlying business processes of a financial institution. However, there is significant content overlap in the lower-level details of the LDM entities/attributes. This is captured in the form of attribute level mappings between LDM entities/attributes and their corresponding equivalents in the OFSDF staging area data model. This mapping is made available as part of the OFSDF release packaging.

- **Synchronized Releases:** The Staging and Reporting models that are part of an OFSDF release are updated to reflect prior application-specific releases. This means that the latest release of OFSDF (8.0.6.0.0) reflects all prior application releases across OFSAA from a data model perspective, with respect to the Staging and Reporting models.

2. Infrastructure

- The Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) is the same infrastructure that is used to deploy and manage an OFSDF instance.
- The same collection of tools (OFSAAI) is used to manage a deployed instance of the Oracle Financial Services Data Foundation (OFSDF).

These are the tools used to manage the data lifecycle in OFSDF:

- Model Upload
- Unified Analytical Metadata (UAM)
- Data Quality Framework
- T2T framework

OFSDF Prerequisite Components and Tools

The key prerequisites for running the OFSDF are as listed as follows:

Table 2. Pre-requisites

Component	Provider	Purpose
Oracle Financial Services Analytical Applications Infrastructure version 8.0.6.0.0	Oracle	OFSAAI is the platform on which the OFSDF is deployed and operated. It represents the OFSDF 'runtime' environment, and consists of a number of tools used to manage the data lifecycle within OFSDF, from sourcing to reporting*.
Oracle Database Enterprise Edition 11gR2 and 12cR1	Oracle	OFSDF is certified on Oracle Database releases 11gR2 and higher.
ERwin data modeler version 9.5 and 9.64	Computer Associates (CA)	ERwin is a data modeling tool that provides a visual environment to manage the complex enterprise data environment.

Note: OFSAAI, the infrastructure platform has its own prerequisites as well as supporting documentation.

This chapter explains the background of OFSDF, its functional architecture along with the differences from traditional warehouse architecture. OFSDF product package consists of the Logical Data Model and Analytical Data Warehouse model. The individual components of Physical and Logical Data Model, and Mapping Logical Data Model to OFSDF Staging Area are explained in detail.

This chapter covers the following topics:

- Background
- OFSDF Architecture
- Differences from Traditional Warehouse Architecture
- OFSDF Logical Data Model
- Staging Data Model
- OFSDF Logical Data Model
- Relationship to the OFSDF Physical Model
- OFSDF LDM Content Details
- Mapping from Logical Data Model to OFSDF Staging Area

Background

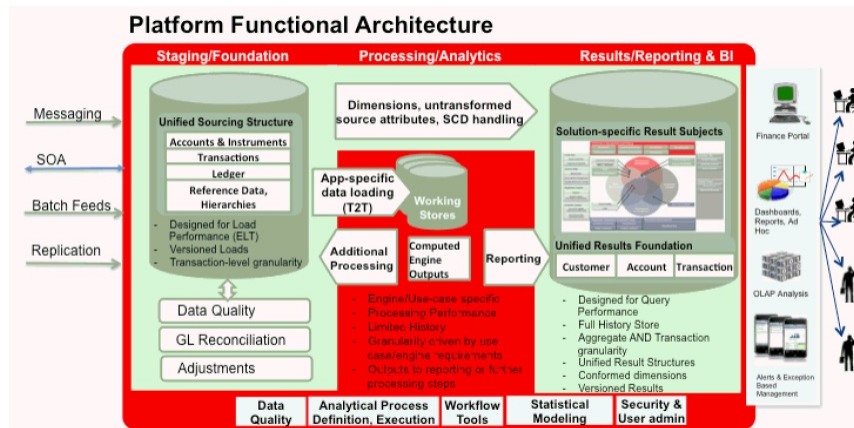
It is important to note that OFSDF architecture differs from 'traditional' data warehouse architecture in key ways.

OFSDF was built to specifically address the key challenges of building a scalable, practical data management platform to support Financial Services Analytics needs. The differences from traditional BI architecture are highlighted and explained in the following sections.

OFSDF Architecture

The following figure depicts the OFSDF Functional Architecture.

Figure 1. Architecture



The OFSDF architecture can be thought of as two distinct, interacting components. The dark green colored cylindrical portions denote the data repository and the red colored portion denotes the management toolkit.

1. **A data repository** whose physical structure is given by the OFSDF physical data model (Here Physical Data Model = OFSDF Staging + OFSDF Reporting). *This user guide only details the structure and organization of the data repository that is covered by the data models in the OFSDF.*
2. **A management toolkit** provided by OFSAAI that is used to manage the data repository, by providing a collection of tools and frameworks based on a common metadata foundation. *This user guide DOES NOT cover the details of the Analytical Application Infrastructure, as that is a separate product with its own documentation..*

The architecture illustrates the following key concepts:

1. **A unified data-sourcing area for analytics:** The OFSDF Staging Data Model provides the basis for central, unified data-sourcing layer for a wide variety of analytical needs. The staging layer faces the operational (OLTP) and front office systems of a bank. It consists of tables to capture key business and operational data from these systems, which is then processed by one or more analytical engines.
2. **A unified reporting/consumption layer:** Analytical results can be simple to complex, quantitative and qualitative measures of a bank's Risk, Compliance, Customer and Financial Performance. The OFSDF Reporting data model is a dimensional data model spanning these key analytical functions. It forms the foundation of OFSAA Business Intelligence applications, but can clearly be used as the result data store for any equivalent engines and processes provided by other vendors, or custom-built solutions. By providing a single data repository for reporting needs, this layer provides a foundation for departmental as well as cross-departmental and cross-functional reporting.

Additionally, the schema of the reporting area is specifically built for Financial Services analytics. As an illustration, the reporting area has a 'Run dimension' that is shared across all BI/reporting use cases. Similarly, the aggregation of shared measures/reporting into a common set of output structures (Account Summary) allows for cross-functional reporting, which is increasingly the norm in Financial Institutions.

3. **Single point of control and operation:** The Oracle Financial Services Analytical Applications Infrastructure is a separate Oracle product that offers a set of tools that are built on a common metadata foundation. These tools are used to control and manage the lifecycle of data from sourcing to reporting. There is a collection of frameworks to manage the following lifecycle steps of data within OFSDF:

-
- Metadata Management
 - Data Quality
 - Data Movement
 - Scheduling and runtime operations
 - Security/User management
 - Analytical Process Definition and Execution

4. **Processing Area:** As explained earlier, the primary purpose of the OFSDF is to serve as a container for analytical processing from sourcing to consumption. Such processing is usually delivered in the form of discrete units called analytical applications spanning different analytical use cases ranging from Finance to Risk to Compliance.

These applications consist of custom-built computational engines and numerical libraries, and may execute processes on the data that range from simple aggregations to complex, multi-step stochastic processes such as Monte-Carlo simulation.

Hence, analytical applications place varying demands on the data infrastructure in terms of volumes and speed, and hence place different demands on data architecture. In practice, the normalized (3NF) design favored for Enterprise Data Warehouses often fails to be efficient or performant when it comes to analytical processing across a wide range of use cases.

Therefore, the OFSDF recognizes the need for distinct application-specific working stores, separate from the staging and reporting area. For example, the OFSAA Asset/Liability Management application (ALM) has a distinct set of ALM-specific tables, as does the Market Risk solution.

Important: The structure of these processing area stores is decided by the actual analytical application and engine used. The OFSAA suite of applications is organized this way, with each application managing a specific set of tables/schemas within the processing area.

The processing area tables/schemas are NOT part of the OFSDF. This is because the OFSDF is intended to be an open platform. Other analytical applications and engines can equally provision data out of OFSDF by mapping their input requirements appropriately to the *OFSDF staging area model*.

Differences from Traditional Warehouse Architecture

This table summarizes the differences of the OFSDF from a traditional Data Warehouse architecture.

Table 3. Differences of the OFSDF

Component	FSDf	Traditional BI Architecture
Staging Layer	Common staging Area (CSA) where data from source systems are staged and is implemented as database schema.	Usually a filesystem-based area where file-based extracts (operational images) from source systems are staged prior to loading into a target schema.
3rd Normal Form Operational Data Store/Enterprise Data Warehouse	Does not provide a physicalized model for a 3NF store. The FS LDM can be physicalized as a 3NF store if desired. Operational/fine-grained reporting will be fulfilled from the reporting area.	Notification sent to all users captured as owners and user who submitted the definition.
Data Marts/Reporting Model	Set of star schemas with conformed dimensions (Ralph Kimball approach)	Set of star schemas

With the preceding understanding in mind, the following sections describe the data models in the OFSDF in greater detail.

1. **Readily Deployable:** The Physical Data Model is a readily deployable physical schema. It is provided as an ERwin data model file (for details on ERwin, see <http://erwin.com/products/modeler/>) and consists of tables grouped into distinct subject areas depending on function. The tables are either used to gather source data (Staging Area), or as containers of outputs/results from analytical processing and engines for reporting purposes (Reporting Area).
2. **Use-case Driven:** The OFSDF Physical Data model is driven by a set of clearly identified analytical use cases spanning Risk, Performance, Customer Insight, and Compliance.
3. **Extensible:** While the OFSDF Physical Data Model satisfies a very large number of analytical use cases across Risk, Finance, Marketing, and Compliance subject areas, customers may find the need to customize the model for a specific installation.

These customizations may be done in accordance with guidelines published in Using OFSDF section of this manual.

The OFSDF Physical Data Model is divided into two primary areas:

Staging Data Model

Overview/Design

The Common Staging Area Model (CSA) represents the point of entry of data into the OFSDF. The CSA provides a simplified, unified data sourcing area for inputs required by analytical applications and engines. It consists of over 900 tables and nearly 9000 columns organized into distinct subjects. The salient features of the CSA are as follows:

1. **Mapping to Analytical Use Cases:** Since the primary purpose of the OFSDF is to be a data repository supporting analytics, each database object in the OFSDF physical data model is necessarily mapped to a corresponding analytical use case.

These mappings are captured in the data model, in the form of additional metadata called User-defined Properties (UDPs), and can be leveraged to reduce the scope of data gathering efforts by focusing on clearly-defined end use cases such as Basel II, Market Risk Analytics, ALM and others.

These mappings can readily be extracted into a Download Specification, which lists the data demands for a specific analytical use case. An example is shown below:

Figure 2. Download Specification

name	datatype	definition	Util	FR	FR	Basel II	CR	ROBA	CCRA	Market Risk	Channel Analytics	Column	Exp-Credit R	Exp-Market Risk	SAAR	Market Risk	DR
ms_date	DATE	This is the date on which the codes are extracted from the source systems.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
rejt_reason_code	VARCHAR2(60)	This stores the description of the application rejection reason code	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
ppm_hdrcty	VARCHAR2(20)	Application rejection reason code like inadequate collateral, industry risks etc	NO	NO	NO	NO	Yes	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO
ppm_hdrcty_code	VARCHAR2(20)	This column stores the application status code	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
ms_date	DATE	This is the date on which the status codes are extracted from the source systems.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
ppm_hdrcty_desc	VARCHAR2(60)	This column stores the description of application status codes	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
ppm_hdrcty_desc	VARCHAR2(20)	This column stores the application type code like fresh, existing and so on.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
ms_date	DATE	This is the date on which the codes are extracted from the source systems.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
ppm_hdrcty_desc	VARCHAR2(30)	This column stores the description of application type.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
tr_reason_code	VARCHAR2(10)	This is the unique code (identifier) assigned to each of the attrition reasons.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
tr_reason_desc	VARCHAR2(60)	This is the date on which processing commences. Ideally, the end of the business day of the month.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
ms_date	DATE	This is the date on which the codes are extracted from the source systems.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
tr_reason_desc	VARCHAR2(60)	This column holds the description of reasons	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
tr_reason_code	CHAR(2)		NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
tr_reason_desc	VARCHAR2(60)		NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
tr_reason_code	VARCHAR2(20)	This column stores the authorization decision reason codes.	NO	NO	NO	NO	Yes	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO
ms_date	DATE	This is the date on which the codes are extracted from the source systems.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
tr_reason_desc	VARCHAR2(60)	This column stores the description for authorization decision reason codes.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
tr_reason_code	VARCHAR2(20)	Stores the bank role for the Securitisation Exposure	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	Yes	NO	NO	NO	NO
ms_date	DATE	The extraction date represents the period for which the data is meant for. When the frequency of data is monthly the extraction date will be the month end date.	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	Yes	NO	NO	NO	NO

The Mappings can be generated from the OFSDF ERwin file using ERwin Data modeling tools.

2. **Schema Design:** The data model of the Common Staging Area is designed to facilitate loading efficiency. This means that it is denormalized.

The primary focus of the staging area is to efficiently deliver data from operational systems such as Core Banking, Trading and Wealth Management to multiple analytical applications that process this data.

Typically, this data is extracted from source systems and loaded into OFSDF directly, or alternatively into a pre-defined file-based operational image area from which it is subsequently loaded into the OFSDF schema. In a large bank, it is not unusual to have 10s to 100s of millions of accounts and transactions.

Standard ETL (Extract, Transform, Load) approaches can fail to address this requirement adequately because of the complexity imposed by the target schema. If the target schema is increasingly normalized, then the ETL into this schema is correspondingly more complex, requiring careful load ordering to prevent integrity-related load failures, as well as integrity validation prior to loading. Such complex ETL processing is time-consuming and is prone to failure.

To address this, the CSA is designed to support a simplified loading process. De-normalized tables allow efficient data loading, and subsequent transformations can be done to verify data integrity through a series of data quality checks. This represents an 'ELT (Extract Load Transform)' approach to data sourcing, which is far more suited for an analytical data repository.

3. **Application-managed Referential Integrity (RI):** In conjunction with the database design of the staging schema, a key feature is the management of Referential Integrity primarily in the application tier, rather than within the database. Rather than imposing foreign key relationships in the database, which could cause

complex loading order dependencies, relationships between staging tables are managed by the Data Quality (DQ) framework, a toolkit within the Analytical Application Infrastructure that captures the relevant data quality checks for a specific table in the staging model. These checks include:

- Value checks (Nulls, Value ranges, business checks on numeric fields)
- Referential Integrity checks (which are otherwise implemented as foreign key constraints)

Important: This is also why the ER model of the staging area in ERwin does not contain any relationships – the staging area is a physical data model, which is deployed using the Analytical Application Infrastructure, which manages it.

In summary – the design of the staging area data model is to allow efficient data loading for analytics. It thus has crucial differences from a general-purpose repository of operational/transactional data across a bank.

Details of the Staging Data Model

The CSA model consists of over 900 tables and nearly 9000 attributes. These tables are organized into multiple 'subjects', currently by analytical use case/solution need.

However, it makes sense to first understand the staging area tables in terms of content before understanding how they map to analytical use cases.

There are two broad categories of staging data, regardless of the use case or analytical application that requires it:

1. **Business Data:** This set of tables captures the actual business events and the resulting state of a bank from those business events. The OLTP systems (or Transactional Systems) capture this information resulting from the execution of the bank's different business processes. Broadly, this information can be categorized as:
 - **Events:** Business transactions, whether financial or non-financial, represent business happenings (events) that are relevant for analytical purposes. For example – a financial transaction by a customer on a current account is a specific event. Events happen at a specific point in time, and are recorded by OLTP systems. In the staging area model, there are several **transaction** tables that capture this detail – for both financial and non-financial transactions.
 - **State:** The net effect of business transactions is to change the bank's overall financial/non-financial state. An example of this – when a customer opens a checking/current account and deposits money into it, the account tracks the net effect of all withdrawals and deposits using a numeric quantity called a **'balance'**. The **account** will also contain a list of all events (Withdrawals, deposits, fees, etc) that resulted in the balance. This state information is typically captured by product-specific systems in a bank or FSI. For example, there is a lending system that captures details of loans, and a current account system that captures details of checking and savings accounts, which are distinct products. In both cases, the accounts are governed by contracts, which refer to the terms and conditions governing business on that account.
2. **Reference/Master Data:** Events and state refer to 'business activities' of a bank or FSI. To provide more detail on these, banks need to capture additional data that provides context for these activities. This data may be variously called as 'reference data' or 'master data', and covers various business dimensions of a giventransaction or account. For example – a bank has a master list of products that it sells to customers (Product Master). Similarly, it has a list of customers (Customer master). A trading firm may hold a list of

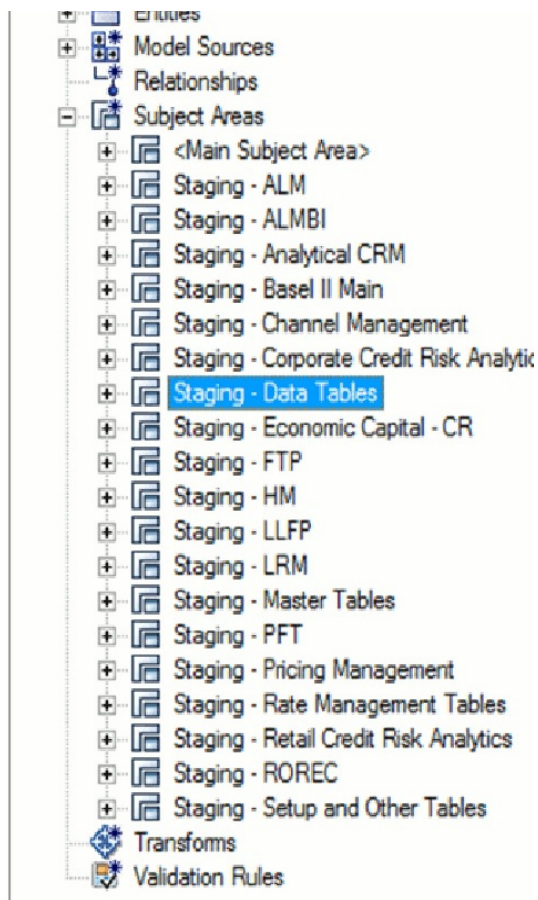
securities it transacts in (Securities master). These and other lists provide context for each business transaction or account. Banks typically maintain 'Master' data for this purpose.

With this background, the following lists the key categories of business data and reference data in the staging data model.

Business Data:

To view the current set of **business data** tables in the staging area, open the OFSDF Staging Area model in the ERwin data modeling tool, and click on subject areas as shown below. Right click and switch to the 'Staging-Data Tables' subject area.

Figure 3. 'Staging-Data Tables' Subject Area



This provides an ordered, visually grouped list of the business data tables in the current staging area model. Above each group in the diagram is the group name as a label, to categorize the tables, as follows:

Figure 4. Business Data Tables



The key groups of business data tables in the model are as follows.

Table 4. Key Groups

Group Name	Purpose
Product Processors	Tables for Financial Instruments and Contracts. Note that these tables can accommodate nearly 80+ types of instruments and derivatives across Banking and Trading books.
Transactions	Tables to hold Transaction/Event level data from the bank's systems. These tables are available by both contract and channel, and there is additionally a set of Transaction Summary tables that contain transaction data in a summarized form required by the Profitability application.
Exposures	Exposures are basically contracts on the asset side of the balance sheet. The tables in this category are primarily used by Risk applications.
Mitigants	Mitigants are used to address specific types of risk posed by exposures. The tables in this category are primarily used by Risk applications.
GL	Contains tables holding information pertaining to the General Ledger.
Customer	Consists of Marketing-related Customer activity and plan data relevant to CRM analytics. Note that core customer information is in the Product Processors, and reference data about customers is elsewhere.
Rates	Economic indicators, Interest Rates and other information relevant to analytical processing for Risk and Asset Liability Management applications.

Product Processors

Product Processors are classified into four categories as depicted in the following table:

Product Category	Entity Name
Asset	Payment Settlement Account Stage Cards Stage Correspondent Accounts Stage Foreign Exchange Contracts Stage Investments Stage Leases Contracts Stage Loan Contracts Stage Managed Investment Account Other Services Stage Merchant Banking Stage Money Market Contracts Stage Over Draft Accounts Stage Repo Contracts Stage Credit Facility Details
Liabilities	Payment Settlement Account Stage Annuity Contracts Stage Borrowings Stage Casa Accounts Stage Correspondent Accounts Stage Custodial Accounts Stage Foreign Exchange Contracts Stage Merchant Cards Stage Money Market Contracts Stage Mutual Funds Stage Prepaid Cards Stage Repo Contracts Stage Retirement Accounts Stage Term Deposit Contracts Stage Trading Account Stage Trusts
Off Balance Sheet	Stage Bill Contracts Stage Borrowing Commitment Contract Stage Commitment Contracts Stage Credit Derivatives Stage Futures Contract Stage Guarantees Stage Letter Of Credit Contracts Stage Option Contracts Stage Repo Contracts Stage Swaps Contracts Stage Forwards Contract
Derivatives	Stage Credit Derivatives Stage Futures Contract Stage Option Contracts Stage Swaps Contracts Stage Forwards Contract

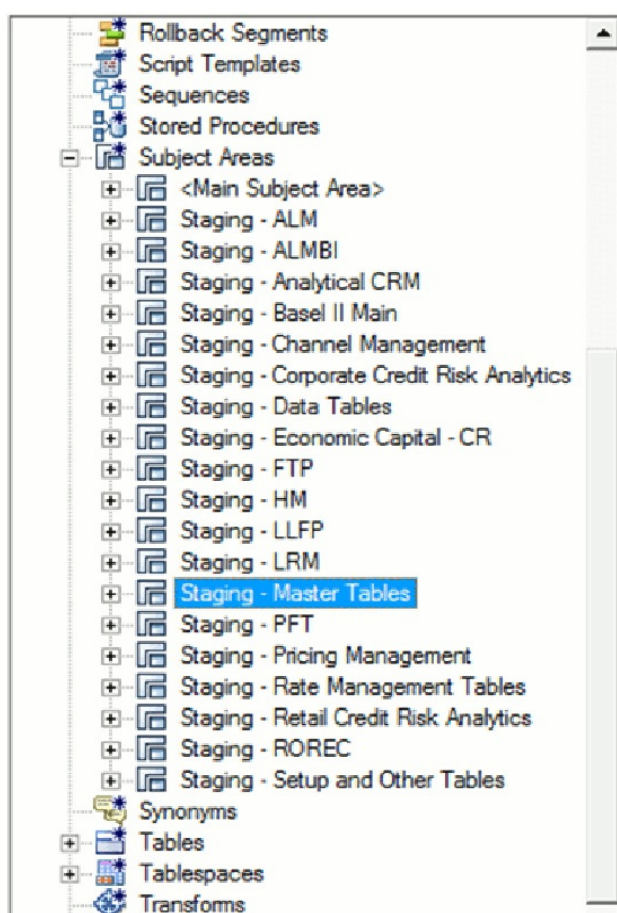
For detailed mapping information, see [Banking Product to Product Processor Mapping](#) document at My Oracle Support. This document provides details regarding which banking product must be sourced to which Product Processor table based on the product functionality

Reference/Master Data

To view the current set of Master/Reference data tables in the staging area, open the OFSDF Staging Area model in the ERwin data modeling tool, and open up the subject areas menu on the left hand side, as shown below.

Right click and switch to the 'Staging-Master Tables' diagram to get the list of the master tables currently in staging.

Table 5. Staging Master Tables



Similar to the Business Data tables, the '**Staging - Master Tables**' subject area provides a single folder view of all of the reference/master information currently required by the staging area.

Reporting Data Model

Overview

The Reporting Data Model is the point where outputs of analytical processing are aggregated for reporting and Business Intelligence (BI) tools.

Similar to the Common Staging Area being the foundation for data provisioning to analytical applications, engines and processes, the Reporting Data Model is the common data store for the **outputs** of these processes. Outputs are computed quantitative measures and Key Performance Indicators that involve simple

to complex, mathematical and statistical processing using the raw data, which is performed by specialized engines and computational models. In the OFSDF design, the Reporting Data Model design ensures that the historical data is maintained.

The key features of the design of the Reporting Area model are as follows:

1. Design:

The Reporting Area data model is a dimensional data model. This means that it consists primarily of central fact tables (de-normalized), related to multiple dimension tables, also called a Star Schema. Additionally, the dimension tables are shared across the star schemas in the reporting mode, meaning they are Conformed Dimensions. This means that Drill-across reporting is naturally supported by the OFSDF design.

Additionally – in keeping with the key principle of the OFSDF, the Reporting Model is organized by use cases to facilitate reporting and BI in a wide variety of areas.

2. Support for multiple scenarios of analysis:

Increasingly as a result of the 2008 crisis, the Financial Services industry is moving towards scenario-based, forward-looking risk analysis instead of retroactive analysis. The reporting data model has been designed to support scenario analysis of the sort required by financial institutions that need to measure and report risk and performance under a variety of economic scenarios.

To facilitate this, the Oracle Financial Services Advanced Analytical Infrastructure (OFSAAI) provides a Stress Testing framework, allowing risk analysis to be performed under a variety of known scenarios corresponding to different input parameter values to risk models.

The reporting model provides support for this kind of analysis via a Run Dimension – it allows analytical engines to load multiple result sets identified by scenarios, and hence permits reporting related to baseline and stress conditions in economic terms.

3. Support for Cross Functional Reporting:

The third critical feature of the Reporting area design is the support for cross-functional reporting.

Typically, Business Intelligence and Reporting Solutions work off a dedicated, purpose-specific data store called a data mart. Data marts are function-specific data stores typically star schemas (eg. Marketing data marts, Risk Data Marts, Customer Data mart), that provide the necessary reporting and analytics relevant to a particular business function in the FSI.

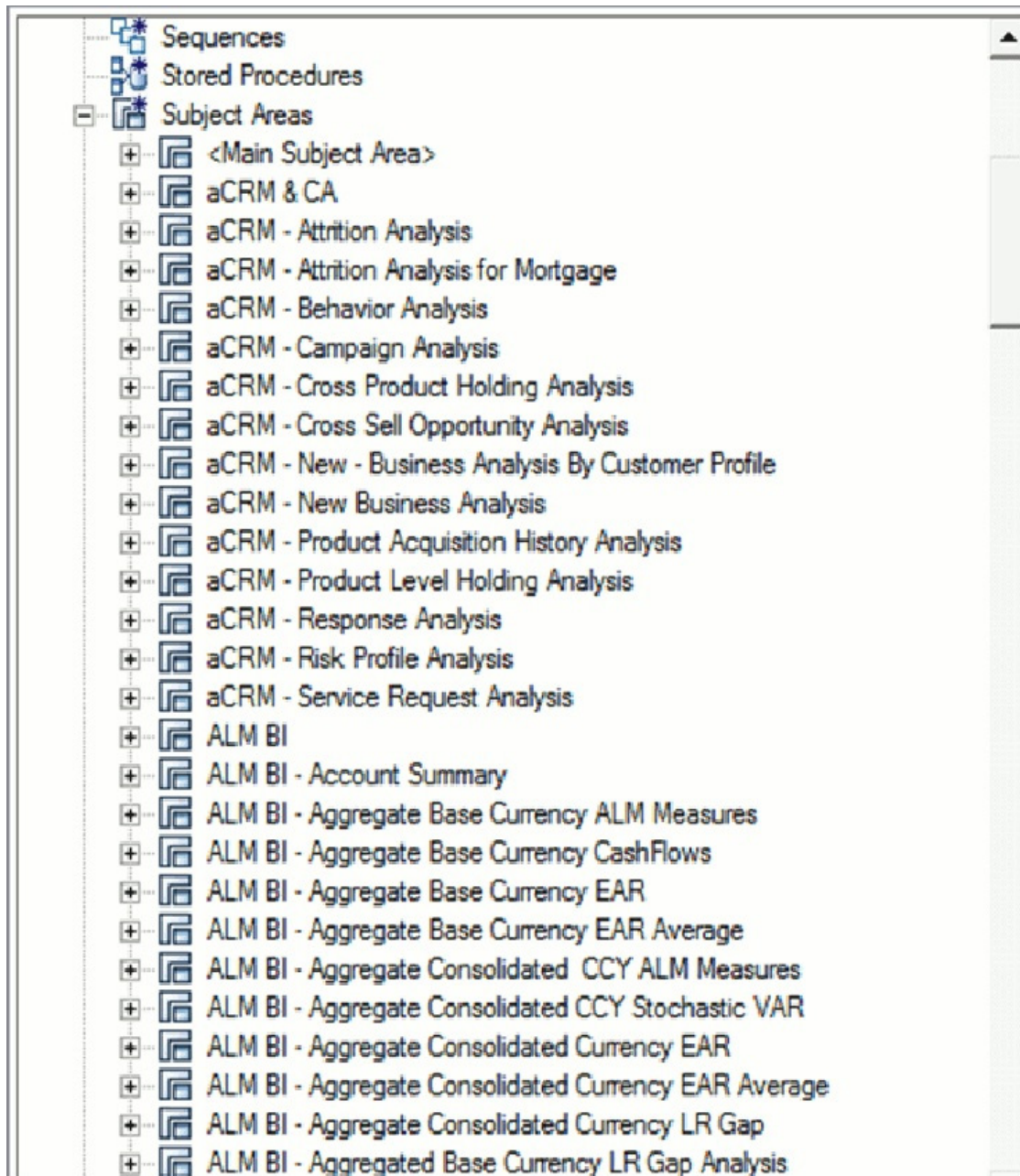
Increasingly, the trend is for cross-functional analytics and reporting. Majority of emerging needs relate to the analytical problems at the intersection of the distinct areas of Risk, Performance, Customer Insight, and Compliance.

Reporting Area Details

Similar to the staging data model, the reporting model is divided into Subject Areas.

These subject areas are visible by clicking on the **Subject Areas** view in the ERwin data modeling tool as show below.

Figure 5. Subject Areas



As can be seen above, the Subject Areas are organized to support detailed level analysis related to a set of higher-level analytical reporting solutions. The list of solutions supported by the OFSDF are:

Table 6. List of Solution Supported by OFSDF

Name	Definition
ALM	Subject Areas corresponding to Asset Liability Management (ALM).
Basel II, Pillar I and III, Pillar II	Subject Areas corresponding to the Basel II Regulatory framework, and its reporting requirements as specified in the framework.
Capital Planning	These Subject areas provide support for reporting related to Capital Planning
Channel Management	Support for analytics related to Channel Management, which is part of the overall Customer Insight solution set
Common Account Summary	A critical subject area that collects account-level results from multiple analytical processes. The Common Account Summary allows for cross-functional Analytics such as Risk Adjusted Performance Measurement by combining outputs from Profitability and Risk solutions
Corporate Credit Risk Analytics	Support for detailed analytics and reporting on Corporate Credit Risk - including Commercial Lending, Credit Facilities, Limits and Collateral
Economic Capital	Support for Aggregated Economic Capital Reporting based on risk assessments for Market, Credit and Operational Risk.
GL Reconciliation	Support for reporting specific to the Oracle GL reconciliation module that is part of the OFSAA product line. This allows a reporting view of the reconciliation processes and outputs
ICAAP	Supports reporting related to the Internal Capital Adequacy Assessment process.
LRM	Supports reporting related to Liquidity Risk Management processes within an organization
CRM	Part of the Corporate Credit Risk Solution, allows reporting on Credit Limits.
CI	Part of the Customer Insight solution set providing support for Retail, Institutional, and Channel Analytics.
Market Risk	Support for Market Risk analytics
PFT BI	Supports reporting related to Profitability analysis, part of the Enterprise Performance Management solution area.
Reputational Risk	Support for Reputational Risk measurement and analysis
Retail Credit Risk	The set of tables/subject areas supporting Retail Credit Risk Analysis.

Table 6. List of Solution Supported by OFSDF

Name	Definition
Retail Pooling	Reporting support for Pooling for Retail Exposures – required by ALM and Retail Credit Risk analysis.
Strategic Risk	Support for reporting related to Strategic Risk – measurement of these risks is a qualitative process, and part of the ICAAP framework.

Within the above higher-level areas, there are over 165 subject areas consisting of star schemas supporting detailed analysis. Each Subject Area typically corresponds to one or possibly more star schemas, depending on the reporting need. There are currently 500+ fact and dimension tables in the reporting area. The details of the various naming conventions used in OFSDF Data Model are explained in Appendix A. For column-level details, see the Technical Whitepaper on *Data Model Document Generation*, which details how to extract the data dictionary from ERwin section.

OFSDF Logical Data Model

The previous sections have discussed the components of the **OFSDF Physical Data Model** (or the OFSDF Analytical Warehouse model) – this is the actual deployable physical database model that addresses a number of analytical needs out of the box, as illustrated earlier.

The Logical Data Model is a reference data model of the Financial Services Domain, which captures the data created by the key business processes in Financial Services. A reference data model is different from the Physical Data Model in these ways:

Table 7. Difference between Reference Data Model and Physical Data Model

Area	FS LDM	OFSDF Physical Model
Purpose	Generic blueprint of the data reduced by the business processes of a Financial Services institution	Designed for analytical processing (Risk, Profitability, Customer Insight)
Format/Structure	Entity Relationship Diagrams organized into key Top Level subject areas	Physical Tables/Columns Definitions
Usage	Cannot be directly used – needs to be used within a physical database design step to arrive at a database schema	Readily deployable
Scope	All the key business processes/activities and their supporting reference data requirements	The business and reference data required for a set of analytical use cases. However, can be extended to serve other needs.

Relationship to the OFSDF Physical Model

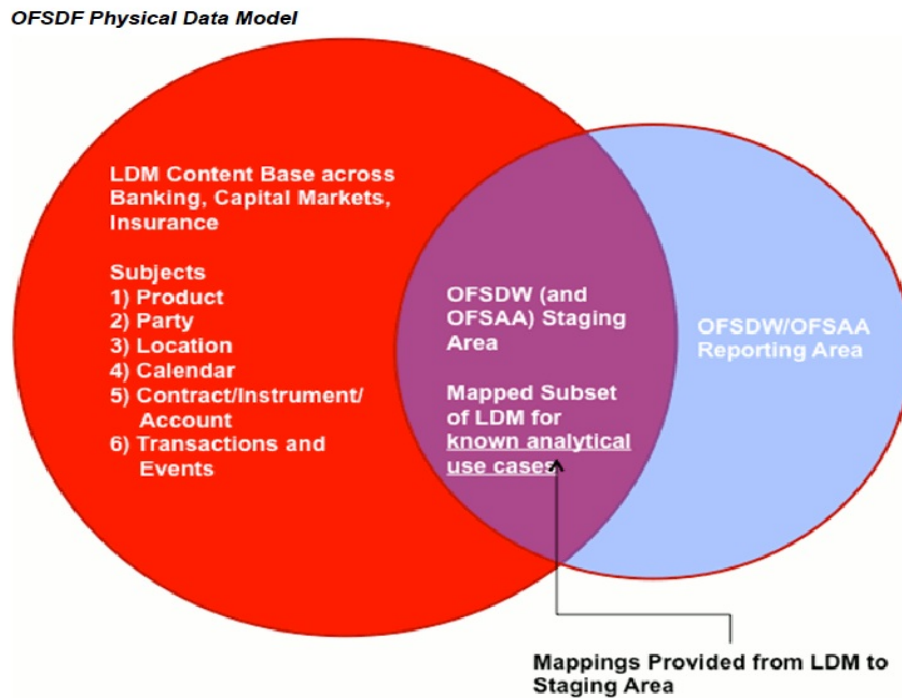
The coverage of the OFS LDM in terms of core business processes is as follows.

Table 8. Coverage of OFS LDM

Business Area	Comments
Retail Banking	End to end content support for Retail Assets and Liabilities – Retail lending across a variety of loan products, Current Accounts, and Term Deposits.
Corporate Banking	End to end process support for Commercial Assets and liabilities – Corporate/Commercial lending operations, Structured Lending, Collateral and Limits models, and Trade Finance.
Capital Markets, Trading and Securities	Process support for the trade lifecycle from quoting to settlement/clearance. Coverage of a wide variety of security and derivative asset classes.
Insurance	Preliminary coverage of Insurance liabilities in the non P & C insurance domains.

The OFSDF physical data model should be seen as a **deployable, physicalized subset** of this reference data model for known analytical use cases spanning Risk, Performance, Customer Insight, and Compliance, as illustrated below.

Figure 6. OFSDF Physical Data Model



OFSDF LDM Content Details

The OFS LDM is organized into a small set of high-level subject areas, each of which represents a key category of data that is produced by the business processes in a bank.

Table 9. Subject Areas and their definitions

Subject Area Name	Subject Area Definition
Accounting	This subject area deals with the structure of the internal accounting of a financial institution, which includes budget , GL & GL mapping to budgets.
Application	This subject area covers various aspects of application across different products. The section helps to analyze potentially risky customers and performance of different sourcing channels.
Calendar	This subject area deals with entities that contain the details for Calendar & Fiscal Date, Month, Quarter, Half year & Year.
Campaign	This subject area provides the information on the campaigns, which are organized by the financial institutions. It covers the details of the campaign treatment, offers given on a campaign, response & delivery status and so on, which helps in analyzing the profitability of a campaign.
Channel	This subject area caters to all the variety of channels, which are involved in the functioning of financial institution in terms of advertisement or collecting response or sending message, or so on.
Claim	This subject area covers the details of claims originated from insurance contracts. It also captures the various claim events, the settlement history & the claim recovery process.
Collateral	This subject area records the details of various physical & financial collateral and also helps in analyzing the quality of the each collateral.
Collections and Recovery	This section captures performance of collection & recovery activities. The section helps ascertain the amount and quality of the collections and the recovery that was made using the secondary means of collecting money (by selling collaterals, and so on).
Common	This subject area covers the entities that are common across subject areas like currency, data origin, and so on.
Contract	This subject area captures contract & account details of various banking & insurance products, which includes Loans, Cards, CASA, TD, Collection, Insurance, Merchant Card, Overdraft, Commitment, Prepaid Cards, Reinsurance, and so on.
Coverage	This subject area deals with the presentation of the Insurance policy that defines and limits the provisions and obligations under an Insurance Policy. Components of the coverage are the provisions, benefits, options, terms and condition specific to offered insurance to an insured party.
Geography	This subject area deals with entities that contain the physical address of the party and offices where they are located or contacted.

Table 9. Subject Areas and their definitions

Subject Area Name	Subject Area Definition
Instrument	This subject area captures the details of various instrument financial organization deals with like Certificate of deposit, common stock, derivatives, equity, fixed income securities, repo, and so on
Limit	This subject area defines the generic limit, which can be associated with an account access restriction, acceptance tolerance on amount difference, age of contribution, approval workflow, number of transactions allowed, lending and credit.
Market	This subject area identifies entities that are of importance with respect to capturing data feed from Order Management and Order Execution systems along with the entities that are involved in the clearing and settlement processes.
Organization	This subject area deals with entities that can perform one or many business functions and it is a unit of a financial institution that can be consolidated in both a managerial and legal hierarchy.
Party	This subject area deals with the entities, which can interact with the Financial Organization for any kind of business/non-business activity and on the same lines the entities with which the bank interacts with. It also includes Bank as a whole and various internal units of the bank mutually involved in certain set of transactions. Party in different contexts can play different roles i.e. Customer for a particular contract and a Vendor in some business relationship (servicing the bank).
Product	This subject area deals with entities which stores the features & details of the products, which are offered by the financial institutions.
Reinsurance	This subject area covers the details of reinsurance contract, which is a contract between reinsurer & an insurer.
Securitization	This subject area covers the details & the complete process of securitization, it also covers on mortgage servicing rights.
Trade	This subject area covers the details of order management, trade execution & trade settlement which helps in analyzing the trade life cycle.
Transaction	This subject area deals with entities and relationships that capture any kind of activity (financial or non-financial) including those on an account or between accounts.

Within the above subject areas, the FS LDM has 800+ entities. The various naming conventions used in OFSDF Logical Data Model are explained in Appendix A. For column-level details, see the Technical Whitepaper on *Data Model Document Generation*, which details how to extract the data dictionary from ERwin section.

Mapping from Logical Data Model to OFSDF Staging Area

In addition to the data dictionary and download specifications, the OFSDF also provides a mapping between the relevant columns in the OFSDF physical staging area model and the OFSDF Logical Data Model. These mappings are currently made available as Excel spreadsheets, and part of the OFSDF package.

Following are the links to the excel spreadsheets. Click the corresponding spreadsheet link that you want to view.

- [LDM Staging mapping Accounting](#)
- [LDM Staging mapping Application](#)
- [LDM Staging mapping Campaign](#)
- [LDM Staging mapping Campaign 7.3](#)
- [LDM Staging mapping Claim](#)
- [LDM Staging mapping Collateral](#)
- [LDM Staging mapping Collections and Recovery](#)
- [LDM Staging mapping Contract](#)
- [LDM Staging mapping Credit Card 7.3](#)
- [LDM Staging mapping Geography](#)
- [LDM Staging mapping Instrument](#)
- [LDM Staging mapping Organization](#)
- [LDM Staging mapping Party](#)
- [LDM Staging mapping Product](#)
- [LDM Staging mapping Transaction](#)

Note: Column description updated as ‘deprecated’ would be dropped from the model in the next major release.

CHAPTER 3***Dimension Loading Process***

This chapter provides information about Dimension Loading Process in the Oracle Financial Services Data Foundation application.

This chapter includes the following topics:

- Dimension Table Population
- Overview of SCD Process
- Prerequisites
- Tables Used by the SCD Component
- Executing the SCD Component
- Checking the Execution Status

Dimension Table Population

Dimension Tables in Data Foundation Solutions are of two types:

- Seeded Dimensions
- SCD Dimensions

Data Foundation solutions use the SCD component to handle dimensional data changes.

Overview of Seeded Dimensions

The Seeded Dimensions are those Dimension tables which are pre-populated with data (which are standard and OFSAA specific codes). The Banks are supposed to refer these dimension tables and its values to configure the ETL Layer of Staging area wherever there is a corresponding reference.

Overview of SCD Process

A Slowly Changing Dimension (SCD) is a dimension that stores and manages both current and historical data over time in a data warehouse. SCDs are dimensions that have data that changes slowly, rather than changing on a time-based, regular schedule. It is considered and implemented as one of the most critical ETL tasks in tracking the history of dimension records. There are three types of SCDs and you can use Warehouse Builder to define, deploy, and load all three types of SCDs.

Type 1 SCDs - Overwriting

The Type 1 methodology overwrites old data with new data, and therefore does not track historical data. This is useful for making changes to dimension data.

Example:

In this example, N_PRODUCT_SKEY is the surrogate key column which is a unique key for each record in the dimension table. V_PRODUCT_NAME is the product name. D_START_DATE indicates the date from which this product record is valid.

Table 10. Type 1 SCD Methodology Example

N_PRODUCT_ SK EY	V_PRODUCT_ NAME	D_START_DATE	D_END_DATE	F_LATEST_RECORD_ INDICATOR
1	PL	5/31/2010	12/31/9999	Y

D_END_DATE indicates the date till which this product record is valid.

F_LATEST_RECORD_INDICATOR with value 'Y', which indicates this is the latest record in the dimension table for this product and 'N' indicates it is not. If the V_PRODUCT_NAME column is set as a Type 1 SCD column and if there is a change in the product name to 'Personal Loan' from 'PL' in the above example, in the next processing period, then when SCD is executed for the new processing period the record in the above example changes to:

Table 11. Record Change Example

N_PRODUCT_ SK EY	V_PRODUCT_ NAME	D_START_DATE	D_END_DATE	F_LATEST_REC ORD_INDICAT OR
1	Personal Loan	6/30/2010	12/31/9999	Y

Type 2 SCDs - Creating another dimension record

The Type 2 method tracks historical data by creating multiple records for a given natural key in the dimensional tables with separate surrogate keys. With Type 2, the historical changes in dimensional data are preserved. In the above example for the change in product name from 'PL' to 'Personal Loan' if history has to be preserved, then the V_PRODUCT_NAME column has to be set as Type 2 when SCD is processed for the processing period and the change inserts a new record as shown in the following example:

Table 12. Type 2 SCD Methodology Example

N_PRODUCT_ SK EY	V_PRODUCT_ NAME	D_START_DATE	D_END_DATE	F_LATEST_RECORD_ INDICATOR
1	PL	5/31/2010	12/31/9999	N
2	Personal Loan	6/30/2010	12/31/9999	Y

A new record is inserted to the product dimension table with the new product name. The latest record indicator for this is set as 'Y', indicating this is the latest record for the personal loan product. The same flag for the earlier record was set to 'N'.

Type 3 SCDs - Creating a current value field

A Type 3 SCD stores two versions of values for certain selected level attributes. Each record stores the previous value and the current value of the selected attribute.

When the value of any of the selected attributes changes, the current value is stored as the old value and the new value becomes the current value.

For more information on SCDs, see:

- *Oracle Data Integrator Best Practices for a Data Warehouse at*
<http://www.oracle.com/technetwork/middleware/data-integrator/overview/odi-best>

[practices-datawarehouse-wbi-129686.pdf](#)

- Oracle® Warehouse Builder Data Modeling, ETL, and Data Quality Guide at http://docs.oracle.com/cd/E11882_01/owb.112/e10935.pdf
[http://docs.oracle.com/cd/E14072_01/owb.112/e10935.pdf]

Additional online sources include:

- http://en.wikipedia.org/wiki/Slowly_changing_dimension
- http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/10g/r2/owb/owb10gr2_gs/owb/lesson3/slowlychangingdimensions.htm
- <http://www.oraclebidwh.com/2008/11/slowly-changing-dimension-scd/>
- <http://www.informationweek.com/news/software/bi/showArticle.jhtml?articleID=204800027&pgno=1>
- <http://www.informationweek.com/news/software/bi/showArticle.jhtml?articleID=59301280>

An excellent published resource that covers SCD in detail is *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling* by Ralph Kimball and Margy Ross.

The SCDs used in Data Foundation solutions are listed in the [Oracle Financial Services Data Foundation - SCD Metadata](#) spreadsheet under *Technical Metadata for FSDF 8.0.6.0.0*.

Prerequisites

1. The SCD executable should be present under <installation home>ficdb/bin. The file name is **scd**.
2. The user executing the SCD component should have execute rights on the file mentioned as prerequisite in point 2.
3. The setup tables accessed by SCD component are SYS_TBL_MASTER and SYS_STG_JOIN_MASTER.

SYS_TBL_MASTER stores the information like which is the source stage table and the target dimension tables. The source sometimes can be the database views which could be simple or a complex view.

SYS_STG_JOIN_MASTER stores the information like which source column is mapped to which column of a target dimension table. It makes use of data base sequence to populate into surrogate key columns of dimension tables.

Tables Used by the SCD Component

The database tables used by the SCD component are:

- SYS_TBL_MASTER

The solution installer will populate one row per dimension for the seeded dimensions in this table.

Table 13. SYS_TBL_MASTER

Column Name	Data Type	Column Description
MAP_REF_NUM	NUMBER(3) NOT NULL	The Mapping Reference Number for this unique mapping of a Source to a Dimension Table.
TBL_NM	VARCHAR2(30) NOT NULL	Dimension Table Name
STG_TBL_NM	VARCHAR2(30) NOT NULL	Staging Table Name
SRC_PRTY	SRC_PRTY NUMBER(2) NULL	Priority of the Source when multiple sources are mapped to the same target.
SRC_PROC_SEQ	NUMBER(2) NOT NULL	The sequence in which the various sources for the DIMENSION will be taken up for processing.
SRC_TYP	VARCHAR2(30) NULL	The type of the Source for a Dimension, that is, Transaction Or Master Source.
DT_OFFSET	NUMBER(2) NULL	The offset for calculating the Start Date based on the Functional Requirements Document (FRD).
SRC_KEY	NUMBER(3) NULL	

Sample Data: This is the row put in by the solution installer for the Line of Business dimension.

Table 14. Sample Data

MAP_REF_NUM	6
TBL_NM	DIM_LOB
STG_TBL_NM	STG_LOB_MASTER
SRC_PRTY	
SRC_PROC_SEQ	23
SRC_TYP	MASTER
DT_OFFSET	0
SRC_KEY	

Note: For any new dimension added, a row will have to be inserted to this table manually.

- SYS_STG_JOIN_MASTER

The solution installer will populate this table for the seeded dimensions.

Table 15. Table for Seeded Dimension

Column Name	Data Type	Column Description
MAP_REF_NUM	NUMBER(3) NOT NULL	The Mapping Reference Number for this unique mapping of a Source to a Dimension Table.
COL_NM	VARCHAR2(30) NOT NULL	Name of the column in the Dimension Table.
COL_TYP	VARCHAR2(30) NOT NULL	Type of column. The possible values are given in the following section.
STG_COL_NM	VARCHAR2(60) NULL	Name of the column in the Staging Table.
SCD_TYP_ID	NUMBER(3) NULL	SCD type for the column.
PRTY_LOOK- UP_REQD_FLG	CHAR(1) NULL	Column to determine whether Lookup is required for Priority of Source against the Source Key Column or not.
COL_DATATYPE	VARCHAR2(15) NULL	The list of possible values are VARCHAR, DATE, NUMBER based on the underlying column datatype.
COL_FORMAT	VARCHAR2(15) NULL	

The possible values for column type (the COL_TYPE column) in SYS_STG_JOIN_MASTER are:

1. PK – Primary Dimension Value (may be multiple for a given "Mapping Reference Number")
2. SK – Surrogate Key
3. DA – Dimensional Attribute (may be multiple for a given "Mapping Reference Number")
4. SD – Start Date
5. ED – End Date
6. LRI – Latest Record Indicator (Current Flag)
7. CSK – Current Surrogate Key
8. PSK – Previous Surrogate Key
9. SS – Source Key
10. LUD – Last Updated Date / Time
11. LUB – Last Updated By
12. NN - Not Null
13. Sample Data: This is the row put in by the solution installer for the Line of Business dimension.

- Click **New Task** ('+' symbol in Task Details container).
- Enter the Task ID and Description.
- Select **Run Executable**, from the Component ID list.
- Click **Parameters**. Select the following from the Dynamic Parameters List and then click **Save**:
 - Datastore Type - Select the appropriate datastore from the list
 - Datastore Name - Select the appropriate name from the list
 - IP address - Select the IP address from the list
 - Executable - scd,<map ref num>

Example:

scd, 61 (Refer the following sections for details)

- **Wait**: When the file is being executed you have the choice to either wait till the execution is complete or proceed with the next task. Click the list box of the field provided for Wait in the Value field to select 'Yes' or 'No'. Clicking **Yes** confirms that you wish to wait for the execution to be complete. Clicking **No** indicates that you wish to proceed.
- **Batch Parameter**: Clicking **Yes** would mean that the batch parameters are also passed to the executable being started; else the batch parameters will not be passed to the executable.

Important: Always select **Y** in Batch Parameter.

For the Parameter Executable earlier mentioned, the map ref num values are

- -1 (if you want to process all the dimensions). The *Executable* parameter mentioned earlier would be scd,-1
- If you want to process for a single dimension, query the database table SYS_TBL_MASTER and give the number in the map_ref_num column for the dimension you want to process. These are the ones which come seeded with the install.
- Execute the batch from Batch Execution by choosing the batch created following the steps mentioned in the preceding sections for a date.

Note: A seeded batch FSDFINFO_DATA_FOUNDATION_SCD is provided which consists of all the required dimensions as different tasks that are part of SCD.

Checking the Execution Status

The status of execution can be monitored using the Batch Monitor screen. You can access this from the Left Hand Side (LHS) menu as follows:

From the **Home** menu, select **Operations**, then select **Batch Monitor**.

Note: For a more comprehensive coverage, see [OFS Analytical Applications Infrastructure User Guide](#).

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going
- F - Failure
- S – Success

The ICC execution log can be accessed on the application server in the following directory:
ftpshare/logs/<Run_Date>/FSDFINFO/RUN EXECUTABLE.

The file name will have the batch execution id.

Sample

/dbfiles/home/oracle/OFSAAI/ficdb/log/ficgen

The detailed SCD component log can be accessed on the application server in the directory \$FIC_HOME, go one folder up from there and then accessing the following

ftpshare/logs/<Run_Date>/FSDFINFO/RUN EXECUTABLE

The file name will have the batch execution id.

Check the **.profile** file in the installation home if you are not able to find the paths mentioned earlier.

LOAD DIM TABLES THROUGH SCD

Batch FSDFINFO_DATA_FOUNDATION_SCD has been introduced with 177 tasks under it. These 177 tasks represent the 177 SCD processes where different staging tables would be the source and Dimension Tables would be the targets. The required SCDs have been introduced into SYS_TBL_MASTER table, and subsequently into SYS_STG_JOIN_MASTER.

Depending on the requirement by an application, a task can be excluded or included from the batch execution

LOAD KEY DIMENSIONS USING DRMLoader AND HIERARCHY FLATTENING

The Dimension Loader functionality in FSDF, enables you to load dimension tables such as DIM_ORG_UNIT, DIM_GL_ACCOUNT, DIM_COMMON_COA, DIM_PRODUCT, DIM_ORG_STRUCTURE.

Loading Data from STG_INTF Tables to DIM_INTF Tables

The following tables are part of staging. Dimension data is stored in the following set of tables:

- STG_<DIMENSION>_B_INTF - Stores leaf and node member codes within the dimension.
- STG_<DIMENSION>_TL_INTF - Stores names of leaf and node and their translations.
- STG_<DIMENSION>_ATTR_INTF - Stores attribute values for the attributes of the dimension.
- STG_<DIMENSION>_HIER_INTF - Stores parent-child relationship of members and nodes that are part of hierarchies.
- STG_HIERARCHIES_INTF - Stores master information related to hierarchies. Data present in the above set of staging dimension tables are loaded into the below set of dimension tables.
- DIM_<DIMENSION>_B - Stores leaf and node member codes within the dimension.
- DIM_<DIMENSION>_TL - Stores names of leaf and node and their translations.
- DIM_<DIMENSION>_ATTR - Stores attribute values for the attributes of the dimension.
- DIM_<DIMENSION>_HIER - Stores parent-child relationship of members and nodes that are part of hierarchies.
- REV_HIERARCHIES - Stores hierarchy related information.

- REV_HIERARCHY_LEVELS - Stores levels of the hierarchy.

For more information on viewing the structure of staging tables, refer to the Oracle Financial Services Analytical Applications Data Model Data Dictionary.

Staging tables are present for all key dimensions that are configured within the OFSAAI framework.

The Dimension IDs used for key dimensions are:

Table 17. Dimension IDs

Dimension ID	Dimension Table	Staging Tables
1	DIM_ORG_UNIT	STG_ORG_UNIT_**_INTF
2	DIM_GL_ACCOUNT	STG_GL_**_INTF
3	DIM_COMMON_COA	STG_COMMON_COA_**_INTF
4	DIM_PRODUCT	STG_PRODUCTS_**_INTF
5	DIM_ORG_STRUCTURE	STG_LEGAL_ENTITY_**_INTF

Executing The Loading Procedure using Batch Framework

The batch for Key Dimension Loader can be executed by executing the task (Task for Loading Data from DRM) present in the seeded batch FSDFINFO_DATA_FOUNDATION_SCD.

Below are the input parameters:

- pDIMENSIONID: This is the dimension ID.
- pSynchFlag: This parameter is used to identify if a complete synchronization of data between staging and fusion table is required. The default value is 'Y'.

For more details refer to Dimension Loaders Section, in Chapter 4 Data Loaders, of the Oracle Financial Services Analytical Applications Data Model Utilities Guide

Executing The Hierarchy Flattening Procedure using Batch Framework

The batch for Dimension Hierarchy flattening can be executed by executing the task (DT for DRM Dimension Hierarchy Flattening) present in the seeded batch FSDFINFO_DATA_FOUNDATION_SCD.

Below are the input parameters:

- pDIMENSIONID: This is the dimension ID.
- pHierarchyId: Enter the Hierarchy ID. If all the hierarchies belonging to a dimension are to be processed then, provide NULL as the parameter value. Else, provide the System Identifier of the hierarchy that needs to be transformed.

For more details refer to Hierarchy Transformation Section, in Chapter 4 Data Loaders, of the Oracle Financial Services Analytical Applications Data Model Utilities Guide.

Executing The SCD for Loading Key Dimensions using Batch Framework

The batch for Key Dimension Loading into final Dimension tables can be executed by executing the seeded batch <Infodom>_POP_KEY_DIMENSION_SCD.

Improving SCD Performance

SCD performance can be improved by providing hints and session alter statements.

This requires the presence of the following four columns in SYS_TBL_MASTER:

- merge_hint
- select_hint
- session_enable_statement
- session_disable_statement

These columns are present in the OFSAAI versions 7.3.2.4.0 and higher. If these have to be used in OFSAAI versions 7.3.2.2.0 or 7.3.2.3.0 and higher, execute the following SQL queries:

```
ALTER TABLE SYS_TBL_MASTER ADD MERGE_HINT VARCHAR2 (255)
/
ALTER TABLE SYS_TBL_MASTER ADD SELECT_HINT VARCHAR2 (255)
/
ALTER TABLE SYS_TBL_MASTER ADD SESSION_ENABLE_STATEMENT VARCHAR2 (255)
/
ALTER TABLE SYS_TBL_MASTER ADD SESSION_DISABLE_STATEMENT VARCHAR2 (255)
/
```

During upgrade to OFSAAI 7.3.2.4.0, ensure to backup SYS_TBL_MASTER table and to drop the preceding four columns, if these scripts are executed in any of the OFSAAI versions prior to 7.3.2.4.0. Otherwise, an upgrade to OFSAAI 7.3.2.4.0 may throw an error, since the columns are existing.

- For improving performance, hints for the MERGE query which is generated internally by the SCD can be provided under MERGE_HINT. Session alters could be mentioned under SESSION_ENABLE_STATEMENT and SESSION_DISABLE_STATEMENT columns.
- SESSION_ENABLE_STATEMENTs will be executed before the MERGE in the SCD and SESSION_DISABLE_STATEMENTs will be executed after the SCD MERGE.
- Since all the tasks under the SCD batch for DIM_ACCOUNT works on the same target, the SESSION_DISABLE_STATEMENTs in SYS_TBL_MASTER cannot be provided when tasks are executed. In this case, there can be a separate SQL file to contain all the SESSION_DISABLE_STATEMENTs to be executed once after all the tasks in the SCD are done. The SESSION_DISABLE_STATEMENT will hold a null in SYS_TBL_MASTER table.
- SESSION_ENABLE_STATEMENTs are required to be mentioned only for the first task in the batch. Here the target is the same for all the tasks under a batch. In case any of the tasks are to be executed separately, then the SESSION_ENABLE_STATEMENTs should be mentioned for any one of the tasks which is included in the batch for the execution.

Example

MERGE_HINT and SESSION_ENABLE_STATEMENT in SYS_TBL_MASTER

Table 18. Merge Hint and Session Enable Statement

Table Name	Stage Table Name	Merge Hint	Session Enable Statement
DIM_ACCOUNT	STG_LOAN_CONTRACTS_V	/*+parallel (DIM_ACCOUNT,10) */	"alter session enable parallel dml query", "alter table DIM_ACCOUNT nologging parallel 10"

- Execute the SQL file with all the SESSION_DISABLE_STATEMENTS, after the successful completion of the SCD batch.

Reclassification of Standard Dimensions

This chapter provides information about Standard Dimension Tables in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes these topics:

- Overview of Reclassification of Standard Dimensions
- Overview of Reclassification of Standard Dimensions Population
- Dimension Data Expectations through SCD
- Overview of Mappers for Reclassification of Standard Dimensions
- Maintenance of Mapper for Reclassification of Standard Dimensions
- Loading Mapper Maintenance through Backend
- Usage of Mapper Tables in Data Flow and Reports

Overview of Reclassification of Standard Dimensions

There are certain Standard Dimensions in Financial Services Data Foundation, which are pre-populated with standard set of values. These values are used by downstream applications for various reporting requirements. There are equivalent customer specific dimension tables which are populated using Slowly Changing Dimension (SCD) process. It is required to reclassify these user specific values to standard specific values as the reporting expects these standard set of values. The reclassification is done using out of the box Mapper Definitions under Mapper Maintenance screen.

Overview of Reclassification of Standard Dimensions Population

These are the out of the box User Specific Dimensions to Standard Dimensions reclassification available in Financial Services Data Foundation:

Table 19. Reclassification of Standard Dimensions

User Specific Dimension		Standard Dimension	
DIM_BALANCE_CATEGORY	Balance Category	DIM_STD_BALANCE_CATEGORY	Standard Balance Category
DIM_CREDIT_LINE_PURPOSE	Credit Line Purpose	DIM_STD_CREDIT_LINE_PURPOSE	Standard Credit Line Purpose
DIM_CREDIT_LINE_TYPE	Credit Line Type	DIM_STD_CREDIT_LINE_TYPE	Standard Credit Line Type
DIM_IRC	Interest Rate Curve	DIM_STANDARD_IRC	Standard Interest Rate Curve

User Specific Dimension		Standard Dimension	
DIM_LOB	Line of Business	DIM_STANDARD_LOB	Standard Line of Business
DIM_MITIGANT_TYPE	Mitigant Type	DIM_STD_MITIGANT_TYPE	Standard Mitigant Type
DIM_PARTY_TYPE	Party Type	DIM_STANDARD_PARTY_TYPE	Standard Party Type
DIM_PRODUCT	Product	DIM_STANDARD_PRODUCT_TYPE	Standard Product Type
DIM_GL_ACCOUNT	General Ledger	DIM_STD_GL_TYPE	Standard General Ledger Type
DIM_VEHICLE_TYPE	Vehicle Type	DIM_STD_VEHICLE_TYPE	Standard Vehicle Type
DIM_WRITE_OFF_REASONS	Write Off Reasons	DIM_STD_WRITE_OFF_REASONS	Standard Write Off Reasons
DIM_RECOVERY_TYPE	Recovery Type	DIM_STD_RECOVERY_TYPE	Standard Recovery Type

Dimension Data Expectations through SCD

By default, all standard dimensions will be pre-populated with seeded data. It is mandatory to have data in user specific dimensions and then maintain the reclassifications. Therefore, you must execute the SCDs and then map the reclassification codes under Mapper Maintenance.

Overview of Mapper for Reclassification of Standard Dimensions

These are out of the box mappers that are available in FSDF for the standard dimension reclassifications:

- MAP_PROD_CODE_STD_PROD_TYPE - Mapper for Product Code to Standard Product Code
- MAP_PARTY_TYP_STD_PARTY_TYP - Mapper for Party Type Code to Standard Party Type Code
- MAP_CRDLN_TYP_STD_CRDLN_TYP - Mapper for Credit Line Type to Standard Credit Line Type
- MAP_DIM_IRC_STD_IRC - Mapper for Interest Rate Code to Standard Interest Rate Code
- MAP_DIM_LOB_STD_LOB - Mapper for Line of Business Code to Standard Line of Business Code
- MAP_BAL_CAT_STD_BAL_CAT - Mapper for Balance Category to Standard Balance Category
- MAP_CRDLN_PUR_STD_CRDLN_PUR - Mapper for Credit Line Purpose to Standard Credit Line Purpose
- MAP_MITG_TYP_STD_MITGN_TYP - Mapper for Mitigant Type to Standard Mitigant Type
- MAP_CREDIT_SCR_MDL_REG_MDL - Mapper for Credit Score Model To Reg Credit Score Model
- MAP_DIM_GL_ACCT_STD_GL_TYPE - Mapper for General Ledger Account to Standard General Ledger Account Type

- MAP_GL_CODE_REP_LINE - Mapper for GL Code to Repline Code
- MAP_RECVR_TYP_STD_RECVR_TYP - Mapper for Common Recovery Type to Standard Recovery Type
- MAP_VEHCL_TYP_STD_VEHCL_TYP - Mapper for Vehicle Type to Standard Vehicle Type
- MAP_WRTOFF_STD_WRTOFF_REASN - Mapper for Write Off Reasons to Standard Write Off Reasons

Maintenance of Mapper for Reclassification of Standard Dimensions

Mapper can be maintained under OFSAAI.

1. Navigate to **OFSAAI > Financial Services Data Foundation > Unified Analytical Metadata > Business Metedata Management > Map Maintenance**.

Map Maintenance

Information Domain: FSDFINFO Segment: FSDFSEG

Default Security Map: Not Set

Name	Version	Description	Dynamic	Inherit member	Map type
1514359600480	1	Mapper for Balance Category to Standard Balance Category	Yes	Yes	Data filt
1524045220417	1	Mapper for Common Recovery Type to Standard Recovery Type	Yes	Yes	Data filt
1511528494678	1	Mapper for Credit Line Purpose to Standard Credit Line Purpose	Yes	Yes	Data filt
1497513837744	1	Mapper for Credit Score Model To Reg Credit Score Model	Yes	Yes	Data filt
1523447233065	1	Mapper for General Ledger Account to Standard General Ledger Account Type	Yes	Yes	Data filt
1494610765133	1	Mapper for GL Code to Repline Code	Yes	Yes	Data filt
1511442223838	1	Mapper for Interest Rate Code to Standard Interest Rate Code	Yes	Yes	Data filt
1511442482993	1	Mapper for Line of Business Code to Standard Line of Business Code	Yes	Yes	Data filt
1514359498413	1	Mapper for Mitigant Type to Standard Mitigant Type	Yes	Yes	Data filt
1511441945154	1	Mapper for Party Type Code to Standard Party Type Code	Yes	Yes	Data filt
1511441227779	1	Mapper for Product Code to Standard Product Code	Yes	Yes	Data filt
1524044256132	1	Mapper for Vehicle Type to Standard Vehicle Type	Yes	Yes	Data filt
1524044617123	1	Mapper for Write Off Reasons to Standard Write Off Reasons	Yes	Yes	Data filt

Page 1 of 1 (1-13 of 13 items)

- For illustration, we have selected **Mapper for Mitigant Type to Standard Mitigant Type**. Click the **Mapper Maintenance** icon.

Map Maintenance

Information Domain: FSDFINFO Segment: FSDFSEG

Default Security Map: Not Set

Mapper Maintenance

<input type="checkbox"/>	Name	Version	Description	Dynamic	Inherit member	Map type	Database View name
<input type="checkbox"/>	1514359600480	1	Mapper for Balance Category to Standard Balance Category	Yes	Yes	Data filter	MAP_BAL_CAT_STD_BAL_CAT
<input type="checkbox"/>	1524045220417	1	Mapper for Common Recovery Type to Standard Recovery Type	Yes	Yes	Data filter	MAP_RECVR_TYP_STD_RECVR_TYP
<input type="checkbox"/>	1511528494678	1	Mapper for Credit Line Purpose to Standard Credit Line Purpose	Yes	Yes	Data filter	MAP_CRDLN_PUR_STD_CRDLN_PUR
<input type="checkbox"/>	1497513837744	1	Mapper for Credit Score Model To Reg Credit Score Model	Yes	Yes	Data filter	MAP_CREDIT_SCR_MDL_REG_MDL
<input type="checkbox"/>	1523447233065	1	Mapper for General Ledger Account to Standard General Ledger Account Type	Yes	Yes	Data filter	MAP_DIM_GL_ACCT_STD_GL_TYPE
<input type="checkbox"/>	1494610765133	1	Mapper for GL Code to Repline Code	Yes	Yes	Data filter	MAP_GL_CODE_REP_LINE
<input type="checkbox"/>	1511442223838	1	Mapper for Interest Rate Code to Standard Interest Rate Code	Yes	Yes	Data filter	MAP_DIM_IRC_STD_IRC
<input type="checkbox"/>	1511442482993	1	Mapper for Line of Business Code to Standard Line of Business Code	Yes	Yes	Data filter	MAP_DIM_LOB_STD_LOB
<input checked="" type="checkbox"/>	1514359498413	1	Mapper for Mitigant Type to Standard Mitigant Type	Yes	Yes	Data filter	MAP_MITG_TYP_STD_MITGN_TYP
<input type="checkbox"/>	1511441945154	1	Mapper for Party Type Code to Standard Party Type Code	Yes	Yes	Data filter	MAP_PARTY_TYP_STD_PARTY_TYP
<input type="checkbox"/>	1511441227779	1	Mapper for Product Code to Standard Product Code	Yes	Yes	Data filter	MAP_PROD_CODE_STD_PROD_TYPE
<input type="checkbox"/>	1524044256132	1	Mapper for Vehicle Type to Standard Vehicle Type	Yes	Yes	Data filter	MAP_VEHCL_TYP_STD_VEHCL_TYP
<input type="checkbox"/>	1524044617123	1	Mapper for Write Off Reasons to Standard Write Off Reasons	Yes	Yes	Data filter	MAP_WRTOFF_STD_WRTOFF_REASN

Page 1 of 1 (1-13 of 13 items) Records Per Page 12

Copyright © 1993, 2018 Oracle and/or its affiliates. All rights reserved.

- FSDF Maps OTH and MSG out-of-the-box for this mapper. The remaining mappings can be maintained by the user according to user specific values.

Mapper Maintenance - Search

Map - Mapper for Mitigant Type to Standard Mitigant Type - 1514359498413 - 1

Hier - Map Common Mitigant Type: [] Hier - Map Common Standard Mitigant Type: []

Excluded: []

Member combinations(2)

<input type="checkbox"/>	Hier - Map Common Mitigant Type	Macro	Hier - Map Common Standard Mitigant Type	Macro	Excluded
<input type="checkbox"/>	MSG -	Self & Desc	MSG - Missing	Self & Desc	N
<input type="checkbox"/>	OTH -	Self & Desc	OTH - Others	Self & Desc	N

Mapped members(2)

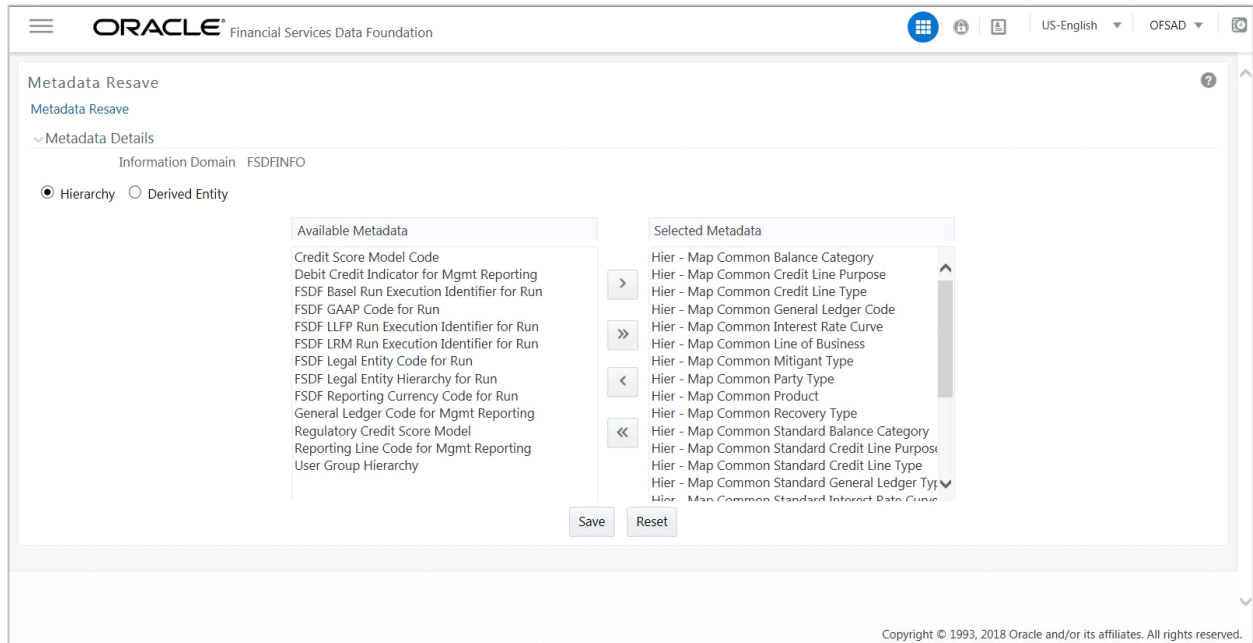
Hier - Map Common Mitigant Type	Hier - Map Common Standard Mitigant Type
MSG -	MSG - Missing
OTH -	OTH - Others

Close

© Oracle. All rights reserved.

Prerequisites for Mapper Maintenance

1. Load all the required user specific dimensions using SCD.
2. Resave these hierarchies:
 - HCMDF001 - Hier - Map Common Product
 - HCMDF002 - Hier - Map Common Standard Product Type
 - HCMDF003 - Hier - Map Common Party Type
 - HCMDF004 - Hier - Map Common Standard Party Type
 - HCMDF005 - Hier - Map Common Interest Rate Curve
 - HCMDF006 - Hier - Map Common Standard Interest Rate Curve
 - HCMDF007 - Hier - Map Common Line of Business
 - HCMDF008 - Hier - Map Common Standard Line of Business
 - HCMDF009 - Hier - Map Common Credit Line Type
 - HCMDF010 - Hier - Map Common Standard Credit Line Type
 - HCMDF011 - Hier - Map Common Credit Line Purpose
 - HCMDF012 - Hier - Map Common Standard Credit Line Purpose
 - HCMDF013 - Hier - Map Common Mitigant Type
 - HCMDF014 - Hier - Map Common Standard Mitigant Type
 - HCMDF015 - Hier - Map Common Balance Category
 - HCMDF016 - Hier - Map Common Standard Balance Category
 - HCMDF017 - Hier - Map Common General Ledger Code
 - HCMDF018 - Hier - Map Common Standard General Ledger Type
 - HCMDF019 - Hier - Map Common Vehicle Type
 - HCMDF020 - Hier - Map Common Standard Vehicle Type
 - HCMDF021 - Hier - Map Common Write Off Reasons
 - HCMDF022 - Hier - Map Common Standard Write Off Reasons
 - HCMDF023 - Hier - Map Common Recovery Type
 - HCMDF024 - Hier - Map Common Standard Recovery Type



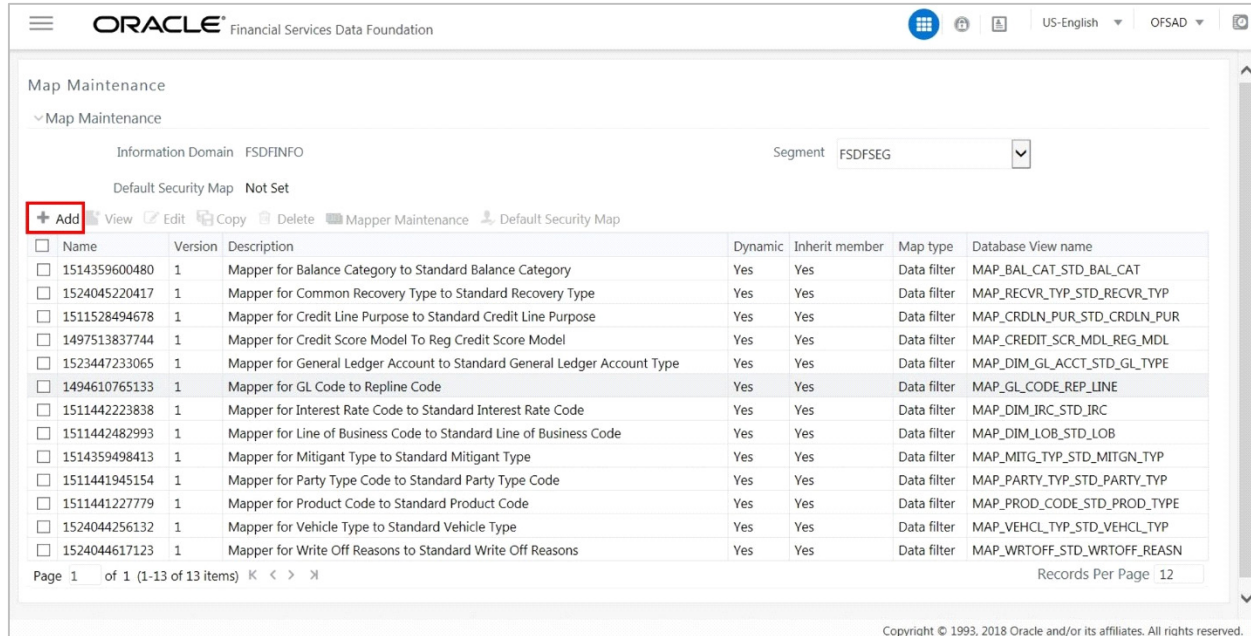
Possible Mapping Combinations

One Standard Dimension table in source can be mapped only to one Standard Dimension table. One to Many or Many to Many mapping leads to error in T2T as the records are duplicated. From the illustration, the possible combinations for Mitigant Type to Standard Mitigant Type mapping are One to One and Many to One mappings.

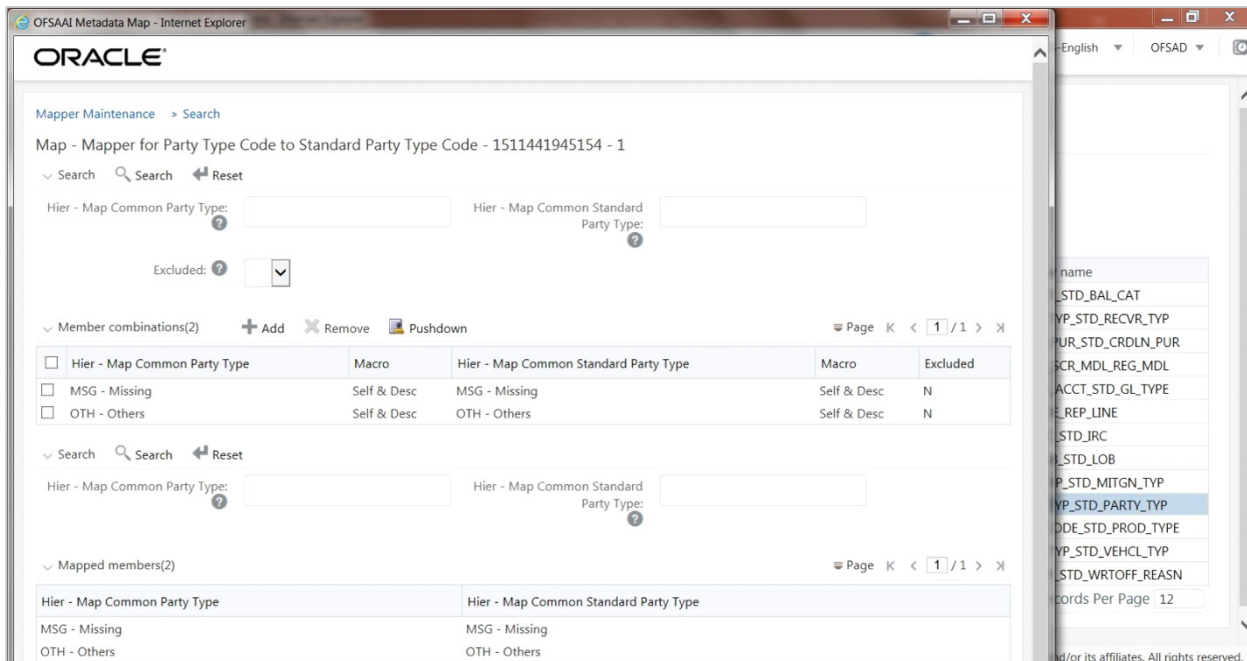
- **One to One Mapping:** You can map one Mitigant Type data model to one Standard Mitigant Type data model using the Mapper Maintenance screen. Here, you must select one value in Mitigant Type data model and one value in Standard Mitigant Type data model.
- **Many to One Mapping:** You can map many values in Mitigant Type data model to one value in Standard Mitigant Type data model using the Mapper Maintenance screen.

To conduct One to One or Many to One mapping:

1. Navigate to **OFSAAI > Financial Services Data Foundation > Unified Analytical Metadata > Business Metadata Management > Map Maintenance**.

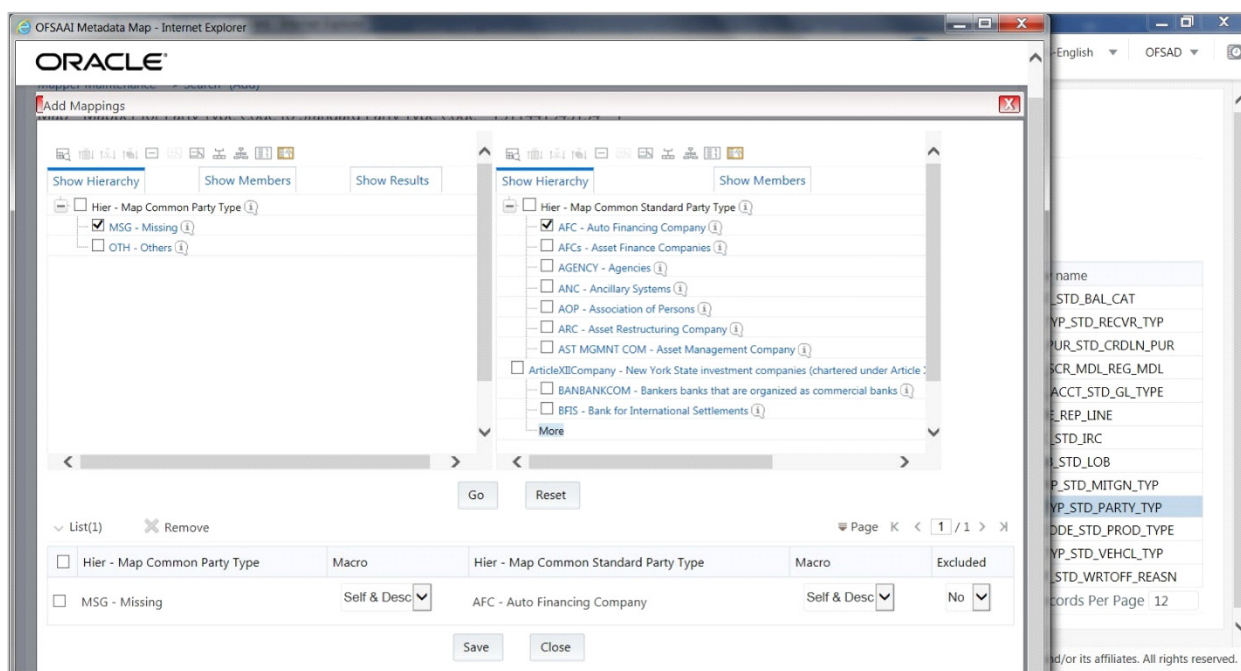


2. Click **Create new Map** icon to create a new map or select an existing Map. For illustration, **Mapper for Party Type Code to Standard Party Type Code** value is selected. Click **Mapper Maintenance** icon.
3. The **Mapper Maintenance** window opens (In this illustration, the **Map - Mapper for Party Type Code to Standard Party Type Code** window opens). To conduct One to One or Many to One mapping, in the Member Combinations section, click **Add**.



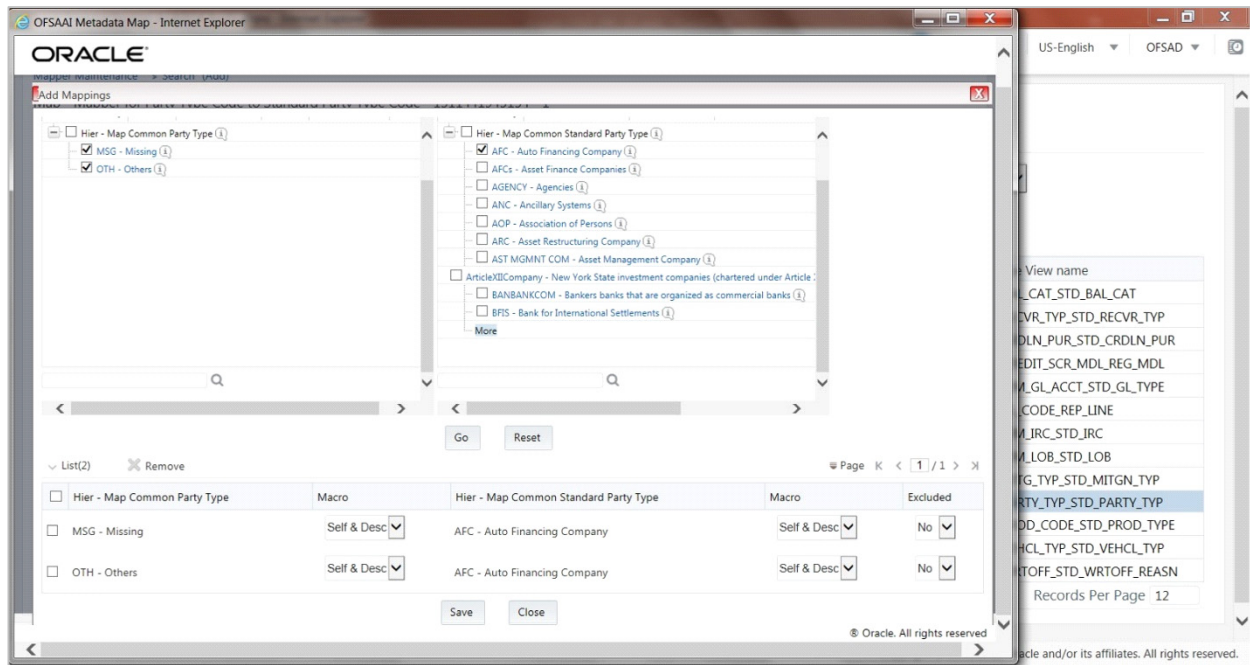
4. The **Add Mappings** pop-up window opens. In this illustration:

- To map One to One, select one value in the Hier - Map Common Mitigant Type data model and one value in the Hier - Map Common Standard Mitigant Type data model, and click **Go**. Repeat this step for each One to One data model mapping, and then click **Save**. In this illustration, **MSG - Missing** is mapped to **AFC - Auto Financing Company**.



- To map Many to One, select multiple (two in this illustration) values in the Hier - Map Common Mitigant Type data model and one value in the Hier - Map Common Standard Mitigant Type data model, and then

click **Go**. Click **Save**. In this illustration, **MSG-Missing** and **OTH-Others** are mapped to **AFC-Auto Financing Company**.



5. An acknowledgment is displayed: *Confirm Save?* To confirm saving data, click **Yes**. In the **Mapper Maintenance** window, in the Mapped combinations and the Mapped members sections, you can see the newly conducted mapping.

Loading Mapper Maintenance through Backend

Load each Physical table in Atomic Schema with V_MAP_ID as mentioned against each mapper, V_MEMBER_1 => Customer Specific Value Dimension's Member Code, V_MEMBER_2 => Standard Dimension's Member Code.

This is the list of Mapper Physical Tables and required details:

PHYSICAL TABLE	V_MAP_ID
MAP_MITG_TYP_STD_MITGN_TYP	1514359498413
MAP_DIM_IRC_STD_IRC	1511442223838
MAP_PROD_CODE_STD_PROD_TYPE	1511441227779
MAP_DIM_LOB_STD_LOB	1511442482993
MAP_CRDLN_PUR_STD_CRDLN_PUR	1511528494678
MAP_PARTY_TYP_STD_PARTY_TYP	1511441945154
MAP_BAL_CAT_STD_BAL_CAT	1514359600480
MAP_CRDLN_TYP_STD_CRDLN_TYP	1511527713328
MAP_CREDIT_SCR_MDL_REG_MDL	1497513837744
MAP_DIM_GL_ACCT_STD_GL_TYPE	1523447233065
MAP_GL_CODE_REP_LINE	1494610765133
MAP_RECVR_TYP_STD_RECVR_TYP	1524045220417
MAP_VEHCL_TYP_STD_VEHCL_TYP	1524044256132
MAP_WRTOFF_STD_WRTOFF_REASN	1524044617123

Usage of Mapper Tables in Data Flow and Reports

The mapper maintenance output is always physically stored in underlying tables. These tables are registered in OFSAA as an object. Therefore, these tables can be used, without any restrictions, in any of the metadata that requires reclassification. Financial Services Data Foundation Data Flows (T2Ts) make use of this information to populate the Standard Dimension Surrogate Keys of Results area tables.

Introduction

The reporting bank may be a part of a financial group that has multiple legal entities such as parent or child entities (subsidiaries) under its name. User can select the entity for which processing is to happen, and whether a ‘Solo’ or ‘Consolidation’ execution is to be done. Select these options using the Run Execution screen, but if it is executed using RRF execution then these options have to setup using the rule ‘Capital Consolidation Level Selection’ in the process ‘CAPITAL_CONSOLIDATION’.

CAPITAL_CONSOLIDATION is the first process to be added in all the runs defined through Run Rule Framework except the ones for staging data population. Run Management screen selects this process by default.

Run Parameters Assignment:

Basel Accord mentions about different approaches for calculating RWA. Run Management framework in the product allows the reporting bank to define and execute a Run by selecting a combination of different Basel II approaches for RWA computation.

Run parameter Assignment is also part of Capital Consolidation process. The rule ‘Run Definition User Defined Run Param Assignment’ is used to assign the run parameters in case of a run executed through Run Rule Framework. But if the execution is through run management, the parameters are populated based on the run defined in the run definition screen.

Consolidation Procedures

Following listed are the Consolidation procedures:

- Combine like items of assets, liabilities, equity, income, expenses and cash flows of the parent with those of its subsidiaries
- Offset (eliminate) the carrying amount of the parent's investment in each subsidiary and the parent's portion of equity of each subsidiary (IFRS 3 Business Combinations explain how to account for any related goodwill)
- Eliminate or retain in full intragroup assets and liabilities, equity, income, expenses and cash flows relating to transactions between entities of the group (profits or losses resulting from intragroup transactions that are recognized in assets, such as inventory and fixed assets, are eliminated or retained in full). During Consolidation Run, a parameter INTRAFLAG is introduced for Intra Company Elimination at the setup level with YES and NO values. The Intra Company Elimination can be chosen by the customer to eliminate (YES) or skip the elimination (NO) of Intra Company Accounts.

Types of Consolidation

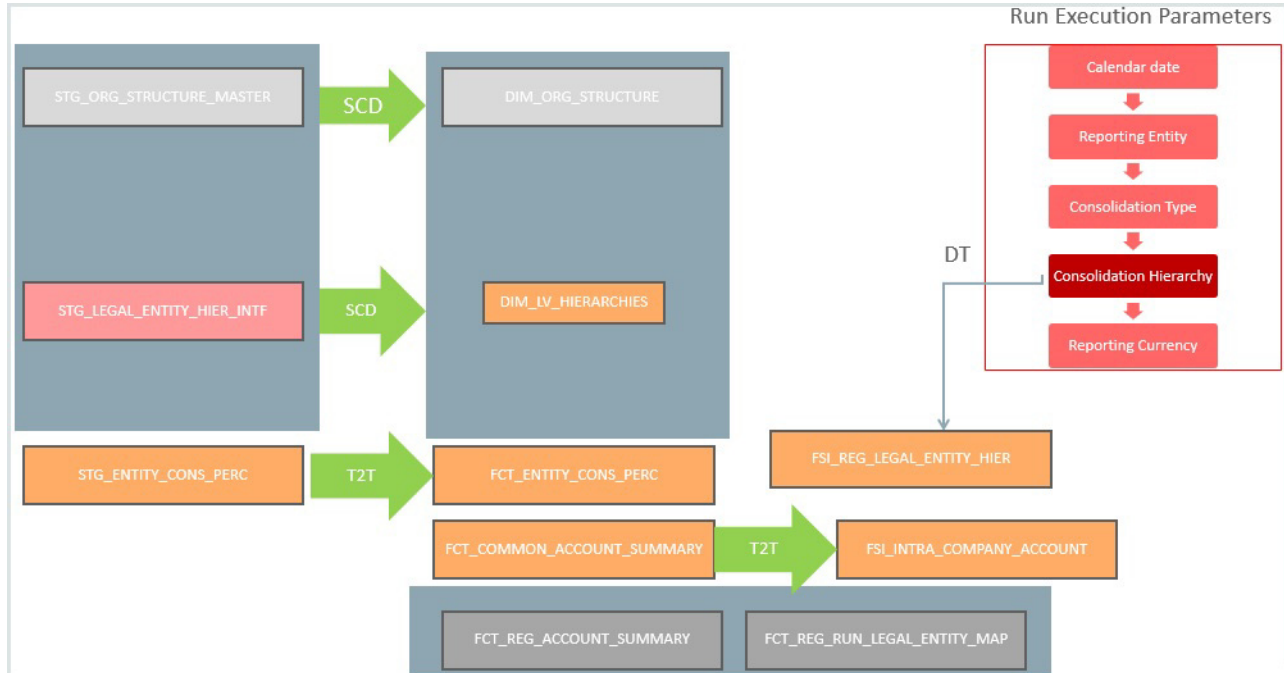
Following listed are the types of Consolidation:

- Simple Aggregation: As name suggests Simply aggregate across entities without any elimination
- Full Consolidation : Aggregate. Eliminate Intra group transactions.
- Proportionate Consolidation : Aggregate. Eliminate Intra group transactions and balances reflect consolidation percentage owned by parent in subsidiary (This can be configured as per the customer requirement).

Consolidation Activities

Scope Of consolidation is about list of Entities which participate in consolidation. Legal Entity Structure is looked through Organization Structure Dimension. This stores parent child relationship. This is stored only once. While moving the data, Legal Entity can move related entities to processing/reporting area. Legal structure being finalized once, this structure only stores one parent child relationship.

Data Flow



Executing Run through Run Management

Introduction

Starting from FSDF 8.0.3.1.0 release, we are packaging two out of the box Runs for data loading. Same can be executed through the Run Management screen. The following are the two runs that are packaged as part of Installer.

- Financial Services Data Foundation Sourced Run: This Run can be executed once per day for Data Movement from Staging Area to Results Area for Non-RUN SKEY tables.
- Financial Services Data Foundation Execution Run: This Run can be executed any number of times per day with each unique RUN SKEY for Data Movement in Run enabled tables.

Summary and Details Page

Upon initially navigating to **Run Management > Run Management**, a summary page is displayed showing all the defined Runs. By selecting a Run or by using search criteria, you can control the set of Runs that are displayed. This page displays the list of runs defined in the Run Rule (RRF) Framework except those with Immediate Execution Option **Yes** in the grid.

Navigation within the Summary Page

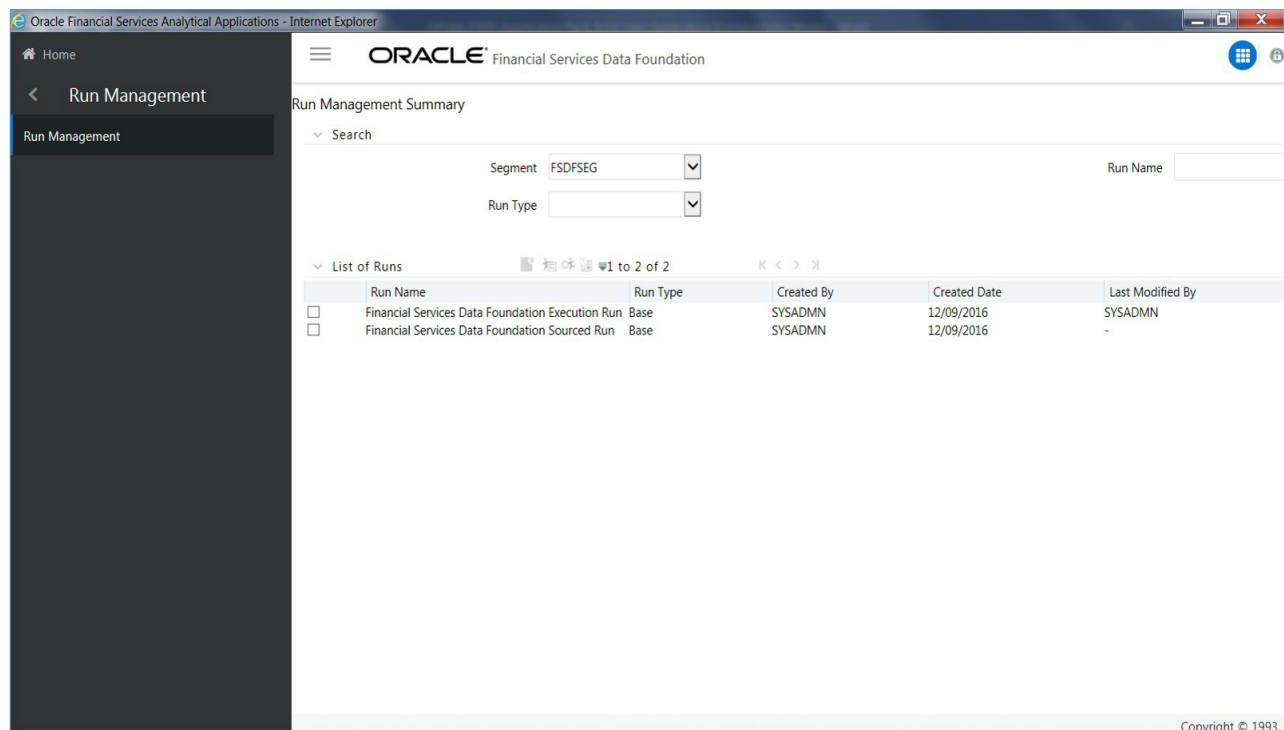
When you first navigate to the Run Management summary page, the Runs defined in the RRF are presented in a summary grid. The Run Management summary page has two sections:

- Search
- List of Runs

Search Section

Among other properties, each Run possesses a segment, a Run Name, and a Run Type. You may search on any of these properties in the Search section.

Figure 1. Search



List of Runs Section

The List of Runs section presents a grid containing all of the Runs that meet your search criteria. This summary grid offers several icons that allow you to perform different functions when a Run is selected.

To select a Run, click the check box in the first column of the grid.

- **View:** Selecting a single row out of the grid enables the **View** icon. Clicking the **View** icon allows you to view the detailed definition of a Run on a read-only basis. The **View** icon is only enabled when a single Run has been selected.
- **Run Default Parameters:** Selecting a single row out of the grid enables you to define the default parameters of a Run.
- **Run Execution Parameters:** Selecting a single row out of the grid enables you to define the execution parameters of a Run.
- **Run Execution Summary:** Selecting a single row out of the grid enables you to view the status of the Run executed in the Run Execution parameters window.

List of Runs Summary Grid

The following columns categorize each Run in the summary grid:

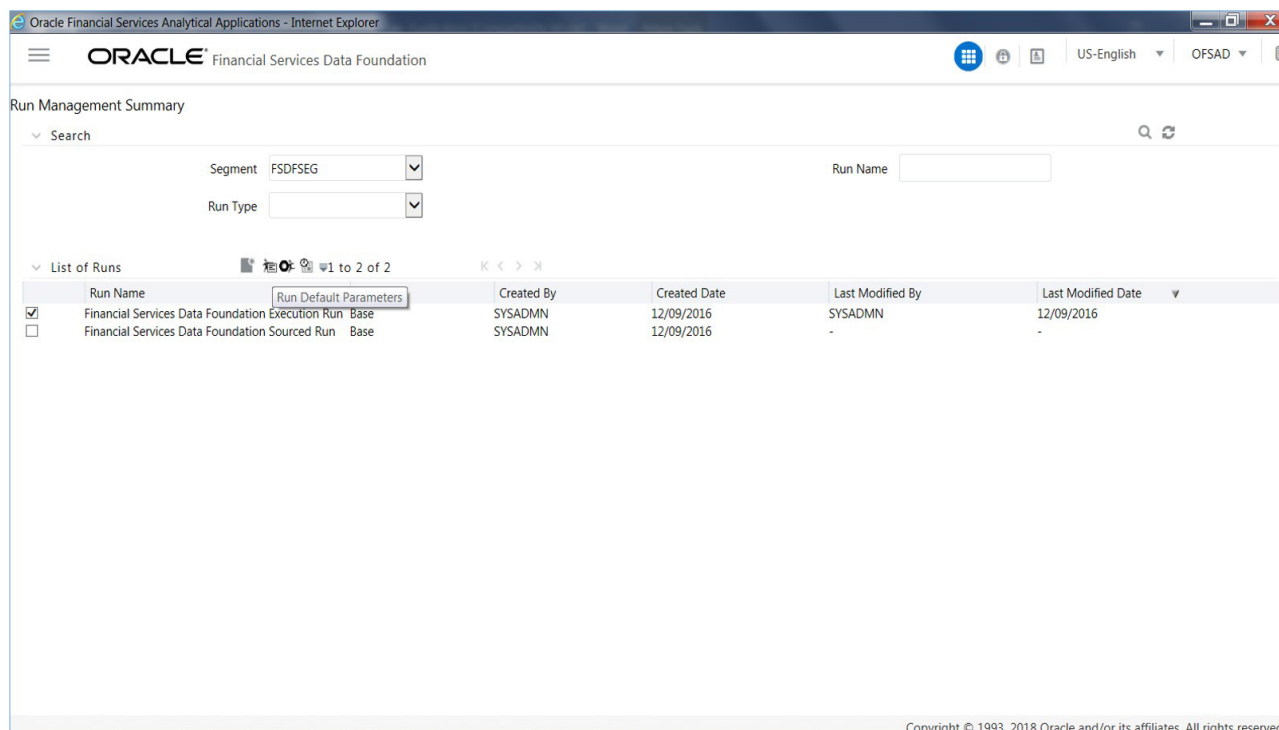
- **Run Name:** Displays the short name of the Run.
- **Run Type:** Displays the type of Run, Simulation or Baseline Run.
- **Created By:** Displays the name of the User who defined the Run.
- **Creation Date:** Displays the date on which the Run was created.

- Last Modified By: Displays the name of the user who has performed any modifications to the Original Run details.
- Last Modified Date: Displays the date on which the Original Run details were modified.

Navigation within Run Default Parameters Window

Click the Run Default Parameters icon on the navigation bar of the **Run Management Summary** Window to input the Run level parameters. The Run Parameters Window is displayed.

Figure 2. Run Management Summary



Note: To modify or view the parameters the Modify Run Parameters role should be mapped to that relevant user profile.

This window consists of two sections Run Details and Run Execution Parameters.

Run Details Section

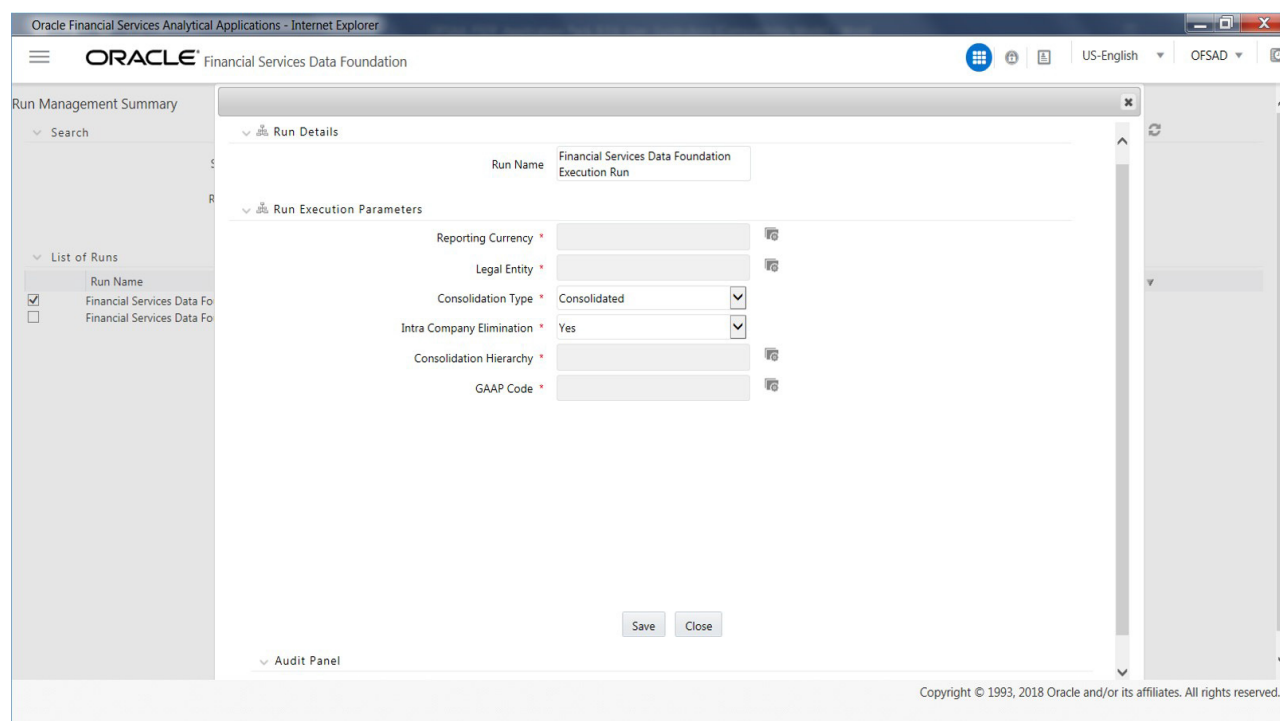
This section displays the name of the Run which is a read only value.

Run Execution Parameters Section

In this section you can update the following:

- Reporting Currency: Reporting Currency Code parameter is used for calculation of amounts in Reporting Currency during Data Population.
- Legal Entity: Legal Entity Code parameter is used for identifying the legal entity, which is used for the Run.

- **Consolidation Type:** Consolidation Type parameter is used for selecting legal entities on a solo or consolidation basis. In a solo run, only the selected legal entity will be used. In a consolidated run, along with the selected legal entity, all its child legal entities are also used.
- **Intra Company Elimination:** Intra Company Elimination can be chosen to eliminate (YES) or skip the elimination (NO) of Intra Company Accounts during a Consolidated Run.
- **Consolidation Hierarchy:** Legal Entity Hierarchy is used for selecting the required hierarchy for the consolidated run. This parameter is not required for solo run.



Before proceeding further, to ensure that you do not lose the updated data, click the **Save** button.

Note: To get the values for Reporting Currency parameter and Legal Entity parameter, you need to save the following hierarchies under Save Metadata screen:

- Legal Entity Code for Run (HFSDF001)
- Reporting Currency Code for Run (HFSDF002)
- Legal Entity Hierarchy for Run (HFSDF003)
- User Group Hierarchy (H_GROUP)

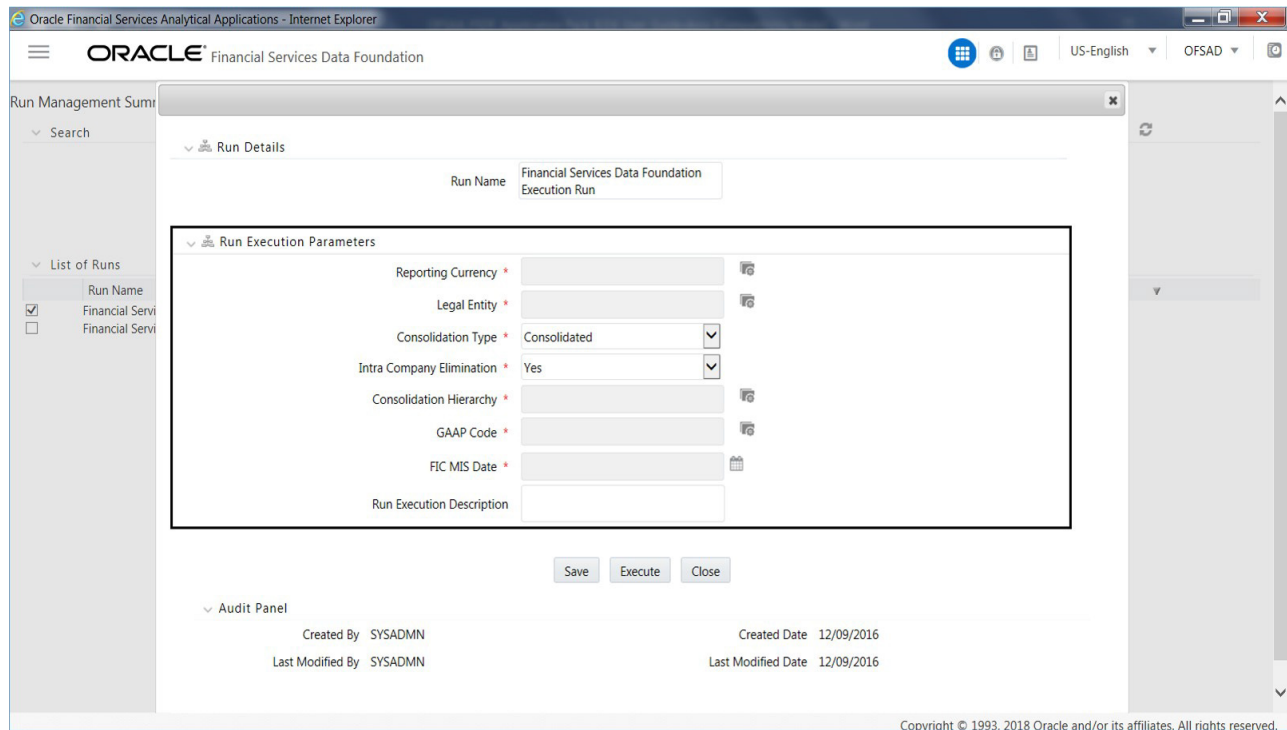
Note: For further details on Save Hierarchy, see [OFS Analytical Applications Infrastructure User Guide](#).

The values selected for reporting currency and Legal entity for the selected Run will be shown as the default selected value in the Run Execution Parameters screen.

Navigation within Run Execution Parameters Window

Click the **Run Execution Parameters** icon on the navigation bar of the **Run Management Summary** window. The **Run Execution Parameter** window allows you to enter and save the Run execution parameters.

Figure 3. Run Details

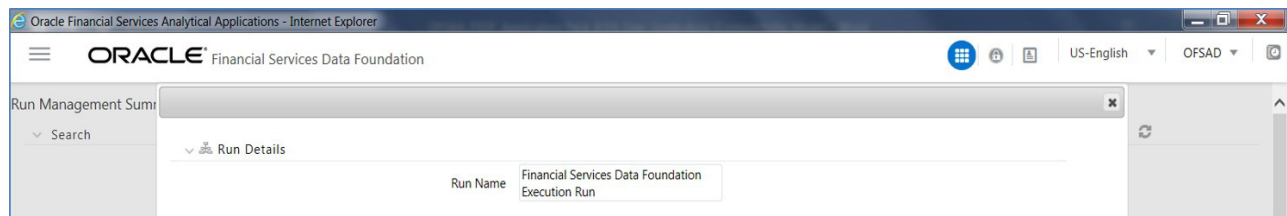


The **Run Execution Parameters** window consists of two sections **Run Details** and **Run Execution Parameters**.

Run Details Section

This section displays the name of the Run which is a read only value.

Figure 4. Run Details



Run Execution Parameters Section

The following Run execution parameters can be updated:

- Reporting Currency: Reporting Currency Code parameter is used for calculation of amounts in Reporting Currency during Data Population.

-
- **Legal Entity:** Legal Entity Code parameter is used for identifying the legal entity, which is used for the Run.
 - **Consolidation Type:** Consolidation Type parameter is used for selecting legal entities on a solo or consolidation basis. In a solo run, only the selected legal entity will be used. In a consolidated run, along with the selected legal entity, all its child legal entities are also used.
 - **Intra Company Elimination:** Intra Company Elimination can be chosen to eliminate (YES) or skip the elimination (NO) of Intra Company Accounts during a Consolidated Run.
 - **Consolidation Hierarchy:** Legal Entity Hierarchy is used for selecting the required hierarchy for the consolidated run. This parameter is not required for solo run.
 - **FIC MIS Date:** Enter the extraction date in this field.
 - **Run Execution Description:** Enter a longer description of the Run.

Note: To get the values for Reporting Currency parameter and Legal Entity parameter, you need to save the following hierarchies under Save Metadata screen:

- Legal Entity Code for Run (HFSDF001)
- Reporting Currency Code for Run (HFSDF002)

By clicking the **Save** button; a batch with the defined Run execution parameters is created. The batch created can be executed from the **Batch Execution** screen.

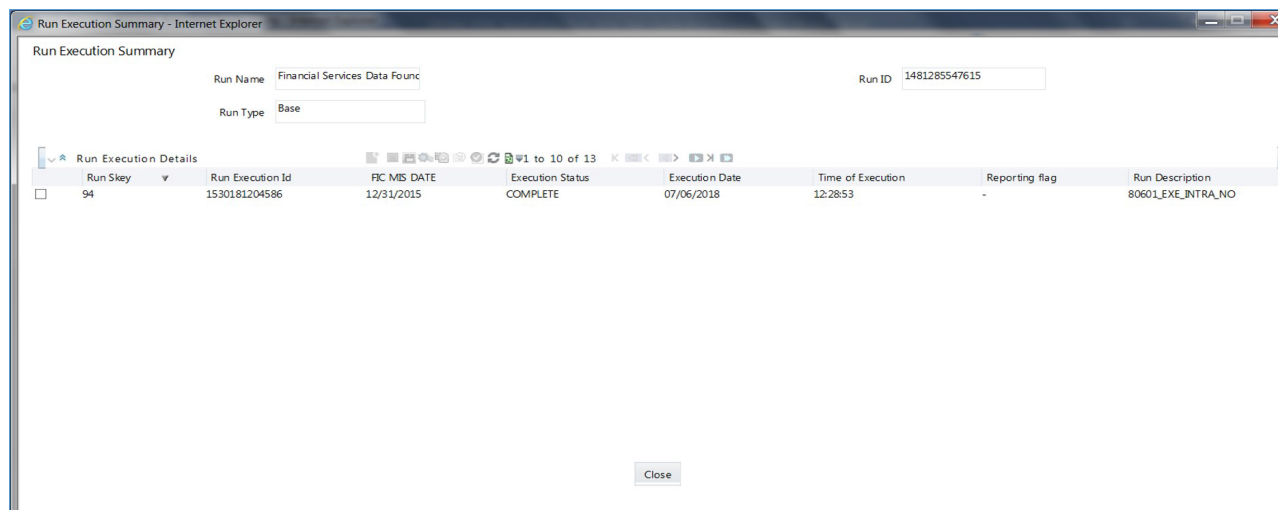
By clicking the **Execute** button, a batch with the defined Run execution parameters is created and executed immediately. Status of the executed run can be seen in **Batch Monitor** screen or **Run Execution Summary** page.

Note: For further details on Save Hierarchy and Batch Execution, refer to Oracle Financial Services Advanced Analytical Applications Infrastructure Application Pack 8.0.6.0.0 on [OHC](#). To execute a Run, the execute run role should be mapped to your user profile. Currently, the users mapped under FSDF Admin or FSDF Operator User Groups automatically have this role.

Navigation within Run Execution Summary Page

Select a Run from the Run Management Summary page and click the Run Execution Summary icon to display the Run Execution Summary page where the following sections are displayed.

Figure 5. Run Execution Summary



This section consists of the two sections Run Execution Summary and Run Execution Details.

Run Execution Summary Section

The **Run Execution Summary** displays the following details:

- **Run Name:** Displays the name of the Run.
- **Run Type:** Displays the type of Run, baseline or Simulation.
- **Run ID:** Displays the Run Execution ID.

Run Execution Details Section

The **Run Execution Details** section presents a grid containing all of the executions of Run and status of a particular execution of the Run. The menu bar in this grid offers several icons that allow you to perform different functions when a Run Execution is selected. To select a Run Execution, click the check box in the first column of the grid. More than one Run Execution can be selected at a time but this will cause some of the icons to become disabled.

- **Parameter details:** Click this icon to view the Run execution and Run default parameter details in read-only mode.
- **Copy:** Click **Copy** icon, to copy the parameters as defined in the **Run Execution Parameter** window to create a new batch.
- **Execute:** Click **Execute** icon to trigger the batch which has been created from the Run Execution Parameter window. The status of the triggered batch is displayed. In the Execution Summary page, multiple selections of the execution ids are available to trigger a batch.
- **Request Report Flag:** To request for a Report Flag, select a Run Execution ID in the **Run Execution Summary** page and click the **Request for Reporting Execution** icon. A dialog box will appear for you to input your comments. Click **Submit** and the status of this Run is displayed in the **Report Flag** section. Only a successful execution can be requested for reporting. For the selected Run and Execution date, there can be only one reporting flag.

-
- **Override Report Flag:** Any reporting execution can be overwritten with another execution. Select a successfully triggered batch in the **Run Execution Summary** page. The **Override Report Flag** icon is enabled, if an execution is already marked as a **Report Flag**. You can override the execution by updating your comments. This should be approved by the approver and the procedure is similar to the procedure detailed in the **Approve Report Flag** section.
 - **Approve Report Flag:** After submitting the Reporting Run in the earlier section, the **Approve Report Flag** icon is enabled. On clicking the icon a dialog box with the User Comments and Approver Comments is displayed. The Approver can update the comments in the Approver Comments field and then click the Approve or Reject button accordingly.

Run Execution Grid

The **Run Execution Details** displays the following details:

- Run Skey: Displays the Run skey of an individual execution.
- Run Execution Id: Displays the execution ID of the Run.
- FIC MIS DATE: Enter the extraction date in this field.
- Execution Status: Displays the status of the execution which is failed or complete.
- Execution Date: Displays the date when the Run was executed.
- Time of Execution: Displays the time when the Run was executed.

The Run Execution can be performed from the Command Line Interface with the following steps.

1. Navigate to the **\$FIC_HOME/ficdb/conf** directory.
2. Enter the details for the following fields in the **<RUN_NAME>.properties.template** file:

- **FSDF Execution Run - FSDF_EXE_RUN.properties.template**

and

- **FSDF Sourced Run - FSDF_SOURCED_RUN.properties.template**

Table 1. Details for <RUN_NAME>.properties.template file

Name	Description	Example
INFODOM	Specify name of Information Domain (INFODOM) of Run Definition	INFODOM=FSDFINF300
SEGMENT	Specify the Folder Code / Segment Code of Run Definition	SEGMENT=FSDFSEG
RUN_CODE	Specify the Run Code of the Run Definition	RUN_CODE=FSDF_EXE_RUN Or RUN_CODE=FSDF_SOURCED_RUN
USER_ID	Specify the OFSAAI User ID for the Run Execution	USER_ID=rrruser
HIER_RCY	Specify the Reporting Currency Hierarchy Code for the Run Execution	HIER_RCY=[HFSDF002].[USD] (default value)
HIER_LE	Specify the Legal Entity Code for the Run Execution	HIER_LE=WFCB
HIER_CONSOHIER	Specify the Consolidation Hierarchy for the Run Execution	HIER_CONSOHIER=[HFSDF003].[Default Org Structure Hierarchy] (default value)
LIST_CONSOTYPE	Specify the Consolidation Type for the Run Execution	LIST_CONSOTYPE=<value> List of values accepted are: 1. CONSL: Consolidated Run 2. SOLO: Solo Run (default value)
HIER_GAAP	Specify the GAAP Code Hierarchy for the Run Execution	HIER_GAAP=[HFSDF005].[USGAAP]

Table 1. Details for <RUN_NAME>.properties.template file

Name	Description	Example
HIER_BASELEXEID	This is an optional Parameter. Only if OFSAA - BASEL Run Execution is completed. Specify the BASEL Run Execution Identifier Hierarchy for the BASEL Processed Output to FSDF Results. Format. [Hierarchy].[V_RUN_ID].[N_RUN_SKEY]	HIER_BASELEXEID=[HFSD004].[1234567890].[100]
HIER_LRMEXEID	This is an optional Parameter. Only if OFSAA - LRM Run Execution is completed. Specify the LRM Run Execution Identifier Hierarchy for the LRM Processed Output to FSDF Results. Format. [Hierarchy].[V_RUN_ID].[N_RUN_SKEY]	HIER_LRMEXEID=[HFSD006].[1234567891].[101]
HIER_LLFPPEXEID	This is an optional Parameter. Only if OFSAA - LLFP Run Execution is completed. Specify the LLFP Run Execution Identifier Hierarchy for the LLFP Processed Output to FSDF Results. Format. [Hierarchy].[V_RUN_ID].[N_RUN_SKEY].	HIER_LLFPPEXEID=[HFSD007].[1234567892].[102]
RUN_EXE_COMMENTS	Specify the Comments for Run Execution	RUN_EXE_COMMENTS=FSDF Results Run
REQ_TYPE	Specify the Type of Execution for Run	REQ_TYPE=E List of Values Accepted are: 1. E: Create Batch and Execute (default value) 2. S: Create Batch

3. Navigate to **\$FIC_HOME/ficdb/bin** directory.
4. Rename the **.properties.template** file to **.properties**. Changed file names are:
 - **FSDF Execution Run - FSDF_EXE_RUN.properties**
 - and
 - **FSDF Sourced Run - FSDF_SOURCED_RUN.properties**
5. Execute the following **.sh** file by passing two arguments.
ExecuteRunManagement.sh <file name> <execution date>

Note: The execution date is in YYYYMMDD format. For example: **20171130**

For example: **ExecuteRunManagement.sh FSDF_EXE_RUN.properties 20171130**

6. After successfully invoking the Run, the following messages are displayed.

Note: Wait till the newly created batch execution completes, and returns the status code for success or failure. Therefore, further action can be taken using autosys batch scheduler.

With REQ_TYPE=S, the message “*Successfully created batch*” is displayed in the console.

```
/scratch/ofsadb/OFSAA/ficdb/bin>ExecuteRunManagement.sh RNUS_REG_RUN.properties 20171206
sDynamParam: HIER#LE-WFCB, HIER#CONSOLIER-[HFSDF003].[Default Org Structure Hierarchy], HIER#RCY-[HFSDF002].[USD], HIER#GAAP-[HFSDF005].[USGAAP], LIST#CONSOTYPE-SOLO
responseStatus: 200
responsePhrase: OK
response: successfully created batch
```

With REQ_TYPE=E, the message “*Batch Triggered Successfully*” is displayed in the console.

```
sDynamParam: HIER#LE-WFCB, HIER#CONSOLIER-[HFSDF003].[Default Org Structure Hierarchy], HIER#RCY-[HFSDF002].[USD], HIER#GAAP-[HFSDF005].[USGAAP], LIST#CONSOTYPE-SOLO
responseStatus: 200
responsePhrase: OK
response: Batch Triggered Successfully
```

Note: If any of the parameters in the **FSDF_EXE_RUN.properties** file is not entered correctly, the execution does not display the success message as above.

The Batch execution status can be monitored through the Batch Monitor link from the OFSAA Application Interface and the relevant logs are generated under the **\$FIC_HOME/ficdb/log** directory.

The screenshot shows the 'Batch Monitor' interface within the 'Financial Services Data Foundation > Operations > Batch Monitor' navigation path. The interface includes a search section with fields for 'Batch ID Like' (FSDFINF300), 'Batch Description Like' (WFCB_RUN_20171206_1), 'Module', 'Status', 'Start Date', and 'End Date'. Below the search section is a 'Batch Details' table with columns for 'Batch ID' and 'Batch Description', showing a single entry: FSDFINF300_1512552621486 with description WFCB_RUN_20171206_1. The 'Batch Run Details' section shows 'Information Date' as 20171206, 'Monitor Refresh Rate (seconds)' as 5, and 'Batch Run ID' as FSDFINF300_1512552621486_20171206_1. The 'Batch Status' section shows the 'Batch Run ID' and 'Batch Status' as FSDFINF300_1512552621486_20171206_1 and Ongoing, respectively.

This chapter provides information about Big Data processing supported in the Oracle Financial Services Data Foundation application.

Overview

FSDF supports Big Data processing on HDFS (Hadoop Distributed File System) using Hive as Query Engine. The AAI application components such as SCD, H2H, and DQs provide equivalent Hive support. For more information, see [OFS Analytical Applications Infrastructure User Guide](#).

FSDF Big Data Architecture

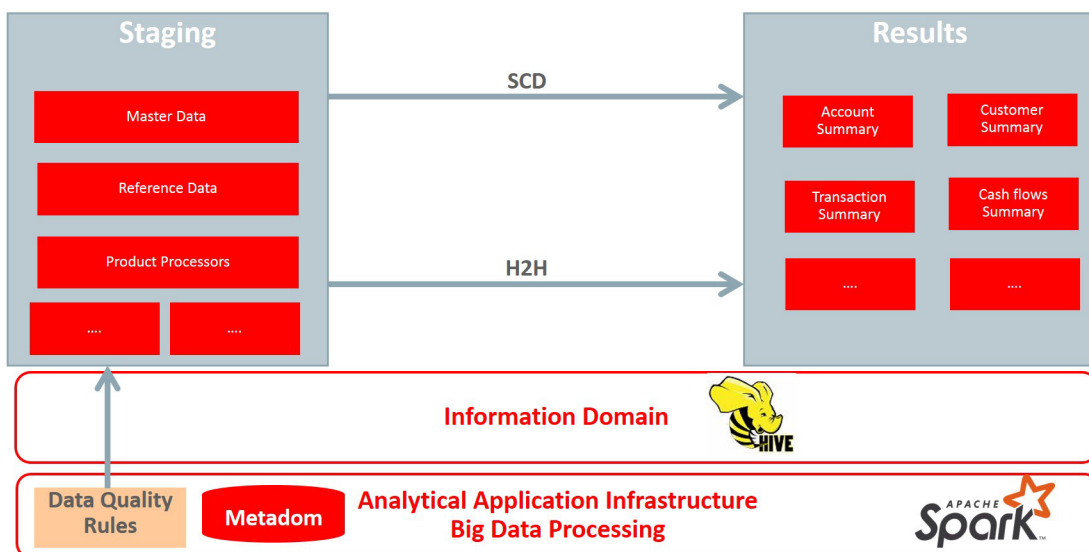
In Big Data processing, FSDF provides two types of deployment processes for Hive support. They are:

- Staging and Results on Hive
- Staging on Hive and Results on Oracle

Staging and Results on Hive

In the Staging and Results on Hive deployment process, the Staging, and Results model elements are a part of Hive known as Hive Datadom. The config schema definitions used by AAI and certain metadata definitions that were part of Atomic schema now resides in a RDBMS schema known as Metadom. Refer to the below architecture diagram for the representation of this deployment process.

Figure: FSDF Big Data Architecture - Staging and Results on Hive



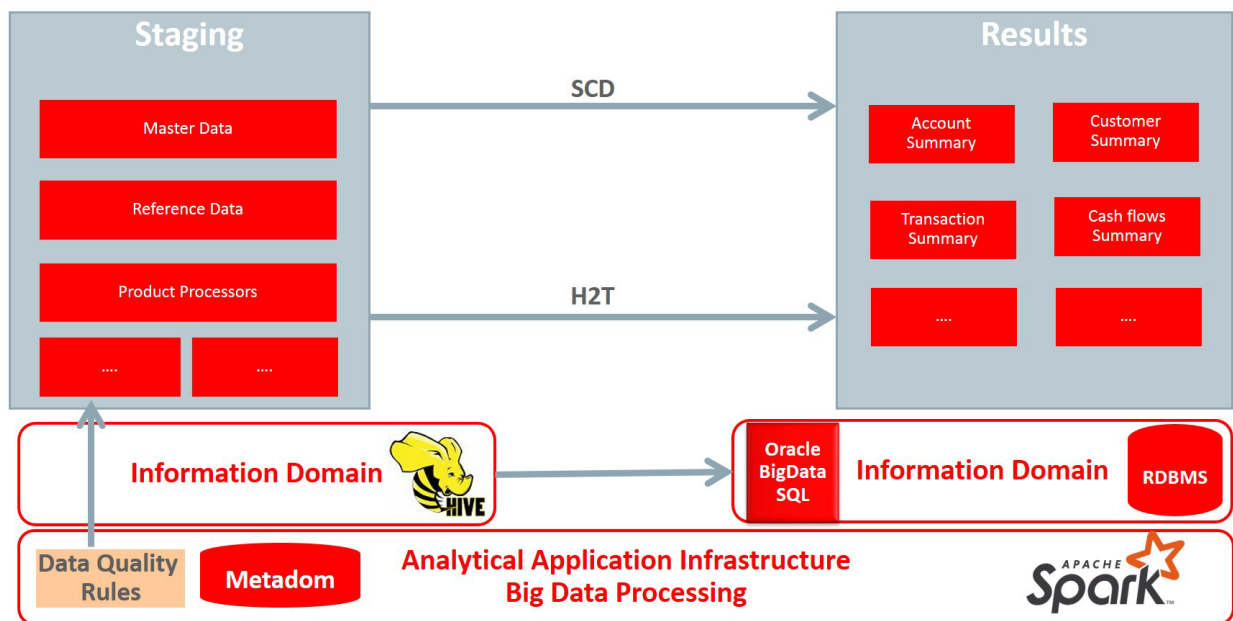
For the list of supported FSDF Hive Metadata definitions, refer to Run Chart, SCD Metadata, and Technical Metadata (Staging Source) documents at [My Oracle Support](#).

Note: Ensure that String values, which are null, must contain \n and then source to Hive stage tables.

Staging on Hive and Results on RDBMS

In the Staging on Hive and Results on RDBMS deployment process, the Staging occurs on Hive, known as Hive Datadom, and the Results model elements are a part of RDBMS, known as RDBMS Datadom. In this deployment process, an additional software is required, which is Oracle BigData SQL (For more information, see [Financial Services Data Foundation Installation Guide Release 8.0.6.0.0](#)). Oracle BigData SQL software resides on Hive and RDBMS Datadoms, and enables the user to create a link to Hive Stage tables as external tables in RDBMS. This software manages the data representation of the Hive tables in RDBMS Datadom and can be accessed as external table objects. As a result, enables the use of RDBMS infrastructure components of SCD, T2T, and DQs. For more information on External Tables, see the section Oracle's External Tables Utility in the [Financial Services Data Foundation Installation Guide Release 8.0.6.0.0](#). Refer to the below architecture diagram for the representation of this deployment process.

Figure: FSDF Big Data Architecture - Staging on Hive and Results on RDBMS



Staging and Results on Hive

This section provides information about the Staging and Results on Hive.

Modifications

Modifications in OFSDF Hive (Staging and Results on Hive deployment process) compared to OFSDF RDBMS are:

- Dim Dates Population

The Dim Dates functionality in Hive is similar to that of RDBMS. The difference exists in the date input format on the UI. In OFSDF Hive, the Date Range input format is, an executable textbox contains .sh file name followed by the FROM and TO dates.

For example:

```
load-dimdates-run.sh,20110101,20110105
```

Workarounds

The list of workarounds in OFSDF Hive (Staging and Results on Hive deployment process) is:

1. The process of data access, modification, and representation is different in HDFS compared to that of RDBMS. Therefore, the mechanism for Data Protection also varies.

Workaround:

- Data Redaction

The process of enabling Data Redaction for Hive can be performed using CDH Manager (Cloudera Distribution Hadoop Manager). For more information, see the section *Sensitive Data Redaction* in the [Cloudera Security Guide](#).

- Right to Forget

The Right to Forget feature enhancement is planned for future releases. However, prior to loading in the Hive instance, Anonymization of Party PII data can be done at the source. This process provides the flexibility to secure the PII information, and in addition, to drop the PII upon a request from the Party.

2. Mappers are supported using AM/HM screens in OFSDF (RDBMS). However, development of the underlying functionality for AM/HM feature in OFSDF Hive is planned for future releases. Therefore, the Mappers will be supported during future releases.

Workaround:

The Views defined on these Mappers in RDBMS are converted to Tables in Hive. Therefore, these Mappers must be loaded manually:

- MAP_BAL_CAT_STD_BAL_CAT
- MAP_CRDLN_PUR_STD_CRDLN_PUR
- MAP_CRDLN_TYP_STD_CRDLN_TYP
- MAP_CREDIT_SCR_MDL_REG_MDL_DDL
- MAP_CREDIT_SCR_MDL_REG_MDL_VWI
- MAP_DIM_IRC_STD_IRC
- MAP_DIM_LOB_STD_LOB
- MAP_GL_CODE_REP_LINE
- MAP_MITG_TYP_STD_MITGN_TYP
- MAP_PARTY_TYP_STD_PARTY_TYP
- MAP_PROD_CODE_STD_PROD_TYPE
- MAP_DIM_GL_ACCT_STD_GL_TYPE

- MAP_VEHCL_TYP_STD_VEHCL_TYP
- MAP_RECVR_TYP_STD_RECVR_TYP
- MAP_WRTOFF_STD_WRTOFF_REASN

3. Data is not populated in the target table FSI_INTRA_COMPANY_ACCOUNT.

Workaround:

FSI_REG_LEGAL_ENTITY_HIER load references POP_REG_LE_HIER DT, which is not supported in OFSDF Hive. FSI_REG_LEGAL_ENTITY_HIER must be loaded to make use of the T2T for FSI_REG_LEGAL_ENTITY_HIER because that T2T consists of an inner join on FSI_REG_LEGAL_ENTITY_HIER.

Similarly, Dimension tables for Unsupported SCDs must be loaded to make use of the T2Ts that reference them.

4. Hive does not completely support the exchange rates population functionality for cross currencies. In Hive, the exchange rates population for cross currencies cannot be derived using exchange rates values for a base currency.

Workaround:

Source the corresponding Exchange Rate value.

5. When performing Big Data installation for OFSDF Hive, the following error is logged in the file OFS_BFND_installation.log:

```
Error:ORA-00942: table or view does not exist
ORA-06512: at "BSYMETA806.FSI_CREATE_SEQUENCE", line 6
```

Workaround:

This error can be ignored.

List of Supported SCDs and T2Ts

This section provides the list of supported SCDs and T2Ts for OFSDF Hive (Staging and Results on Hive deployment process).

List of Supported SCDs

The SCDs for Hive used in Data Foundation solutions are listed in the Oracle Financial Services Data Foundation for Hive - SCD Metadata for Hive spreadsheet under [Technical Metadata for FSDF HIVE 8.0.6.0.0](#).

Run Enabled T2Ts

To execute SOURCED_RUN and EXE_RUN in Run Management, follow these steps:

1. Navigate to the **Run Management** section. Select the required Run (SOURCED_RUN or EXE_RUN) to execute.
2. Select all required values.
3. Save the batch.
4. Query DIM_RUN in Metadom. Copy the resultant entry and insert in the DIM_RUN table of Datadom.
5. In the **Batch Execution** screen, execute Run.

List of Supported T2Ts

The T2Ts for Hive used in Data Foundation solutions are listed in the Oracle Financial Services Data Foundation for Hive - Technical Metadata (Staging Source) spreadsheet under [Technical Metadata for FSDF HIVE 8.0.6.0.0](#).

List of Unsupported SCDs and T2Ts

This section provides the list of unsupported SCDs and T2Ts for OFSDF Hive (Staging and Results on Hive deployment process).

List of Unsupported SCDs

This is the list of unsupported SCDs with corresponding Map Reference Numbers, for OFSDF Hive (Staging and Results on Hive deployment process):

- SCD-126
- SCD-127
- SCD-128
- SCD-129
- SCD-131
- SCD-132
- SCD-195
- SCD-196
- SCD-205
- SCD-208
- SCD-267
- SCD-270
- SCD-332
- SCD-335
- SCD-381
- SCD-411
- SCD-419
- SCD-465
- SCD-467

Note: Hierarchies are supported using AM/HM screens in OFSDF (RDBMS). However, development of the underlying functionality for AM/HM feature in OFSDF Hive is planned for future releases. As a result, the tables REV_BIHIER and REV_LOCALE_HIER will be available in Hive, when the Hierarchies will be supported during future releases.

List of Unsupported T2Ts

This is the list of unsupported T2Ts for OFSDF Hive (Staging and Results on Hive deployment process):

-
- T2T_FCT_COMMON_CUSTOMER
 - T2T_FSI_EXCHANGE_RATES
 - T2T_FCT_ACCT_RECOVERY_DETAILS
 - T2T_FCT_ACCT_WRITE_OFF_DETAILS
 - T2T_FCT_CARDS_SUMMARY
 - T2T_FCT_CREDIT_PARTCPN_TRNCH_DETL
 - T2T_FCT_FIXED_ASSETS
 - T2T_FCT_FUND_CIS_COMPOSITION
 - T2T_FCT_HEDGE_PORTFL_SET_ACCT_MAP
 - T2T_FCT_LITIGATION_DETAILS
 - T2T_FCT_LOANS_SERVICED
 - T2T_FCT_MERCHANT_BANKING
 - T2T_FCT_PAYMENTS_SUMMARY
 - T2T_FCT_SPEND_OBLIGATIONS
 - T2T_FFSIS_STG_MUTUAL_FUNDS
 - T2T_FFSIS_STG_RETIREMENT_ACCOUNTS
 - T2T_FFSIS_STG_TD_CONTRACTS
 - T2T_FFSIS_STG_TRUSTS
 - T2T_FLAS_STG_LEASES_CONTRACTS
 - T2T_FLAS_STG_LOAN_CONTRACTS
 - T2T_FLAS_STG_OD_ACCOUNTS
 - T2T_FCT_FORECAST_REG_CAP_SUMMARY
 - T2T_FCT_MITIGANT_REG_CAPITAL
 - T2T_FCT_MR_CAPITAL_SUMMARY_FMRCC
 - T2T_FCT_MR_CAPITAL_SUMMARY_FMREQC
 - T2T_FCT_MR_CAPITAL_SUMMARY_FMRFRXC
 - T2T_FCT_MR_CAPITAL_SUMMARY_FMRIRC
 - T2T_FCT_MR_VAR_PORTFOLIO_SUMMARY
 - T2T_FCT_MR_VAR_SUMMARY
 - T2T_FCT_REG_ACCT_MITIGANT_MAPPING
 - T2T_FCT_REG_CAP_PARTY_GRP_MMBR_MAP
 - T2T_FCT_REG_CAP_PLCD_COLL_SUMMARY
 - T2T_FCT_REG_CAP_POOL_SUMMARY

-
- T2T_FCT_REG_CP_CAPITAL_SUMMARY
 - T2T_FCT_REG_LARGE_EXP_CP_LIMITS
 - T2T_FCT_REG_LE_CAPITAL_SUMMARY
 - T2T_FCT_REG_MARKET_RISK_EXPOSURES
 - T2T_FCT_REG_OR_CAPITAL_SUMMARY
 - T2T_FCT_REG_POOL_MITIGANT_MAP
 - T2T_FCT_REG_SEC_POOL_SUMMARY
 - T2T_FRCAS_FCT_NON_SEC_EXPOSURES
 - T2T_FRCAS_FCT_NON_SEC_EXPOSURES_CHILD
 - T2T_FRCAS_FCT_NON_SEC_EXPOSURES_PARENT
 - T2T_FRCAS_FCT_SEC_EXPOSURES
 - T2T_FRCAS_FCT_SEC_EXPOSURES_CHILD
 - T2T_FRCAS_FCT_SEC_EXPOSURES_PARENT
 - T2T_FRCCS_FCT_CCP_DETAILS
 - T2T_FCT_LRM_ACCOUNT_SUMMARY
 - T2T_FDB_STG_BORROWINGS
 - T2T_FDB_STG_CASA
 - T2T_FDB_STG_TD_CONTRACTS
 - T2T_FFSIS_STG_CASA
 - T2T_FFSIS_STG_CUSTODIAL_ACCOUNTS
 - T2T_FFSIS_STG_INVESTMENTS
 - T2T_FFSIS_STG_MANAGED_INV_ADV

Executing Run through Run Management for Hive

To load data in OFSDF Hive, use Solo or Consolidated run execution in Run Management. For detailed information, see the chapter *Executing Run through Run Management*.

This chapter provides information about Account Dimension Loading Process in the Oracle Financial Services Data Foundation application.

This chapter includes the following topics:

- Overview of Account Dimension
- Overview of Account Dimension Population
- All the above-mentioned Product Processor tables are mutually exclusive in terms of functionality. Therefore, V_ACCOUNT_NUMBER or V_CONTRACT_CODE (Primary Key of each Product Processor) must also have mutually exclusive values across all the 40 tables. The same account number cannot be part of multiple Product Processor tables.

Overview of Account Dimension

The SCD population in DIM_ACCOUNT table generates individual numeric Surrogate Keys for every account number with an additional leg key. Below are the columns that will be loaded during SCD population:

- V_ACCOUNT_NUMBER
- N_ACCT_SKEY
- N_RCV_LEG_ACCT_SKEY
- FIC_MIS_DATE
- V_PRODUCT_PROCESSOR_NAME
- V_ORIGINAL_ACCOUNT_NUMBER
- F_LATEST_RECORD_INDICATOR
- V_APPLN_NUMBER
- V_CCY_CODE

Overview of Account Dimension Population

SCD seeded definitions are provided for loading data into the target table Account Dimension (DIM_ACCOUNT):

Table 2. Table to Table Seeded Definitions

Map Reference No.	Source View Name
188	STG_ANNUITY_CONTRACTS_V
189	STG_BILLS_CONTRACTS_V
190	STG_BORROWING_V
191	STG_CARDS_V
192	STG_CASA_V

Table 2. Table to Table Seeded Definitions

Map Reference No.	Source View Name
193	STG_COMMITMENT_CONTRACTS_V
194	STG_CREDIT_DERIVATIVES_V
195	STG_FUTURES_V
196	STG_FX_CONTRACTS_V
197	STG_GUARANTEES_V
198	STG_INVESTMENTS_V
199	STG_LC_CONTRACTS_V
200	STG_LEASES_CONTRACTS_V
201	STG_LOAN_CONTRACTS_V
202	STG_MM_CONTRACTS_V
203	STG_MUTUAL_FUNDS_V
204	STG_OD_ACCOUNTS_V
205	STG_OPTION_CONTRACTS_V
206	STG_REPO_CONTRACTS_V
207	STG_RETIREMENT_ACCOUNTS_V
208	STG_SWAPS_CONTRACTS_V
209	STG_TD_CONTRACTS_V
210	STG_TRUSTS_V
211	STG_ISTISNA_V
212	STG_IJARAH_V
213	STG_MUDARABAH_V
214	STG_MURABAHAH_V
215	STG_MUSHARAKAH_V
216	STG_SALAM_V
217	STG_SUKUK_V
262	STG_BORROWING_COMMITMENTS_V
263	STG_CORRESPONDENT_ACCOUNT_V
264	STG_TRADING_ACCOUNT_V
341	STG_PAYMENT_SETTLEMENT_ACCT_V
349	STG_COMMODITIES_V
350	STG_CUSTODIAL_ACCOUNTS_V
351	STG_MANAGED_INV_ADV_V

Table 2. Table to Table Seeded Definitions

Map Reference No.	Source View Name
352	STG_MERCHANT_BANKING_V
353	STG_PREPAID_CARDS_V
354	STG_SPEND_OBLIGATIONS_V
400	STG_MERCHANT_CARDS_V
465	STG_FORWARDS_V
467	STG_ASSETS_SOLD_V

All the above-mentioned Product Processor tables are mutually exclusive in terms of functionality. Therefore, V_ACCOUNT_NUMBER or V_CONTRACT_CODE (Primary Key of each Product Processor) must also have mutually exclusive values across all the 40 tables. The same account number cannot be part of multiple Product Processor tables.

Executing the Account Dimension SCD

Batch FSDFINFO_DATA_FOUNDATION_SCD has been introduced with 40 tasks under it.

These 40 tasks represent the 40 SCD processes where different product processors would be the source and DIM_ACCOUNT would be the target. MAP_REF_NUMs 188 to 217, 262 to 264, 341, 349 to 354, 400, 465, 467 have been introduced into SYS_TBL_MASTER table, and subsequently into SYS_STG_JOIN_MASTER.

Depending on the requirement by an application, a task can be excluded or included from the batch execution.

- SCD execution occurs based on the GAAP code which is configured in SETUP_MASTER table. This has been introduced to tackle the scenario of multiple GAAP codes. Whether or not there exist multiple GAAP codes, SETUP_MASTER should be manually configured as follows:
- All the tasks can be executed in parallel. This might cause the N_RCV_LEG_ACCT_SKEY to have an incremental value as compared to N_ACCT_SKEY.

Note: By default, FSDF installer will seed the following entry into SETUP_MASTER.

Table 3. Configuration

V_COMPONENT_CODE	V_COMPONENT_DESC	V_COMPONENT_VALUE
DEFAULT_GAAP	DEFAULT_GAAP	USGAAP

For all other GAAP codes, we need to update SETUP_MASTER manually before running DIM_ACCOUNT SCD.

Handle Multiple GAAP Codes for Same Account Number for the Same MIS Date in SCD

When multiple GAAP codes exist for the same account number for the same MIS date, configure the SETUP_MASTER table manually as mentioned in the preceding section:

V_COMPONENT_VALUE will hold the GAAP code for which the SCD is to be executed.

If there are different GAAP codes for two distinct account numbers for the same MIS date, then the SCD has to be executed for each GAAP code by changing the V_COMPONENT_VALUE manually in setup_master table. The SETUP_MASTER table should have only one record WHERE V_COMPONENT_DESC = 'DEFAULT_GAAP'.

Handle Multiple Load Runs for Same Account Number for the Same MIS Date in SCD

When multiple Load Runs exist for the same account number for the same MIS date, ensure FSI_ACCOUNT_LOAD_RUN_MAP is populated with Account Numbers having Latest Load Run Flag = Y. For further details, please refer Loading Multiple Load Runs in OFSAA.

This chapter provides information about Customer Dimension Loading Process in the Oracle Financial Services Data Foundation application.

This chapter includes the following topics:

- Customer Dimension Loading Overview
- Customer Dimension Population using Stage Party Master Entity
- Customer Dimension Population using Stage Customer Master Entity
- Execution of Customer Dimension Population Batch

Customer Dimension Loading Overview

The Customer Dimension populates DIM_CUSTOMER table by generating individual numeric Surrogate Keys for every customer reference code. This table will be used in Reporting by joining with other results tables.

Customer Dimension Population using Stage Party Master Entity

The Customer Dimension can be populated from Stage Party Master Entity using the SCD packaged in FSDF with map ref number 335.

Prerequisites

- STG_PARTY_MASTER should be loaded with all records which are required
- STG_PARTY_ROLE_MAP should be loaded with all customer records and Party Role should be 'CUSTOMER'

Customer Dimension Population using Stage Customer Master Entity

The Customer Dimension can be populated from Stage Customer Master Entity using the SCD packaged in FSDF with map ref number 32.

Prerequisites

- STG_CUSTOMER_MASTER should be loaded with all records which are required

Execution of Customer Dimension Population Batch

- For Party Master Flow

Customer Dimension SCD from Party Master can be executed by executing task present in the seeded batch FSDFINFO_DATA_FOUNDATION_SCD.

Following steps will help you to execute the batch:

1. Navigate to the *Batch Execution* screen.
2. Select the seeded batch FSDFINFO_DATA_FOUNDATION_SCD.
3. Select the AS_OF_DATE for which source customer information is required to be loaded into the table.

-
4. Click **Execute Batch**.
 5. Monitor the status of the batch from *Batch Monitor* screen of OFSAAI.

Error Messages

Following are the most common error message which will be logged in the T2T log file present in the *ftpshare/logs/<Run_Date>/FSDFINFO/RUN EXECUTABLE* folder:

Unique Constraint Violation : This occurs when attempting re-load or loading existing records for the already executed AS_OF_DATE.

- For Customer Master Flow

Customer Dimension SCD from Customer Master can be executed by executing task present in the seeded batch FSDFINFO_DATA_FOUNDATION_SCD.

Following steps will help you to execute the batch:

1. Navigate to the *Batch Execution* screen.
2. Select the seeded batch FSDFINFO_DATA_FOUNDATION_SCD.
3. Select the AS_OF_DATE for which source customer information is required to be loaded into the table.
4. Click **Execute Batch**.
5. Monitor the status of the batch from *Batch Monitor* screen of OFSAAI.

Error Messages

Following are the most common error message which will be logged in the T2T log file present in the *ftpshare/logs/<Run_Date>/FSDFINFO/RUN EXECUTABLE* folder:

- **Unique Constraint Violation** : This occurs when attempting re-load or loading existing records for the already executed AS_OF_DATE.

Note: Following are the two customer based dimensions that are used across various OFSAA applications for their processing and reporting requirements:

- DIM_PARTY
- DIM_CUSTOMER

In the earlier releases, both the dimensions sourced data from separate staging tables i.e. STG_PARTY_MASTER and STG_CUSTOMER_MASTER respectively. This design however enforced the population of customer data in both staging tables.

To address bug - **Bug 20486362 - SCD TO POPULATE DIM_CUSTOMER FROM**

STG_PARTY_MASTER TO BE ADDED, a new SCD (MAP_REF_NUM = 335) was introduced in FSDF release 8.0.1.0.0 to load DIM_CUSTOMER using STG_PARTY_MASTER and STG_PARTY_ROLE_MAP as the source. Customers, who use applications that have a dependence on DIM_PARTY, are advised to use this SCD instead of SCD, 32. Execute batch DIM_CUSTOMER_SCD_PARTY to populate DIM_CUSTOMER using STG_PARTY_MASTER and STG_PARTY_ROLE_MAP as the source. SCD, 32 will be deprecated in a future release. There are two flows available for DIM_CUSTOMER population. Only one need to be executed accordingly. Following are the two flows:

- Customer Dimension Populating using Customer Master
- Customer Dimension Populating using Party Master

Loading Multiple Load Runs in OFSAA

This chapter provides information about Loading Multiple Load Runs in OFSAA in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview
- Objective
- Design Details
- Implementation
- Loading OFSAA Staging Tables
- Post Stage Load Process
- Loading Data into OFSAA Results Tables from Staging Tables

Overview

Multiple load run enables data to be loaded multiple times during the day for staggered processing of data by analytical applications. The degree of complexity of data required by analytical apps vary from one to the other, the load run ensures that the customer can process the data as soon as it is ready for an app to uptake. This reduces the turnaround time to reporting, by avoiding the 'end of day' type of processing of information as part of the original design.

Note: As a work around, currently you can use the following methods to load the results table using different Load Run IDs

Objective

1. To optimize the end-to-end data flow and the need for intra-day reporting, institutions could load intra-day records into OFSAA. Current application can only handle one set of records per date (incremental loads are not possible).
2. Users need to adjust and reload data (either full or partial) for the current date.
3. Users need to adjust and reload data (either full or partial) for any of past dates.
4. Support incremental consumption of data from staging area.

Note: The load run is enabled only in the model and is '0' by default in the model. This does not impact data previously available. The enhancements to the OFSAA batch and run framework to cover all the use cases will be taken in a future release.

FSDF 8.0.6 staging model provides a flexibility to load multiple snapshots of the data in the staging tables (Product Processors). A column named `n_load_run_id` is introduced as part of the primary key of the product processor tables to enable this feature. However, the full fledged functionality to load and manage these snapshots will be part of the platform release at a later stage. In order to leverage this design in 8.0 release, the below mentioned changes

should be performed as a workaround to load multiple snapshot of data from staging to results tables such as Fact Common Account Summary.

For Loading multiple snapshots of data for the same FIC_MIS_DATE, the existing T2T's should be executed through the Run Rule Framework . Additionally, the load should be filtered accordingly for each load run via the run filter.

To execute this run:

1. Navigate to `$FIC_HOME/ficweb/webroot/conf`
2. Edit the file: `excludeURLList.cfg`
3. Add the following entry at the end of the file: `[SQLIA] ./pr2`

Note: There should not be any blank line in the file.

Design Details

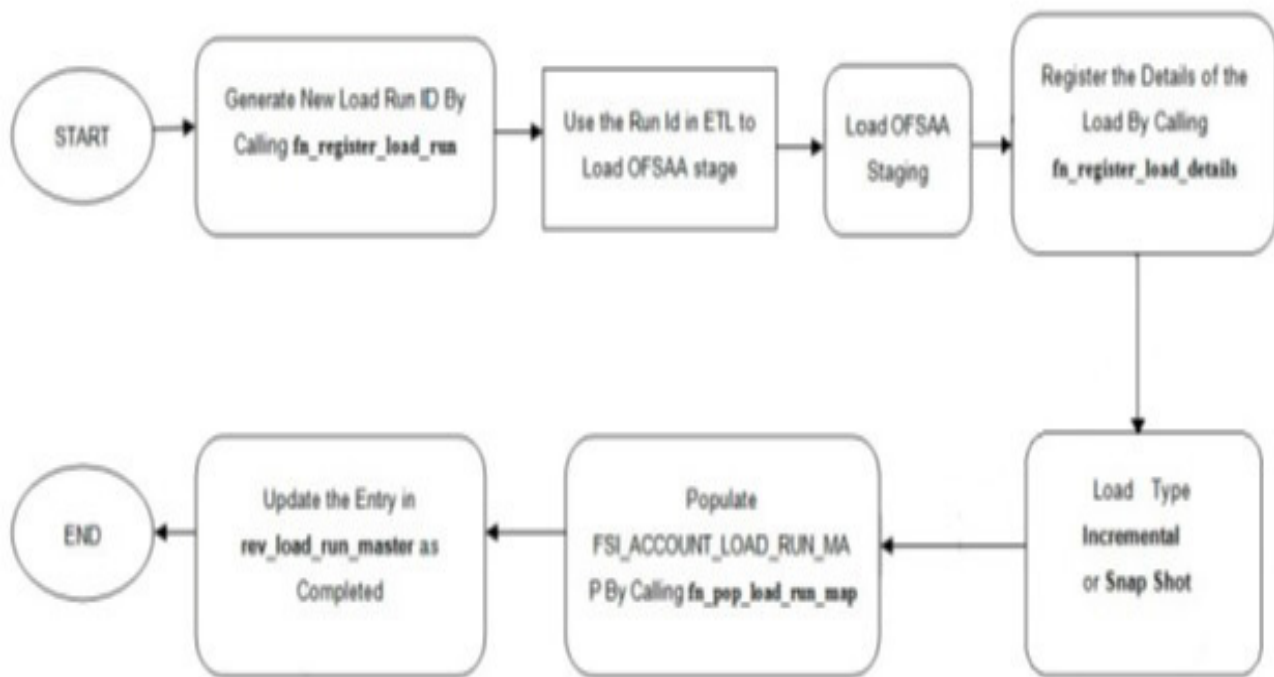
Loading of data into OFSAA can be in any of the following ways:

- ETL Tool
- OFSAA F2T
- OFSAA T2T
- OFSAA Excel upload
- OFSAA DIH

OFSAA data model includes load run identifier as part of the primary key for a set of staging tables. This enables data to be stored for multiple load runs for any date. OFSAA data model also has a table to maintain master information about load run and can be used for identifying/filtering load run during run execution within OFSAA.

Implementation

Before loading data into the staging table, you should generate a Load Run Identifier to stamp the records from the source. These records could be a complete snapshot or could be partial/incremental data too. This load run identifier can be generated by calling the function in the OFSAA atomic schema named `fn_register_load_run`. The function expects some input parameters and returns a unique load run identifier back to the calling program.



Note: Column n_load_run_id should always be populated only by the value returned by fn_register_load_run.

Function - Register Load Run (fn_register_load_run)

Parameters - Batch ID, MIS-Date, Load Run Name, Load Run Purpose, Load Run Type

Parameters	Source of Values	Example Values
Batch ID	Auto generated if you are using OFSAA Framework	OFSBFNDIN-FO_20150101_1
MIS-Date	Input from Customer	01/01/2015
Load Run Name	Input from Customer	Daily EOD Load
Load Run Purpose	Input from Customer	BA/BS (Basel Advanced Approach, Basel Standard)
Load Run Type	Input from Customer	B - Base, A - Adjustments, P- Backdated Adjustments

Example:

Declare

Result number;

Begin

```

Result := fn_register_load_run
('OFSBFNDINFO_20150101_1','20150101','FSDF_LOAD','BA','A');
  
```


End;

The function registers the request in the table name rev_load_run_master and marks load as "In progress". You can use columns LOAD_RUN_NAME and LOAD_PURPOSE as per the requirement.

Column Load Type must have only the permissible value such as:

- "B - Base
- "A - Adjustments
- "P- Backdated Adjustments

LOAD_RUN_ID	MIS_DATE	LOAD_TYPE	LOAD_PURPOSE	START_DT_TIME	LOAD_RUN_NAME	BATCH_ID	LOAD_RUN_STATUS
1	01-JAN-15	A	BA	01-JAN-15	FSDF_Load	OFSBFNDIN-FO_20150101_1	In Progress

Note: Multiple calls to the procedure can be made to the function for given FIC_MIS_DATE. Each call will return a number which will be unique across the FIC_MIS_DATE/Extraction date. You can use this load identifier to load either one or more staging tables.

Loading OFSAA Staging Tables

After the load run ID is generated as described above, you can use the same in external ETL process to stamp the records from the source system before them loading either in one or the multiple staging tables of OFSAA Staging area.

Load strategy at a customer site falls across two categories:

- Complete Snapshot
- Incremental Snapshot

Complete Snapshot Load Example

For example, if we have three Loan Contract accounts in a bank system which is supposed to be loaded into OFSAA Stage Loan Contracts using the Load Run ID = 1 for Basel Standard Approach, the data after loading staging table will appear as below:

FIC_MIS_DATE	V_ACCOUNT_NUMBER	V_GAAP_CODE	N_LOAD_RUN_ID	N_EOP_BAL
01-JAN-15	LOAN1000	USGAAP	1	4066.213
01-JAN-15	LOAN1001	USGAAP	1	34538.905
01-JAN-15	LOAN1002	USGAAP	1	667.357

Note: After each load you need to run fn_register_load_details function mentioned in Post Stage Load Process and Updating Load as Completed which is explained in the following sections.

In order to enable downstream applications to consume only the latest set of record, you need to call another function named `fn_pop_load_run_map`. This is mandatory in case of incremental snapshot load scenario. This function populates a intermediate processing table that keep track of latest incoming record identifier.

Function - Populate Load Run Map

Parameters - Batch Id, MIS Date, Stage Table Name, Load Run Id, Load Run Name

Parameters	Source of Values	Example Values
Batch ID	Auto generated if you are using OFSAA Framework	OFSBFNDIN-FO_20150101_1
MIS-Date	Input from Customer	01/01/2015
Stage Table Name	Input from Customer	STG_CASA
Load Run ID	Input from Customer	1
Load Run Name,	Input from Customer	FSDF_Load

Example

```

Declare
    Result number;
Begin
    Result: =
fn_pop_load_run_map('OFSBFNDINFO_20150101_1', '20150101', 'STG_CASA', 1, 'FSDF_LOAD');
END;
```

Note : For troubleshooting any errors while making the function calls , refer to `fsi_message_log` table for more details.

For the example mentioned above, records in `FSI_ACCOUNT_LOAD_RUN_MAP` table will appear as below:

FIC_MIS_DATE	V_ACCOUNT_NUMBER	V_GAAP_CODE	N_LOAD_RUN_ID	F_LATEST_LOAD_RUN_FLAG
1-Jan-15	LOAN1000	USGAAP	1	Y
1-Jan-15	LOAN1001	USGAAP	1	N
1-Jan-15	LOAN1002	USGAAP	1	N
1-Jan-15	LOAN1001	USGAAP	2	Y
1-Jan-15	LOAN1002	USGAAP	2	Y
1-Jan-15	LOAN1000	USGAAP	2	Y

There may be a requirement to reload a complete snapshot of data in the OFSAA staging again. This could either be to satisfy a intraday reporting requirement, or to load corrected source records in the OFSAA staging table. The earlier design forced users to truncate staging table to accommodate the new set of date. However with the introduction of Load Run identifier concept , you can retain both set of data in the staging area and allow the downstream application choose the correct set for processing. This will involve making another call to `fn_register_load_run` function.

For example in the below table, LOAN1001 and LOAN1002 have some changes since the previous load and will now need be loaded again staging with a different load run identifier . Additionally, strategy is to load the complete snapshot again to staging, all the records from the source such as, both changed and unchanged records will need to stamped with the new load run identifier

STG_LOAN_CONTRACTS after load will appear as below:

FIC_MIS_DATE	V_ACCOUNT_NUMBER	V_GAAP_CODE	N_LOAD_RUN_ID	N_EOP_BAL
01-JAN-15	LOAN1000	USGAAP	1	4066.213
01-JAN-15	LOAN1001	USGAAP	1	34538.905
01-JAN-15	LOAN1002	USGAAP	1	667.357
01-JAN-15	LOAN1000	USGAAP	2	4066.213
01-JAN-15	LOAN1001	USGAAP	2	34540.000
01-JAN-15	LOAN1002	USGAAP	2	670.000

REV_LOAD_RUN_MASTER after second function call will appear as below:

LOAD_RUN_ID	MIS_DATE	LOAD_TYPE	LOAD_PURPOSE	START_DT_TIME	LOAD_RUN_NAME	BATCH_ID	LOAD_RUN_STATUS
1	01-JAN-15	B	BA	01-JAN-15 13:00 PM	FSDF_Load	OFSBFND- INFO_20150 101_1	Complete
2	01-JAN-15	B	BA	01-JAN-15 23:00 PM	Loan Correc- tions	OFSBFND- INFO_20150 101_2	In Progress

Following tables require full snapshot mandatorily:

- STAGE ACCOUNT WRITE OFF DETAILS
- STAGE ACCOUNT RECOVERY DETAILS
- STAGE PARTY RATING DETAILS
- STAGE INSTRUMENT RATING DETAILS
- STAGE ACCOUNT RATING DETAILS

If the source system is unable to provide snapshots on a daily basis due to certain limitations, we can customise the T2Ts, and prepare the snapshot data out of incremental data. Data Loading scenarios are as follows to prepare the snapshot data set:

1. Day 1: Load full snapshot data as part of day 1 load to the Staging area and process the same to Results area.
2. Day 2: Receive incremental data in the Staging area and process the same to Results area.
3. Day 2: Create a customised T2T to copy the previous day data from results table excluding the records that are no more part of the load, for which you have received data as part of today's incremental load, and reload the same to results table with Current Date Surrogate key and Run Surrogate key.

Day 2 steps should be followed for subsequent loads.

Incremental Snapshot Load Example

This scenario is applicable when source may to portions of data at different point in time, or handover only the records changed since the last load. This is contrary to example explained under the Complete snapshot load section.

The same scenario in case of incremental snapshot load will appear as below.

Note : Only the changed source records are stamped with the new load run identifier.

FIC_MIS_DATE	V_ACCOUNT_NUMBER	V_GAAP_CODE	N_LOAD_RUN_ID	N_EOP_BAL
01-JAN-15	LOAN1000	USGAAP	1	4066.213
01-JAN-15	LOAN1001	USGAAP	1	34538.905
01-JAN-15	LOAN1002	USGAAP	1	667.357
01-JAN-15	LOAN1001	USGAAP	2	34540.000
01-JAN-15	LOAN1002	USGAAP	2	670.000

Note: After each load you need to run `fn_register_load_details` function mentioned in Post Stage Load Process and Updating Load as Completed which is explained in the following sections.

In order to enable downstream applications to consume only the latest set of record, you need to call another function named `fn_pop_load_run_map`. This is mandatory in case of incremental snapshot load scenario. This function populates a intermediate processing table that keep track of latest incoming record identifier.

Function - Populate Load Run Map

Parameters - Batch Id, MIS Date, Stage Table Name, Load Run Id, Load Run Name

Parameters	Source of Values	Example Values
Batch ID	Auto generated if you are using OFSAA Framework	OFSBFNDIN-FO_20150101_1
MIS-Date	Input from Customer	01/01/2015
Stage Table Name	Input from Customer	STG_CASA
Load Run ID	Input from Customer	1
Load Run Name,	Input from Customer	FSDF_Load

Example

```

Declare
    Result number;
Begin
    Result: =
fn_pop_load_run_map('OFSBFNDINFO_20150101_1', '20150101', 'STG_CASA', 1, 'FSDF
_LOAD');
END;

```

Note : For troubleshooting any errors while making the function calls , refer to fsi_message_log table for more details.

For the example mentioned above, records in FSI_ACCOUNT_LOAD_RUN_MAP table will appear as below:

FIC_MIS_DATE	V_ACCOUNT_NUMBER	V_GAAP_CODE	N_LOAD_RUN_ID	F_LATEST_LOAD_RUN_FLAG
1-Jan-15	LOAN1000	USGAAP	1	Y
1-Jan-15	LOAN1001	USGAAP	1	N
1-Jan-15	LOAN1002	USGAAP	1	N
1-Jan-15	LOAN1001	USGAAP	2	Y
1-Jan-15	LOAN1002	USGAAP	2	Y

Post Stage Load Process

Once you load the OFSAA Stage tables successfully using the load run ID which is generated from Load Run Map function, you need to perform certain post load processes in order to complete the loading.

Register Load Run Details

Once you load the OFSAA Stage tables successfully using the load run ID which is generated from Load Run Map function, you need to register the load run details by calling the following function with the load type whether it was incremental or full snap shot.

Function - Register Load Run Details

Parameters - batch id, mis-date, load run name, load run id, stage table name, load type

Parameters	Source of Values	Example Values
Batch ID	Auto generated if you are using OFSAA Framework	OFSBFNDIN-FO_20150101_1
MIS-Date	Input from Customer	01/01/2015
Stage Table Name	Input from Customer	STG_CASA

Parameters	Source of Values	Example Values
Load_Run_Id	Input from Customer	1
Load Run Name,	Input from Customer	FSDF_Load
Load Type	Input from Customer	S - Full Snap Shot I - Incremental

Example:

Declare

```
Result number;
```

Begin

```
Result: =
```

```
fn_register_load_details('OFSBFNDINFO_20150101_1','20150101',
'STG_CASA',1,'FSDF_LOAD', 'I');
```

END;

This function populates a table named rev_load_run_details. The columns load type can have only 2 values, such as:

- "S - Complete/Full Snapshot
- "I - Incremental Snapshot

LOAD_RUN_ID	MIS_DATE	STAGE_TABLE_NAME	LOAD_TYPE
1	1-Jan-15	STG_LOAN_CONTRACTS	I

Updating Load as Completed

Once you complete these steps, update the status of the record inside rev_load_run_master as Completed.

```
UPDATE rev_load_run_master
SET LOAD_RUN_STATUS = 'Completed'
WHERE pMis_Date = '01-Jan-15'
AND pLoad_Run_Name = 'FSDF_LOAD' and LOAD_RUN_ID = 1 ;
```

Loading data into OFSAA Results Tables from Staging Tables

OFSAAI does not support load run versioning for Data Movement from Stage to Results using Batch/Run Framework. The same feature is expected as an enhancement in future release of AAI.

As a work around, currently you can use the following methods to load the results table using different Load Run IDs

Complete Snapshot Load Scenario

To enable the Data Load into Results table from staging using the Load Run concept in case of Full Snap Shot, You can use the following Task Level Parameter in the T2T Filter condition and can pass the load run id to be passed in Batch Framework or Run Framework

For example: FCAS T2T for Stage Loan Contracts

- "Modify Filter Condition inside T2T:

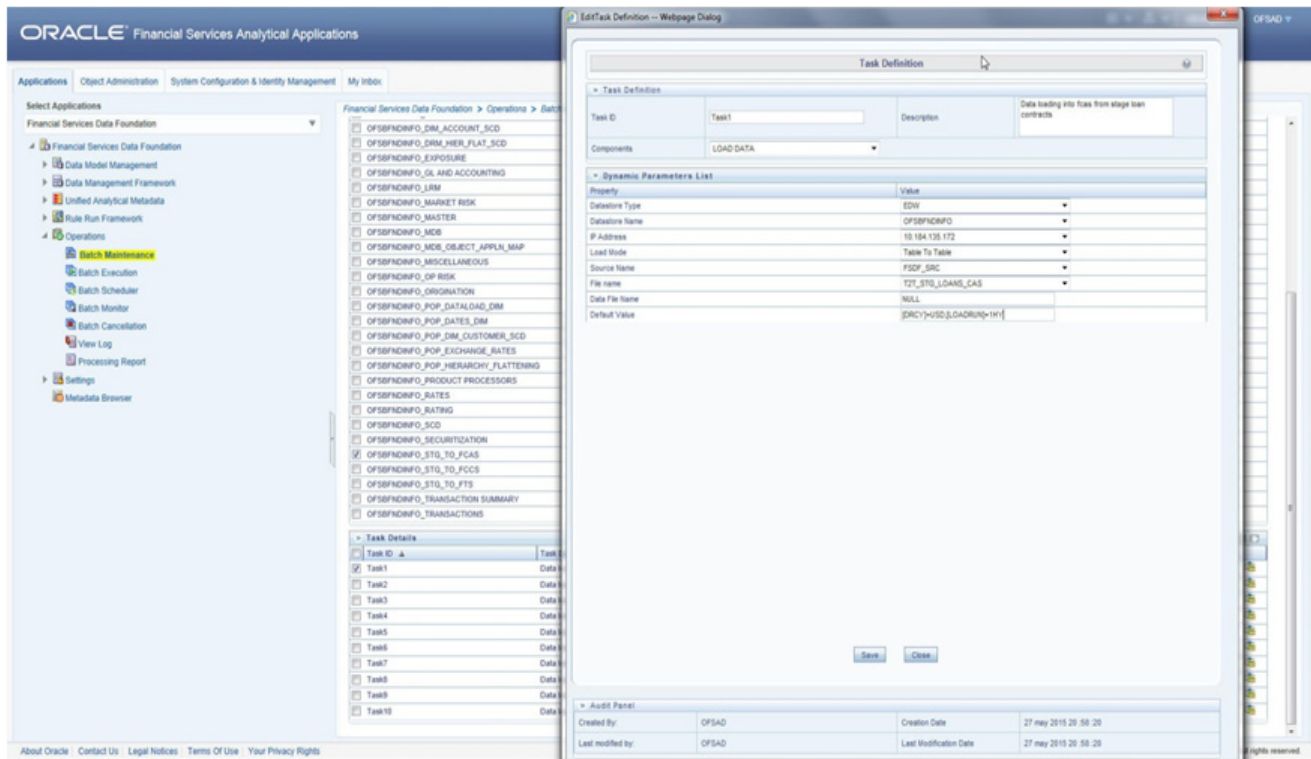
Filter Condition to be used in T2T: The highlighted condition required to be added in T2T

```
Filter : UPPER(STG_LOAN_CONTRACTS.v_gaap_code)='USGAAP' AND  
STG_LOAN_CONTRACTS.fic_mis_date=$MISDATE AND  
STG_LOAN_CONTRACTS.N_LOAD_RUN_ID=' [LOADRUN] '
```

- "Modify the Corresponding Batch Task Each Time with Load Run Idxxx

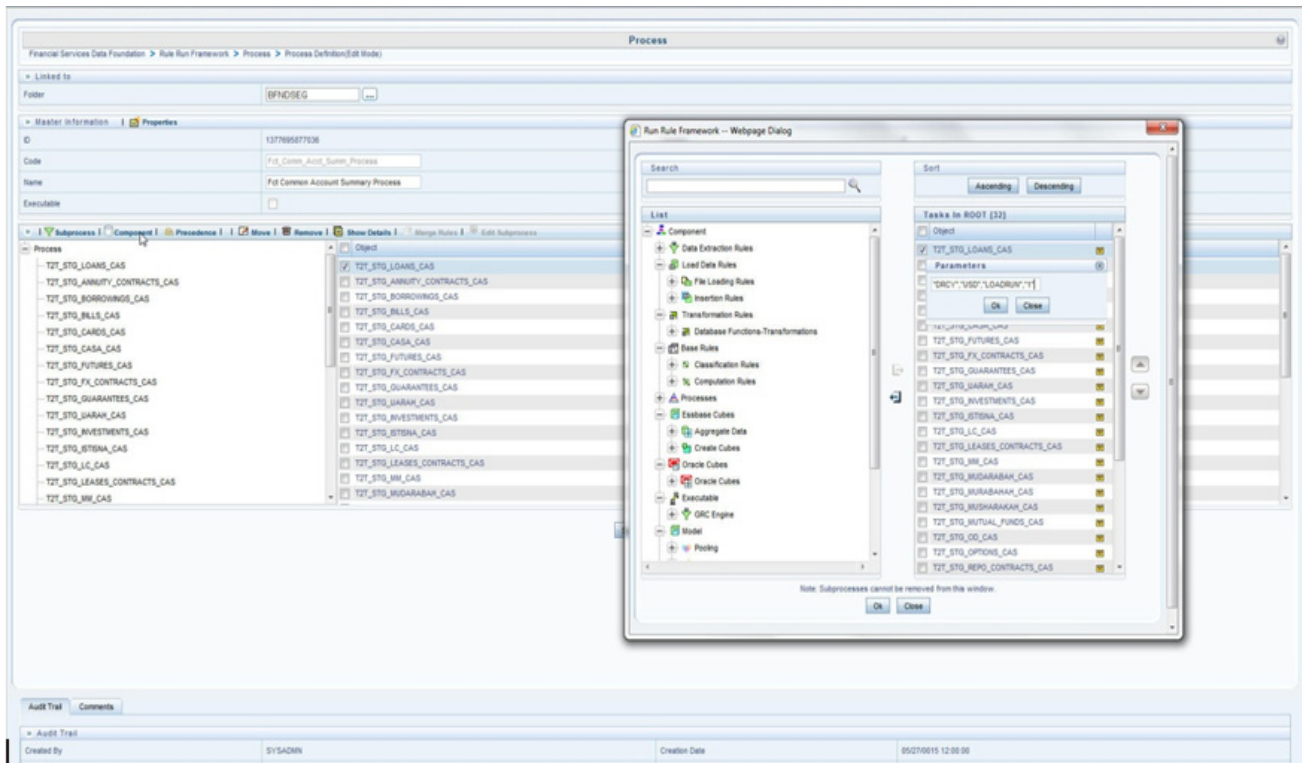
1. Select Batch, Task (T2T_STG_LOANS_CAS)
2. Click Edit
3. Add Highlighted Condition in Default Value and Save (Each Time we need to provide the Load Run ID which are supposed to use. Here in the below example we are using 1)

```
[DRCY]=USD, [LOADRUN]=1
```



Execute the batch after this change; it will load the Result table with the given Load Run Id Records
 Modify the Corresponding Process Task of a Run Each Time with Load Run Id

1. Select Process, Click Edit
2. Choose the Components, Select the Object (T2T_STG_LOANS_CAS) and Click Components
3. Click the Yellow Drop Button near the T2T
4. Add the Highlighted Condition with the text, and Save : "DRCY","USD","LOADRUN","1"



Fire the Run after this change. It loads the Result table with the given Load Run ID Records.

Incremental Load Scenario

To enable the Data Load into Results table from staging using the Load Run concept in case of incremental load, You can use the following T2T join change concept in the T2T ANSI join and can use the Batch Framework or Run Framework to execute the same

For example, FCAS T2T for Stage CASA

Modify Join Condition inside T2T:

Join Condition to be added in all relevant T2T

For example, STG_CASA T2T join

```
INNER JOIN FSI_ACCOUNT_LOAD_RUN_MAP
```

```
    ON FSI_ACCOUNT_LOAD_RUN_MAP.V_ACCOUNT_NUMBER =
       STG_CASA.V_ACCOUNT_NUMBER
```

```
    AND FSI_ACCOUNT_LOAD_RUN_MAP.N_LOAD_RUN_ID = STG_CASA.N_LOAD_RUN_ID
```

```
    AND FSI_ACCOUNT_LOAD_RUN_MAP.FIC_MIS_DATE = STG_CASA.FIC_MIS_DATE
```

```
    AND FSI_ACCOUNT_LOAD_RUN_MAP.V_GAAP_CODE = STG_CASA.V_GAAP_CODE
```

```
AND FSI_ACCOUNT_LOAD_RUN_MAP.F_LATEST_LOAD_RUN_FLAG = 'Y'
```

Execute the batch / fire the Run after the above step.

This chapter provides information about Exchange Rates Population in the Oracle Financial Services Data Foundation Application and step-by step instructions to use this section.

This chapter includes the following topics:

- Overview of Exchange Rates Table
- Overview of Exchange Rates Population
- Handling Alternate Currency
- Execution of Currency Exchange Rates Population T2T

Overview of Exchange Rates Table

Exchange Rates Table is loaded from a view VW_FSI_RATE_TRIANGULATION which is created on top of Stage table STG_EXCHANGE_RATE_HIST using the Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework.

In an integrated environment, there may be a Scenario where customer is using two applications and each of the applications refer to STG_FORWARD_EXCHG_RATES or STG_EXCHANGE_RATE_HIST.

In that case, the customer may be faced with duplicate data loads into these two tables. For this scenario, following is the recommendation:

- STG_EXCHANGE_RATE_HIST will take precedence over STG_FORWARD_EXCHG_RATES for Dataload
- The implementation team shall make sure that the T2T for STG_EXCHANGE_RATE_HIST dataload will be used
- From STG_EXCHANGE_RATE_HIST the FSI_EXCHANGE_RATES TABLE will be loaded

Overview of Exchange Rates Population

Table to Table seeded definition is provided for loading data into the target table Exchange Rates (FSI_EXCHANGE_RATES):

Table 4. Table to Table Seeded Definitions

Sl. No.	Source View Name	T2T Definition Name
1	VW_FSI_RATE_TRIANGULATION	T2T_FSI_EXCHANGE_RATES

Handling Alternate Currency

Exchange rates are provided by the customers as download. The applications may require the information in different format, the current exchange rate population has been enhanced to store the following variations.

Inverse Rate – When the exchange rate information of two currencies are provided for a base and a counter currency, the inverse rates if not available is derived and populated.

Triangulation Rate – When exchange rate information of two pairs are available as part of download, with a common currency in each of the pair, exchange rate for currencies not common as part of the download pair is derived and populated.

Multiple Execution of T2T in Run Chart

The batch FSDF_SOURCED_RUN is used to populate fsi_exchange_rates with RUN Skey defaulted to -1.

Note: These exchange rates are used for populating those FSDF Results tables which do not have any RUN SKEY. FSDF Source Run is used for populating these results tables using the exchange rates which were populated against Run Skey -1.

The T2T is also a part of Seeded Run Financial Services Data Foundation Execution Run. While executing through Run, the RUN Skey is auto-generated and stamped against each record. The FSDF Results tables that have a valid Run Skey in their primary key are a part of this run. The exchange rates which were populated against each Run Skey are used populating the results area of the same Run Skey.

Note: The T2T for Exchange Rates are part of Run Chart Twice.

1. Through a Batch (Will insert Run Skey as -1 in Exchange Rates Table).
2. Through a Run (Will insert Run Skey as actual Run Skey in Exchange Rates Table).

FCT_COMMON_ACCOUNT_SUMMARY is not a Run Enabled Table, hence the join is with -1, whereas FCT_REG_ACCOUNT_SUMMARY is Run Enabled Table hence the join is with Actual Run Skey.

So All T2Ts which are part of FSDF_EXE_RUN will use the exchange rate with actual Run Skey, other T2Ts will use -1.

Execution of Currency Exchange Rates Population T2T

Execution through Batch

The batch FSDF_SOURCED_RUN needs to be executed to populate fsi_exchange_rates as the entries in setup_master are seeded during installation.

Note: FSI_EXCHANGE_RATES table has to be loaded prior loading any other Results tables. While executing throughbatch, the RUN Skey will be defaulted to -1.

Execution through Run Management

The T2T is part of *Financial Services Data Foundation Sourced Run*. The process can be executed through the Seeded Run *Financial Services Data Foundation Execution Run*.

Note: While executing through Run, the RUN Skey will be auto-generated and stamped against each record.

Currency Execution Rates - Batch Execution

A seeded batch, FSDF_SOURCED_RUN must be executed for the required MIS Date.

Alternatively, following steps will help you create a new batch:

1. From the Home menu, click Operations and select Batch Maintenance.
2. Click **New Batch** ('+' symbol in Batch Name container). Enter the Batch Name and Description.
3. Click **Save**.
4. Click the check box in the Batch Name container to select the Batch, you created in the earlier step.

5. Enter the **Task ID** and **Description**.

6. Select Load Data from the Components list.

7. Select the following from the Dynamic Parameters List:

- Datastore Type - Select the appropriate datastore from the list.
- Datastore Name - Select the appropriate name from the list.
- IP address - Select the IP address from the list.
- Load Mode - Select Table to Table from the list.
- Source Name - Select <T2T Source Name> from the list.
- File Name - Select the T2T name for the source stage channel table you want to process.

8. Click **Save**.

Data file name will be blank for any Table to Table Load mode. Default value refers to currency calculation. If there is any need for currency conversion in T2T transactions, Default value has to be provided.

Execute the batch created in the preceding steps.

This chapter provides information about Account Summary Population in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview of Account Summary Tables
- Overview of Account Summary Population
- Executing the Account Summary Population T2T
- Checking the Execution Status
- Account Summary T2Ts

Overview of Account Summary Tables

Account Summary tables are loaded from the staging product processor tables using the Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework.

Customer account level data from the Oracle Financial Services Analytical Applications (OFSA) staging product processor tables must be consolidated into a standardized relational Business Intelligence (BI) data model. This consolidation is done to have all the staging product processor table data in a single Fact table. The Account Summary table data can be used for building cubes which allow rollup of data for a dimension or a combination of dimensions. This relational BI model consists of the following vertically partitioned Account Summary tables that are organized by application subject area.

- FCT_CRM_ACCOUNT_SUMMARY
- FCT_PFT_ACCOUNT_SUMMARY
- FCT_FTP_ACCOUNT_SUMMARY
- FCT_REG_CAP_ACCOUNT_SUMMARY
- FCT_ECO_CAPITAL_ACCOUNT_SUMMARY

The preceding Account Summary tables are part of data model but there are no seeded T2T definitions available to populate these tables. T2T processes must be custom configured to populate these tables to use measures defined on these tables for reporting.

Overview of Account Summary Population

Table to Table seeded definitions are provided for loading data into the target table Fct Common Account Summary (FCT_COMMON_ACCOUNT_SUMMARY):

Table 5. Table to Table Seeded Definitions

Sl. No.	Source Table Name	T2T Definition Name
1	STG_ANNUIITY_CONTRACTS	T2T_STG_ANNUIITY_CONTRACTS_CAS
2	STG_BILLS_CONTRACTS	T2T_STG_BILLS_CAS
3	STG_BORROWING_COMMITMENTS	T2T_STG_BORROWING_COMMITMENT S_CAS
4	STG_BORROWINGS	T2T_STG_BORROWINGS_CAS
5	STG_CARDS	T2T_STG_CARDS_CAS
6	STG_CASA	T2T_STG_CASA_CAS
7	STG_COMMITMENT_CONTRACTS	T2T_STG_COMMITMENT_CONTRACTS_ CAS
8	STG_CORRESPONDENT_ACCOUNT	T2T_STG_CORRESPONDENT_ACCOUNT _CAS
9	STG_CREDIT_DERIVATIVES	T2T_STG_CREDIT_DERIVATIVES_CAS
10	STG_FUTURES	T2T_STG_FUTURES_CAS
11	STG_FX_CONTRACTS	T2T_STG_FX_CONTRACTS_CAS
12	STG_GUARANTEES	T2T_STG_GUARANTEES_CAS
13	STG_IJARAH	T2T_STG_IJARAH_CAS
14	STG_INVESTMENTS	T2T_STG_INVESTMENTS_CAS
15	STG_ISTISNA	T2T_STG_ISTISNA_CAS
16	STG_LC_CONTRACTS	T2T_STG_LC_CAS
17	STG_LEASES_CONTRACTS	T2T_STG_LEASES_CONTRACTS_CAS
18	STG_LOAN_CONTRACTS	T2T_STG_LOANS_CAS
19	STG_MM_CONTRACTS	T2T_STG_MM_CAS
20	STG_MUDARABAH	T2T_STG_MUDARABAH_CAS
21	STG_MURABAHAH	T2T_STG_MURABAHAH_CAS
22	STG_MUSHARAKAH	T2T_STG_MUSHARAKAH_CAS
23	STG_MUTUAL_FUNDS	T2T_STG_MUTUAL_FUNDS_CAS
24	STG_OD_ACCOUNTS	T2T_STG_OD_CAS
25	STG_OPTION_CONTRACTS	T2T_STG_OPTIONS_CAS
26	STG_REPO_CONTRACTS	T2T_STG_REPO_CONTRACTS_CAS
27	STG_RETIREMENT_ACCOUNTS	T2T_STG_RETIREMENT_ACCOUNTS_C AS
28	STG_SALAM	T2T_STG_SALAM_CAS
29	STG_SUKUK	T2T_STG_SUKUK_CAS

Table 5. Table to Table Seeded Definitions

Sl. No.	Source Table Name	T2T Definition Name
30	STG_SWAPS_CONTRACTS	T2T_STG_SWAPS_CONTRACTS_CAS
31	STG_TD_CONTRACTS	T2T_STG_TD_CONTRACTS_CAS
32	STG_TRUSTS	T2T_STG_TRUSTS_CAS
33	STG_COMMODITIES	T2T_STG_COMMODITIES_CAS
34	STG_CUSTODIAL_ACCOUNTS	T2T_STG_CUSTODIAL_ACCOUNTS_CAS
35	STG_MANAGED_INV_ADV	T2T_STG_MANAGED_INV_ADV_CAS
36	STG_PREPAID_CARDS	T2T_STG_PREPAID_CARDS_CAS
37	STG_TRADING_ACCOUNT	T2T_STG_TRADING_ACCOUNT_CAS

Executing the Account Summary Population T2T

Executing through Batch

Fact Common Account Summary table has to be loaded prior loading any of the other Account Summary tables. You can execute the T2T component from OFSAA Infrastructure ICC framework (accessed through the application Batch Operations screen).

Note: Before executing Account Summary Population T2Ts, we need to manually configure the setup_master table with required GAAP_CODES. For an account we can load only one GAAP_CODE to Fact Common Account Summary. By default, FSDF installer will seed the following entry into SETUP_MASTER. While executing through batch, the RUN Skey will be defaulted to -1.

Execution through Run Management

The T2T is part of *Financial Services Data Foundation Sourced Run*. The process can be executed through the Seeded Run *Financial Services Data Foundation Execution Run*.

Note: While executing through Run, the RUN Skey will be autogenerated and stamped against each record.

Table 6. Configuration

V_COMPONENT_CODE	V_COMPONENT_DESC	V_COMPONENT_VALUE
DEFAULT_GAAP	DEFAULT_GAAP	USGAAP

For all other GAAP codes, we need to update SETUP_MASTER manually before running each Account Summary Population T2Ts.

Fact Common Account Summary - Batch Execution

A seeded batch, FSDF_SOURCED_RUN has to be executed for the required MIS Date.

Alternatively, following steps will help you create a new batch:

1. From the Home menu, click **Operations** and select **Batch Maintenance**.

-
2. Click New Batch ('+' symbol in Batch Name container). Enter the **Batch Name** and **Description**.
 3. Click **Save**.
 4. Click the check box in the Batch Name container to select the Batch, you created in the earlier step.
 5. Enter the **Task ID** and **Description**.
 6. Select **Load Data** from the Components list.
 7. Select the following from the Dynamic Parameters List:
 - Datastore Type - Select the appropriate datastore from the list.
 - Datastore Name - Select the appropriate name from the list.
 - IP address - Select the IP address from the list.
 - Load Mode - Select Table to Table from the list.
 - Source Name - Select <T2T Source Name> from the list.
 - File Name - Select the T2T name for the source stage channel table you want to process.
 8. Click **Save**.

Data file name will be blank for any Table to Table Load mode.

Default value refers to currency calculation. If there is any need for currency conversion in T2T transactions, Default value has to be provided.

For example, default value is [DRCY]='USD' Here 'USD' acts as reporting currency parameter to T2T.

9. Repeat steps 4 to 8 for adding the remaining T2Ts within the same batch definition.
10. Execute the batch created in the preceding steps.

Fact Common Account Summary - Run Execution

Alternatively, Stage Common Account Summary can be executed through Run Rule Framework.

Following steps will help you create a new batch:

1. From the Home menu, click Rules Framework and select Process.
2. Select **Fct_Comm_Acct_Summ_Process Process** and click **Edit**.
3. Click on **Component** if you wish to change the Parameter.
4. Select the required T2T and click on the Yellow Drop Down Button. Repeat as many as T2Ts need to be changed.
5. Update the Parameter Value and click **OK**. For example, default value is "DRCY","USD". Here 'USD' acts as reporting currency parameters to T2T.
6. Save the Process, and click **Save**.
7. Click **NO** in the Dialogue Box as we need to save it as the same version.
8. Close the Dialogue Box.
9. Open the RUN menu from Rules Frame Work.
10. Select **Comm Acc Summ Load Run** and click on **Fire Run** button.
11. Select the **Create & Execute** Option from Batch Menu and select the required MIS Date.

-
12. Click **OK** to execute the RUN.
 13. It will show the Batch execution is in Progress Dialogue. Close the Dialogue Box.
 14. Click **Batch Progressing Report** from **Operations** menu, and note down the Batch Run ID.
 15. Click on **Batch Monitor** from Operations menu.
 16. Type the noted Run Id in Batch ID Like Column and click on **Search** Button.
 17. Select the respective batch, Select Information Date and Batch Run ID from Batch Details tab and click on Start Monitoring Button.
 18. Check the Status of Each Task in the Run.

Checking the Execution Status

The status of execution can be monitored using the *Batch Monitor* screen.

Note: For a more comprehensive coverage of configuration and execution of a batch, see [OFS Analytical Applications Infrastructure User Guide](#).

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going
- F - Failure
- S – Success

The execution log can be accessed on the application server in the following directory *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA*. The file name will have the batch execution id.

The error log table in atomic schema is: FCT_COMMON_ACCOUNT_SUMMARY\$

Account Summary T2Ts

T2T definitions can be retrieved as an excel document for reference from the metadata browser of the Unified Metadata Manager (UMM) component of OFSAAL.

This chapter provides information about Customer Summary Population in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview of Common Customer Summary Tables
- Prerequisites
- Executing the Customer Summary Population T2T

Overview of Common Customer Summary Tables

Fact Common Customer Summary table stores attributes pertaining to customer related data on an 'as-is' basis received from the source system. Data is populated into this table using T2T. Customer balances are derived from account summary. Customer relationship table drives the relationship between accounts and customers. Common customer summary data is populated for all the active customers in customer dimension.

Prerequisites

Following are the lists of tables used in the population of Fact Common Customer Summary and these tables are required to be loaded prior to running the T2T:

- DIM_CUSTOMER
- DIM_BANDS
- DIM_EDUCATION
- DIM_CUSTOMER_TYPE
- DIM_GENDER
- DIM_INDUSTRY
- DIM_CHANNEL
- DIM_GEOGRAPHY
- DIM_MARITAL_STATUS
- DIM_MANAGEMENT
- DIM_PROFESSION
- DIM_CREDIT_RATING
- DIM_VINTAGE
- DIM_MIGRATION_REASONS
- FCT_COMMON_ACCOUNT_SUMMARY
- FCT_LIMITS_SUMMARY

-
- STG_CUSTOMER_DETAILS
 - STG_PARTY_RATING_DETAILS
 - STG_PARTY_FINANCIALS

Dimensions tables are loaded through the SCD process. The fact tables FCT_COMMON_ACCOUNT_SUMMARY is loaded from the respective T2T processes.

Executing the Customer Summary Population T2T

Execution through Batch

Fact Common Customer Summary T2T can be executed by executing task present in the seeded batch FSDF_SOURCED_RUN.

Following steps will help you to execute the batch:

1. Navigate to the *Batch Execution* screen.
2. Select the seeded batch FSDF_SOURCED_RUN.
3. Select the AS_OF_DATE for which source customer information is required to be loaded into the table.
4. Click **Execute Batch**.
5. Monitor the status of the batch from *Batch Monitor* screen of OFSAAI.

Note: While executing through batch, the RUN Skey will be defaulted to -1.

Execution through Run Management

The T2T is part of *Financial Services Data Foundation Sourced Run*. The process can be executed through the Seeded Run *Financial Services Data Foundation Execution Run*.

Note: While executing through Run, the RUN Skey will be autogenerated and stamped against each record

Error Messages

Following are the most common error message which will be logged in the T2T log file present in the *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA* folder:

- **Unique Constraint Violation:** This occurs when attempting re-load or loading existing records for the already executed AS_OF_DATE.

This chapter provides information about Fact Transaction Summary in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

The Fact Transaction Summary stores data from the stage transactions table for further operation reporting. The data is moved through a T2T process from stage to fact, which ensures that the stage data is available in a single table in the result area.

This chapter includes the following topics:

- Table to Table
- Executing the Fact Transaction Summary

Table to Table

Table to Table seeded definitions are provided for loading data into target table Fact Transaction Summary (FCT_TRANSACTION_SUMMARY)

Table 7. Table to Table Seeded Dimensions

SI No.	Source Table Name	T2T Definition Name
1	STG_ANNUITY_TXNS	T2T_STG_ANNUITY_TXNS_FTS
2	STG_BILL_CONTRACTS_TXNS	T2T_STG_BILL_CONTRACTS_TXNS_FTS
3	STG_BORROWING_COMMITMENT_TXNS	T2T_STG_BORROWING_COMMITMENT_TXNS_FTS
4	STG_BORROWINGS_TXNS	T2T_STG_BORROWINGS_TXNS_FTS
5	STG_CARDS_PAYMENT_TXNS	T2T_STG_CARDS_PAYMENT_TXNS_FTS
6	STG_CARDS_SETTLEMENT_TXNS	T2T_STG_CARDS_SETTLEMENT_TXNS_FTS
7	STG_CASA_TXNS	T2T_STG_CASA_TXNS_FTS
8	STG_COMMITMENT_CONTRACT_TXNS	T2T_STG_COMMITMENT_CONTRACT_TXNS_FTS
9	STG_CORRESPONDENT_ACCT_TXNS	T2T_STG_CORRESPONDENT_ACCT_TXNS_FTS
10	STG_CREDIT_DERIVATIVES_TXNS	T2T_STG_CREDIT_DERIVATIVES_TXNS_FTS
11	STG_FOREX_TXNS	T2T_STG_FOREX_TXNS_FTS
12	STG_FUTURES_TXNS	T2T_STG_FUTURES_TXNS_FTS
13	STG_GUARANTEES_TXNS	T2T_STG_GUARANTEES_TXNS_FTS
14	STG_IJARAH_TXNS	T2T_STG_IJARAH_TXNS_FTS
15	STG_INVESTMENT_TXNS	T2T_STG_INVESTMENT_TXNS_FTS

Table 7. Table to Table Seeded Dimensions

SI No.	Source Table Name	T2T Definition Name
16	STG_ISTISNA_TXNS	T2T_STG_ISTISNA_TXNS_FTS
17	STG_LC_TXNS	T2T_STG_LC_TXNS_FTS
18	STG_LEASES_TXNS	T2T_STG_LEASES_TXNS_FTS
19	STG_LOAN_CONTRACT_TXNS	T2T_STG_LOAN_CONTRACT_TXNS_FTS
20	STG_MM_TXNS	T2T_STG_MM_TXNS_FTS
21	STG_MUDARABAH_TXNS	T2T_STG_MUDARABAH_TXNS_FTS
22	STG_MURABAHAH_TXNS	T2T_STG_MURABAHAH_TXNS_FTS
23	STG_MUSHARAKAH_TXNS	T2T_STG_MUSHARAKAH_TXNS_FTS
24	STG_MUTUAL_FUNDS_TXNS	T2T_STG_MUTUAL_FUNDS_TXNS_FTS
25	STG_OD_ACCOUNTS_TXNS	T2T_STG_OD_ACCOUNTS_TXNS_FTS
26	STG_OPTION_CONTRACTS_TXNS	T2T_STG_OPTION_CONTRACTS_TXNS_FTS
27	STG_RETIREMENT_ACCOUNTS_TXNS	T2T_STG_RETIREMENT_ACCOUNTS_TXNS_FTS
28	STG_SALAM_TXNS	T2T_STG_SALAM_TXNS_FTS
29	STG_SUKUK_TXNS	T2T_STG_SUKUK_TXNS_FTS
30	STG_SWAP_ACCOUNT_TXNS	T2T_STG_SWAP_ACCOUNT_TXNS_FTS
31	STG_TERMDEPOSITS_TXNS	T2T_STG_TERMDEPOSITS_TXNS_FTS
32	STG_TRADING_ACCOUNT_TXNS	T2T_STG_TRADING_ACCOUNT_TXNS_FTS
33	STG_TRUSTS_TXNS	T2T_STG_TRUSTS_TXNS_FTS
34	STG_COMMODITIES_TXNS	T2T_STG_COMMODITIES_TXNS_FTS
35	STG_CUSTODIAN_ACCOUNT_TXNS	T2T_STG_CUSTODIAN_ACCOUNT_TXNS_FTS
36	STG_PREPAID_CARDS_TXNS	T2T_STG_PREPAID_CARDS_TXNS_FTS
37	STG_REPO_TRANSACTIONS	T2T_STG_REPO_TRANSACTIONS_FTS

Executing the Fact Transaction Summary

Execution through Batch

A seeded batch, FSDF_SOURCED_RUN has to be executed for the required MIS Date. Alternatively, following steps will help you create a new batch:

1. From the Home menu, click **Operations** and select **Batch Maintenance**.
2. Click New Batch ('+' symbol in Batch Name container). Enter the **Batch Name** and **Description**.

-
3. Click **Save**
 4. Click the check box in the Batch Name container to select the Batch, you created in the earlier step.
 5. Enter the **Task ID** and **Description**.
 6. Select **Load Data** from the Components list.
 7. Select the following from the Dynamic Parameters List:
 - Data Store Type
 - Datastore Name - Select the appropriate name from the list.
 - IP address - Select the IP address from the list. • Load Mode - Select Table to Table from the list.
 - Source Name - Select <T2T Source Name> from the list.
 - File Name - Select the T2T name for the source stage channel table you want to process.
 8. Click **Save**.

Data file name will be blank for any Table to Table Load mode.

Default value refers to currency calculation. If there is any need for currency conversion in T2T transactions, Default value has to be provided.

For example, default value is [DRCY]='USD' Here 'USD' acts as reporting currency parameter to T2T.

9. Repeat steps 4 to 8 for adding the remaining T2Ts within the same batch definition.
10. Execute the batch created in the preceding steps.

Note: While executing through batch, the RUN Skey will be defaulted to -1.

Execution through Run Management

The T2T is part of *Financial Services Data Foundation Sourced Run*. The process can be executed through the Seeded Run *Financial Services Data Foundation Execution Run*.

Note: While executing through Run, the RUNSkey will be auto-generated and stamped against each record

This chapter provides information about Loan Account Summary Population in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview of Loan Account Summary Tables
- Overview of Loan Account Summary Population
- Executing the Loan Account Summary Population T2T
- Checking the Execution Status
- Loan Account Summary T2Ts

Overview of Loan Account Summary Tables

Loan Account Summary table is loaded from the staging product processor tables using the Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework.

The Fact Loan Account Summary stores data from the Stage Loan Contracts, Stage Leases Contracts and Stage OD Accounts for further operation and regulatory reporting.

Overview of Loan Account Summary Population

Table to Table seeded definitions are provided for loading data into the target table Fct Loan Account Summary (FCT_LOAN_ACCOUNT_SUMMARY):

Table 8. Table to Table Seeded Definitions

Sl. No.	Source Table Name	T2T Definition Name
1	STG_LEASES_CONTRACTS	T2T_FLAS_STG_LEASES_CONTRACTS
2	STG_LOAN_CONTRACTS	T2T_FLAS_STG_LOAN_CONTRACTS
3	STG_OD_ACCOUNTS	T2T_FLAS_STG_OD_ACCOUNTS

Executing the Loan Account Summary Population T2T

Execution through Batch

Fact Loan Account Summary T2T can be executed by executing task present in the seeded batch FSDF_SOURCED_RUN.

Following steps will help you to execute the batch:

1. Navigate to the **Batch Execution** screen.
2. Select the seeded batch FSDF_SOURCED_RUN.
3. Select the AS_OF_DATE for which source customer information is required to be loaded into the table.

4. Click **Execute Batch**.

5. Monitor the status of the batch from *Batch Monitor* screen of OFSAAL.

Note: While executing through batch, the RUN Skey will be defaulted to -1.

Execution through Run Management

The T2T is part of *Financial Services Data Foundation Sourced Run*. The process can be executed through the Seeded Run *Financial Services Data Foundation Execution Run*.

Note: While executing through Run, the RUNSkey will be auto-generated and stamped against each record.

Error Messages

Following are the most common error message which will be logged in the T2T log file present in the *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA* folder:

- **Unique Constraint Violation** : This occurs when attempting re-load or loading existing records for the already executed AS_OF_DATE.

Checking the Execution Status

The status of execution can be monitored using the *Batch Monitor* screen.

For a more comprehensive coverage of configuration and execution of a batch, see [OFS Analytical Applications Infrastructure User Guide](#).

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going
- F - Failure
- S - Success

The execution log can be accessed on the application server in the following directory *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA*. The file name will have the batch execution id.

The error log table in atomic schema is: FCT_LOAN_ACCOUNT_SUMMARY\$.

Loan Account Summary T2Ts

T2T definitions can be retrieved as an excel document for reference from the metadata browser of the Unified Metadata Manager (UMM) component of OFSAAL.

Deposit Borrowings Summary Population

This chapter provides information about Deposit Borrowings Summary Population in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview of Deposit Borrowings Summary Tables
- Overview of Deposit Borrowings Summary Population
- Executing the Deposit Borrowings Summary Population T2T
- Checking the Execution Status
- Deposit Borrowings Summary T2Ts

Overview of Deposit Borrowings Summary Tables

Deposit Borrowings Summary table is loaded from the staging product processor tables using the Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework.

The Fact Deposit Summary stores data from the Stage Borrowings, Stage CASA and Stage TD Contracts for further operation and regulatory reporting.

Overview of Deposit Borrowings Summary Population

Table to Table seeded definitions are provided for loading data into the target table Fct Deposit Summary (FCT_DEPOSITS_BORROWINGS):

Table 9. Table to Table Seeded Definitions

Sl. No.	Source Table Name	T2T Definition Name
1	STG_BORROWINGS	T2T_FDB_STG_BORROWINGS
2	STG_CASA	T2T_FDB_STG_CASA
3	STG_TD_CONTRACTS	T2T_FDB_STG_TD_CONTRACTS

Executing the Deposit Borrowings Summary Population T2T

Execution through Batch

Fact Deposit Summary T2T can be executed by executing task present in the seeded batch FSDF_SOURCED_RUN.

Following steps will help you to execute the batch:

1. Navigate to the **Batch Execution** screen.
2. Select the seeded batch FSDF_SOURCED_RUN.

-
3. Select the AS_OF_DATE for which source customer information is required to be loaded into the table.
 4. Click **Execute Batch**.
 5. Monitor the status of the batch from **Batch Monitor** screen of OFSAAI.

Note: When executing through batch, the RUNSkey will be defaulted to -1.

Execution through Run Management

The T2T is part of *Financial Services Data Foundation Sourced Run*. The process can be executed through the Seeded Run *Financial Services Data Foundation Execution Run*.

Note: While executing through Run, the RUNSkey will be auto-generated and stamped against each record

Error Messages

Following are the most common error message which will be logged in the T2T log file present in the *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA* folder:

- **Unique Constraint Violation** : This occurs when attempting re-load or loading existing records for the already executed AS_OF_DATE.

Checking the Execution Status

The status of execution can be monitored using the *Batch Monitor* screen.

Note: For a more comprehensive coverage of configuration and execution of a batch, see [OFS Analytical Applications Infrastructure User Guide](#).

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going
- F - Failure
- S - Success

The execution log can be accessed on the application server in the following directory *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA*. The file name will have the batch execution id.

The error log table in atomic schema is: FCT_DEPOSITS_BORROWINGS\$.

Deposit Borrowings Summary T2Ts

T2T definitions can be retrieved as an excel document for reference from the metadata browser of the Unified Metadata Manager (UMM) component of OFSAAI.

This chapter provides information about Cards Summary Population in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview of Cards Summary Tables
- Overview of Cards Summary Population
- Executing the Cards Summary Population T2T
- Checking the Execution Status
- Cards Summary T2Ts

Overview of Cards Summary Tables

Cards Summary table is loaded from the staging product processor table and Cards Account Mapping table is loaded from respective staging table using the Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework.

The Fact Cards Summary stores data from the Stage Cards for further operation and regulatory reporting.

Overview of Cards Summary Population

Table to Table seeded definitions are provided for loading data into the target tables Fct Cards Summary (FCT_CARDS_SUMMARY) and Fct Cards Account Mapping (FCT_CARD_ACCT_MAPPING):

Table 10. Table to Table Seeded Definitions

Sl. No.	Source Table Name	T2T Definition Name
1	STG_CARDS, STG_CARDS_MASTER	T2T_FCT_CARDS_SUMMARY
2	STG_CARD_ACCT_MAPPING	T2T_FCT_CARD_ACCT_MAPPING
3	STG_CARDS_BALANCE_SUMMARY	T2T_FCT_CARDS_BALANCE_SUMMARY

Executing the Cards Summary Population T2T

Executing through Batch

Fact Cards Summary T2T can be executed by executing task present in the seeded batch FSDF_SOURCED_RUN.

Following steps will help you to execute the batch:

1. Navigate to the **Batch Execution** screen.
2. Select the seeded batch FSDF_SOURCED_RUN.

-
3. Select the AS_OF_DATE for which source customer information is required to be loaded into the table.
 4. Click **Execute Batch**.
 5. Monitor the status of the batch from **Batch Monitor** screen of OFSAAI.

Note: While executing through batch, the RUNSkey will be defaulted to -1.

Execution through Run Management

The T2Ts are part of *Financial Services Data Foundation Sourced Run*. The process can be executed through the Seeded Run *Financial Services Data Foundation Execution Run*.

- T2T_FCT_CARDS_SUMMARY
- T2T_FCT_CARDS_BALANCE_SUMMARY

Note: While executing through Run, the RUNSkey will be auto-generated and stamped against each record

Error Messages

Following are the most common error message which will be logged in the T2T log file present in the *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA* folder:

- **Unique Constraint Violation** : This occurs when attempting re-load or loading existing records for the already executed AS_OF_DATE.

Checking the Execution Status

The status of execution can be monitored using the *Batch Monitor* screen.

Note: For a more comprehensive coverage of configuration and execution of a batch, see [OFS Analytical Applications Infrastructure User Guide](#).

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going
- F - Failure
- S - Success

The execution log can be accessed on the application server in the following directory *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA*. The file name will have the batch execution id.

The error log table in atomic schema are:

- FCT_CARDS_SUMMARY\$
- FCT_CARD_ACCT_MAPPING\$
- FCT_CARDS_BALANCE_SUMMARY\$

Cards Summary T2Ts

T2T definitions can be retrieved as an excel document for reference from the metadata browser of the Unified Metadata Manager (UMM) component of OFSAAI.

This chapter provides information about Populating Mitigants Results Tables in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview of Mitigants Results Tables
- Overview of Mitigants Results Population
- Executing the Populating Mitigants Results T2T
- Checking the Execution Status
- Mitigants Results T2Ts

Overview of Mitigants Results Tables

Mitigants Results Tables are loaded from respective Stage tables using the Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework. Following are the Results Tables that stores Mitigants Information:

- FCT_MITIGANTS
- FCT_ACCOUNT_MITIGANT_MAP
- FCT_LOAN_SERVICED_MITIGANT_MAP

Overview of Mitigants Results Population

Table to Table seeded definitions are provided for loading data into the target tables:

Executing the Populating Mitigants Results T2T

Table 11. Table to Table Seeded Definitions

Sl. No.	Source Table Name	Target Table Name	T2T Definition Name
1	STG_MITIGANTS	FCT_MITIGANTS	T2T_FCT_MITIGANTS
2	STG_ACCOUNT_MITIGANT_MAP	FCT_ACCOUNT_MITIGANT_MAP	T2T_FCT_ACCOUNT_MITIGANT_MAP
3	STG_LOAN_SERVICED_MITIGANT_MAP	FCT_LOAN_SERVICED_MITIGANT_MAP	T2T_FCT_LOAN_SERVICED_MITIGANT_MAP

Executing through Batch

Mitigants Results T2T can be executed by executing task present in the seeded batch FSDF_SOURCED_RUN.

Following steps will help you to execute the batch:

1. Navigate to the **Batch Execution** screen.
2. Select the seeded batch FSDF_SOURCED_RUN.

-
3. Select the AS_OF_DATE for which source customer information is required to be loaded into the table.
 4. Click **Execute Batch**.
 5. Monitor the status of the batch from **Batch Monitor** screen of OFSAAI.

Note: While executing through batch, the RUNSkey will be defaulted to -1.

Execution through Run Management

The T2Ts are part of *Financial Services Data Foundation Sourced Run*. The process can be executed through the Seeded Run *Financial Services Data Foundation Execution Run*.

Note: When executing through Run, the RUNSkey will be auto-generated and stamped against each record.

Error Messages

Following are the most common error message which will be logged in the T2T log file present in the *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA* folder:

- **Unique Constraint Violation** : This occurs when attempting re-load or loading existing records for the already executed AS_OF_DATE.

Checking the Execution Status

The status of execution can be monitored using the *Batch Monitor* screen.

Note: For a more comprehensive coverage of configuration and execution of a batch, see [OFS Analytical Applications Infrastructure User Guide](#).

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going
- F - Failure
- S - Success

The execution log can be accessed on the application server in the following directory *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA*. The file name will have the batch execution id.

Following are the error log tables in atomic schema:

- FCT_MITIGANTS\$
- FCT_ACCOUNT_MITIGANT_MAP\$
- FCT_LOAN_SERVICED_MITIGANT_MAP\$

Mitigants Results T2Ts

T2T definitions can be retrieved as an excel document for reference from the metadata browser of the Unified Metadata Manager (UMM) component of OFSAAI.

Fiduciary Services Investment Summary Population

This chapter provides information about Fiduciary Services Investment Summary Population in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview of Fiduciary Services Investment Summary Table
- Overview of Fiduciary Services Investment Summary Population
- Executing the Fiduciary Services Investment Summary Population T2T
- Checking the Execution Status
- Fiduciary Services Investment Summary T2Ts

Overview of Fiduciary Services Investment Summary Table

Fiduciary Services Investment Summary table is loaded from the staging product processor table using the Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework.

A fiduciary is responsible for managing the assets of another person, or of a group of people. The fiduciary is expected to manage the assets for the benefit of the customer. The customer can open a trust account that would have the different investments that the person would do. The customer would have different assets under the trust such as Investments, Mutual Funds, CASA, Term Deposit etc. So, in these T2Ts the target table is loaded only with those records which are of fiduciary service type which is identified using Parent Account Number in each Product Processor table.

The Fact Fiduciary Services Investment Summary stores records from the Stage CASA, Stage Custodial Accounts, Stage Investments, Stage Managed Inv Adv, Stage Mutual Funds, Stage Retirement Accounts, Stage TD Contracts and Stage Trusts, that has valid Parent Account Number for further operation and regulatory reporting.

Overview of Fiduciary Services Investment Summary Population

Table to Table seeded definitions are provided for loading data into the target table Fct Fiduciary Serv Invst Summary (FCT_FIDUCIARY_SERV_INVST_SUMM):

Table 12. Table to Table Seeded Definitions

Sl. No.	Source Table Name	T2T Definition Name
1	STG_CASA	T2T_FFSIS_STG_CASA
2	STG_CUSTODIAL_ACCOUNTS	T2T_FFSIS_STG_CUSTODIAL_ACCOUNTS
3	STG_INVESTMENTS	T2T_FFSIS_STG_INVESTMENTS
4	STG_MANAGED_INV_ADV	T2T_FFSIS_STG_MANAGED_INV_ADV
5	STG_MUTUAL_FUNDS	T2T_FFSIS_STG_MUTUAL_FUNDS

Table 12. Table to Table Seeded Definitions

Sl. No.	Source Table Name	T2T Definition Name
6	STG_RETIREMENT_ACCOUNTS	T2T_FFSIS_STG_RETIREMENT_ACCOUNTS
7	STG_TD_CONTRACTS	T2T_FFSIS_STG_TD_CONTRACTS
8	STG_TRUSTS	T2T_FFSIS_STG_TRUSTS

Executing the Fiduciary Services Investment Summary Population T2T

Executing through Batch Framework

Fact Fiduciary Services Investment Summary T2T can be executed by executing task present in the seeded batch FSDF_SOURCED_RUN

Following steps will help you to execute the batch:

1. Navigate to the **Batch Execution** screen.
2. Select the seeded batch FSDF_SOURCED_RUN.
3. Select the AS_OF_DATE for which source customer information is required to be loaded into the table.
4. Click **Execute Batch**.
5. Monitor the status of the batch from **Batch Monitor** screen of OFSAAL.

Note: When executing through batch, the RUNSkey will be defaulted to -1.

Execution through Run Management

The T2T is part of *Financial Services Data Foundation Sourced Run*. The process can be executed through the Seeded Run *Financial Services Data Foundation Execution Run*.

Note: While executing through Run, the RUNSkey will be auto-generated and stamped against each record

Error Messages

Following are the most common error message which will be logged in the T2T log file present in the *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA* folder:

- **Unique Constraint Violation** : This occurs when attempting re-load or loading existing records for the already executed AS_OF_DATE.

Checking the Execution Status

The status of execution can be monitored using the *Batch Monitor* screen.

Note: For a more comprehensive coverage of configuration and execution of a batch, see [OFS Analytical Applications Infrastructure User Guide](#).

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going
- F - Failure

-
- S - Success

The execution log can be accessed on the application server in the following directory *fipshare/logs/<Run_Date>/FSDFINFO/LOAD DATA*. The file name will have the batch execution id.

The error log table in atomic schema is: FCT_FIDUCIARY_SERV_INVST_SUMM\$.

Fiduciary Services Investment Summary T2Ts

T2T definitions can be retrieved as an excel document for reference from the metadata browser of the Unified Metadata Manager (UMM) component of OFSAAL.

This chapter provides information about Populating various Party Attributes Results Tables in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview of Party Attributes Results Tables
- Overview of Party Attributes Results Population
- Executing the Populating Party Attributes T2T
- Checking the Execution Status
- Party Attributes Results T2Ts

Overview of Party Attributes Results Tables

Party Attribute Results Tables are loaded from respective Stage tables using the Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework. Following are the Results Tables that stores Party Attributes:

- FCT_PARTY_ADDRESS_MAP
- FCT_PARTY_EMAIL_MAP
- FCT_PARTY_PHONE_MAP
- FCT_PARTY_FINANCIAL_DETAIL
- FCT_PARTY_RATING_DETAILS
- FCT_ACCOUNT_RATING_DETAILS
- FCT_CREDIT_SCORE_DETAILS

Overview of Party Attributes Results Population

Table to Table seeded definitions are provided for loading data into the target tables:

Table 13. Table to Table Seeded Definitions

Sl. No.	Source Table Name	Target Table Name	T2T Definition Name
1	STG_PARTY_ADDRESS_MAP	FCT_PARTY_ADDRESS_MAP	T2T_FCT_PARTY_ADDRESS_MAP
2	STG_PARTY_EMAIL_MAP	FCT_PARTY_EMAIL_MAP	T2T_FCT_PARTY_EMAIL_MAP
3	STG_PARTY_PHONE_MAP	FCT_PARTY_PHONE_MAP	T2T_FCT_PARTY_PHONE_MAP

Table 13. Table to Table Seeded Definitions

Sl. No.	Source Table Name	Target Table Name	T2T Definition Name
4	STG_PARTY_FINANCIALS	FCT_PARTY_FINANCIAL_DE TAIL	T2T_FCT_PARTY_FINAN CIAL_DETAIL
5	STG_PARTY_FINANCIALS	FCT_PARTY_FINANCIALS	T2T_FCT_PARTY_FINAN CIALS
6	STG_PARTY_RATING_DETA ILS	FCT_PARTY_RATING_DETAI LS	T2T_FCT_PARTY_RATIN G_DETAILS
7	STG_ACCOUNT_RATING_DE TAILS	FCT_ACCOUNT_RATING_DE TAILS	T2T_FCT_ACCOUNT_RA TING_DETAILS
8	STG_CREDIT_SCORE_DETAI LS	FCT_CREDIT_SCORE_DETAI LS	T2T_FCT_CREDIT_SCOR E_DETAILS

Executing the Populating Party Attributes T2T

Executing through Batch Framework

Party Attributes Results T2T can be executed by executing task present in the seeded batch FSDF_SOURCED_RUN.

Following steps will help you to execute the batch:

1. Navigate to the *Batch Execution* screen.
2. Select the seeded batch FSDF_SOURCED_RUN.
3. Select the AS_OF_DATE for which source customer information is required to be loaded into the table.
4. Click **Execute Batch**.
5. Monitor the status of the batch from **Batch Monitor** screen of OFSAAL.

Note: When executing through batch, the RUNSkey will be defaulted to -1.

Execution through Run Management

The T2T is part of *Financial Services Data Foundation Sourced Run*. The process can be executed through the Seeded Run *Financial Services Data Foundation Execution Run*.

Note: When executing through Run, the RUNSkey will be auto-generated and stamped against each record.

Error Messages

Following are the most common error message which will be logged in the T2T log file present in the *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA* folder:

- **Unique Constraint Violation** : This occurs when attempting re-load or loading existing records for the already executed AS_OF_DATE.

Checking the Execution Status

The status of execution can be monitored using the *Batch Monitor* screen.

Note: For a more comprehensive coverage of configuration and execution of a batch, see [OFS Analytical Applications Infrastructure User Guide](#).

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going
- F - Failure
- S - Success

The execution log can be accessed on the application server in the following directory *fipshare/logs/<Run_Date>/FSDFINFO/LOAD DATA*. The file name will have the batch execution id.

Following are the error log tables in atomic schema:

- FCT_PARTY_ADDRESS_MAP\$
- FCT_PARTY_EMAIL_MAP\$
- FCT_PARTY_PHONE_MAP\$
- FCT_PARTY_FINANCIAL_DETAIL\$
- FCT_PARTY_RATING_DETAILS\$
- FCT_ACCOUNT_RATING_DETAILS\$
- FCT_CREDIT_SCORE_DETAILS\$

Party Attributes Results T2Ts

T2T definitions can be retrieved as an excel document for reference from the metadata browser of the Unified Metadata Manager (UMM) component of OFSAAL.

This chapter provides information about Populating Placed Collateral Tables in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview of Placed Collateral Population Tables
- Overview of Placed Collateral Population
- Executing the Placed Collateral Population T2Ts
- Checking the Execution Status
- Placed Collateral Population Results T2Ts

Overview of Placed Collateral Population Tables

Placed Collateral Population Tables are loaded from respective Stage tables using the Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework. Following are the Results Tables that stores Placed Collateral Population:

- FCT_PLACED_COLLATERAL
- FCT_ACCT_PLACED_COLL_MAP

Overview of Placed Collateral Population

Table to Table seeded definitions are provided for loading data into the target tables:

Table 14. Table to Table Seeded Definitions

Sl. No.	Source Table Name	Target Table Name	T2T Definition Name
1	STG_PLACED_COLLATERAL	FCT_PLACED_COLLATERAL	T2T_FCT_PLACED_COLLATERAL
2	STG_ACCT_PLACED_COLL_MAP	FCT_ACCT_PLACED_COLL_MAP	T2T_FCT_ACCT_PLACED_COLL_MAP

Executing the Placed Collateral Population T2Ts

Executing through Batch Framework

Placed Collateral Populations T2T can be executed by executing task present in the seeded batch FSDF_SOURCED_RUN.

Following steps will help you to execute the batch:

1. Navigate to the **Batch Execution** screen.

-
2. Select the seeded batch FSDf_SOURCEd_RUN.
 3. Select the AS_OF_DATE for which source customer information is required to be loaded into the table.
 4. Click **Execute Batch**.
 5. Monitor the status of the batch from **Batch Monitor** screen of OFSAAI.

Note: When executing through batch, the RUNSkey will be defaulted to -1.

Execution through Run Management

The T2T is part of *Financial Services Data Foundation Sourced Run*. The process can be executed through the Seeded Run *Financial Services Data Foundation Execution Run*.

Note: When executing through Run, the RUNSkey will be auto-generated and stamped against each record.

Error Messages

Following are the most common error message which will be logged in the T2T log file present in the *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA* folder:

- **Unique Constraint Violation** : This occurs when attempting re-load or loading existing records for the already executed AS_OF_DATE.

Checking the Execution Status

The status of execution can be monitored using the *Batch Monitor* screen.

Note: For a more comprehensive coverage of configuration and execution of a batch, see [OFS Analytical Applications Infrastructure User Guide](#).

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going
- F - Failure
- S - Success

The execution log can be accessed on the application server in the following directory *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA*. The file name will have the batch execution id.

Following are the error log tables in atomic schema:

- FCT_PLACED_COLLATERAL\$
- FCT_ACCT_PLACED_COLL_MAP\$

Placed Collateral Population Results T2Ts

T2T definitions can be retrieved as an excel document for reference from the metadata browser of the Unified Metadata Manager (UMM) component of OFSAAI.

Credit Score Model and Probability of Default Model

This chapter provides information about Credit Score Model and Probability of default Model Tables in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview of Credit Score Model and Probability of Default Model
- Overview of Credit Score Model and Probability of Default Model Population
- Staging Data Expectation for Credit Score Model
- Overview of Mapper for Credit Score Model to Regulatory Credit Score Model
- Maintenance of Mapper for Credit Score Model to Regulatory Credit Score Model
- Loading Mapper Maintenance through Backend
- Executing the Credit Score Model and Probability of Default Model T2Ts
- Checking the Execution Status
- Credit Score Model and Probability of Default Model Population T2Ts

Overview of Credit Score Model and Probability of Default Model

Credit Score Model and Probability of default Model Tables are loaded from respective Stage tables using the Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework. Following are the Results Tables that stores Party Attributes:

- FCT_ACCT_CREDIT_SCORE_DETAILS
- FCT_SERV_ACCT_CREDIT_SCORE_DTL
- FCT_PARTY_PD_DETAILS
- FCT_INSTRUMENT_PD_DETAILS

Overview of Credit Score Model and Probability of Default Model Population

Sl. No.	Source Table Name	Target Table Name	T2T Definition Name
1	STG_ACCT_CREDIT_SCORE_-DETAILS	FCT_ACCT_CREDIT_SCORE_-DETAILS	T2T_FCT_ACCT_CREDIT_-SCORE_DETAILS
2	STG_SERV_ACCT_CREDIT_-SCORE_DTL	FCT_SERV_ACCT_CREDIT_-SCORE_DTL	T2T_FCT_SERV_ACCT_CREDIT_SCORE_DTL
3	STG_PARTY_PD_DETAILS	FCT_PARTY_PD_DETAILS	T2T_FCT_PARTY_PD_DETAILS
4	STG_INSTRUMENT_PD_DE-TAILS	FCT_INSTRUMENT_PD_DE-TAILS	T2T_FCT_INSTRUMENT_PD_DE-TAILS

Staging Data Expectation for Credit Score Model

Stage Account Credit Score Details and Stage Service Account Credit Score Details tables expect data incrementally and not in snapshot mode. For example: Account-1 loaded on Day-1 with Model-1 and Score-1 need not be loaded everyday till data gets changed. But Reporting tables: Fact Account Credit Score Details and Fact Service Account Credit Score Details are mapped to reports in snapshot mode. In T2T, latest records available on a daily basis for a given account and model are picked and loaded to the Reporting tables.

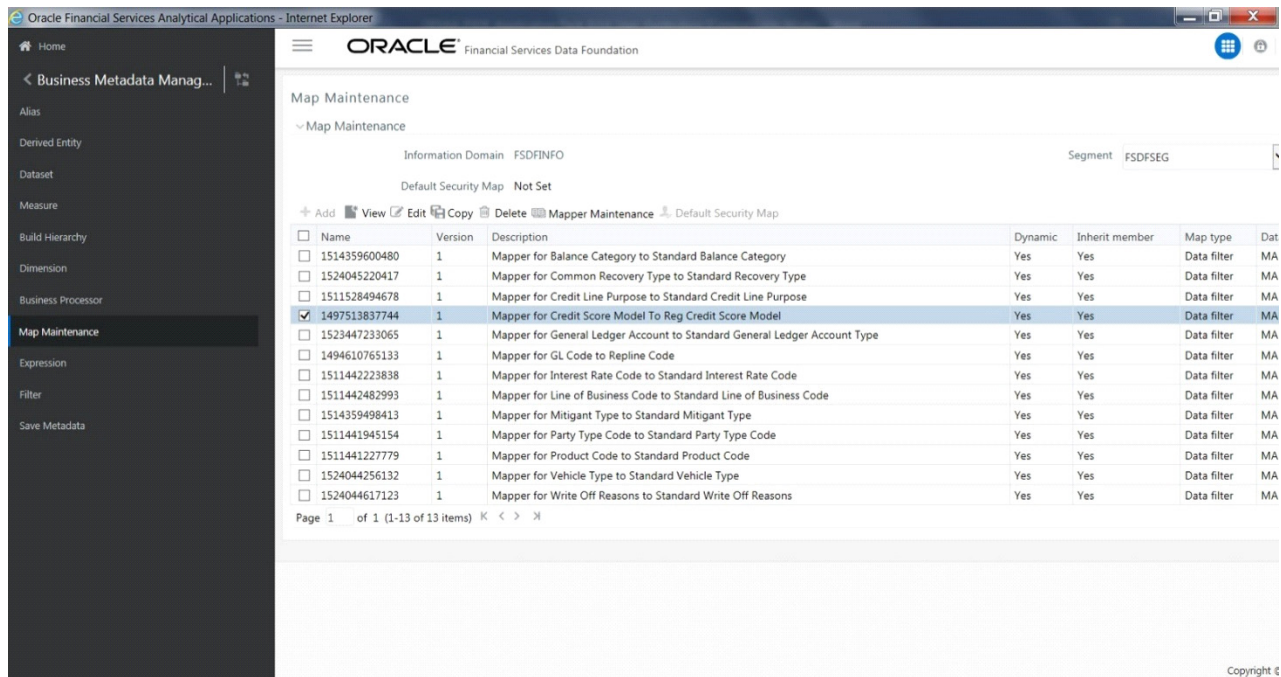
Overview of Mapper for Credit Score Model to Regulatory Credit Score Model

Credit Score Model dimension table is a slowly changing dimension (SCD), which has user-specific values for available credit score models. Regulatory Credit Score Model dimension table is a seeded dimension, which has the reporting specific values for regulatory reporting requirements. There is one column in Fact Account Credit Score Details table and Fact Service Account Credit Score Details, which stores Regulatory Credit Score Model Surrogate key (Skey). This column is populated through a mapper table maintained through Mapper Maintenance Component of OFSAAI.

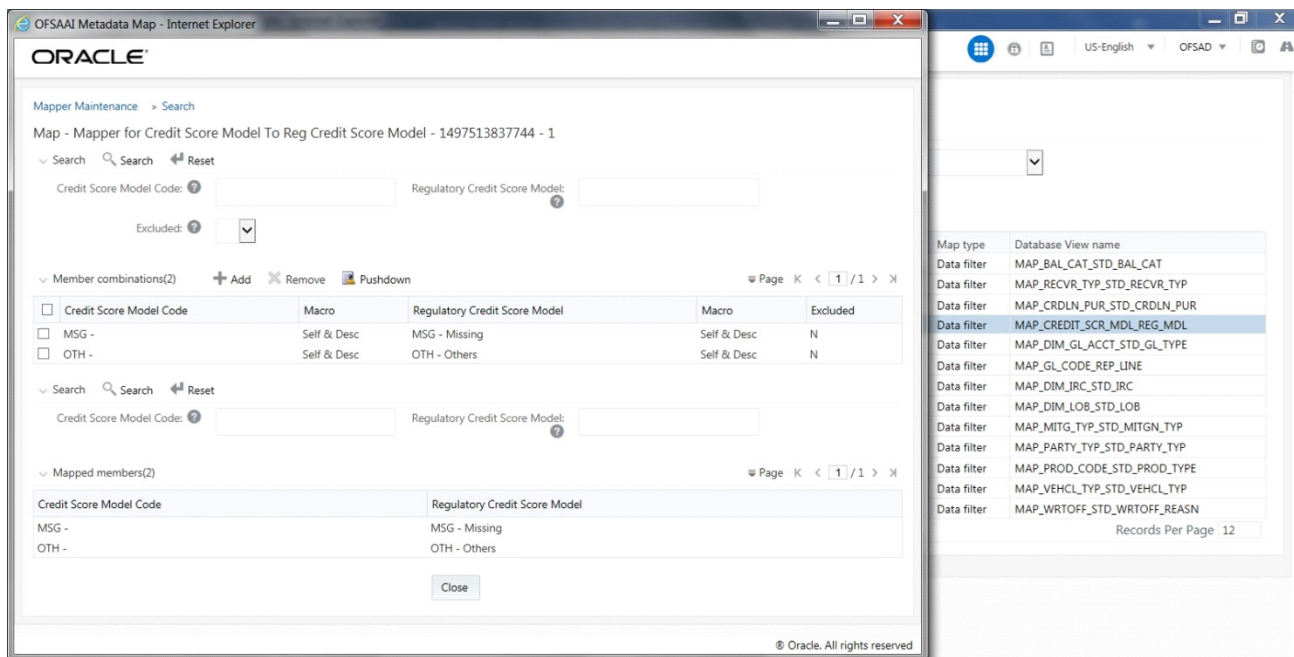
Maintenance of Mapper for Credit Score Model to Regulatory Credit Score Model

Mapper can be maintained under OFSAAI.

1. Navigate to **OFSAAI > Financial Services data Foundation > Unified Analytical Metadata > Business Metadata Management > Map Maintenance**. Select **Mapper for Credit Score Model to Regulatory Credit Score Model**. Click Mapper Maintenance icon.



2. FSDF Maps OTH and MSG out-of-the-box for this mapper. The remaining Mappings can be maintained by the user according to user specific values.



Prerequisites for Mapper Maintenance

1. Load Credit Score Model Dimension using SCD.

2. Resave the following hierarchies:

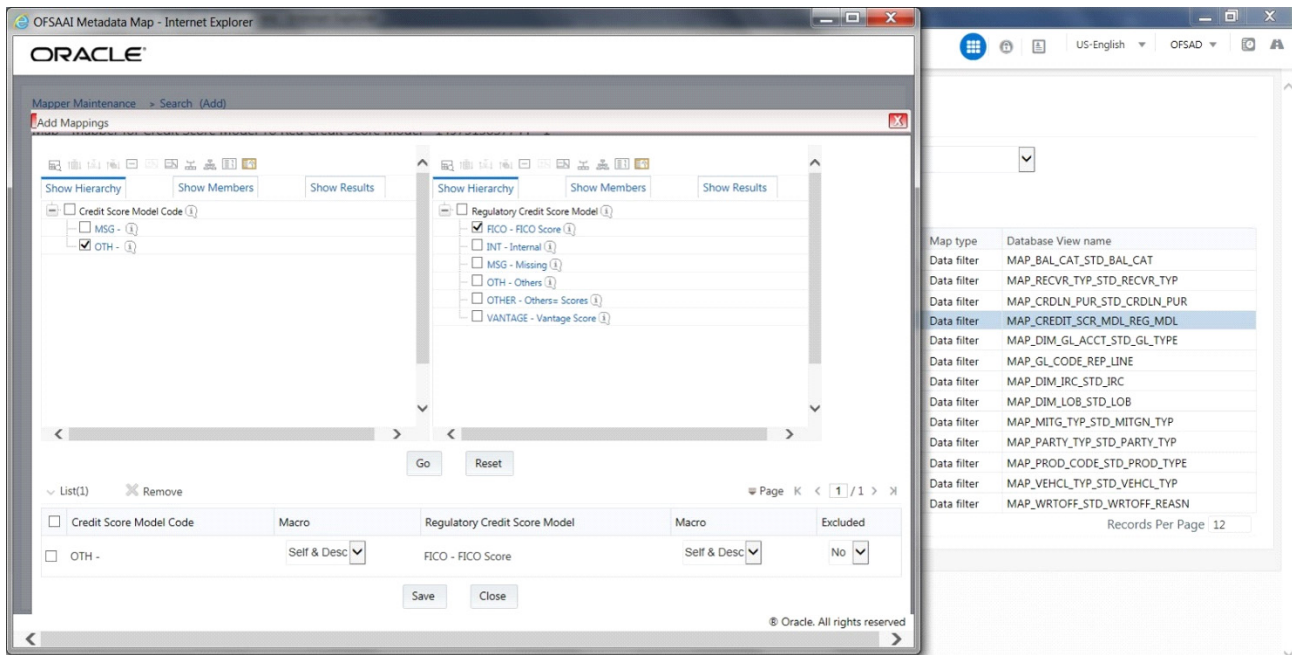
- HSCRMDL1 - Credit Score Model Code
- HSCRMDL2 - Regulatory Credit Score Model

Possible Mapping Combinations

One Credit Score Model in source can be mapped only to one Regulatory Credit Score Model. One to Many or Many to Many mapping will lead to error in T2T as the records will be duplicated. The possible combinations for Credit Score Model to Regulatory Credit Score Model mapping are One to One and Many to One mappings.

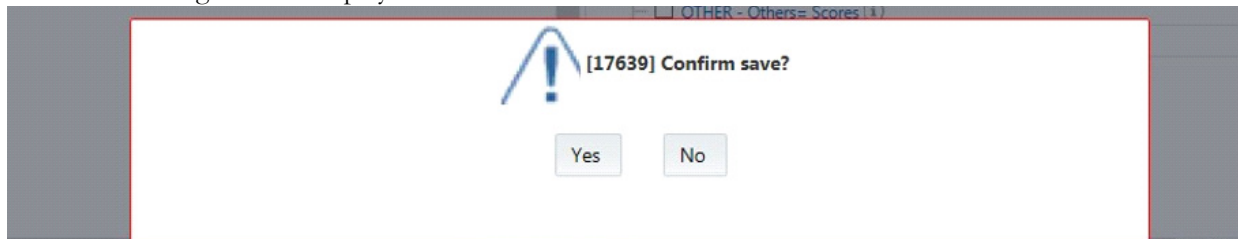
- **One to One Mapping:** One Credit Score Model can be mapped to one Regulatory Credit Score Model using the Mapper Maintenance screen. Here, you need to select one Credit Score Model and one Regulatory Credit Score Model. This means the corresponding data in Stage Account Credit Score Details Credit Score Model column for the selected Credit Score Model will be loaded into Fact Account Credit Score Details Regulatory Credit Score Model column while loading the T2T.

1. In the **Mapper Maintenance** window, click **Add**. The **Add Mappings** pop-up window is displayed.
2. In this example, OTH - Others is mapped to FICO - FICO Score. To map, click **Go** and then to save the mapping, click **Save**.

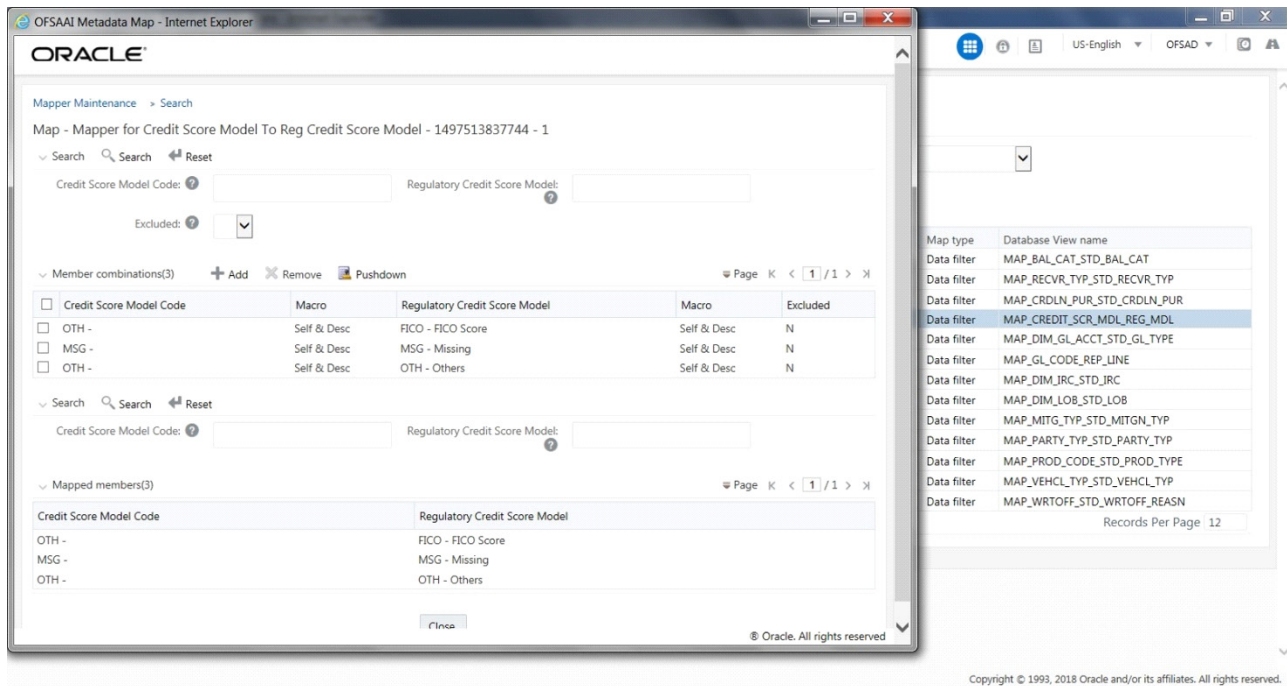


Copyright © 1993, 2018 Oracle and/or its affiliates. All rights reserved.

3. An acknowledgement is displayed. Click **Yes** to confirm.

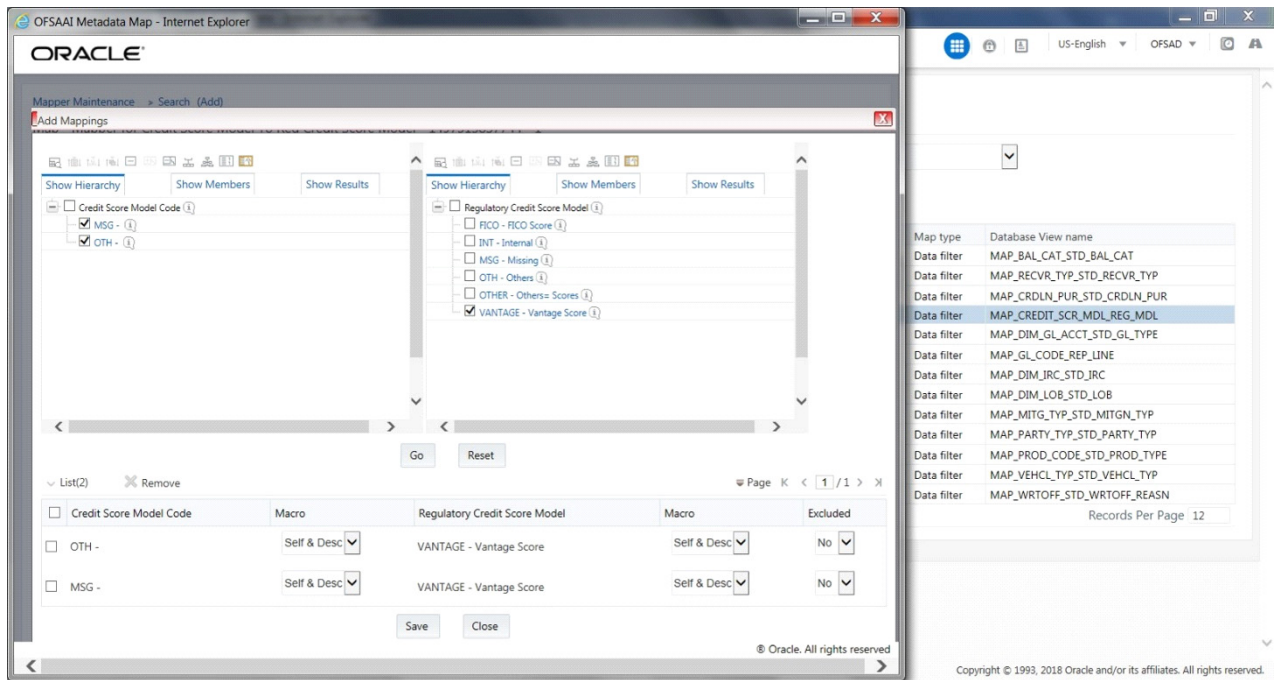


4. The Mapped Members are displayed in the Mapper Maintenance Screen.

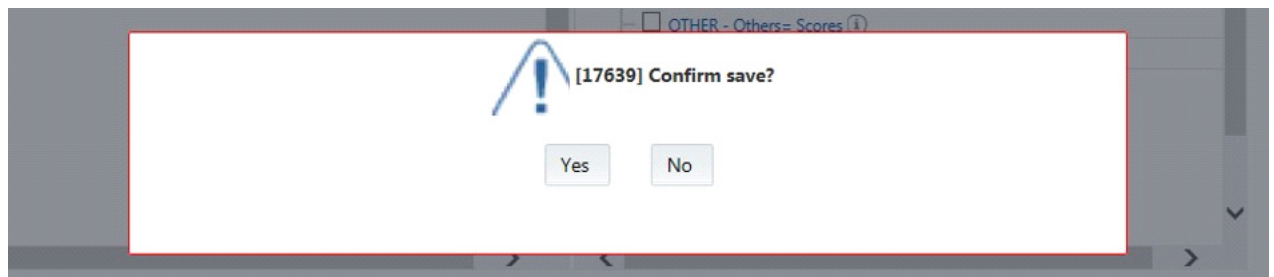


- **Many to One Mapping:** Many Credit Score Models can be mapped to one Regulatory Credit Score Model using the Mapper Maintenance screen. Here, you need to select two or more Credit Score Models and one Regulatory Credit Score Model. This means the corresponding data in Stage Account Credit Score Details Credit Score Model column for the selected Credit Score Models will be loaded into Fact Account Credit Score Details Regulatory Credit Score Model column while loading the T2T.

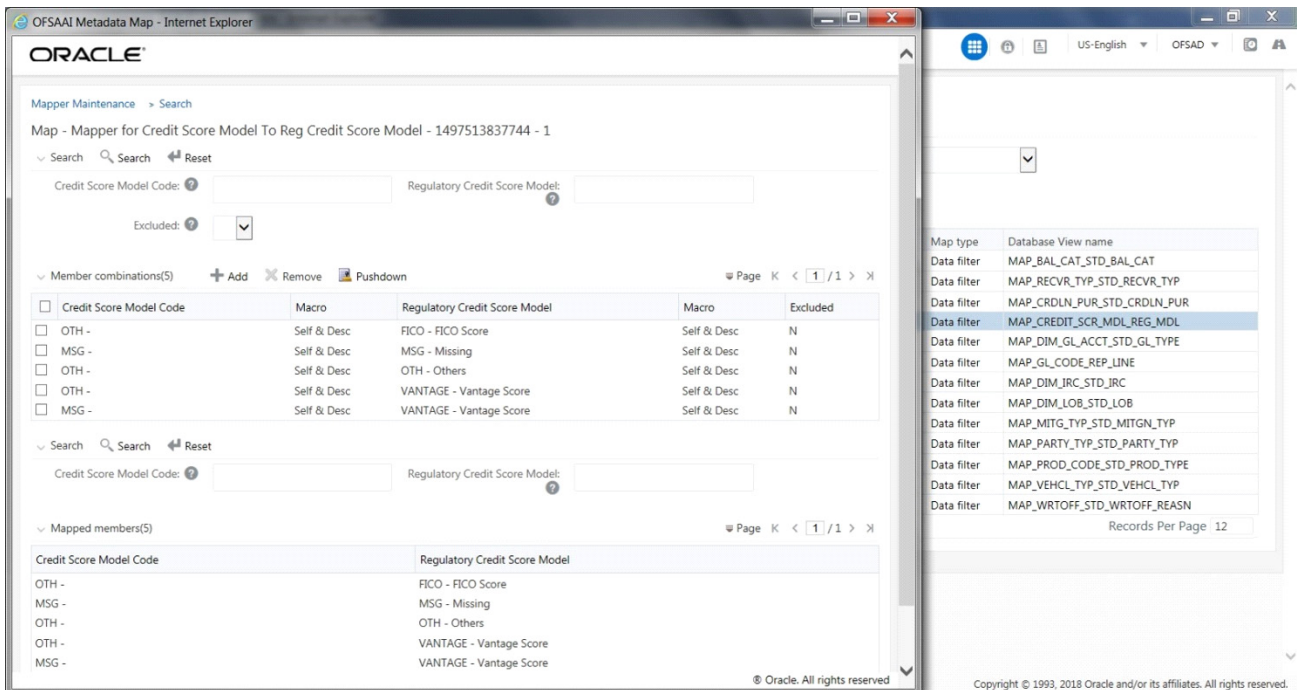
1. In the **Mapper Maintenance** window, click **Add**. The **Add Mappings** pop-up window is displayed.
2. In this example, **MSG – Missing** and **OTH - Others** are mapped to **VANTAGE - Vantage Score**.



3. To map, click **Go** and then to save the mapping, click **Save**. An acknowledgement is displayed. Click **Yes** to confirm.



4. The Mapped Members are displayed in the Mapper Maintenance Screen.



Loading Mapper Maintenance through Backend

Load the MAP_CREDIT_SCR_MDL_REG_MDL table in Atomic Schema with V_MAP_ID as 1497513837744, V_MEMBER_1 => Credit Score Model Code (values from DIM_CREDIT_SCORE_MODEL.V_CREDIT_SCORE_MODEL_CODE), V_MEMBER_2 => Regulatory Credit Score Model Code (values from DIM_REG_CREDIT_SCORE_MODEL.V_REG_CREDIT_SCORE_MODEL_CODE).

Note: These values are actual business key columns and not display codes.

```
SELECT * FROM MAP_CREDIT_SCR_MDL_REG_MDL;
```

	V_MAP_ID	N_MAP_ID	N_INHERIT_MAP_ID	V_MEMBER_1	V_MEMBER_2	V_MEMBER_3	V_MEMBER_4	V_MEMBER_5	V_MEMBER_6	V_MEMBER_7	V_MEMBER_8	V_MEMBER_9
1	1497513837744			VAN	VANTAGE							
2	1497513837744			OTH	OTH							
3	1497513837744			FICO2	FICO							
4	1497513837744			FICO1	FICO							
5	1497513837744			FICO3	FICO							
6	1497513837744			MSG	MSG							

Executing the Credit Score Model and Probability of Default Model T2Ts

Execution through Run Management

The T2Ts are part of *Financial Services Data Foundation Sourced Run*. The process can be executed through the Seeded Run *Financial Services Data Foundation Execution Run*.

Note: When executing through Run, the RUNSkey will be auto-generated and stamped against each record.

Error Messages

Following are the most common error message which will be logged in the T2T log file present in the *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA* folder:

- **Unique Constraint Violation:** This occurs when attempting re-load or loading existing records for the already executed AS_OF_DATE.

Checking the Execution Status

The status of execution can be monitored using the *Batch Monitor* screen.

Note: For a more comprehensive coverage of configuration and execution of a batch, see [OFS Analytical Applications Infrastructure User Guide](#).

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going
- F - Failure
- S - Success

The execution log can be accessed on the application server in the following directory *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA*. The file name will have the batch execution id.

Following are the error log tables in atomic schema:

- FCT_ACCT_CREDIT_SCORE_DETAILS\$
- FCT_SERV_ACCT_CREDIT_SCORE_DTL\$
- FCT_PARTY_PD_DETAILS\$
- FCT_INSTRUMENT_PD_DETAILS\$

Credit Score Model and Probability of Default Model Population T2Ts

T2T definitions can be retrieved as an excel document for reference from the metadata browser of the Unified Metadata Manager (UMM) component of OFSAAL.

This chapter provides information about GL to Management Reporting Tables in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview of GL to Management Reporting
- Overview of GL to Management Reporting Population
- Overview of Mapper for GL to Management Reporting
- Maintenance of Mapper for GL to Management Reporting
- Loading Mapper Maintenance through Backend
- Executing the GL to Management Reporting T2Ts
- Checking the Execution Status
- GL to Management Reporting T2Ts

Overview of GL to Management Reporting

Fact Management Reporting Table is loaded from respective Stage GL Data using a Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework and a Map Maintenance component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI). Following is the Results Table that stores Management Reporting Data:

- FCT_MGMT_REPORTING

Overview of GL to Management Reporting Population

SI No	Source Table Name	Mapper Table Name	Target Table Name	T2T Definition Name
1	STG_GL_DATA	MAP_GL_CODE_ REP_LINE	FCT_MGMT_RE- PORTING	T2T_FCT_MGMT_REPORT- ING_STG_GL_DATA

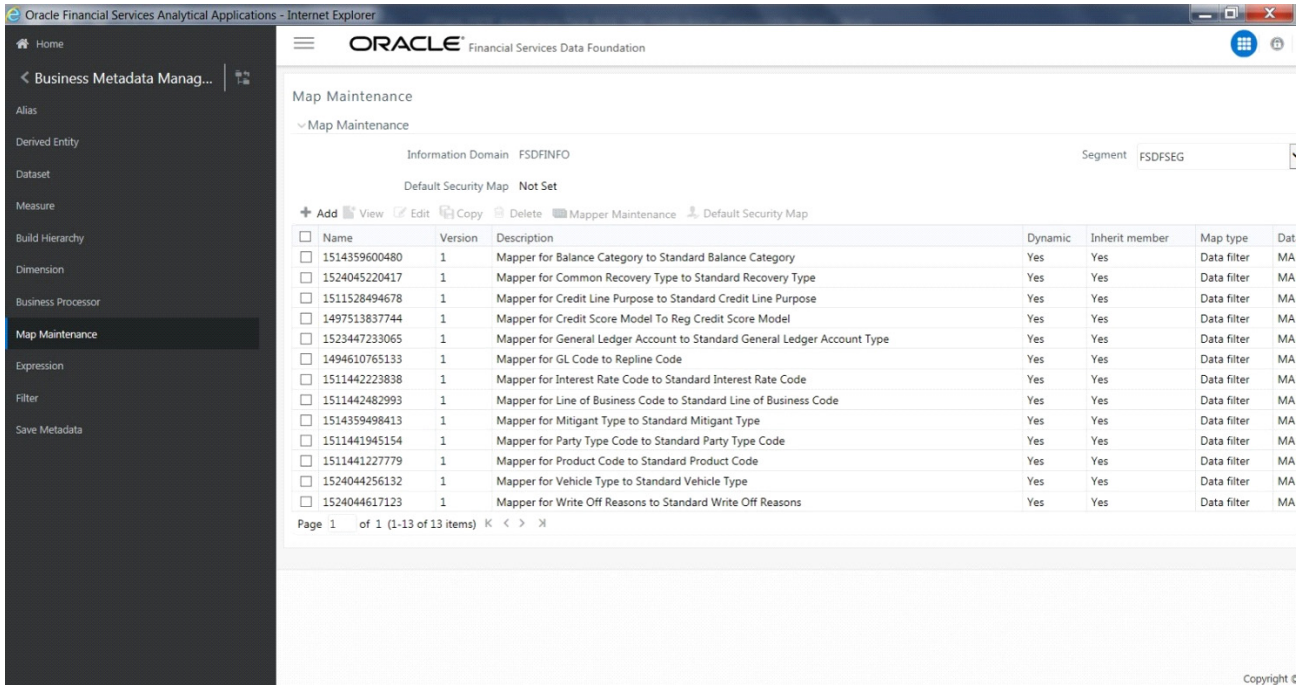
Overview of Mapper for GL to Management Reporting

Fact Management Reporting is a reporting table that uses reporting line codes which are OFSAA specific values that represent each line item of reporting requirements. On the other hand, the Stage GL Data is a staging table which uses the GL codes that are user specific values that represent each General Ledger Codes of users books. To load Fact Management Reporting, it needs mapping between GL Codes and Reporting Line Codes. These mappings are maintained through Mapper Maintenance component of OFSAAI.

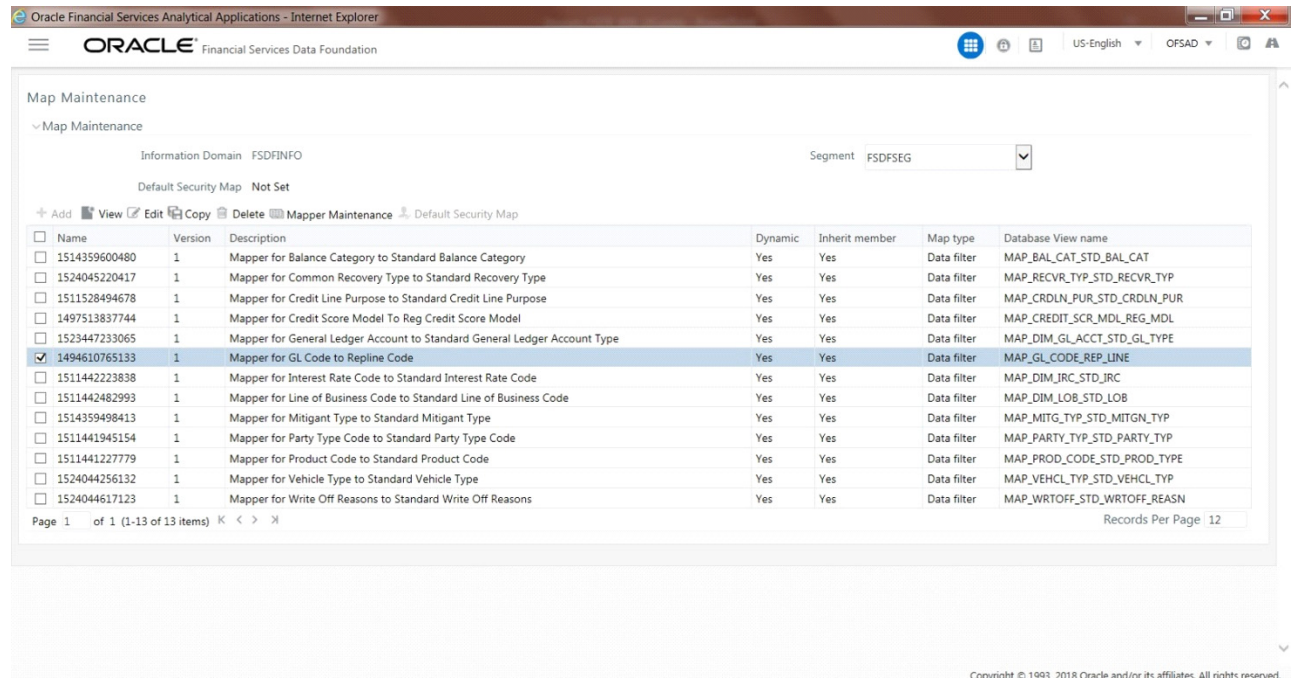
Maintenance of Mapper for GL to Management Reporting

Mapper can be maintained under OFSAAI.

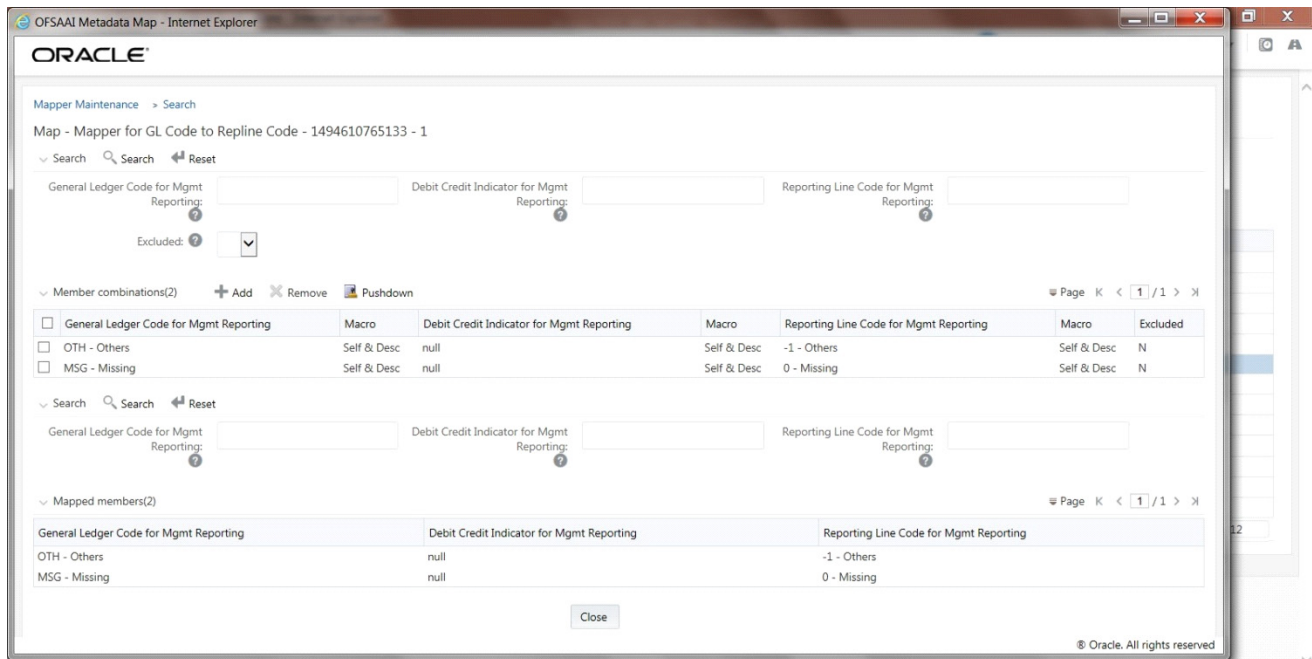
1. Navigate to **OFSAAI > Financial Services data Foundation > Unified Analytical Metadata > Business Metadata Management > Map Maintenance**.



2. Select Mapper for GL Code to Repline Code. Click Mapper Maintenance icon.



3. FSDF Maps OTH and MSG out-of-the-box for this mapper. The remaining Mappings can be maintained by the user according to user specific values.



Copyright © 1993, 2018 Oracle and/or its affiliates. All rights reserved.

Prerequisites for Mapper Maintenance

1. Load GL Dimension using SCD.
2. Resave the following hierarchies:
 - HMGMT001 - General Ledger Code for Mgmt Reporting
 - HMGMT002 - Debit Credit Indicator for Mgmt Reporting
 - HMGMT003 - Reporting Line Code for Mgmt Reporting

Possible Mapping Combinations

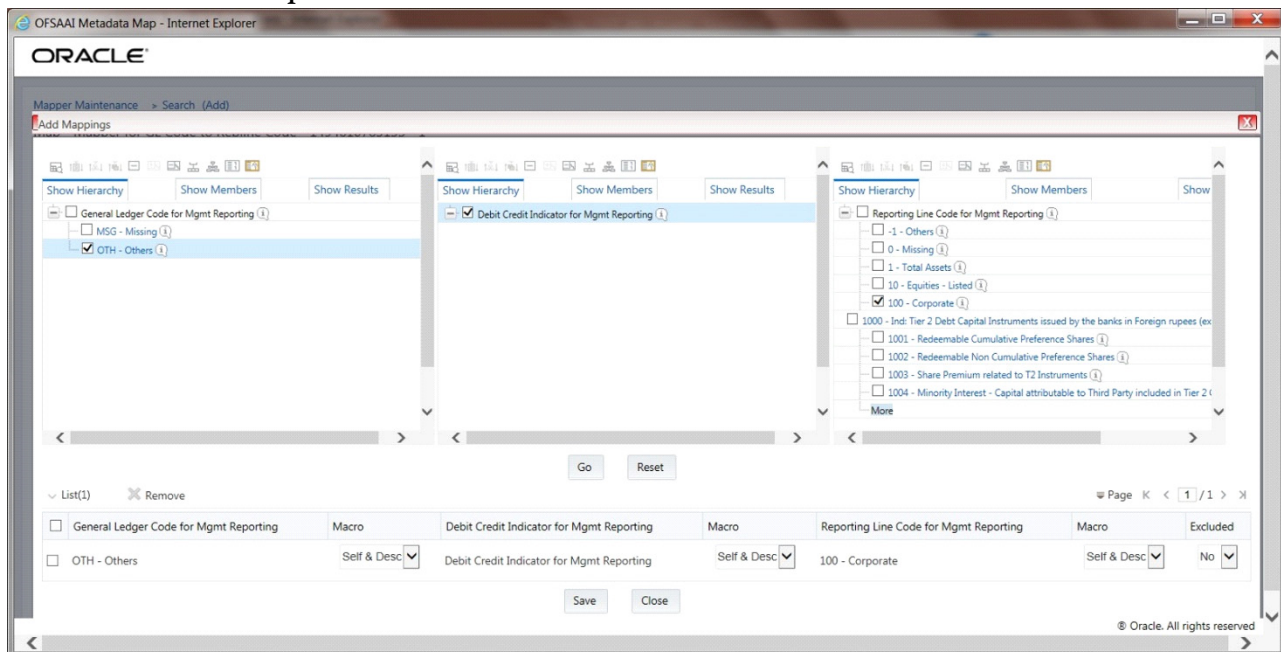
Following are the various mapping combinations for General Ledger Code for Mgmt Reporting Hierarchy individual notes:

- **One to One Mapping with/without Debit Credit Indicator:** One GL Code can be mapped to one Reporting Line Code using the Mapper Maintenance screen. Here, you need to select one GL Code, Debit Credit Indicator as either 'Debit' or 'Credit' or 'Missing' and one Reporting Line Code. This means the corresponding data in Stage GL data for the selected GL Code will be loaded into Fact Management Reporting while loading the T2T.

Note: The Debit Credit Indicator column of Stage GL data:

- Should be null, if mapped data for Debit Credit Indicator is selected as 'Missing'
 - Should be 'D', if mapped data for Debit Credit Indicator is selected as 'Debit'
 - Should be 'C', if mapped data for Debit Credit Indicator is selected as 'Credit'
1. Click **Add** button to open the *Mapper Maintenance Screen*.

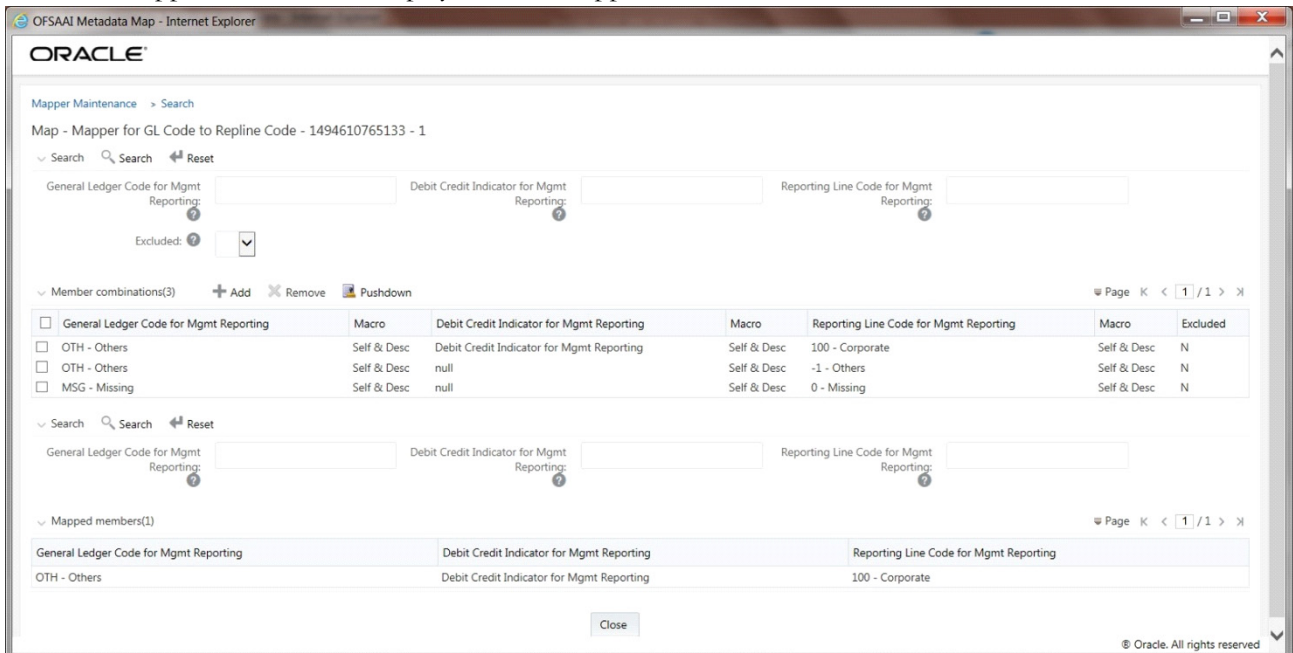
2. In this example, **OTH - Others** and **Debit Credit Indicator for Mgmt Reporting** is mapped to Reporting Line Code **100 - Corporate**.



3. To map, click **Go**. To save the mapping, click **Save**. An acknowledgement is displayed. Click **Yes** to confirm.



4. The Mapped Members are displayed in the Mapper Maintenance Screen.



- **Many to One with/without Debit Credit Indicator:** Many GL Codes can be mapped to one Reporting Line Code using the Mapper Maintenance screen. Here, you need to select two or more GL Codes, Debit Credit Indicator as either 'Debit' or 'Credit' or 'Missing' and one Reporting Line Code. This means the corresponding data in Stage GL data for the selected GL Codes will be aggregated and loaded into Fact Management Reporting while loading the T2T.

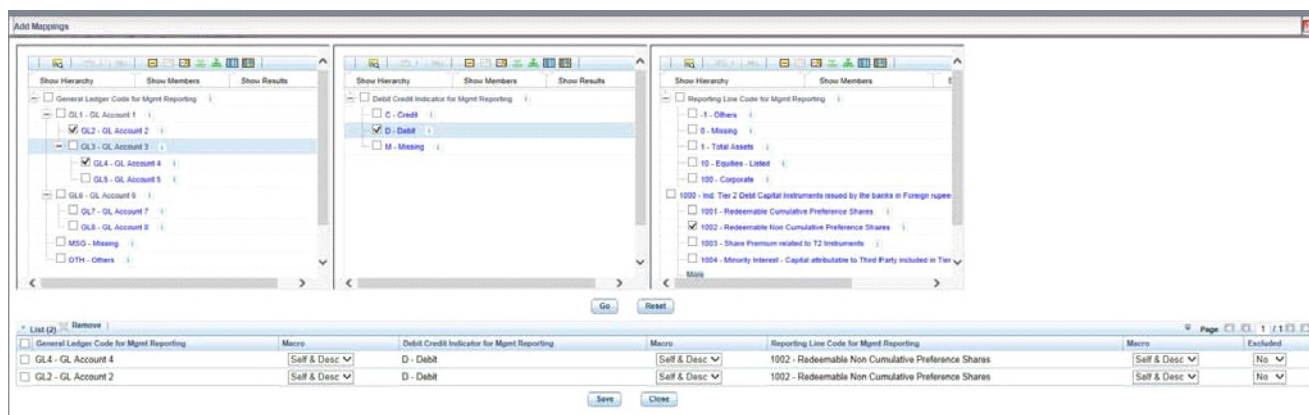
Note: The aggregation happens on all Measure Columns in T2T wrt all primary keys of Stage GL data. The Debit Credit Indicator column of Stage GL data:

- Should be null, if mapped data for Debit Credit Indicator is selected as 'Missing'
- Should be 'D', if mapped data for Debit Credit Indicator is selected as 'Debit'
- Should be 'C', if mapped data for Debit Credit Indicator is selected as 'Credit'

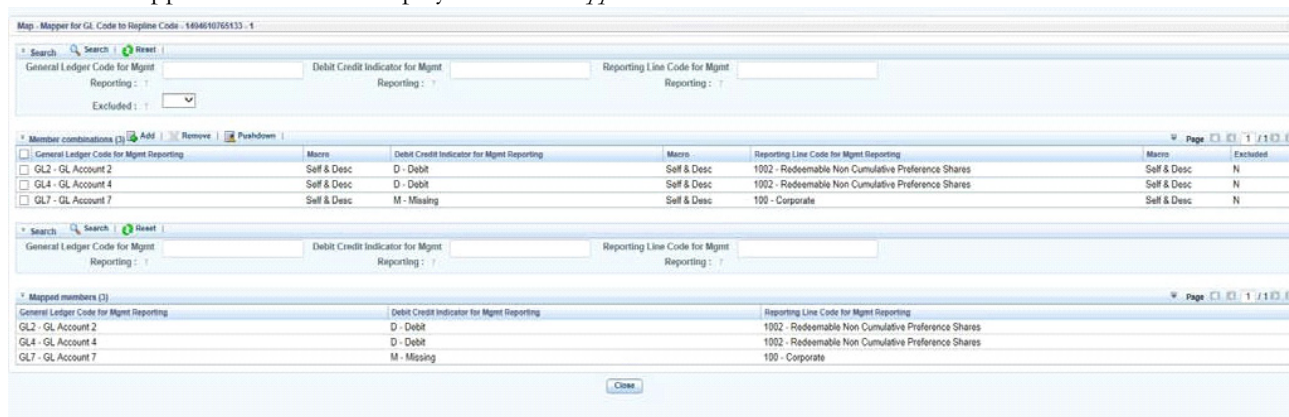
1. Click **Add** button to open the *Mapper Maintenance Screen*.

In this example, **GL2 - GL Account 2**, **GL4 - GL Account 4** and **D - Debit** is mapped to Reporting Line Code **1002 - Redeemable Non Cumulative Preference Shares**.

2. Click **Go** and **Save**. Click **Yes** to confirm.



3. The Mapped Members are displayed in the *Mapper Maintenance Screen*.



- **Many to Many with/without Debit Credit Indicator:** Many GL Codes can be mapped to many Reporting Line Codes using the Mapper Maintenance screen. Here, you need to select two or more GL Codes, Debit Credit Indicator as either 'Debit' or 'Credit' or 'Missing' and two or more Reporting Line Codes. This means the corresponding data in Stage GL data for the selected GL Codes will be aggregated and loaded into Fact Management Reporting for all Reporting Lines Codes selected while loading the T2T.

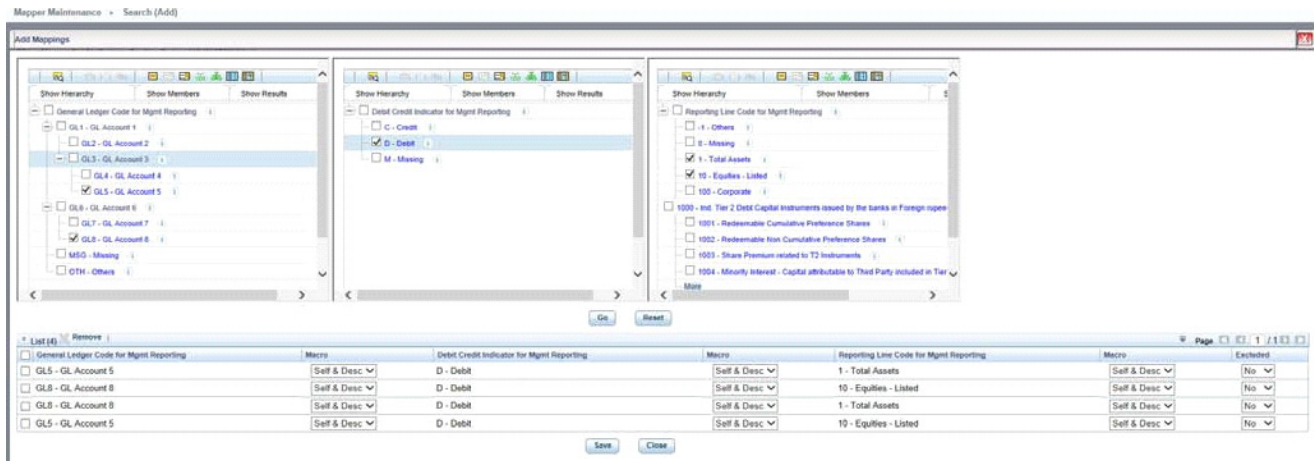
Note: The Debit Credit Indicator column of Stage GL data:

- Should be null, if mapped data for Debit Credit Indicator is selected as ‘Missing’
- Should be ‘D’, if mapped data for Debit Credit Indicator is selected as ‘Debit’
- Should be ‘C’, if mapped data for Debit Credit Indicator is selected as ‘Credit’

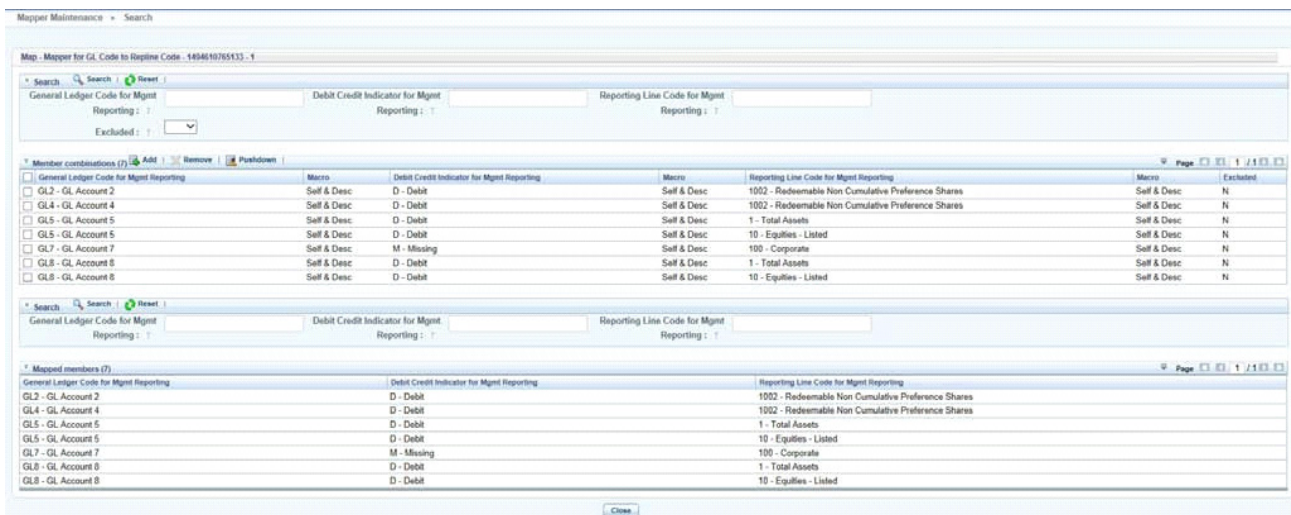
1. Click **Add** button to open the *Mapper Maintenance Screen*.

In this example, **GL5 - GL Account 5, GL8 - GL Account 8** and **D - Debit** is mapped to Reporting Line Code **1 - Total Assets** and **10 - Equities Listed**.

2. Click **Go** and **Save**. Click **Yes** to confirm.



3. The Mapped Members are displayed in the *Mapper Maintenance Screen*.



Following are the various mapping combinations for General Ledger Code for Mgmt Reporting Hierarchy is maintained as Parent Child Hierarchy:

- **One Parent to One Reporting Line Code Mapping with/without Debit Credit Indicator:** One parent GL Code can be mapped to one Reporting Line Code using the Mapper Maintenance screen. Here, you need to select one parent GL Code, Debit Credit Indicator as either ‘Debit’ or ‘Credit’ or ‘Missing’ and one Reporting Line Code. This means the corresponding data in Stage GL data for all the parent GL Code and all

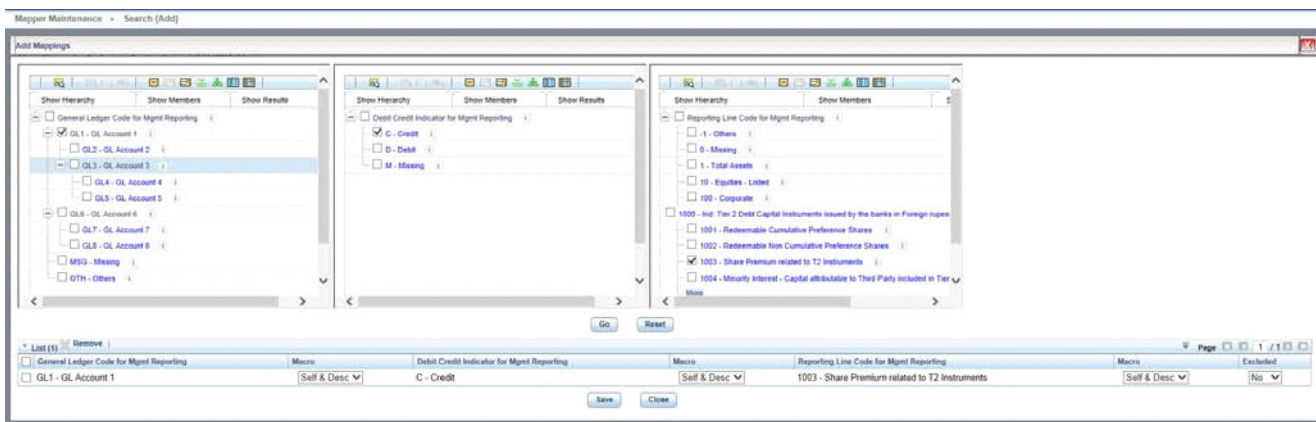
its descendants GL Codes will be aggregated and loaded into Fact Management Reporting while loading the T2T.

Note: The Debit Credit Indicator column of Stage GL data:

- Should be null, if mapped data for Debit Credit Indicator is selected as 'Missing'
- Should be 'D', if mapped data for Debit Credit Indicator is selected as 'Debit'
- Should be 'C', if mapped data for Debit Credit Indicator is selected as 'Credit'

1. Click **Add** button to open the *Mapper Maintenance Screen*.

In this example, **GL1 - GL Account 1** and **C - Credit** is mapped to Reporting Line Code **1003 - Share Premium related to T2 Instruments**.



2. Click **Go** and **Save**. Click **Yes** to confirm.

The Mapped Members are displayed in the *Mapper Maintenance Screen*. You will be able to see that all the descendants of the parent GL1 - GL Account 1 is mapped.



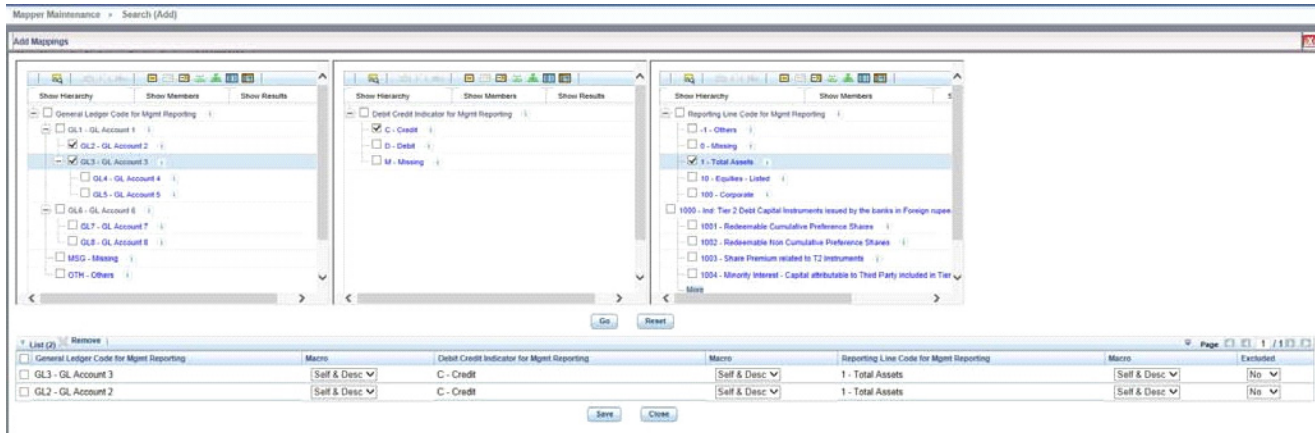
- **Many Parents to One Reporting Line Code with/without Debit Credit Indicator:** Many GL Codes can be mapped to one Reporting Line Code using the Mapper Maintenance screen. Here, you need to select two or more GL Codes, Debit Credit Indicator as either 'Debit' or 'Credit' or 'Missing' and one Reporting Line Code. This means the corresponding data in Stage GL data for the selected GL Codes will be aggregated and loaded into Fact Management Reporting while loading the T2T.

Note: The aggregation happens on all Measure Columns in T2T in reference to all primary keys of Stage GL data. The Debit Credit Indicator column of Stage GL data:

- Should be null, if mapped data for Debit Credit Indicator is selected as 'Missing'
- Should be 'D', if mapped data for Debit Credit Indicator is selected as 'Debit'
- Should be 'C', if mapped data for Debit Credit Indicator is selected as 'Credit'

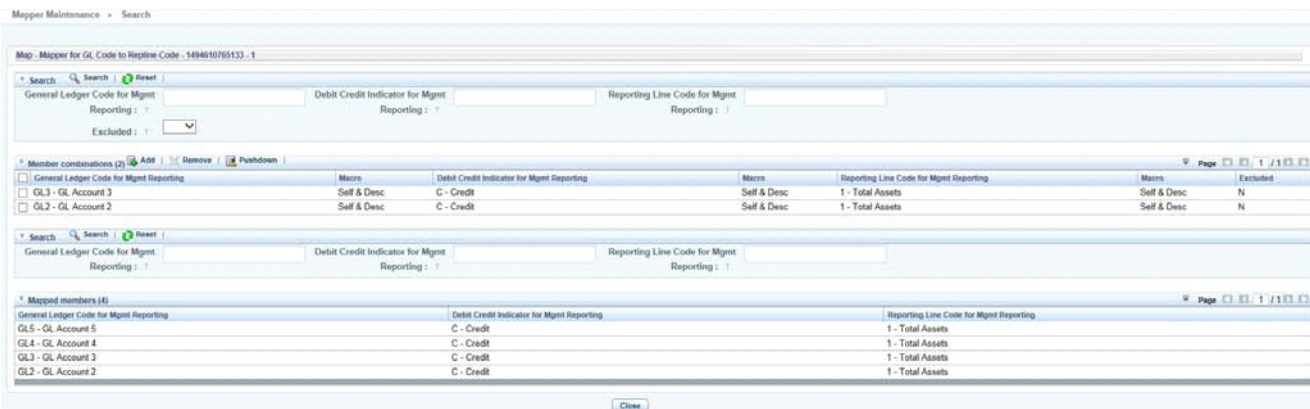
1. Click **Add** button to open the *Mapper Maintenance Screen*.

In this example, **GL2 - GL Account 2**, **GL3 - GL Account 3** and **C - Credit** is mapped to Reporting Line Code **1 - Total Assets**.



2. Click **Go** and **Save**. Click **Yes** to confirm.

The Mapped Members are displayed in the *Mapper Maintenance Screen*.



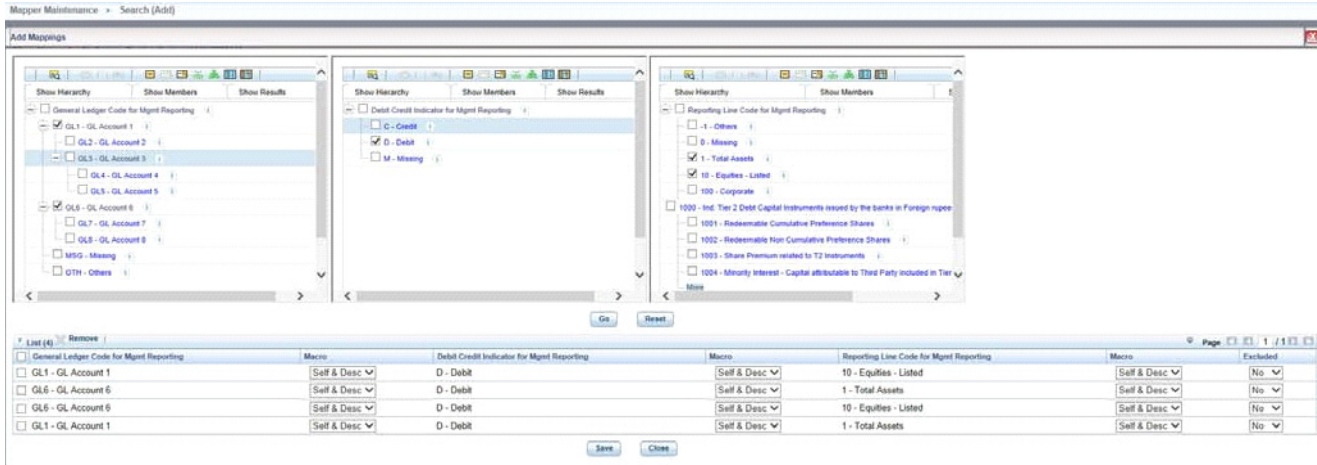
- **Many Parents to Many Reporting Line Codes with/without Debit Credit Indicator:** Many GL Codes can be mapped to many Reporting Line Codes using the Mapper Maintenance screen. Here, you need to select two or more GL Codes, Debit Credit Indicator as either 'Debit' or 'Credit' or 'Missing' and two or more Reporting Line Codes. This means the corresponding data in Stage GL data for the selected GL Codes will be aggregated and loaded into Fact Management Reporting for all Reporting Lines Codes selected while loading the T2T.

Note: The Debit Credit Indicator column of Stage GL data:

- Should be null, if mapped data for Debit Credit Indicator is selected as 'Missing'
- Should be 'D', if mapped data for Debit Credit Indicator is selected as 'Debit'
- Should be 'C', if mapped data for Debit Credit Indicator is selected as 'Credit'

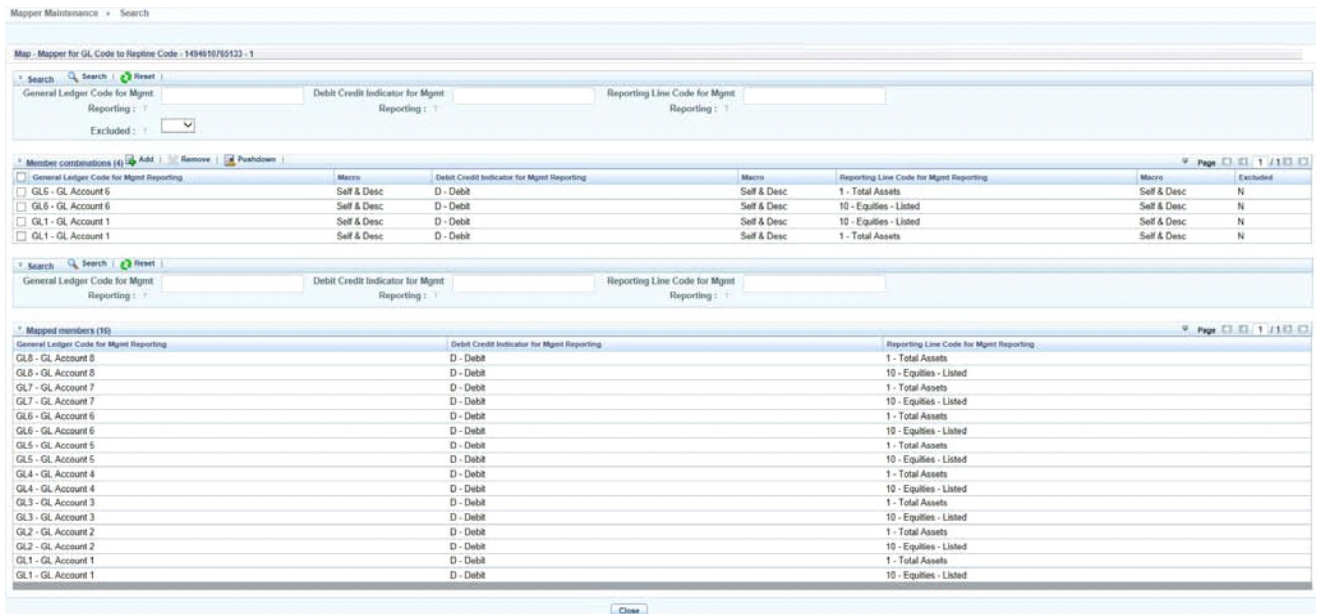
1. Click **Add** button to open the *Mapper Maintenance Screen*.

In this example, **GL1 - GL Account 1**, **GL6 - GL Account 6** and **D - Debit** is mapped to Reporting Line Code **1 - Total Assets** and **10 - Equities Listed**.



2. Click **Go** and **Save**. Click **Yes** to confirm.

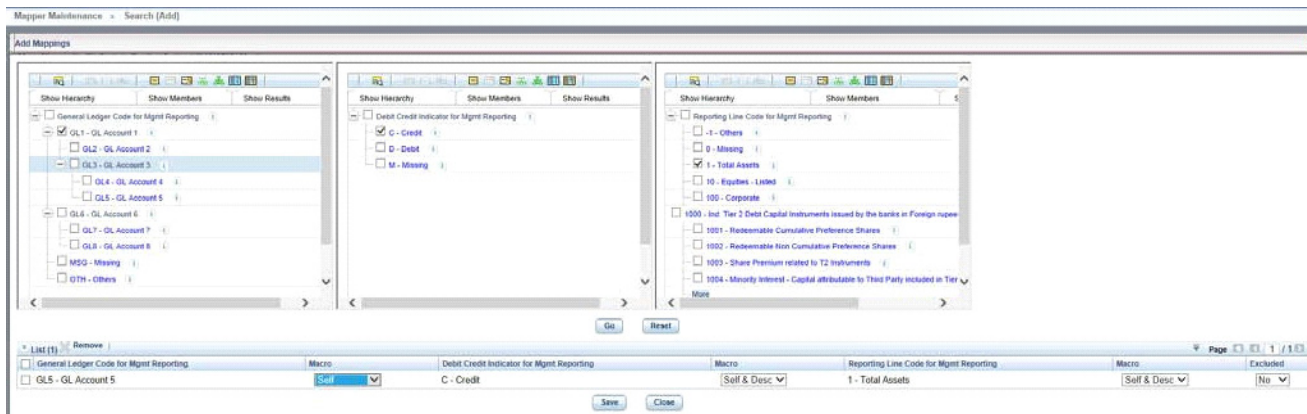
The Mapped Members are displayed in the *Mapper Maintenance Screen*.



- **One Parent to One or Many Reporting Line Codes without Descendants:** One parent GL Code can be mapped to one or many Reporting Line Codes without including any descendants by selecting 'Self' from Macro for GL Hierarchy under List grid which appears after selection of Mapping.

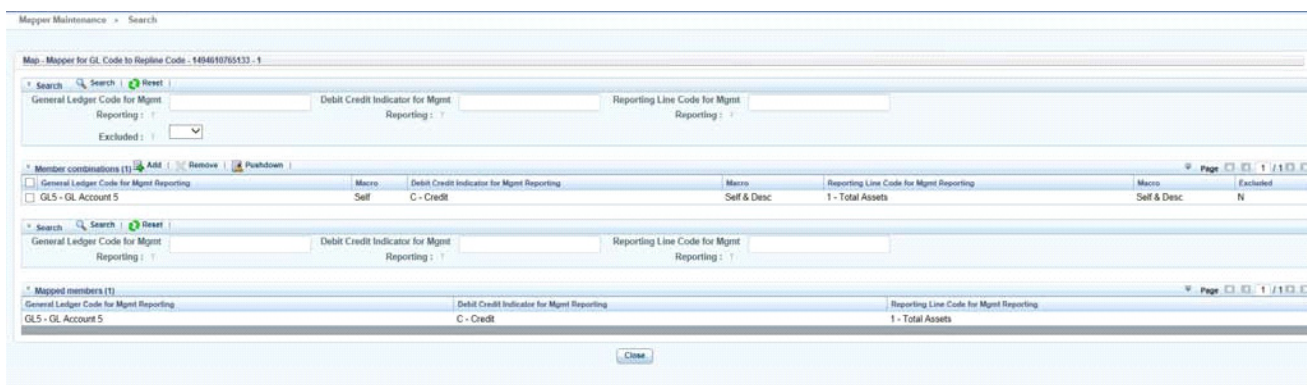
1. Click **Add** button to open the *Mapper Maintenance Screen*.

In this example, **GL1 - GL Account 1** and **C - Credit** is mapped to Reporting Line Code **1 - Total Assets**.



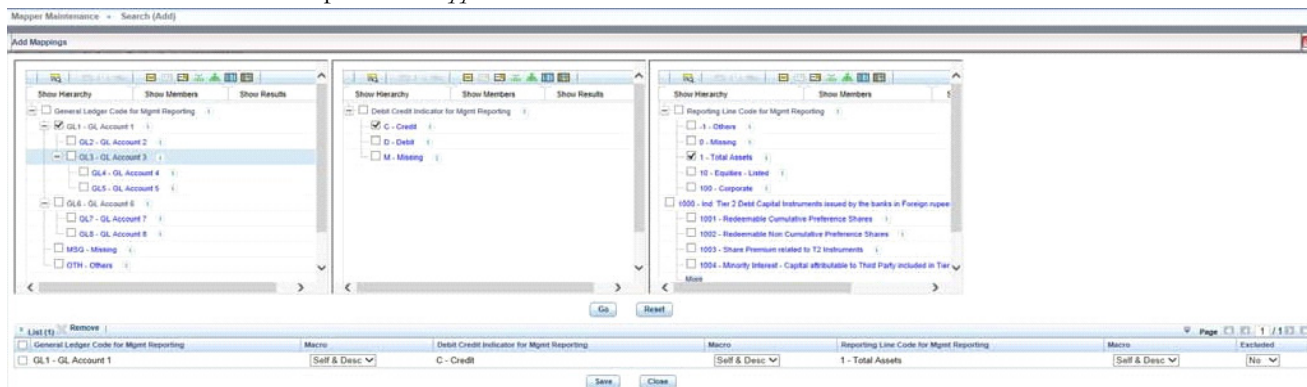
2. Click **Go** and **Save**. Click **Yes** to confirm.

The Mapped Members are displayed in the *Mapper Maintenance Screen*.



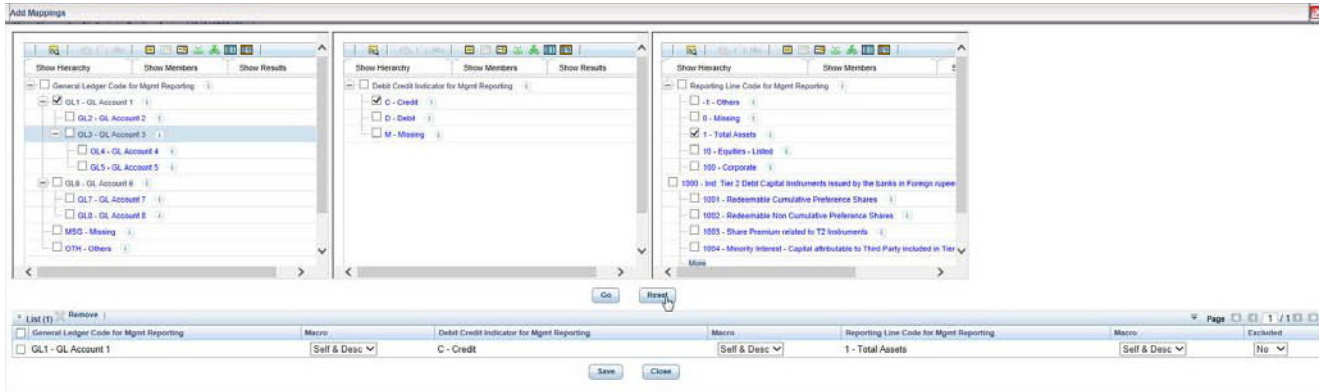
- **One Parent to One or Many Reporting Line Codes by Removing a few Descendants:** One parent GL Code can be mapped to one or many Reporting Line Codes and few descendants can be removed by selecting 'Self and Descendants' from Macro for GL Hierarchy under List grid which appears after selection of Mapping. Select either 'Yes' to Exclude or 'No' to include from the Exclude dropdown present in the List grid.

1. Click **Add** button to open the *Mapper Maintenance Screen*.

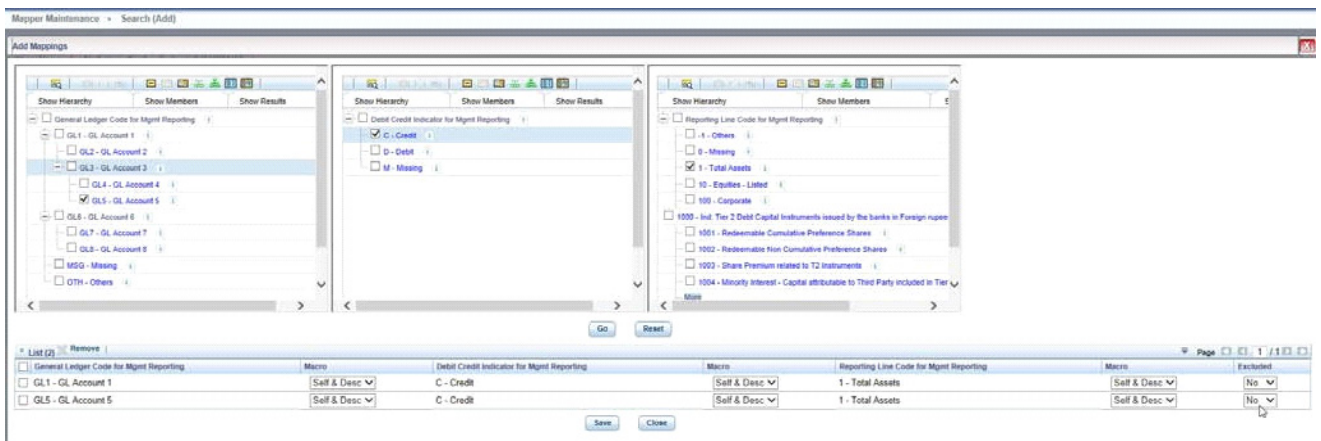


In this example, **GL1 - GL Account 1** and **C - Credit** is mapped to Reporting Line Code **1 - Total Assets**.

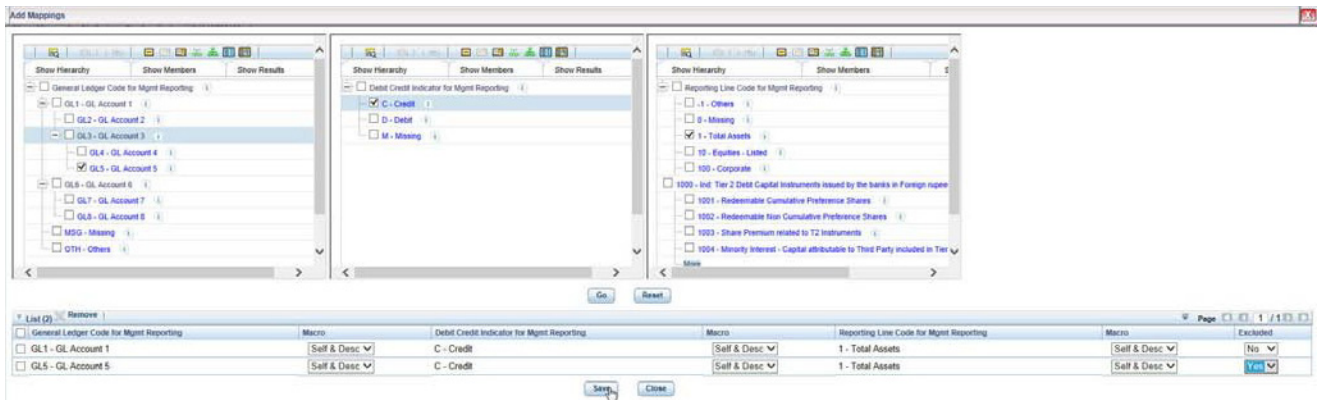
2. Click **Reset** to add another Mapping without saving the previous selection.



3. For second mapping, **GL5 - GL Account 5** and **C - Credit** is mapped to Reporting Line Code **1 - Total Assets**.

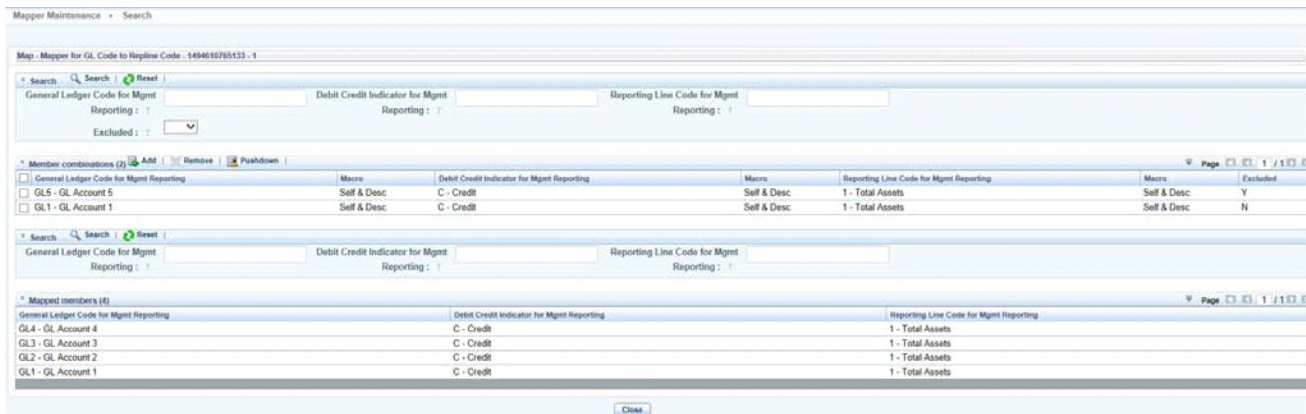


4. Select either 'Yes' to Exclude or 'No' to include from the Exclude dropdown present in the List grid..



5. Click **Save**. Click **Yes** to confirm.

The Mapped Members are displayed in the *Mapper Maintenance Screen*.



Loading Mapper Maintenance through Backend

Load the MAP_GL_CODE_REP_LINE table in Atomic Schema with V_MAP_ID as 194610765133, V_MEMBER_1 => GL Code (values from DIM_GL_ACCOUNT.V_GL_ACCOUNT_CODE), V_MEMBER_2 => Debit Credit Indicator (values should be C or D or M), V_MEMBER_3 => Reporting Line Code (values from DIM_REP_LINE.N_REP_LINE_CD).

Note: These values are actual business key columns and not display codes.

```
SELECT * FROM MAP_GL_CODE_REP_LINE
```

	V_MAP_ID	N_MAP_ID	N_INHERIT_MAP_ID	V_MEMBER_1	V_MEMBER_2	V_MEMBER_3	V_MEMBER_4	V_MEMBER_5	V_MEMBER_6	V_MEMBER_7	V_MEMBER_8	V_MEMBER_9
1	1494610765133			GL1	C	1						
2	1494610765133			GL2	C	1						
3	1494610765133			GL3	C	1						
4	1494610765133			GL4	C	1						

Executing the GL to Management Reporting T2Ts

Execution through Run Management

The T2T is part of *Financial Services Data Foundation Sourced Run*. The process can be executed through the Seeded Run *Financial Services Data Foundation Execution Run*.

Note: When executing through Run, the RUNSkey will be auto-generated and stamped against each record.

Error Messages

Following are the most common error message which will be logged in the T2T log file present in the *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA* folder:

- **Unique Constraint Violation:** This occurs when attempting re-load or loading existing records for the already executed AS_OF_DATE.

Checking the Execution Status

The status of execution can be monitored using the *Batch Monitor* screen.

Note: For a more comprehensive coverage of configuration and execution of a batch, see [OFS Analytical Applications Infrastructure User Guide](#).

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going
- F - Failure
- S - Success

The execution log can be accessed on the application server in the following directory *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA*. The file name will have the batch execution id.

Following are the error log tables in atomic schema:

- FCT_MGMT_REPORTING\$

GL to Management Reporting T2Ts

T2T definitions can be retrieved as an excel document for reference from the metadata browser of the Unified Metadata Manager (UMM) component of OFSAAL.

This chapter provides information about Populating various other Results Tables in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview of Other Results Tables
- Overview of Other Results Population
- Table to Table seeded definitions are provided in the target Dimension tables:
- Checking the Execution Status
- Various Other Results T2Ts

Overview of Other Results Tables

Other Results Tables are loaded from respective Stage tables using the Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework. Following are the Results Tables that stores other results:

- FCT_CUST_IDENTIFICATION_DOC
- FCT_ENTITY_PARENT_INFO
- FCT_INTRA_COMPANY_ACCT_SUMMARY
- FCT_ISSUED_INSTR_POSITIONS
- FCT_ACCOUNT_POSITION_PAIR
- FCT_ACCT_RECOVERY_DETAILS
- FCT_ACCT_WRITE_OFF_DETAILS
- FCT_ASSETS_SOLD
- FCT_CAP_INSTR_POSITIONS
- FCT_CAP_INSTR_TXNS
- FCT_CREDIT_LINE
- FCT_FIXED_ASSETS
- FCT_FUND_CIS_COMPOSITION
- FCT_INSTR_PROPOSED_TXNS
- FCT_LEGAL_ENTITY_DETAILS
- FCT_LITIGATION_DETAILS
- FCT_LOANS_SERVICED
- FCT_MERCHANT_BANKING
- FCT_PAYMENTS_SUMMARY

- FCT_RECOVERY
- FCT_SPEND_OBLIGATIONS
- FCT_TRD_ACCOUNT_TXN_SUMMARY
- FCT_SERV_LN_CUST_RELATIONSHIP
- FCT_ACCT_CUST_RELATIONSHIP
- FCT_CORPORATE_ACTIONS
- FCT_TRADE_EXECUTION
- FCT_SECURITIZATION_POOL
- DIM_CONSENT_PURPOSE
- DIM_CREDIT_REASON
- DIM_COMMODITY_GRADE
- DIM_SUB_NETTING_AGREEMENT
- DIM_LEGAL_PROCEEDING_STATUS

Overview of Other Results Population

Table to Table seeded definitions are provided for loading data into the target Fact tables:

Table 15. Table to Table Seeded Definitions for Fact Target Tables

Sl. No.	Source Table Name	Target Table Name	T2T Definition Name
1	STG_ASSETS_SOLD	FCT_ACCOUNT_POSTION_PAIR	T2T_FCT_ASSETS_SOLD
2	STG_ENTITY_PARENT_DETAILS	FCT_ENTITY_PARENT_INFO	T2T_FCT_ENTITY_PARENT_INFO
3	STG_ISSUED_INSTR_POSITIONS	FCT_ISSUED_INSTR_POSITIONS	T2T_FCT_ISSUED_INSTR_POSITIONS
4	STG_CUSTOMER_IDENTIFICATION_DOC	FCT_CUST_IDENTIFICATION_DOC	T2T_FCT_CUST_IDENTIFICATION_DOC
5	FCT_COMMON_ACCOUNT_SUMMARY	FCT_INTRA_COMPANY_ACCOUNT_SUMMARY	T2T_FCT_INTRA_COMPANY_ACCT_SUMMARY
6	STG_LOANS_SERVICED	FCT_ACCT_RECOVERY_DETAILS	T2T_FCT_LOANS_SERVICED
7	STG_FIXED_ASSETS_DETAILS	FCT_ACCT_WRITE_OFF_DETAILS	T2T_FCT_FIXED_ASSETS
8	STG_CREDIT_LINE_DETAILS	FCT_ASSETS_SOLD	T2T_FCT_CREDIT_LINE
9	STG_LEGAL_ENTITY_DETAILS	FCT_CAP_INSTR_POSITIONS	T2T_FCT_LEGAL_ENTITY_DETAILS

Table 15. Table to Table Seeded Definitions for Fact Target Tables

Sl. No.	Source Table Name	Target Table Name	T2T Definition Name
10	STG_CAP_INSTR_POSITIONS	FFCT_CAP_INSTR_TXNS	T2T_FCT_CAP_INSTR_POSITIONS
11	STG_CAP_INSTR_TXNS	FCT_CAP_INSTR_TXNS	T2T_FCT_CAP_INSTR_TXNS
12	STG_FUND_CIS_COMPOSITION	FCT_FUND_CIS_COMPOSITION	T2T_FCT_FUND_CIS_COMPOSITION
13	STG_MERCHANT_BANKING	FCT_MERCHANT_BANKING	T2T_FCT_MERCHANT_BANKING
14	STG_SPEND_OBLIGATIONS	FCT_SPEND_OBLIGATIONS	T2T_FCT_SPEND_OBLIGATIONS
15	STG_ACCT_WRITE_OFF_DETAILS	FCT_ACCT_WRITE_OFF_DETAILS	T2T_FCT_ACCT_WRITE_OFF_DETAILS
16	STG_RECOVERIES	FCT_ACCT_RECOVERY_DETAILS	T2T_FCT_ACCT_RECOVERY_DETAILS
17	STG_ACCT_RECOVERY_DETAILS	FCT_RECOVERY	T2T_FCT_RECOVERY
18	STG_INSTR_PROPOSED_TXNS	FCT_INSTR_PROPOSED_TXNS	T2T_FCT_INSTR_PROPOSED_TXNS
19	STG_LITIGATION_DETAILS	FCT_LITIGATION_DETAILS	T2T_FCT_LITIGATION_DETAILS
20	STG_SERV_LN_CUST_RELATIONSHIP	FCT_SERV_LN_CUST_RELATIONSHIP	T2T_FCT_SERV_LN_CUST_RELATIONSHIP
21	STG_PARTY_ACCOUNT_ROLE_MAP	FCT_ACCT_CUST_RELATIONSHIP	T2T_FACR_STG_PARTY_ACCOUNT_ROLE_MAP
22	STG_CORPORATE_ACTIONS	FCT_CORPORATE_ACTIONS	T2T_FCT_CORPORATE_ACTIONS
23	STG_TRADE_EXECUTION	FCT_TRADE_EXECUTION	T2T_FCT_TRADE_EXECUTION
24	STG_SECURITIZATION_POOL	FCT_SECURITIZATION_POOL	T2T_FCT_SECURITIZATION_POOL

Table to Table seeded definitions are provided in the target Dimension tables:

Table 14. The SCDs for the Target Dimension Tables

SCD Map Reference Number	Source Table Name	Target Dimension Table Name
469	STG_CREDIT_REASON_MASTER	DIM_CREDIT_REASON
470	STG_COMMODITY_GRADE_MASTER	DIM_COMMODITY_GRADE
471	STG_SUB_NETTING_AGREEMENT	DIM_SUB_NETTING_AGREEMENT
472	STG_LEGAL_PROCEEDING_STATUS_MASTER	DIM_LEGAL_PROCEEDING_STATUS

Executing the Various Other T2T Batches

Executing T2Ts through Batch Framework

Other Results T2T can be executed by executing task present in the seeded batch FSDF_SOURCED_RUN.

Executing SCDs through Seeded Batches

Other Results Dimension SCDs can be executed by executing task present in the SCD batch FSDFINFO_DATA_FOUNDATION_SCD.

Following steps will help you to execute the batch:

1. Navigate to the *Batch Execution* screen.
2. Select the seeded batch
 - FSDFINFO_DATA_FOUNDATION_SCD for target Dimension tables
 - FSDF_SOURCED_RUN for target Fact tables without RUNSkey
 - FSDF_EXE_RUN for FCT_SECURITIZATION_POOL which consists of RUNSkey.
3. Select the AS_OF_DATE for which source customer information is required to be loaded into the table.
4. Click **Execute Batch**.
5. Monitor the status of the batch from *Batch Monitor* screen of OFSAAI.

Note: While executing through batch, the RUN Skey will be defaulted to -1.

Execution through Run Management

These T2Ts are part of *Financial Services Data Foundation Sourced Run*. The process can be executed through the Seeded Run *Financial Services Data Foundation Execution Run*.

- T2T_FCT_ENTITY_PARENT_INFO
- T2T_FCT_ISSUED_INSTR_POSITIONS
- T2T_FCT_CUST_IDENTIFICATION_DOC

-
- T2T_FCT_INTRA_COMPANY_ACCT_SUMMARY
 - T2T_FCT_ACCT_RECOVERY_DETAILS
 - T2T_FCT_ACCT_WRITE_OFF_DETAILS
 - T2T_FCT_ASSETS_SOLD
 - T2T_FCT_CAP_INSTR_POSITIONS
 - T2T_FCT_CAP_INSTR_TXNS
 - T2T_FCT_CREDIT_LINE
 - T2T_FCT_FIXED_ASSETS
 - T2T_FCT_FUND_CIS_COMPOSITION
 - T2T_FCT_INSTR_PROPOSED_TXNS
 - T2T_FCT_LEGAL_ENTITY_DETAILS
 - T2T_FCT_LITIGATION_DETAILS
 - T2T_FCT_LOANS_SERVICED
 - T2T_FCT_MERCHANT_BANKING
 - T2T_FCT_PAYMENTS_SUMMARY
 - T2T_FCT_RECOVERY
 - T2T_FCT_SPEND_OBLIGATIONS
 - T2T_FCT_SERV_LN_CUST_RELATIONSHIP
 - T2T_FACR_STG_PARTY_ACCOUNT_ROLE_MAP
 - T2T_FCT_CORPORATE_ACTIONS
 - T2T_FCT_TRADE_EXECUTION
 - T2T_FCT_SECURITIZATION_POOL

Note: When executing through Run, the RUNSkey will be auto-generated and stamped against each record.

Error Messages

In the log file present in the:

- *ftpshare/logs/<Run_Date>/FSDFINFO/RUN EXECUTABLE* folder for Dimension tables
- *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA* folder for T2T's

This is the most common error message:

- **Unique Constraint Violation** : This occurs when attempting re-load or loading existing records for the already executed AS_OF_DATE.

Checking the Execution Status

The status of execution can be monitored in the **Batch Monitor** screen.

Note: For a more comprehensive coverage of configuration and execution of a batch, see [OFS Analytical Applications Infrastructure User Guide](#).

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going
- F - Failure
- S - Success

The execution log for Dimension tables can be accessed on the application server in the directory *ftpshare/logs/<Run_Date>/FSDFINFO/RUN EXECUTABLE*. The file name consists of the Batch Execution ID.

The execution log for T2Ts can be accessed on the application server in the directory *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA*. The file name consists of the Batch Execution ID.

Following are the error log tables in atomic schema:

- FCT_CUST_IDENTIFICATION_DOC\$
- FCT_ENTITY_PARENT_INFO\$
- FCT_INTRA_COMPANY_ACCT_SUMMARY\$
- FCT_ISSUED_INSTR_POSITIONS\$
- FCT_ACCOUNT_POSTION_PAIR\$
- FCT_ACCT_RECOVERY_DETAILS\$
- FCT_ACCT_WRITE_OFF_DETAILS\$
- FCT_ASSETS_SOLD\$
- FCT_CAP_INSTR_POSITIONS\$
- FCT_CAP_INSTR_TXNS\$
- FCT_CREDIT_LINE\$
- FCT_FIXED_ASSETS\$
- FCT_FUND_CIS_COMPOSITION\$
- FCT_INSTR_PROPOSED_TXNS\$
- FCT_LEGAL_ENTITY_DETAILS\$
- FCT_LITIGATION_DETAILS\$
- FCT_LOANS_SERVICED\$
- FCT_MERCHANT_BANKING\$
- FCT_PAYMENTS_SUMMARY\$
- FCT_RECOVERY\$
- FCT_SPEND_OBLIGATIONS\$
- FCT_SERV_LN_CUST_RELATIONSHIP\$

-
- FCT_ACCT_CUST_RELATIONSHIP\$
 - FCT_CORPORATE_ACTIONS\$
 - FCT_TRADE_EXECUTION\$
 - FCT_SECURITIZATION_POOL\$

Various Other Results T2Ts

T2T definitions can be retrieved as an excel document for reference from the metadata browser of the Unified Metadata Manager (UMM) component of OFSAAL.

Basel Processing to FSDF Results Integration

This chapter provides information about Basel Processing to FSDF Results Integration in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview of Basel Processing to FSDF Results Integration Tables
- Overview of Basel Processing to FSDF Results Integration
- Executing the BASEL Processing to FSDF Results Integration T2Ts
- Checking the Execution Status
- BASEL Processing to FSDF Results Integration Results T2Ts

Overview of Basel Processing to FSDF Results Integration Tables

As part of Basel processing to FSDF results integration, FSDF tables are loaded from Basel Processing tables using Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework. Following are the Results Tables that stores integrated results:

- FCT_FORECAST_REG_CAP_SUMMARY
- FCT_MITIGANT_REG_CAPITAL
- FCT_MR_CAPITAL_SUMMARY
- FCT_MR_VAR_PORTFOLIO_SUMMARY
- FCT_MR_VAR_SUMMARY
- FCT_REG_ACCT_MITIGANT_MAPPING
- FCT_REG_CAP_PLCD_COLL_SUMMARY
- FCT_REG_CAP_POOL_SUMMARY
- FCT_REG_CP_CAPITAL_SUMMARY
- FCT_REG_LE_CAPITAL_SUMMARY
- FCT_REG_OR_CAPITAL_SUMMARY
- FCT_REG_POOL_MITIGANT_MAP
- FCT_REG_CAP_ACCOUNT_SUMMARY

Note: As part of Basel processing results to FSDF integration, FSDF is packaging the forementioned T2Ts. These are optional T2Ts that will be deployed only when OFS_CAP_ADQ_PACK is installed.

Overview of Basel Processing to FSDF Results Integration

Table-to-Table seeded definitions are provided for loading data into the target tables:

Table 16. Table to Table Seeded Definitions

Sl. No.	Source Table Name	Target Table Name	T2T Definition Name
1	FSI_FORECAST_RWA, FSI_FORECAST_RWA_ALL OC_REP	FCT_FORECAST_REG_CA P_SUMMARY	T2T_FCT_FORECAST_REG _CAP_SUMMARY
2	FCT_MITIGANTS, FCT_SUB_EXPOSURES	FCT_MITIGANT_REG_CAP ITAL	T2T_FCT_MITIGANT_REG _CAPITAL
3	FCT_MARKET_RISK_COM _CAPITAL, FCT_MARKET_RISK_EXP OSURES	FCT_MR_CAPITAL_SUMM ARY	T2T_FCT_MR_CAPITAL_S UMMARY_FMRCC
4	FCT_MARKET_RISK_EQ_C APITAL, FCT_MARKET_RISK_EXP OSURES	FCT_MR_CAPITAL_SUMM ARY	T2T_FCT_MR_CAPITAL_S UMMARY_FMREQC
5	FCT_MARKET_RISK_FOR EX_CAPITAL, FCT_MARKET_RISK_EXP OSURES	FCT_MR_CAPITAL_SUMM ARY	T2T_FCT_MR_CAPITAL_S UMMARY_FMRFRXC
6	FCT_MARKET_RISK_IR_C APITAL, FCT_MARKET_RISK_EXP OSURES	FCT_MR_CAPITAL_SUMM ARY	T2T_FCT_MR_CAPITAL_S UMMARY_FMRIRC
7	FCT_MR_VAR_SUMMARY _DATA	FCT_MR_VAR_PORTFOLI O_SUMMARY	T2T_FCT_MR_VAR_PORTF OLIO_SUMMARY
8	FCT_MR_VAR_SUMMARY _DATA, FCT_MR_VAR_TOTAL_DA TA	FCT_MR_VAR_SUMMARY	T2T_FCT_MR_VAR_SUMM ARY
9	EXP_MITIGANT_MAPPING	FCT_REG_ACCT_MITIGAN T_MAPPING	T2T_FCT_REG_ACCT_MIT IGANT_MAPPING
10	FSI_PLACED_COLLATERA L	FCT_REG_CAP_PLCD_COL L_SUMMARY	T2T_FCT_REG_CAP_PLCD _COLL_SUMMARY
11	FCT_NETTABLE_POOL	FCT_REG_CAP_POOL_SU MMARY	T2T_FCT_REG_CAP_POOL _SUMMARY
12	FCT_REG_COUNTERPART Y_CVA, FCT_NETTABLE_POOL	FCT_REG_CP_CAPITAL_S UMMARY	T2T_FCT_REG_CP_CAPIT AL_SUMMARY
13	FCT_STANDARD_ACCT_H EAD	FCT_REG_LE_CAPITAL_S UMMARY	T2T_FCT_REG_LE_CAPIT AL_SUMMARY

Sl. No.	Source Table Name	Target Table Name	T2T Definition Name
14	FCT_OPS_RISK_DATA	FCT_REG_OR_CAPITAL_SUMMARY	T2T_FCT_REG_OR_CAPITAL_SUMMARY
15	EXP_MITIGANT_MAPPING	FCT_REG_POOL_MITIGANT_MAP	T2T_FCT_REG_POOL_MITIGANT_MAP
16	FCT_NON_SEC_EXPOSURES, FCT_SUB_EXPOSURES	FCT_REG_CAP_ACCOUNT_SUMMARY	T2T_FRCAS_FCT_NON_SEC_EXPOSURES
17	FCT_SEC_EXPOSURES, FCT_SUB_EXPOSURES	FCT_REG_CAP_ACCOUNT_SUMMARY	T2T_FRCAS_FCT_SEC_EXPOSURES
18	FCT_NON_SEC_EXPOSURES,	FCT_REG_CAP_ACCOUNT_SUMMARY	T2T_FRCAS_FCT_NON_SEC_EXPOSURES_CHILD
19	FCT_NON_SEC_EXPOSURES	FCT_REG_CAP_ACCOUNT_SUMMARY	T2T_FRCAS_FCT_NON_SEC_EXPOSURES_PARENT
20	FCT_SEC_EXPOSURES	FCT_REG_CAP_ACCOUNT_SUMMARY	T2T_FRCAS_FCT_SEC_EXPOSURES_CHILD
21	FCT_SEC_EXPOSURES	FCT_REG_CAP_ACCOUNT_SUMMARY	T2T_FRCAS_FCT_SEC_EXPOSURES_PARENT
22	FCT_MARKET_RISK_EXPOSURES	FCT_REG_MARKET_RISK_EXPOSURES	T2T_FCT_REG_MARKET_RISK_EXPOSURES

Executing the BASEL Processing to FSDF Results Integration T2Ts

For Basel FSDF integration, you must have FSDF, and Basel installed on the same INFODOM. Also, you must ensure that FSDF and Basel are running the same version.

There are two ways to integrate Basel and FSDF:

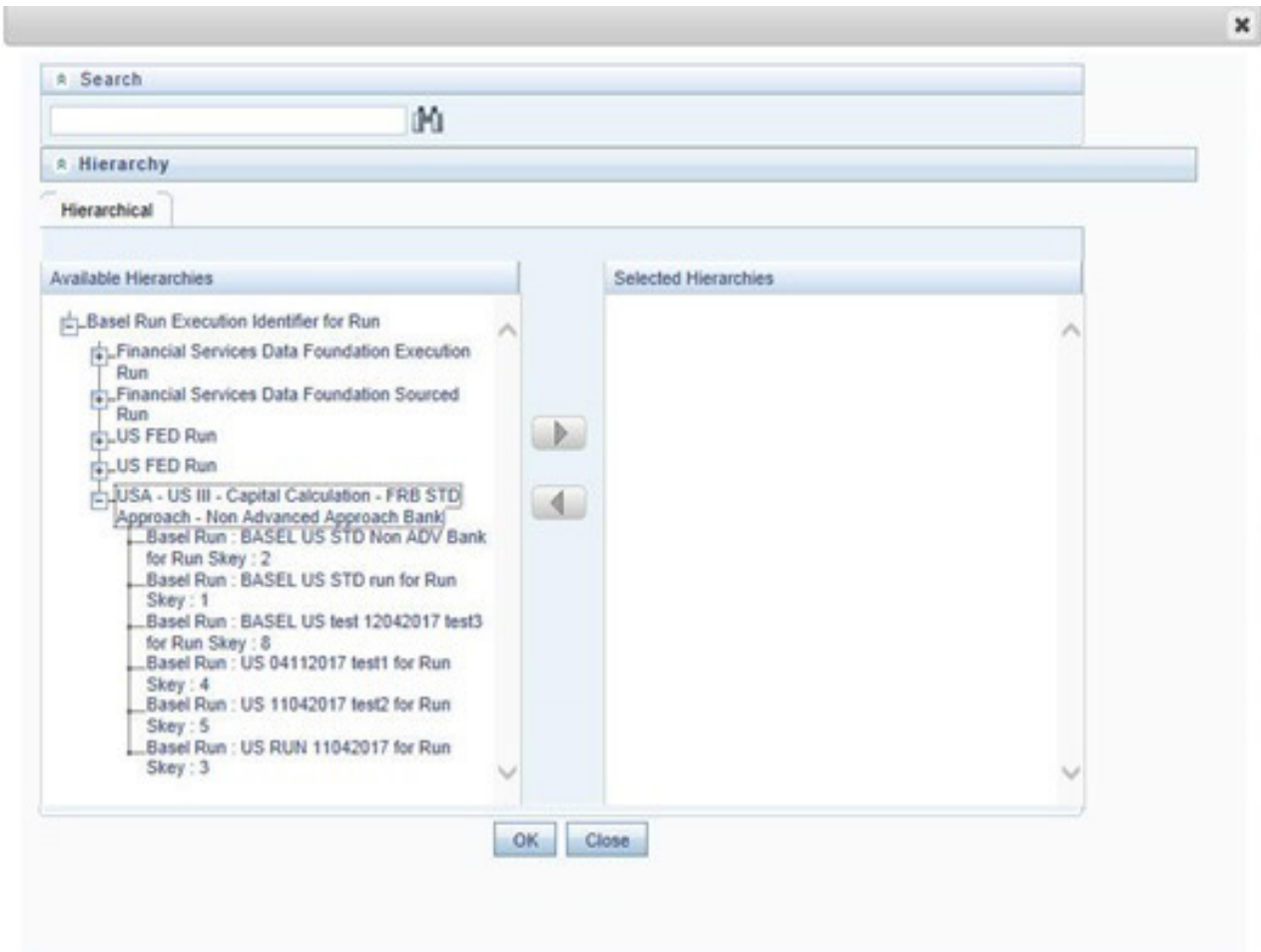
1. Creating Integrated Run at Implementation Site: During implementation, you can merge the tasks of both BASEL and FSDF and create an integrated Run to execute each time. The processes inside Run should be ordered as Basel first, then FSDF, and finally the Basel FSDF Integration process. In this Run, the Basel processing area and the FSDF results area tables must have the same Run SKEY across all tables.

Note: For BASEL-FSDF Integration Run, please use the FSDF Run Management screen as the Request Report Flag, Override Report Flag, and Approve Report Flag options are not available in the Basel Run Management Screen to enable the Reporting Flag.

2. Using approved Basel Run Execution ID in FSDF Run: In this case, you can use the out-of-the-box Basel Run as is for execution. After the execution, if the values are correct, you can execute the out-of-the-box FSDF Run by selecting the required Basel Run SKEY from Run Management screen. In this case, Basel processing area has one RUN SKEY and for the same data, FSDF has a different RUN SKEY in FSDF results area tables, where the data is getting reported. Sample report generation is as follows:
 - a. Login to Oracle Financial Services Analytical Applications interface with your credentials.
 - b. Navigate to **Applications > Financial Services Data Foundation > Run Management > Run Management**.
 - c. Select **Run** and click **Run Execution Summary** icon.

- d. The *Run Details and Run Execution Parameters* window is displayed.
- e. Enter the **Run Name** and **Run Execution Description**. The **Basel Run Execution Identifier** and **FIC MIS Date** is auto-populated from the Basel Run report used.
- f. Click **Execute**.

Note: Resave Hierarchy **HFSDF004** (FSDF Basel Run Execution Identifier for Run) after Basel execution for getting values in this Basel Run Execution Identifier.



- g. Select only one Basel Run from the Available Hierarchies for the execution and click **OK**. The *Run Management Summary* window is displayed.

Checking the Execution Status

The status of execution can be monitored using the *Batch Monitor* screen.

Note: For a more comprehensive coverage of configuration and execution of a batch, see [OFS Analytical Applications Infrastructure User Guide](#).

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going

-
- F - Failure
 - S - Success

The execution log can be accessed on the application server in the following directory *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA*. The file name will have the batch execution ID.

Following are the error log tables in atomic schema:

- FCT_FORECAST_REG_CAP_SUMMARY\$
- FCT_MITIGANT_REG_CAPITAL\$
- FCT_MR_CAPITAL_SUMMARY\$
- FCT_MR_VAR_PORTFOLIO_SUMMARY\$
- FCT_MR_VAR_SUMMARY\$
- FCT_REG_ACCT_MITIGANT_MAPPING\$
- FCT_REG_CAP_PLCD_COLL_SUMMARY\$
- FCT_REG_CAP_POOL_SUMMARY\$
- FCT_REG_CP_CAPITAL_SUMMARY\$
- FCT_REG_LE_CAPITAL_SUMMARY\$
- FCT_REG_OR_CAPITAL_SUMMARY\$
- FCT_REG_POOL_MITIGANT_MAP\$
- FCT_REG_CAP_ACCOUNT_SUMMARY\$

BASEL Processing to FSDF Results Integration Results T2Ts

T2T definitions can be retrieved as an excel document for reference from the metadata browser of the Unified Metadata Manager (UMM) component of OFSAAI.

LLFP Processing to FSDF Results Integration

This chapter provides information about LLFP Processing to FSDF Results Integration in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview of LLFP Processing to FSDF Results Integration Tables
- Overview of LLFP Processing to FSDF Results Integration
- Executing the LLFP Processing to FSDF Results Integration T2Ts
- Checking the Execution Status
- LLFP Processing to FSDF Results Integration Results T2Ts

Overview of LLFP Processing to FSDF Results Integration Tables

As part of LLFP processing to FSDF results integration, FSDF tables are loaded from LLFP Processing tables using Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework. Following are the Results Tables that stores integrated results:

- FCT_LLFP_ACCOUNT_SUMMARY

Note: As part of LLFP processing results to FSDF integration, FSDF is packaging the forementioned T2Ts. These are optional T2Ts that will be deployed only when OFS_IFRS_PACK is installed.

Overview of LLFP Processing to FSDF Results Integration

Table-to-Table seeded definitions are provided for loading data into the target tables.

Table 17. Table to Table Seeded Definitions

SI. No.	Source Table Name	Target Table Name	T2T Definition Name
1	FCT_ACCOUNT_DETAILS	FCT_LLFP_ACCOUNT_SUMMARY	T2T_FCT_LLFP_ACCOUNT_SUMMARY

Executing the LLFP Processing to FSDF Results Integration T2Ts

For LLFP FSDF integration, you must have FSDF and LLFP installed on the same INFODOM.

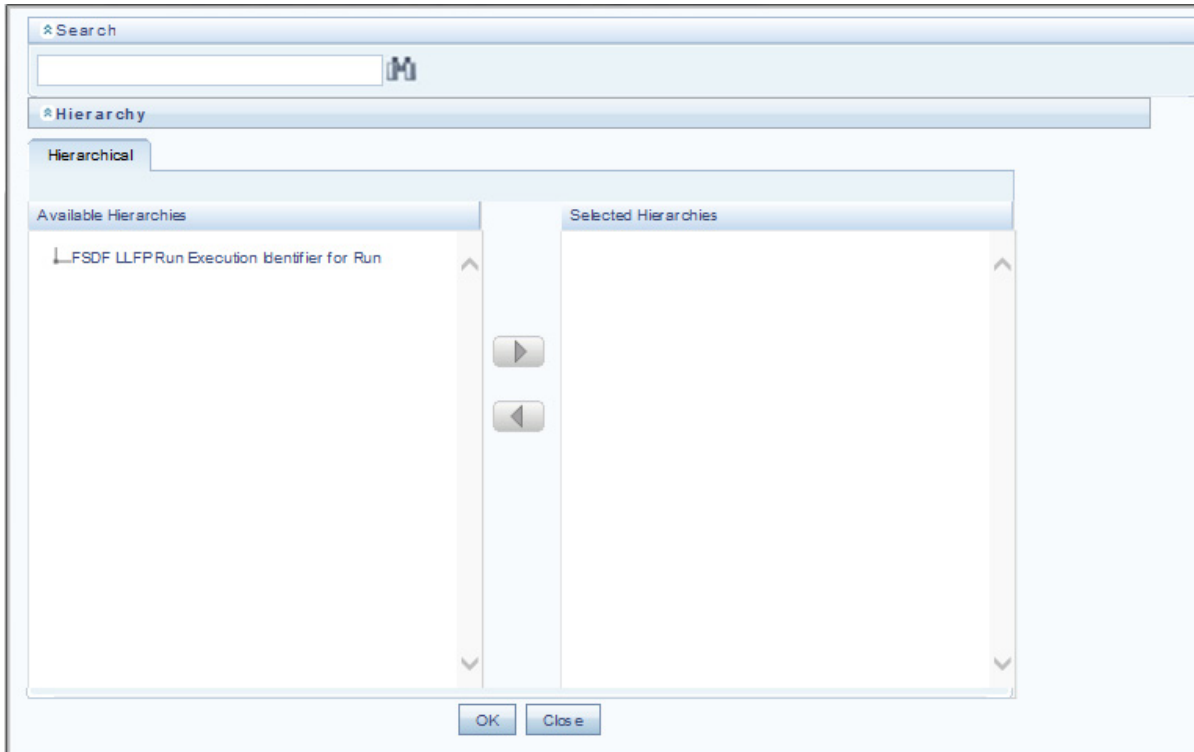
There are two ways to integrate LLFP and FSDF:

1. **Creating Integrated Run at Implementation Site:** During implementation, you can merge the tasks of both LLFP and FSDF and create an integrated Run to execute each time. The processes inside Run should be ordered as LLFP first, then FSDF, and finally the LLFP FSDF Integration process. In this Run, the LLFP processing area and the FSDF results area tables must have the same Run SKEY across all tables.

Note: For LLFP-FSDF Integration Run, please use the FSDF Run Management screen as the Request Report Flag, Override Report Flag, and Approve Report Flag options are not available in the LLFP Run Management Screen to enable the Reporting Flag.

2. Using approved LLFP Run Execution ID in FSDF Run: In this case, you can use the out-of-the-box LLFP Run as is for execution. After the execution, if the values are correct, you can execute the out-of-the-box FSDF Run by selecting the required LLFP Run SKEY from Run Management screen. In this case, LLFP processing area has one RUN SKEY and for the same data, FSDF has a different RUN SKEY in FSDF results area tables, where the data is getting reported. Sample report generation is as follows:
 - a. Login to Oracle Financial Services Analytical Applications interface with your credentials.
 - b. Navigate to **Applications > Financial Services Data Foundation > Run Management > Run Management**.
 - c. Select **Run** and click **Run Execution Summary** icon.
 - d. The *Run Details and Run Execution Parameters* window is displayed.
 - e. Enter the **Run Name** and **Run Execution Description**. The **LLFP Run Execution Identifier** and **FIC MIS Date** is auto-populated from the LLFP Run report used.
 - f. Click **Execute**.

Note: Resave Hierarchy **HFSD007** (FSDF LLFP Run Execution Identifier for Run) after LLFP execution for getting values in this LLFP Run Execution Identifier.



- g. Select only one LLFP Run from the Available Hierarchies for the execution and click **OK**. The *Run Management Summary* window is displayed.

Checking the Execution Status

The status of execution can be monitored using the *Batch Monitor* screen.

Note: For a more comprehensive coverage of configuration and execution of a batch, see [OFS Analytical Applications Infrastructure User Guide](#).

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going
- F - Failure
- S - Success

The execution log can be accessed on the application server in the following directory *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA*. The file name will have the batch execution ID.

Following is the error log table in the atomic schema:

- FCT_LLFP_ACCOUNT_SUMMARY\$

LLFP Processing to FSDF Results Integration Results T2Ts

T2T definitions can be retrieved as an excel document for reference from the metadata browser of the Unified Metadata Manager (UMM) component of OFSAAI.

LRM Processing to FSDF Results Integration

This chapter provides information about LRM Processing to FSDF Results Integration in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview of LRM Processing to FSDF Results Integration Tables
- Overview of LRM Processing to FSDF Results Integration
- Executing the LRM Processing to FSDF Results Integration T2Ts
- Checking the Execution Status
- LRM Processing to FSDF Results Integration Results T2Ts

Overview of LRM Processing to FSDF Results Integration Tables

As part of LRM processing to FSDF results integration, FSDF tables are loaded from LRM Processing tables using Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework. Following are the Results Tables that stores integrated results:

- FCT_LRM_ACCOUNT_SUMMARY

Note: As part of LRM processing results to FSDF integration, FSDF is packaging the forementioned T2Ts. These are optional T2Ts that will be deployed only when OFS_TR_PACK is installed.

Overview of LRM Processing to FSDF Results Integration

Table-to-Table seeded definitions are provided for loading data into the target tables.

Table 18. Table to Table Seeded Definitions

SI. No.	Source Table Name	Target Table Name	T2T Definition Name
1	FSI_LRM_INSTRUMENT	FCT_LRM_ACCOUNT_SUMMARY	T2T_FCT_LRM_ACCOUNT_SUMMARY

Executing the LRM Processing to FSDF Results Integration T2Ts

For LRM FSDF integration, you must have FSDF and LRM installed on the same INFODOM.

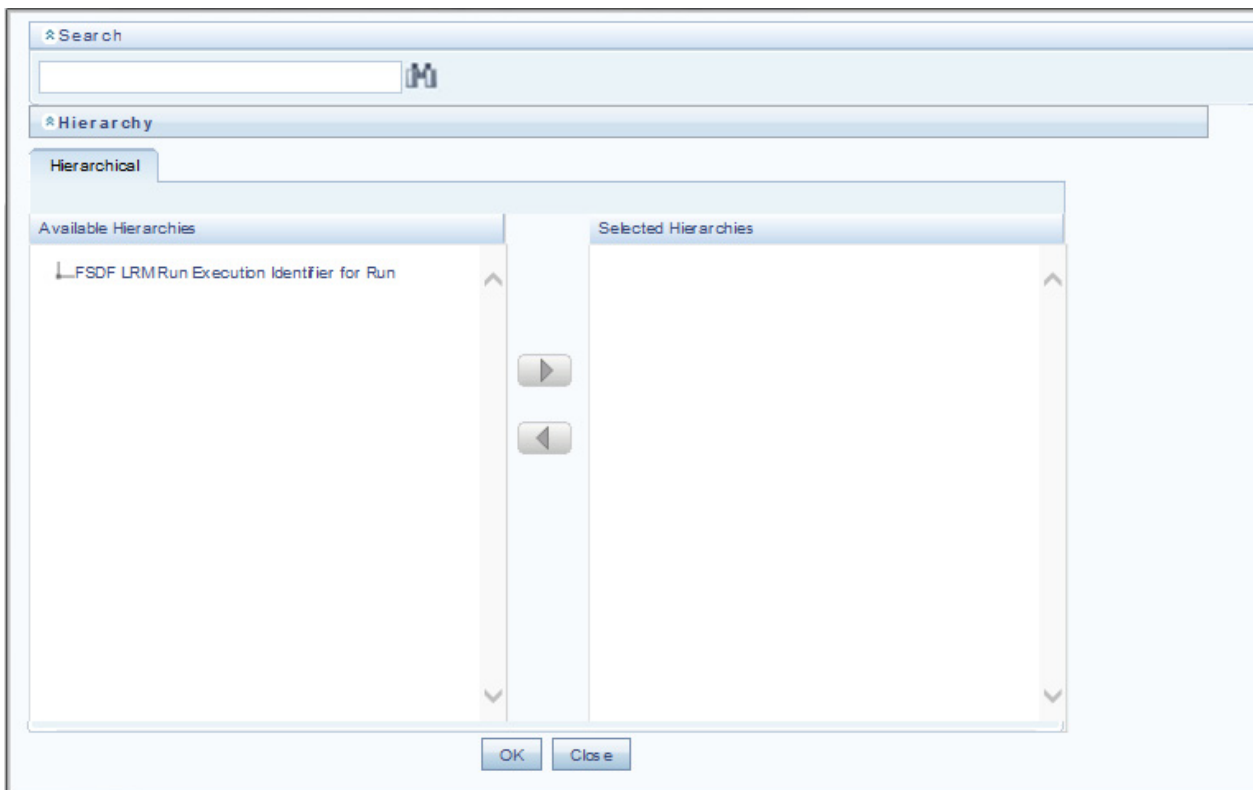
There are two ways to integrate LRM and FSDF:

1. Creating Integrated Run at Implementation Site: During implementation, you can merge the tasks of both LRM and FSDF and create an integrated Run to execute each time. The processes inside Run should be ordered as LRM first, then FSDF, and finally the LRM FSDF Integration process. In this Run, the LRM processing area and the FSDF results area tables must have the same Run SKEY across all tables.

Note: For LRM-FSDF Integration Run, please use the FSDF Run Management screen as the Request Report Flag, Override Report Flag, and Approve Report Flag options are not available in the LRM Run Management Screen to enable the Reporting Flag.

2. Using approved LRM Run Execution ID in FSDF Run: In this case, you can use the out-of-the-box LRM Run as is for execution. After the execution, if the values are correct, you can execute the out-of-the-box FSDF Run by selecting the required LRM Run SKEY from Run Management screen. In this case, LRM processing area has one RUN SKEY and for the same data, FSDF has a different RUN SKEY in FSDF results area tables, where the data is getting reported. Sample report generation is as follows:
 - a. Login to Oracle Financial Services Analytical Applications interface with your credentials.
 - b. Navigate to **Applications > Financial Services Data Foundation > Run Management > Run Management**.
 - c. Select **Run** and click **Run Execution Summary** icon.
 - d. The *Run Details and Run Execution Parameters* window is displayed.
 - e. Enter the **Run Name** and **Run Execution Description**. The **LRM Run Execution Identifier** and **FIC MIS Date** is auto-populated from the LRM Run report used.
 - f. Click **Execute**.

Note: Resave Hierarchy **HFSD006** (FSDF LRM Run Execution Identifier for Run) after LRM execution for getting values in this LRM Run Execution Identifier.



- g. Select only one LRM Run from the Available Hierarchies for the execution and click **OK**. The *Run Management Summary* window is displayed.

Checking the Execution Status

The status of execution can be monitored using the *Batch Monitor* screen.

Note: For a more comprehensive coverage of configuration and execution of a batch, see [OFS Analytical Applications Infrastructure User Guide](#).

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going
- F - Failure
- S - Success

The execution log can be accessed on the application server in the following directory *ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA*. The file name will have the batch execution ID.

Following is the error log table in the atomic schema:

- FCT_LRM_ACCOUNT_SUMMARY\$

LRM Processing to FSDF Results Integration Results T2Ts

T2T definitions can be retrieved as an excel document for reference from the metadata browser of the Unified Metadata Manager (UMM) component of OFSAAI.

This chapter provides information about Metadata Browser in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Overview
- Object View
- Metadata Publish
- Metadata Object to Application Map

Overview

Metadata Browser (MDB) with an Object and an Application view, provides a common repository of metadata objects created in Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) and Oracle Financial Services Analytical Applications (OFSA) hosted in OFSAAI. Using this view, you can identify the usage of base objects in higher level objects and the mapping of Objects to Application. It enables traceability and impact analysis to the user. It also allows users to view the data flow and the work flow of the application and understand the usage of objects within the application.

The visualization of MDB supports Application view and Object view. In Application view, you can browse through the metadata created using the Applications hosted in OFSAAI. In object view, you can view the metadata created in OFSAAI.

Object View

Object view provides the detailed view of the Object. It comprises of basic details, detailed properties, dependencies, usage in higher level Object and Applications consuming the Metadata objects. Object view provides the option to navigate to its dependent or higher level usage objects. Object view enables the user to identify the usage and its dependencies across other objects.

Following are the steps to be followed to migrate the Objects to the metamodel structure:

The Object view will provide the following areas in Financial Service Data Foundation:

- Data Foundation Metadata
- Target Data Model
- Data Mapping
- Data File Mapping
- Data Transformation
- Data Quality Rules
- Data Quality Groups
- Process Metadata
- Process

Metadata Publish

In order to publish the relevant metadata object to the metamodel structure, a seeded batch name <Infodom>MDB has to be executed.

Post successful publish, all the objects can be viewed in the Metadata browser. This step will need to be done on a regular basis to ensure that the metamodel is in sync with the underlying metadata.

Metadata Object to Application Map

To map all the objects of the Financial Service data foundation application, a batch name <Infodom>_MDB_OBJECT_APPLN_MAP has to be executed.

This chapter provides information about Time Dimension Population in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

Business data commonly represents information as of a point in time (for example, a balance as of a point in time) or as of a particular span of time (for example, income for the month of March). The rollup of a particular balance depending on their nature could be a simple additive rollup wherein the child member balances are added up to arrive at the parent node balance (for example, Ending Balance) or non additive rollups wherein a node formula is used to specify how to rollup the child member balances (for example, 3 month rolling average).

This chapter includes the following topics:

- Overview of Time Dimension Population
- Prerequisites
- Tables Used by the Time Dimension Population Transformation
- Executing the Time Dimension Population Transformation
- Checking the Execution Status

Overview of Time Dimension Population

The Time dimension population transformation is used to populate the DIM_DATES table with values between two dates specified by the user.

The database components, used by the transformations are:

1. Database function FN_DIM_DATES
2. Database procedure PROC_DIM_DATES_POPULATION that is called by the function FN_DIM_DATES mentioned earlier.

Prerequisites

The following are the prerequisites for Time dimension population.

1. All the post install steps mentioned in the [OFS Analytical Applications Infrastructure User Guide](#) and the solution installation manual of [Oracle Financial Services Data Foundation](#) must be completed successfully.
2. Application User must be mapped to a role that has seeded batch execution function (BATPRO).
3. Before executing a Batch, check if the following services are running on the application server:
 - a. Iccserver
 - b. Router
 - c. AM Server
 - d. Message Server
 - e. OLAP Server

For more information on how to check if the services are up and on and how to start the services if you find them not running, refer to *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

6. Batches will have to be created for executing the function. For more details, refer to section How to Define a Batch.

Tables Used by the Time Dimension Population Transformation

For more details on viewing the structure of earlier tables, refer to *Oracle Financial Services Analytical Applications Data Model Data Dictionary* or the *OFSDf Erwin Data Model*.

Executing the Time Dimension Population Transformation

You can execute the function from the *Operations* (formerly Information Command Center (ICC) framework) module of OFSAAI.

This component for OFSDf 8.0.6.0.0 has been seeded with the Batch ID `FSDFINFO_DATA_FOUNDATION_SCD`, which can be executed from Batch Execution section of OFSAAI. In the Parameter List, enter the Start Date and End Date. For example: '19940101', '19941231'.

Note: You can load DIM_DATES for a fiscal year for ONE jurisdiction at a time. However, if the dates are populating incorrectly for the selected Jurisdiction, you should revisit the values entered in the DIM_FINANCIAL_YEARS TABLE and then repopulate the DIM_DATES.

You can also define a new Batch and an underlying Task definition from the *Batch Maintenance* window of OFSAAI. For more information on defining a new Batch, refer to section How to Define a Batch.

To define a new task for a Batch definition:

1. Select the check box adjacent to the newly created Batch Name in the *Batch Maintenance* window.
2. Click **Add (+)** button from the Task Details grid.

The *Task Definition* window is displayed.

3. Enter the **Task ID** and **Description**.
4. Select **Transform Data** component from the drop down list.
5. Select the following from the **Dynamic Parameters** list:
 - a. **Datastore Type** - Select the appropriate datastore type from the list.
 - b. **Datastore Name** - Select the appropriate datastore name from the list.
 - c. **IP address** - Select the IP address from the list.
 - d. **Rule Name** - Select `fn_DimDates` from the drop down list of available transformations. (This is a seeded Data Transformation which is installed as part of the OFSDf solution installer. If you don't see this in the list, contact Oracle support)
 - e. **Parameter List** – Enter the Start Date and End Date.
 - **Start Date** – This is the starting date, from which the Transformation will populate DIM_DATES table. This date should be specified in 'YYYYMMDD' format.
For example, '20081131'.
 - **End Date** - This is the end date, to which the Transformation will populate DIM_DATES table. This date should also be specified in 'YYYYMMDD' format.

For example, '20091231'.

6. Click Save.

The Task definition is saved for the selected Batch.

7. Execute the batch.

You can execute a Batch definition from the Batch Execution section of OFSAAI Operations module. The function can also be executed directly on the database through SQLPLUS Details are:

Function Name: FN_DIM_DATES

Parameters: P_BATCH_RUN_ID, P_AS_OF_DATE, P_ST_DT, and P_ED_DT

Sample Parameter Values: 'Batch1', '20091231', '20081131', and '20091231'

Note: This DT should be executed for each year for which data is present in the source table.

Checking the Execution Status

The Batch execution status can be monitored through *Batch Monitor* section of *OFSAAI Operations* module.

The status messages in batch monitor are:

- N - Not Started
- O - On Going
- F - Failure
- S – Success

The execution log can also be accessed on the application server in the directory *ftpshare/logs/<Run_Date>/FSDFINFO/RUN EXECUTABLE*, where file name will have the Batch Execution ID.

You can access the database level operations log by querying the FSI_MESSAGE_LOG table. Filter the Batch Run ID column for identifying the relevant log.

Note: Check the **.profile** file in the installation home if you are unable to find the above mentioned path.

This chapter provides information about performing a backdated Run in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

There are scenarios that require Run executions for a prior date due to reasons such as, Backdated Regulatory Return Submission or Backdated Management Report Generation, etc.

This chapter includes the following topics:

- Overview of Backdated Run Execution
- Required Changes
- Recommendations

Overview of Backdated Run Execution

Backdated Run Execution is similar to any regular Run Execution in FSDF. You must reload or correct the data, which must be loaded for the given prior date. Refer to FSDF Runchart and execute the relevant Batches/Runs for the required prior date.

Required Changes

The following are the prerequisites for Backdated Run Execution:

1. Handling Slowly Changing Dimensions (SCDs)

Ensure that all SCD executions for the given prior date or period is happened / completed with valid records in all the SCD dimensions. Record Start Date and Record End Date columns must have values.

2. Handling T2T Joins

By default, all out-of-the-box T2Ts have Joins with SCD dimensions using latest record indicator. This must be modified to pick the correct record using Record Start Date and Record End Date columns of SCD dimension.

Recommendations

Modifying T2T Joins using Record Start Date and Record End Date columns causes poor performance due to comparison of multiple dates in the query. The following are the recommendations:

1. Copy the existing T2T definitions and modify the Joins to pick Record Start Date and Record End Date instead of latest record indicator.
2. Copy Record Start Date and Record End Date of the existing FSDF Process (under Run Rule Framework) which loads T2T and replace them with new T2T definitions changed above.
3. Whenever a backdated Run is required, modify the Run definition without changing the Run ID to pick the modified Process.

Note: It is recommended to use this only when there is Backdated Run requirement. Any normal sequential days execution can be performed using out-of-the-box T2Ts and Runs as it results in better performance.

Backdated Run Execution using Latest Record Indicator (LRI) batch

A new Data Transformation batch UPDATE_BACK_DATED_DIM_LRI is created. This batch supports Backdated Run execution in the Latest Record Indicator (LRI) dimension table. The task in the UPDATE_BACK_DATED_DIM_LRI batch updates LRI Dimension Table for a given Dimension Table and for a given FIC_MIS_DATE. When this batch is triggered for a given FIC_MIS_DATE, the SKeys are updated with the new LRI flag value (F_LATEST_RECORD_INDICATOR = Y/N).

To execute Backdated Run using LRI batch:

1. Add the required data records to their Master table and for the required FIC_MIS_DATE. For example, added two records to STG_PARTY_MASTER. The resultant image is shown below.

```
select * from stg_party_master
```

	V_PARTY_NAME	V_DATA_ORIGIN	V_PARTY_ID	FIC_MIS_DATE	N_JOINING_AGE	V_D
1	Avani Rai	...	Avani001	1/1/2001		
2	Avani Sharma	...	Avani001	1/1/2002		

Note: Ensure that V_PARTY_ID is same in this step and the LRI batch execution step.

2. Execute the SCD batch to add the required records from Master table to its Dimension table with the required FIC_MIS_DATES. For example, executed SCD batch to add records from STG_PARTY_MASTER to DIM_PARTY. The resultant data record in the DIM_PARTY is as shown below:
 - After adding the first record to DIM_PARTY for the FIC_MIS_DATE 1/1/2001, the flag F_LATEST_RECORD_INDICATOR associated with this data record is set to Y.

```
select d.n_party_skey,d.v_party_name,d.fic_mis_date,d.v_party_id,d.f_latest_record_indicator from dim_party d
```

	N_PARTY_SKEY	V_PARTY_NAME	FIC_MIS_DATE	V_PARTY_ID	F_LATEST_RECORD_INDICATOR
1	-1 Others	...	1/1/1900	OTH	Y
2	0 Missing	...	1/1/1900	MSG	Y
3	1 Avani Rai	...	1/1/2001	Avani001	Y

- After adding the second record to DIM_PARTY for the FIC_MIS_DATE 1/1/2002, the flag F_LATEST_RECORD_INDICATOR associated with this data record is set to Y, and the flag

F_LATEST_RECORD_INDICATOR associated with the previous data record (with FIC_MIS_DATE 1/1/2001) is set to N automatically.

```
select d.n_party_key,d.v_party_name,d.fic_mis_date,d.v_party_id,d.f_latest_record_indicator from dim_party d
```

	N_PARTY_SKEY	V_PARTY_NAME	FIC_MIS_DATE	V_PARTY_ID	F_LATEST_RECORD_INDICATOR
1	-1	Others	1/1/1900	OTH	Y
2	0	Missing	1/1/1900	MSG	Y
3	1	Avani Rai	1/1/2001	Avani001	N
4	2	Avani Sharma	1/1/2002	Avani001	Y

- In Batch Maintenance, add a Task for the LRI Dimension table. In the Parameter List field, mention the Dimension table name ('DIM_TABLE_NAME') for which Backdated Run using LRI must be executed. In this example, 'DIM_TABLE_NAME' = 'DIM_PARTY'.

Save
Close

Task Definition

Task ID: Description:

Components:

Dynamic Parameters List

Property	Value
Datastore Type	EDW
Datastore Name	FSDFINFO
Primary IP For Runtime Processes	10.184.157.123
Rule Name	fn_update_back_dated_lri_wrap
Parameter List	'DIM_TABLE_NAME'

Audit Panel

Created By: OFSAD Creation Date: 05 Jul 2018 23:01:09

Last modified by: OFSAD Last Modification Date: 05 Jul 2018 23:01:09

4. In the Batch Execution window, execute the UPDATE_BACK_DATED_DIM_LRI batch for different FIC_MIS_DATES.

Batch Execution

Batch Mode: Mode Run Restart Rerun

Search: Batch ID Like FSDFINFO_ [] Batch Description Like LRI []
 Module [] Last Modification Date Between [] And []

Batch Details Schedule Batch

Batch ID	Batch Description
<input checked="" type="checkbox"/> FSDFINFO_UPDATE_BACK_DATED_DIM_LRI	This Batch Updates SCD Dimension Table LRI for given FICMIS DATE

Page 1 of 1 (1-1 of 1 items) K < > X

Task Details Exclude/Include Hold/Release

Task ID	Task Description	Metadata Value	Component ID	Precedence
task1	This Task Updates SCD Dimension Table LRI for given FICMIS DATE	fn_update_back_dated_lri_wrap	TRANSFORM DATA	N

Page 1 of 1 (1-1 of 1 items) K < > X

Information Date: Date 01/01/2001 []

The following are the examples of data record after running the LRI batch with different FIC_MIS_DATES:

- The image below depicts execution of the UPDATE_BACK_DATED_DIM_LRI batch for a backdated record for its FIC_MIS_DATE. Therefore, this record is the Latest Record. In the following image, the LRI batch is executed for the data record with historical FIC_MIS_DATE 1/1/2001. Therefore, the flag F_LATEST_RECORD_INDICATOR associated with FIC_MIS_DATE 1/1/2001 is set to Y. This data record is flagged as the Latest Record, and will be used as default by the system for all future transactions and processes.

```
select d.n_party_skey,d.v_party_name,d.fic_mis_date,d.v_party_id,d.f_latest_record_indicator from dim_party d
```

	N_PARTY_SKEY	V_PARTY_NAME	FIC_MIS_DATE	V_PARTY_ID	F_LATEST_RECORD_INDICATOR
1	-1	Others	1/1/1900	OTH	Y
2	0	Missing	1/1/1900	MSG	Y
3	1	Avani Rai	1/1/2001	Avani001	Y
4	2	Avani Sharma	1/1/2002	Avani001	N

- To change the default data selection, execute the UPDATE_BACK_DATED_DIM_LRI batch for the required data record for its FIC_MIS_DATE. In the following image, the batch is executed for the data record with FIC_MIS_DATE 1/1/2002. Therefore, the flag F_LATEST_RECORD_INDICATOR associated with this FIC_MIS_DATE 1/1/2002 is set to Y and this data record is now flagged as the Latest Record. This record will be used as default by the system for all future transactions and processes. The flag

F_LATEST_RECORD_INDICATOR associated with the earlier default data record (with FIC_MIS_DATE 1/1/2001) is set automatically to N, indicating that the record with FIC_MIS_DATE 1/1/2001 is not the Latest Record anymore.

```
select d.n_party_skey,d.v_party_name,d.fic_mis_date,d.v_party_id,d.f_latest_record_indicator from dim_party d
```

	N_PARTY_SKEY	V_PARTY_NAME	FIC_MIS_DATE	V_PARTY_ID	F_LATEST_RECORD_INDICATOR
1	-1	Others	1/1/1900	OTH	Y
2	0	Missing	1/1/1900	MSG	Y
3	1	Avani Rai	1/1/2001	Avani001	N
4	2	Avani Sharma	1/1/2002	Avani001	Y

5. Monitor the status of the batch in the Batch Monitor screen.

This chapter provides information about Using OFSDF in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter details on how the OFSDF models are delivered and how they can be installed and configured into the required environment. The first two sections gives you an understanding of the Delivery Mechanism and OFSDF Installation. The Data Dictionary and Download Specifications sections explains how the self-documenting ERwin file includes the data dictionary and Download Specifications within ERwin itself.

In addition, the Extending Data Model section has guidelines for customization and designing the Staging and Results Area of Physical Data Model.

This chapter includes the following topics:

- Delivery Mechanism
- Installing OFSDF
- OFSDF Supporting Documentation
- Data Dictionary
- Download Specifications
- Extending OFSDF Physical Data Model

Delivery Mechanism

OFSDF being a collection of data model artifacts, includes both a readily deployable model (the OFSDF Physical Data Model) as well as a reference data model (the OFSDF Logical Data Model). Both the data models (Physical and Logical) are delivered as ERwin files. The OFSDF hence requires a license of the ERwin Data modeling tool.

ERwin is the current and only supported modeling tool to view and edit the model. Currently, the minimum versions of ERwin supported are 9.5 and 9.64.

Note: OFS AAI supports data model upload for data models generated using ERwin 7.1.x, 7.2.x, 7.3.x, 9.0.x, 9.2.x, and 9.5.x versions.

Installing OFSDF

As detailed earlier, OFSDF requires the Oracle Financial Services Analytical Application Infrastructure release 8.0.6.0.0 to deploy and operate.

Please refer to the separate **OFSDF Install Guide** for step-wise instructions how to configure and install OFSDF into an AAI instance.

OFSDF Supporting Documentation

The preceding sections have provided an overview of the organization of the OFSDF, and its various component data models. Appendix A explains the naming conventions used in OFSDF data model.

The OFSDF is a detailed model, with nearly 850 entities across both the Staging and Results Area in the physical data model, with another 800+ entities in the Logical Data Model.

Since it is delivered as an ERwin file, all the detailed metadata for the model (Table, Column, Entity, Attribute, Relationship) definitions are embedded in the file itself. The advantage of this approach is that any site-specific customizations to OFSDF can be performed within ERwin, and the updated documentation is retained in the file in the form of additional metadata.

The 2 key detailed artifacts of OFSDF documentation that can be extracted from within the ERwin data model are:

1. Data Dictionary
2. Download Specifications

For more information on Dimension Management and AMHM, refer to the *Dimension Management* chapter in [Oracle Financial Services Analytical Applications Infrastructure User Guide](#) and *Dimension Load Procedure* section in Oracle Financial Services Analytical Applications Data Model Utilities 7.1 / 7.2 User Guide.

Data Dictionary

The data dictionary for OFSDF can be extracted from the ERwin file using ERwin's reporting capability, using a pre-built set of templates for data extraction.

Instructions for how to do so are provided in a separate accompanying document that provides step-by-step instructions. See the Technical Whitepaper on *Data Model Document Generation*, which details how to extract the data dictionary from ERwin section.

Download Specifications

As detailed in the staging area section, the mapping from the Staging Data Model to use cases, called a **download specification** provides an efficient way to manage the sourcing of data into the OFSDF staging area. This is done by mapping the staging model at a column level to use cases. This mapping information is embedded in ERwin at a column level using metadata called **User Defined Properties (UDPs)**.

The Download specifications can be extracted using pre-built templates, in a manner similar to the Data Dictionary. Instructions for how to do so are also provided in the Technical Whitepaper on *Data Model Document Generation*, which details how to extract the data dictionary from ERwin section.

Extending OFSDF Physical Data Model

Oracle Financial Services Data Foundation (OFSDF) Physical Data Model (PDM) design evolves as the analytical use cases covered by the OFSDF and enhanced as improvements are engineered as a part of the product lifecycle. While the model satisfies a very large number of analytical use cases across Risk, Finance, Marketing, and Compliance subject areas, customers may need to customize the model for a specific installation. These custom changes however may impact the ability of the OFSDF installation to be upgraded to later versions of the product. The guidelines outlined in this section will help minimize the impact of custom changes to the model when the installation needs to be upgraded to a later version of OFSDF.

This section consists of the following sub-sections:

- Customization Process Guidelines
- Staging Area Design Guidelines
- Results Area Design Guidelines
- Upgrading Data Model

Customization Process Guidelines

It is strongly recommended to consult OFSAA Support / Field Engineers / Consulting Staff before making any changes to the PDM for the following reasons:

- Tables in the PDM common Staging Area are designed to meet the complex needs of data sourcing for many different financial services analytical use cases and as such have a large number of columns, and the need for the modification should be reviewed with OFSAA consultants.

The Results Area star schemas have been designed with a set of common fact tables and dimension tables to support integration of results from multiple analytical applications and any customization should be reviewed in order to ensure that the unified reporting capabilities of the model are preserved.

After a review with OFSAA field consultants, an extension to the model should first be logged as a request for product enhancement via the standard support process. This allows:

1. Product support and product management teams to identify if a similar enhancement request was submitted on behalf of another customer so that a uniform Model Enhancement design recommendation can be provided to all customers.
2. OFSDF product management to evaluate if the enhancement request is applicable more broadly to other customers and if the change should in fact be taken as a design requirement for subsequent releases.

Note: OFS AAI supports data model upload for data models generated using ERwin 7.1.x, 7.2.x, 7.3.x, 9.0.x, 9.2.x, and 9.5.x versions.

Staging Area Design Guidelines

Following are the Staging Area Design Guidelines:

1. Ensure that the naming conventions as detailed in Appendix A section are followed.
2. Entity relationships and constraints are enforced through OFSAAI data management toolkit and are not enforced via database referential integrity checks.

The model should not be changed to enforce referential integrity checks and other data quality checks via database definitions.

3. All Staging Area tables must have a column that identifies the system from where data is sourced (source system ID).
4. The code columns in master data tables and tables that contain dimension data should be designed to hold alphanumeric values.
5. The Domain dictionary maintains the list of attribute domains. New columns must be identified with an existing domain instead of explicitly defining column data type and valid values. See guidelines in Appendix A section on the use of defined Domains.
6. Tables (e.g. reference or look up tables with static data) needed for only a specific application or use case should be a part of the application specific processing area and should not be part of the common Staging Area in OFSDF.
7. OFSDF download specifications identify the tables and columns for which data needs to be sourced for a specific analytical use case. Any new tables and/or column should have its "APPLICATION USAGE" UDP set with the appropriate application value so that the generated download specification includes the customized column and table. The master list of UDP's are maintained as a central dictionary in ERwin.
8. All columns added or modified as a part of the customization should be marked as such:

-
- The column level UDP named "CUSTOM" must be marked YES, identifying the column as a custom property.
 - The "Customization Reason" UDP should be specified. Valid values are provided as a drop down list and can be "Pending Enhancement Request" or "Specific to Customer".

The "Type of Change" UDP should be set to the appropriate type of change as provided in the drop down list (Length, Datatype, Logical Name, Description, and Addition).

Results Area Design Guidelines

The Results Area consists of a set of star schemas with conformed dimensions and common fact tables. Integration of results from multiple application use cases is achieved by having common fact tables for customer and account level measures. The design of the results area allows for drill-down and drill-across BI reporting, which should be preserved after customization.

Following are the Results Area Design Guidelines:

1. Ensure that the naming convention for results tables and columns detailed in Appendix A section is followed.
2. Dimensional conformance should be maintained: The same dimensional information should not be represented in different forms. In addition, dimension table design should be compatible with the slowly changing dimension process design and so should have the required columns.
3. The common accounts summary fact table.

(FCT_COMMON_ACCOUNTS_SUMMARY) consolidates measures at an account level granularity for all applications. Account level attributes captured from source systems in staging and those attributes that do not vary between runs should be part of the common accounts summary table. This enables integrated reporting of account information.

Note: Any account level application specific attributes and measures that are computed by applications should be part of the application specific account summary entities.

4. The common customer summary fact table.

(FCT_COMMON_CUSTOMER_SUMMARY) consolidates measures at a customer level granularity for all applications. Customer level attributes captured from source systems in staging and those attributes that do not vary between runs should be part of the common customer summary table. This enables integrated reporting of customer information.

Note: Any customer level application specific attributes and measures that are computed by applications should be part of the application specific customer summary entities.

5. **Aggregate Entities:** Depending on performance requirements for each application, information can be reported out of aggregate entities. However, a drill through to the base entity from the aggregate entity is mandatory.
6. **Reporting and local currency support:** Include additional attributes in the fact tables to store reporting and local currency equivalent of base measures. These attributes need to be computed by looking into the exchange rates.
7. **Support for full history:** Any new tables in the Results area should be designed to support maintenance of full history.

Upgrading Data Model

The model upgrade process is achieved through the ERwin Model Compare and Merge utility. Refer to ERwin documentation for details on Menu options, process of comparing, and merging models.

This chapter provides information about Data Quality Rules Execution in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

- Data Quality Framework
- Data Quality Rules For Staging Tables
- Data Quality Groups for Staging Tables

Data Quality Framework

Data Quality Framework consists of a scalable rule-based engine which uses a single-pass integration process to standardize, match, and duplicate information across global data. Data Quality Framework within the Infrastructure system facilitates you to define rules and execute them to query, validate, and correct the transformed data existing in an Information Domain.

You can access Data Quality Framework by expanding the Data Integrator Framework within the Unified Metadata Manager section in tree structure of LHS menu.

Data Quality Framework consists of the following sections:

- Data Quality Summary
- Data Quality Group Summary

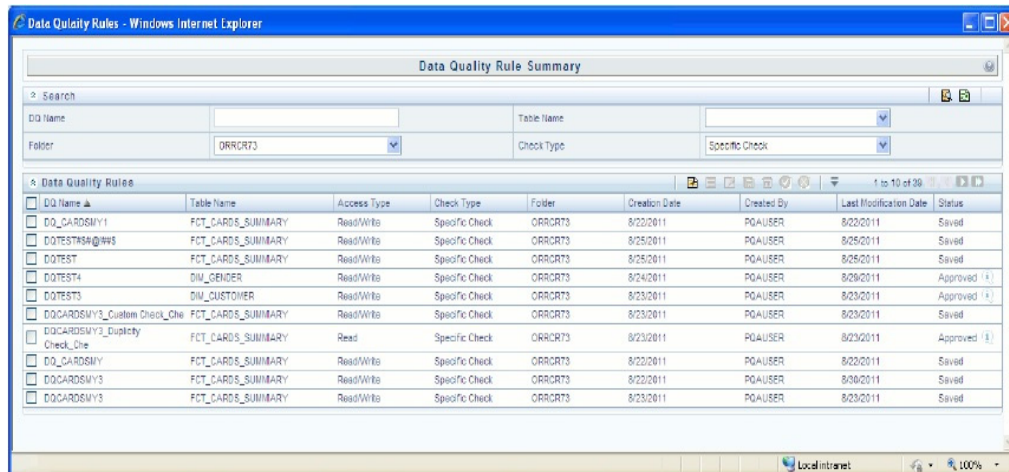
Data Quality Summary

Data Quality Summary within the Data Integrator framework of Infrastructure system facilitates you to create a DQ (Data Quality) definition and define nine specific validation checks based on Range, Data Length, Column Reference/Specific Value, List of Value/Code, Null Value, Blank Value, Referential Integrity, Duplicity, and Custom Check/Business. You can also correct data for range, column reference, list of values, null value, and blank value parameters.

The defined Data Quality Rule checks can be logically grouped and executed together. You (Business Analysts) need to have ETL Analyst function role mapped to access the Data Quality Summary framework within the Infrastructure system.

You can access Data Quality Summary by expanding the Data Quality framework within the Unified Metadata Manager section in tree structure of LHS menu.

Figure 6. Data Quality Rule Summary



The **Data Quality Rule Summary** screen displays the list of pre-defined Data Quality Rules with the other details such as DQ Name, Table Name, Access Type, Check Type, Folder, Creation Date, Created By, Last Modification Date, and Status of the Rule. A defined rule is displayed in Saved status, until it is Approved/Rejected by the approver. An Approved rule can be grouped in order for execution and a Rejected rule is sent back to the user with the Approver comments.

You can add, view, modify, copy, approve/reject, or delete Data Quality Rules within the **Data Quality Rule Summary** screen. You can also make use of Search and Pagination options to search for a Data Quality Rule based on DQ Name, Table Name, Folder, or Check Type and view the existing Data Quality Rules within the system.

For more information, refer to *Oracle Financial Services Analytical Applications Infrastructure User Manual*.

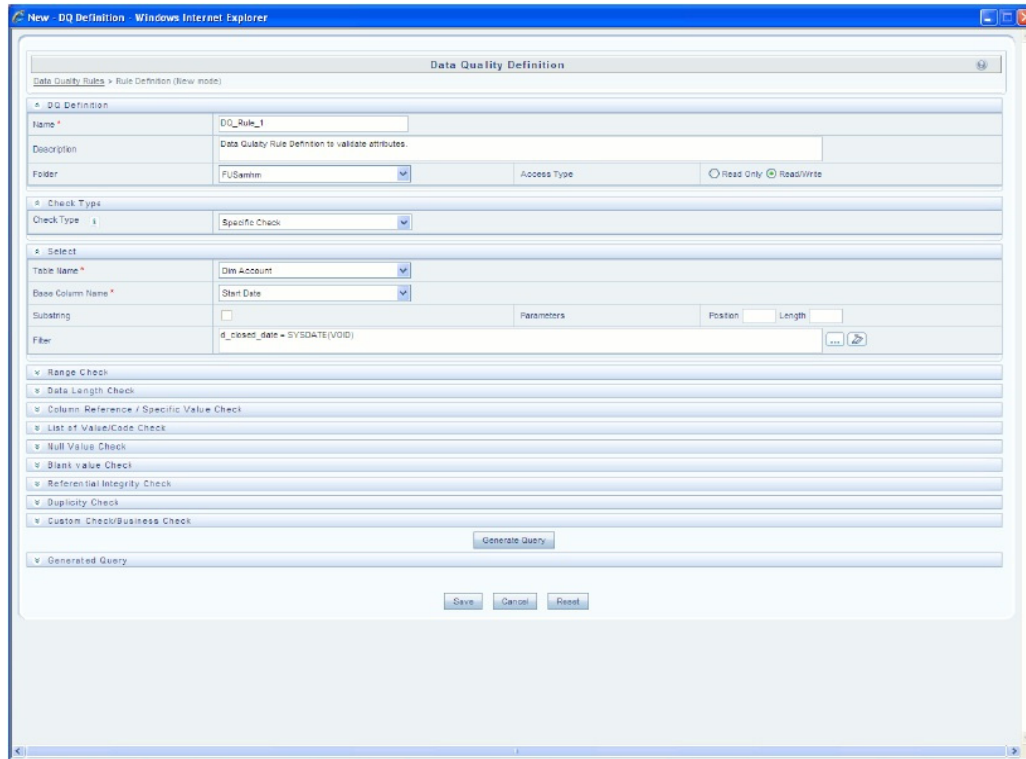
Create Data Quality Rule

You can create a Data Quality Rule definition by specifying the DQ Definition details along with the type of validation check on the required table and defining the required validation conditions to query and correct the transformed data.

To create Data Quality Rule in the **Data Quality Rule Summary** screen:

1. Click **Add** button in the Data Quality Rules tool bar. Add button is disabled if you have selected any check box in the grid. The **Data Quality Definition** screen is displayed.

Figure 7. Data Quality Definition



2. In the DQ definition section, perform the following:

- Enter the **Name** by which you can identify the DQ definition.
- Enter a description or related information about the definition.
- Select the **Folder** (available for selected Information Domain) from the drop down list.
- Select the **Access Type** as either Read Only or Read/Write.

3. Select the **Check Type** from the drop down list. You can mouse-over i icon for information.

- Select **Specific Check**, if the defined conditions are based on individual checks on a single column.
- Select **Generic Check**, if the defined conditions are based on multiple columns of a single base table. These checks are not pre-defined and can be specified (user-defined) as required.

If *Specific Check* is selected, perform the following:

- Select **Table Name** and **Base Column Name** from the drop down list. The list displays all the tables which are marked for Data Quality Rule in a data model, which has the table classification property code set to 340.
- (Optional) If you have selected Base Column of type Varchar/Char, select the **Substring** check box, enter numeric values in **Parameters** Position and Length fields.
- Click the below button and define the **Filter** condition using the **Specify Expression** screen.



- Define the required **Validation Checks** by selecting the appropriate grid and specify the details. You can define nine specific validation checks based on Range, Data Length, Column Reference/Specific Value, List of Value/Code, Null Value, Blank Value, Referential Integrity, Duplicity, and Custom Check/Business.

For more information, refer to *Oracle Financial Services Analytical Applications Infrastructure User Manual*.

Note: A minimum of one Validation check must be defined to generate a query.

- Click **Generate Query**. The details are validated and the validated query along with the status is displayed in the Generated Query section.

If *Generic Check* is selected, perform the following:

- Select **Table Name** from the drop down list. The list displays all the tables which are marked for Data Quality Rule in a data model, which has the table classification property code set to 340.
- Click the below button and define the **Filter** condition using the **Specify Expression** screen.
- Click **Add** button in the **Condition** grid. The Specify Expression screen is displayed. Define the Condition expression.

The Expression is displayed with the "IF" and "Else" conditions along with the **Severity** status as either **Error** or **Warning**. You can change the **Severity** by selecting from the drop down list.

Note: You can add an Assignment only when the Severity is selected as **Warning**. Assignments are added when you want to correct or update record(s) in base column data / selected column data. There can be one or more assignments tagged to a single condition. However, selecting severity as **Error** indicates there are no corrections and only facilitates in reporting the quantity of bad records.

- Select the check box adjacent to the required Condition expression and click **Add** button in the **Assignment** grid. The assignment details are populated.

Note: You can add an Assignment only if the Severity is **Warning**. There can be one or more assignments tagged to a single condition.

- Specify the Assignment details as tabulated.

Table 19. Assignment Details

Field	Description
Column Name	Select the Column Name from the drop down list.
Assignment Type	Select the Assignment Type as one of the following: <ul style="list-style-type: none"> ● No Assignment is the default selected assignment which does not have any target column update, but the message details are pushed. ● Direct Value - enter the Assigned Value ● Another Column - select the required Column as Assigned Value from the drop down list. ● Code - select the required Code as Assigned Value from the drop down list if any code / leaf values exist for the selected base column. <p>If not, you are alerted with a message indicating that <i>No Code values existsfor the selected base column.</i></p>
Assignment Value	Select the Assignment Value from the drop-down list according to the Assignment Type selected.
Message Severity	Select the Message Severity as either 1 or 2 from the drop down list.
Message	Select the required Message for the <i>Severity</i> from the drop down list.

You can also add multiple assignments by clicking **Add** button in **Assignment** grid.

Note: Minimum of one condition needs to be defined to save the Rule.

8. Click **Save**. The defined Data Quality Rule definition is displayed in the **Data Quality Rule Summary** screen with the status as "Saved".

View Data Quality Rule

You can view individual Data Quality Rule definition details at any given point.

To view the existing Data Quality Rule definition in the **Data Quality Rule Summary** screen:

1. Select the check box adjacent to the required DQ Name.
2. Click **View** button from the Data Quality Rules tool bar.

The DQ Definition screen displays the details of the selected Data Quality

definition. The **Audit Trail** section at the bottom of DQ Definition screen displays metadata information about the Data Quality Rule defined.

Modify Data Quality Rule

You can update the existing Data Quality Rule definition details except for the Definition Name, Table, and Base Column selected.

To update the required Data Quality Rule definition details in the **Data Quality Rule Summary** screen:

-
1. Select the check box adjacent to the required DQ Name.

Note: You can only edit those rules which have status either as **Saved** or as **Rejected**.

2. Click **Edit** button from the Data Quality Rules tool bar. The Edit button is disabled if you have selected multiple DQ Names. The DQ Definition screen is displayed.

Update the details as required.

For more information, see Create Data Quality Rule section.

3. Click **Save** to update the changes.

Copy Data Quality Rule

You can copy the existing Data Quality Rule to quickly create a new DQ definition based on the existing rule details or by updating the required parameters.

To copy an existing Data Quality Rule definition in the **Data Quality Rule Summary** screen:

1. Select the check box adjacent to the required DQ Name in the list whose details are to be duplicated.
2. Click **Copy** button from the Data Quality Rules tool bar. **Copy** button is disabled if you have selected multiple check boxes. The DQ Definition screen is displayed.
3. Edit the DQ definition Name and other details as required.

For more information, see Create Data Quality Rule section.

4. Click **Save**. The defined Data Quality Rule definition is displayed in the **Data Quality Rule Summary** screen with the status as "Saved".

Approve/Reject Data Quality Rule

You (Authorizer) can Approve a pre-defined Data Quality Rule definition for further execution or Reject an inappropriate DQ definition listed within the **Data Quality Rule Summary** screen. You should be mapped to DQ Authorizer function role to Approve or Reject a DQ definition.

To Approve/Reject Data Quality Rule in the Data Quality Rule Summary screen:

1. Select the checkbox adjacent to the required DQ Name. Ensure that you select the "Saved" DQ definition based on the **Status** indicated in the **Data Quality Rules** grid.
2. Perform one of the following:
 - To **Approve** the DQ definition, click **Approve** button. The **User Comments** screen is displayed. Enter the notes or additional information to the user and click **OK**. The selected DQ definition is approved and a confirmation dialog is displayed.
 - To **Reject** the DQ definition, click **Reject** button. The **User Comments** screen is displayed. Enter the notes or additional information to the user and click **OK**.

The selected DQ definition is rejected and a confirmation dialog is displayed.

Note: The authorizer can approve/reject only one definition at a time.

The Approved/Rejected status of the DQ definition is indicated in the **Status** column of the **Data Quality Rule Summary** screen. You can mouse-over i button to view the Approver comments in a pop-up.

Delete Data Quality Rule

You can remove Data Quality Rule definition(s) which are created by you and which are no longer required in the system by deleting from **Data Quality Rule Summary** screen.

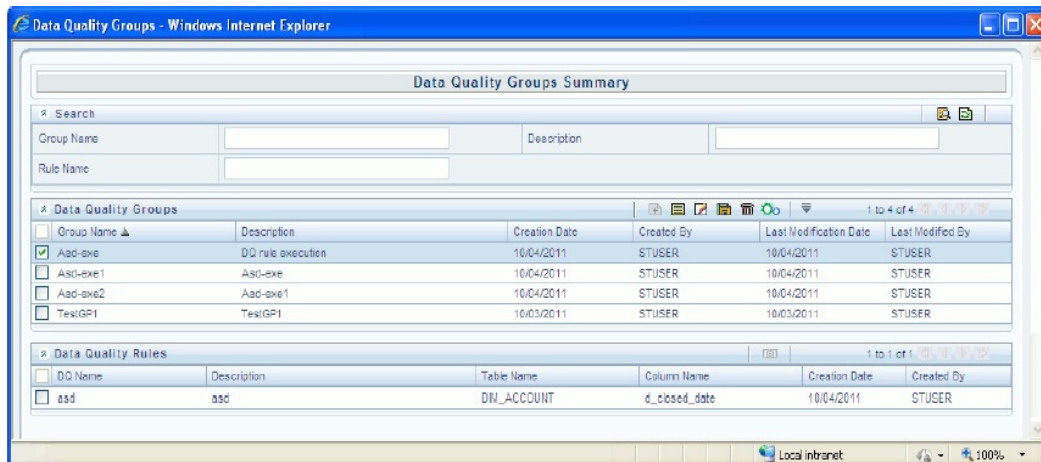
1. Select the check box adjacent to the required DQ Name whose details are to be removed.
2. Click **Delete** button from the Data Quality Rules tool bar.
3. Click **OK** to confirm deletion.

Data Quality Groups Summary

Data Quality Groups Summary within the Data Integrator framework of Infrastructure system facilitates you to logically group the define DQ definitions and schedule for execution. DQ definitions can be executed either through Data Quality Groups Summary screen of Data Integrator framework or in Batch Execution screen of Operations module.

When a Data Quality Group is executed for processing, the details of the execution are captured in a log file. You (Business Analysts) need to have ETL Analyst function role mapped to access the Data Quality Summary framework within the Infrastructure system. You can access Data Quality Groups Summary by expanding the Data Quality framework within the Unified Metadata Manager section in tree structure of LHS menu.

Figure 8. Data Quality Group Summary



The **Data Quality Groups Summary** screen displays the list of pre-defined Data Quality Groups with the other details such as Group Name, Description, Creation Date, Created By, Last Modification Date, and Last Modified By. You can Create and Execute DQ

Group definitions and view, modify, copy, or delete DQ Group definitions within the **Data Quality Groups Summary** screen.

You can also make use of Search and Pagination options to search for a DQ Group definition based on Group Name, Description, or Rule Name and view the existing DQ Group definitions within the system.

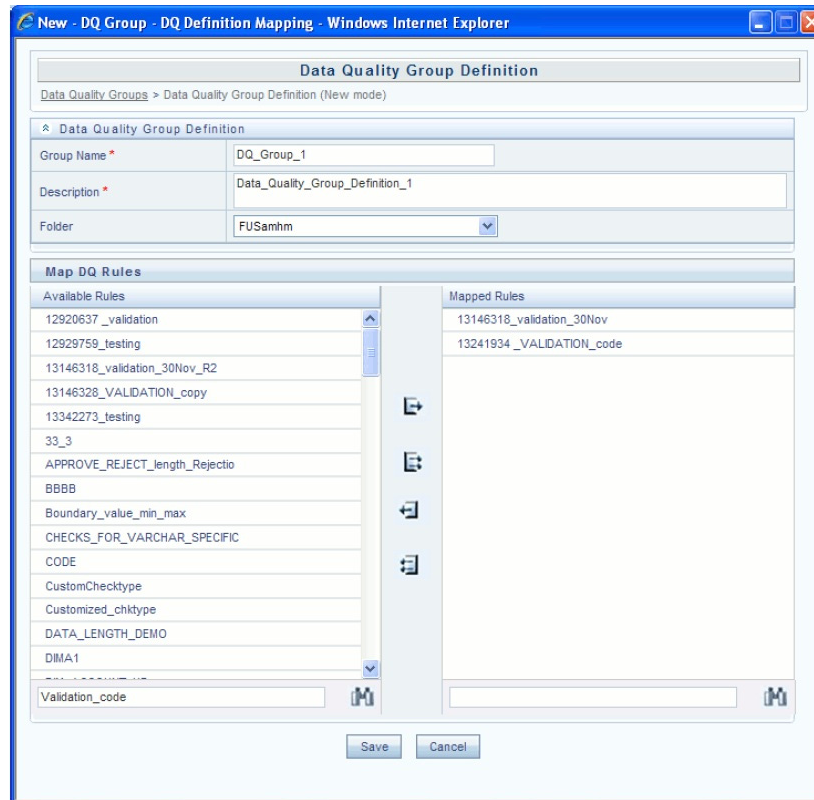
Create Data Quality Group

You can create a DQ Group definition by defining the DQ Definition details and mapping the required DQ Rules which are authorized and approved within the system.

To create DQ Group in the **Data Quality Groups Summary** screen:

1. Click **Add** button in the Data Quality Groups tool bar. Add button is disabled if you have selected any check box in the grid. The **Data Quality Group Definition** screen is displayed.

Figure 9. Data Quality Group Definition



2. In the **Data Quality Group Definition** section, perform the following:
 - Enter the **Group Name** by which you can identify the DQ Group.
 - Enter a description or related information about the DQ Group.
 - Select the **Folder** (available for selected Information Domain) from the drop down list.
3. In the Map DQ Rules section, perform the following:
 - Select the required DQ Rule from the **Available Rules** list and click **Select**. You can also search to select a specific DQ Rule by entering the required keyword and clicking **Find** button.
 - To select all the listed DQ Rules, click **Select All**.

You can also deselect a DQ Rule by selecting from the Mapped Rules list and clicking **Deselect** or clicking **Deselect All** to deselect all the mapped rules. You can search to deselect a specific DQ Rule by entering the keyword and clicking button. You can search to deselect a specific DQ Rule by entering the keyword and clicking **Find** button.

4. Click **Save**. The defined DQ group is listed in the **Data Quality Rule Summary** screen and can be executed for processing.

Execute Data Quality Group

You can execute a defined DQ Group Definitions along with the mapped Rules and validation checks in the **Data Quality Rule Summary** screen. You can also execute a DQ Group in the Batch Execution screen of Operations module.

To Execute a DQ Group in the **Data Quality Rule Summary** screen:

1. Select the checkbox adjacent to the required Group Name.
2. Click **Execute** button from the Data Quality Groups tool bar. Execute button is disabled if you have selected multiple check boxes. The **Group Execution** screen is displayed.
3. In the Batch Details section, perform the following:
 - Select the **MIS Date** using the Calendar. MIS Date refers to the date with which the data for the execution would be filtered. In case MIS date is not present in the target table, execution happens ignoring the date parameter.

Note: The **DQ Batch ID** is auto populated and is not editable.

- Specify the percentage of **Threshold (%)** limit in numeric value. This refers to the maximum percentage of records that can be rejected in a job. If the percentage of failed records exceeds the Rejection Threshold, the job will fail. If the field is left blank, the default the value is set to 100%.
4. Click **Execute**. A confirmation message is displayed and the DQ Group is scheduled for execution.

Once the DQ Group is executed, you can view the details of the execution along with the log information in the **View Logs** screen.

For more information, see View Data Quality Group Summary Log section.

View Data Quality Group

You can view individual Data Quality Group definition details at any given point.

To view the existing DQ Group definition in the **Data Quality Group Summary** screen:

1. Select the check box adjacent to the required Group Name. The mapped DQ Rules are displayed in the Data Quality Rules section.
2. Click **View** button from the Data Quality Groups tool bar. The Data Quality Group Definition screen displays the DQ definition details.

Modify Data Quality Group

You can update the existing DQ Group definition details except for the Group Name.

To updated the required DQ Group definition details in the **Data Quality Groups Summary** screen:

1. Select the check box adjacent to the required Group Name.
2. Click **Edit** button from the Data Quality Groups tool bar. The Data Quality Group Definition screen is displayed.
3. Update the details and click **Save** to update the changes.

Copy Data Quality Group

You can copy the existing DQ Group details to quickly create a new DQ definition based on the existing details or by updating the required parameters.

To copy an existing DQ Group definition in the **Data Quality Groups Summary** screen:

1. Select the check box adjacent to the required Group Name in the list whose details are to be duplicated.
2. Click **Copy** button from the Data Quality Groups tool bar. Copy button is disabled if you have selected multiple check boxes. The Data Quality Group Definition screen is displayed.
3. Edit the DQ Group Name and other details as required.
4. Click **Save**. The new DQ Group definition is displayed in the **Data Quality Groups Summary** screen.

View Data Quality Group Summary Log

You can view the execution log details of Data Quality Rules in the View Logs screen. The **View Logs** screen displays the details such as Check Name, Log Message, Message Date, Message Time, Total Rows, Rows Impacted, Assignment Type, Assignment Severity, and Severity Message of the executed Data Quality Rules.

To view the Data Quality Rule execution log details in the **Data Quality Groups Summary** screen:

1. Select the check box adjacent to the Group Name in the **Data Quality Groups** grid.

The Data Quality Rules associated with the selected Group are displayed in the Data Quality Rules grid.

2. Select the check box adjacent to the DQ Name in the Data Quality Rules grid.
3. Click **View Logs** button. The View Logs screen is displayed.
4. In the View Logs screen, select the **Information Date** from the drop down list.

Based on the selection, you can select the **Group Run ID** and **Iteration ID** from the corresponding drop-down list.

5. Click the below button from the Group Execution details tool bar.



The Data Quality Rule Logs grid displays the execution log details of the selected Data Quality Rule. You can also click **Reset** button in the Group Execution details tool bar to reset the selection.

Delete Data Quality Group

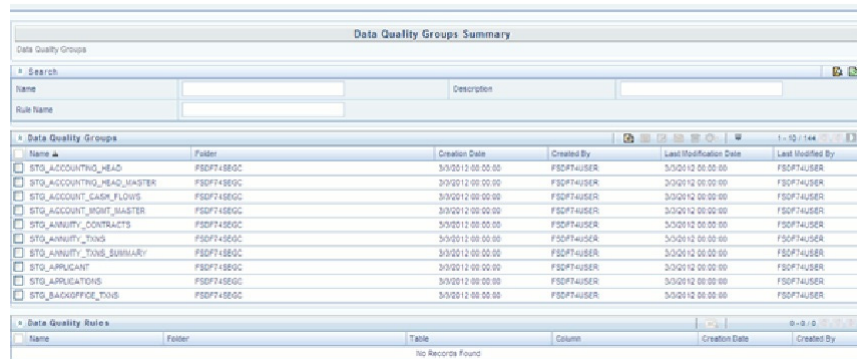
You can remove the DQ Group definition(s) which are created by you and which are no longer required in the system by deleting from Data Quality Groups Summary screen.

To delete:

1. Select the check box adjacent to the required Group Name whose details are to be removed.
2. Click **Delete** button from the Data Quality Groups tool bar.
3. Click **OK** in the information dialog to confirm deletion.

To execute the Data Quality Groups from the *Data Quality Groups Summary* window:

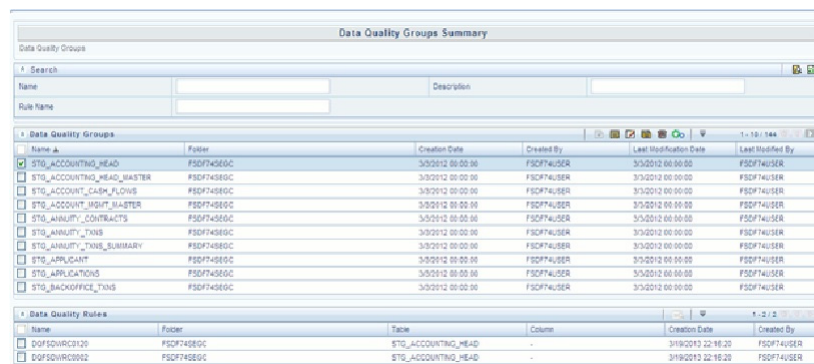
Figure 11. DQ Group Execution



1. Select the check box adjacent to the Group, you want to execute.

The corresponding DQ Rules are displayed in *Data Quality Rules* grid.

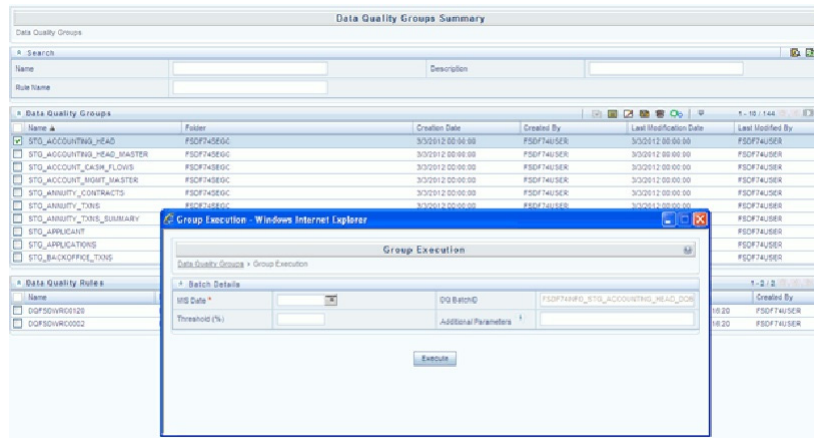
Figure 12. DQ Rules



2. Click **Execute** button.

The *Group Execution* window is displayed.

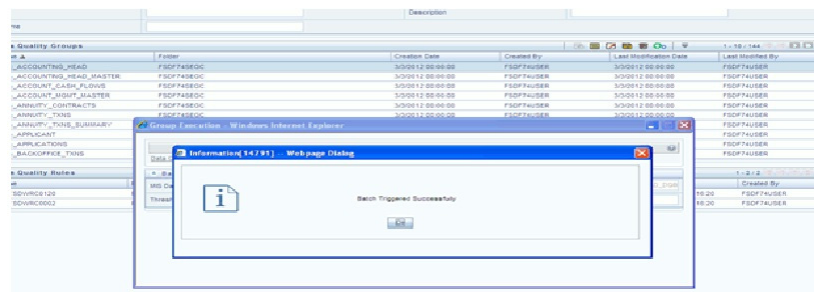
Figure 13. Group Execution



3. Enter the MIS Date.
4. Click **Execute** button from the *Group Execution* window.

The execution is triggered and the following message is displayed.

Figure 14. Execution Message



Once the execution is complete, you can view the details from the *Data Quality Rules Summary* window.

To view the execution logs:

1. Select the check box adjacent to the rule, of which you want to see the execution log.

Figure 15. Data Quality Summary Window

Data Quality Groups Summary						
Data Quality Groups						
Search						
Name			Description			
Rule Name						
Data Quality Groups						
Name	Folder	Creation Date	Created By	Last Modification Date	Last Modified By	
<input checked="" type="checkbox"/>	STG_ACCOUNTING_HEAD	FSDPT4SEGC	3/9/2012 00:00:00	FSDPT4USER	3/9/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_ACCOUNTING_HEAD_MASTER	FSDPT4SEGC	3/9/2012 00:00:00	FSDPT4USER	3/9/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_ACCOUNT_CASH_FLOWS	FSDPT4SEGC	3/9/2012 00:00:00	FSDPT4USER	3/9/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_ACCOUNT_MGMT_MASTER	FSDPT4SEGC	3/9/2012 00:00:00	FSDPT4USER	3/9/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_ANNUITY_CONTRACTS	FSDPT4SEGC	3/9/2012 00:00:00	FSDPT4USER	3/9/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_ANNUITY_TXNS	FSDPT4SEGC	3/9/2012 00:00:00	FSDPT4USER	3/9/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_ANNUITY_TXNS_SUMMARY	FSDPT4SEGC	3/9/2012 00:00:00	FSDPT4USER	3/9/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_APPUCANT	FSDPT4SEGC	3/9/2012 00:00:00	FSDPT4USER	3/9/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_APPUCANTXNS	FSDPT4SEGC	3/9/2012 00:00:00	FSDPT4USER	3/9/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_BADOFFICE_TXNS	FSDPT4SEGC	3/9/2012 00:00:00	FSDPT4USER	3/9/2012 00:00:00	FSDPT4USER
Data Quality Rules						
Name	Folder	Table	Column	Creation Date	Created By	
<input checked="" type="checkbox"/>	DQFSDVRC0120	FSDPT4SEGC	STG_ACCOUNTING_HEAD	-	3/9/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	DQFSDVRC0202	FSDPT4SEGC	STG_ACCOUNTING_HEAD	-	3/18/2013 22:18:21	FSDPT4USER

2. Click **View Logs** button.

The *View Logs* window is displayed.

Figure 16. View Logs

View Logs								
Data Quality Groups > View Logs								
Group Execution details								
Information Date	Select Information Date	Group Run ID	Select Group Run ID					
Iteration ID								
Data Quality Rule Logs								
Check Name	Log Message	Message Date	Message Time	Total Rows	Rows Impacted	Assignment Type	Assignment Severity	Severity Message
No Records Found								

3. Select the **Information Date** from the drop down list.

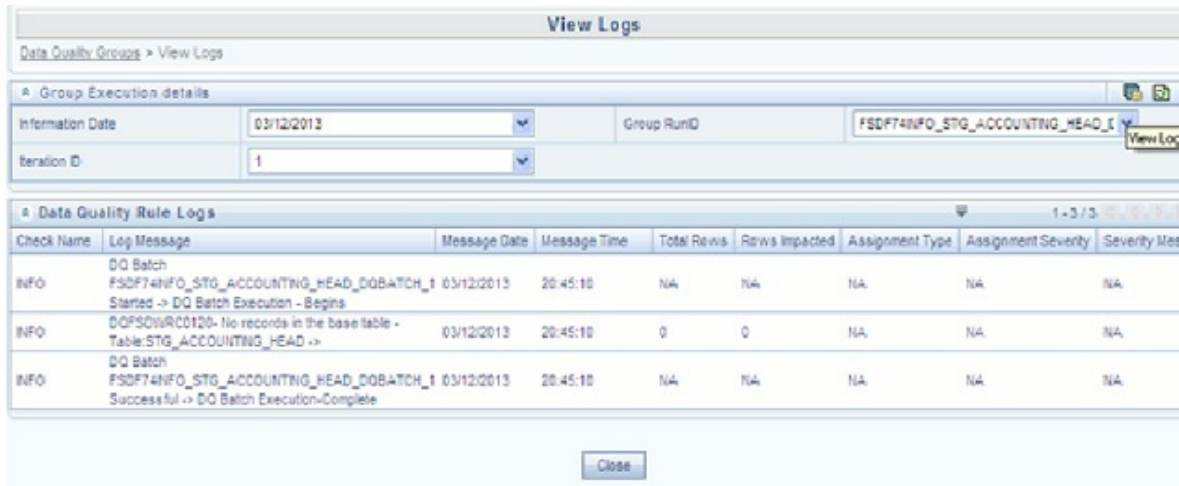
4. Select the **Group Run ID** from the drop down list.

5. Select the **Iteration ID** from the drop down list.

6. Click **View Logs** button.

The details of the selected Group Execution are displayed.

Figure 17. Group Execution Details



Batch Execution of DQ Rules

Create a Batch for executing DQ Rules and add a Task to the selected Batch. Add component as RUN DQ RULE and in Dynamic Parameter List, add a DQ Group.

A single Batch can have multiple number of Tasks and each Task is executing the DQ Group. Batch execution facilitates the execution of multiple Groups simultaneously.

You can also use the **Include** or **Exclude** functionality to determine which all groups have to be executed.

The following batches need to be executed in OFSDF.

Table 20. Batch IDs and their Descriptions

V_BATCH_ID	V_BATCH_DESCRIPTION
<Infodom>_ALM	Data Quality batch for ALM tables
<Infodom>_CAMPAIGN	Data Quality batch for CAMPAIGN tables
<Infodom>_COLLATERAL	Data Quality batch for Collateral tables
<Infodom>_COLLECTION_AND_RECOVERY	Data Quality batch for Collection and Recovery tables
<Infodom>_CRM	Data Quality batch for CRM tables
<Infodom>_CUSTOMER	Data Quality batch for Customer tables
<Infodom>_EXPOSURE	Data Quality batch for EXPOSURE tables
<Infodom>_GL_AND_ACCOUNTING	Data Quality batch for GL and Accounting group
<Infodom>_LRM	Data Quality batch for LRM tables
<Infodom>_MARKET_RISK	Data Quality batch for MARKET RISK tables
<Infodom>_MASTER	Data Quality batch for master tables

Table 20. Batch IDs and their Descriptions

V_BATCH_ID	V_BATCH_DESCRIPTION
<Infodom>_MISCELLANEOUS	Data Quality batch for Miscellaneous tables
<Infodom>_OP_RISK	Data Quality batch for OP RISK tables
<Infodom>_ORIGINATION	Data Quality batch for Origination tables
<Infodom>_PRODUCT_PROCESSORS	Data Quality batch for Product Processors group
<Infodom>_RATES	Data Quality batch for Rates tables
<Infodom>_RATING	Data Quality batch for RATING tables
<Infodom>_SECURITIZATION	Data Quality batch for Securitization tables
<Infodom>_TRANSACTIONS	Data Quality batch for Transactions tables
<Infodom>_TRANSACTION_SUMMARY	Data Quality batch for Transaction Summary tables

The DQ Batches and corresponding groups are detailed in the [DO Batch Group Mapping](#) spreadsheet.

For more information regarding Batch Execution, refer to **Operations** module of *Oracle Financial Services Analytical Applications Infrastructure User Manual*.

This chapter provides information about comparing Data Model Reports of two release versions in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this feature.

The first section gives you an understanding of the Data Model Report extracted from the ERwin Data Modeling tool. The Comparing Data Model Reports section details the steps to use the OFSAA application and download the Difference Report between two Data Model release versions.

This chapter includes the following topics:

- Creating Data Model Report from ERwin
- Extracting Data Model Report from ERwin
- Comparing Data Model Reports

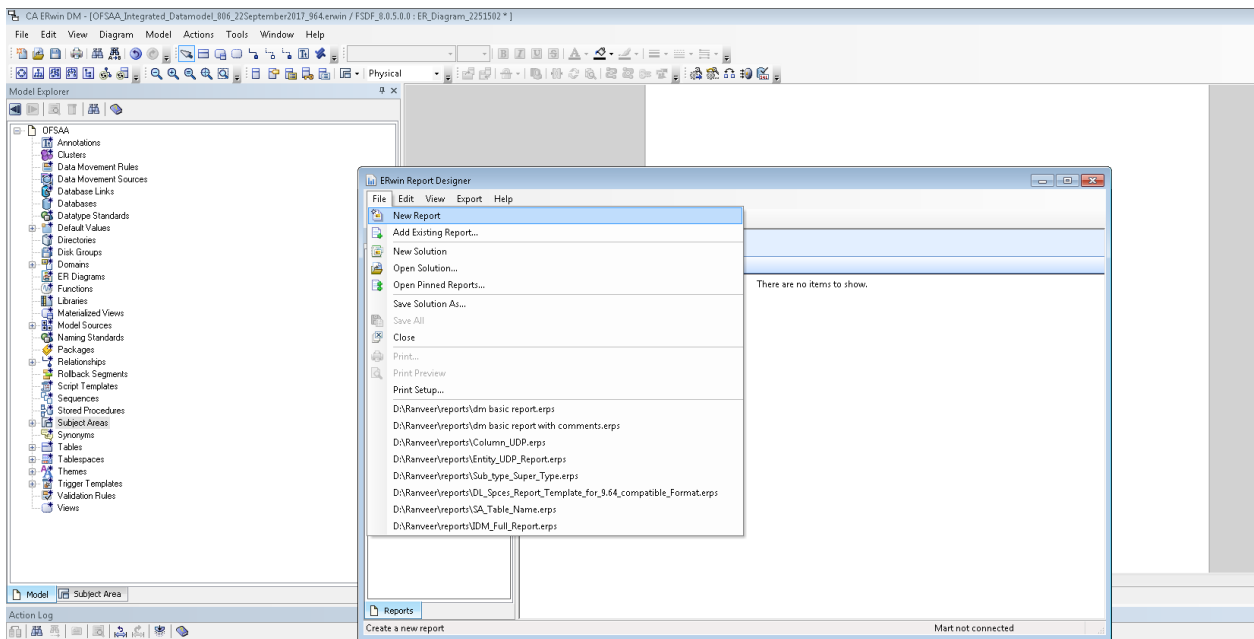
Creating Data Model Report from ERwin

You can create new Data Model Reports from the ERwin Data Modeling tool if there are no **.erps** files to extract the information.

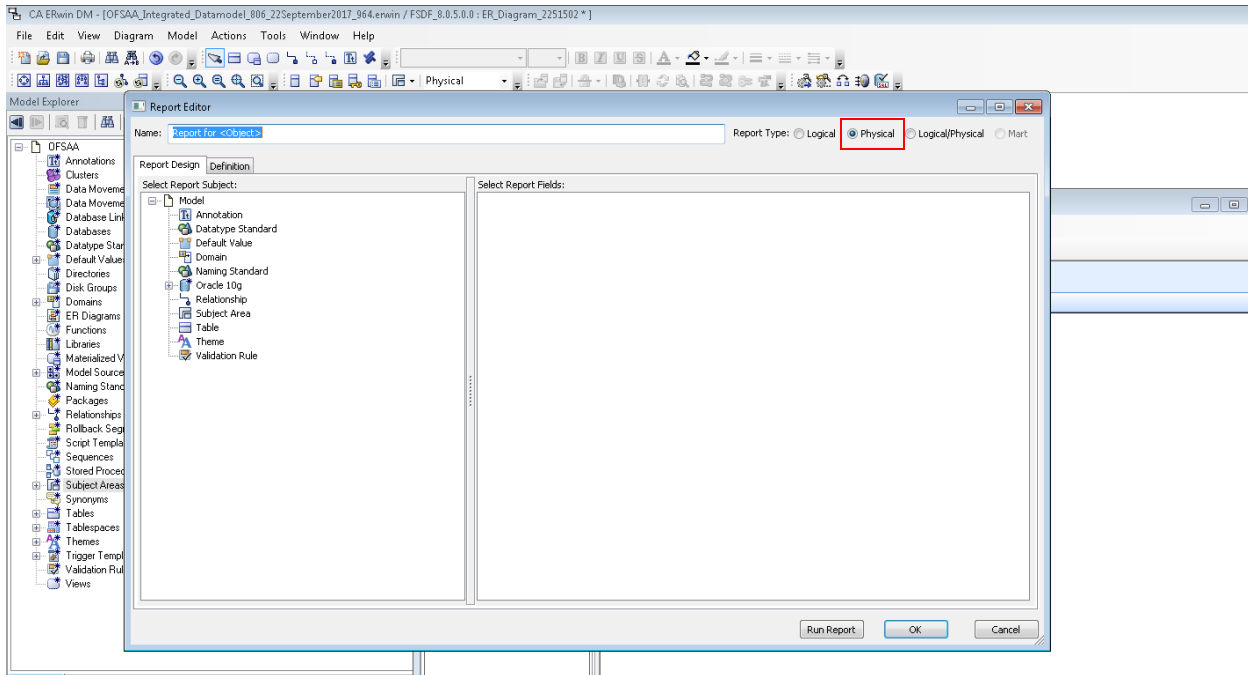
Note: If the Data Model Reports are existing, see Extracting Data Model Report from ERwin for more information.

Perform the following steps to extract the Data Model Report form ERwin Data Modeling tool:

1. In the *ERwin Report Designer* window, select **File --> New Report** (to create the **.erps** report file).



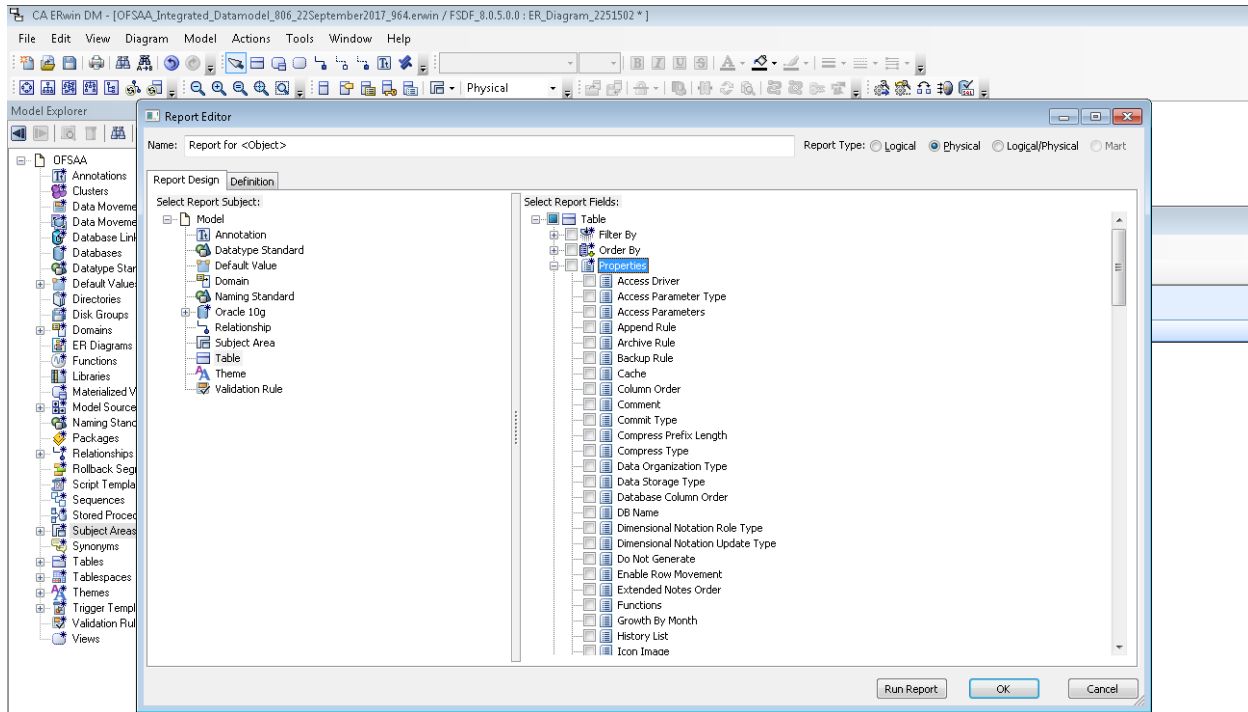
2. In the *Report Editor* window Select **Report Type** as **Physical** --> Select **Report Subject** as **Table**.



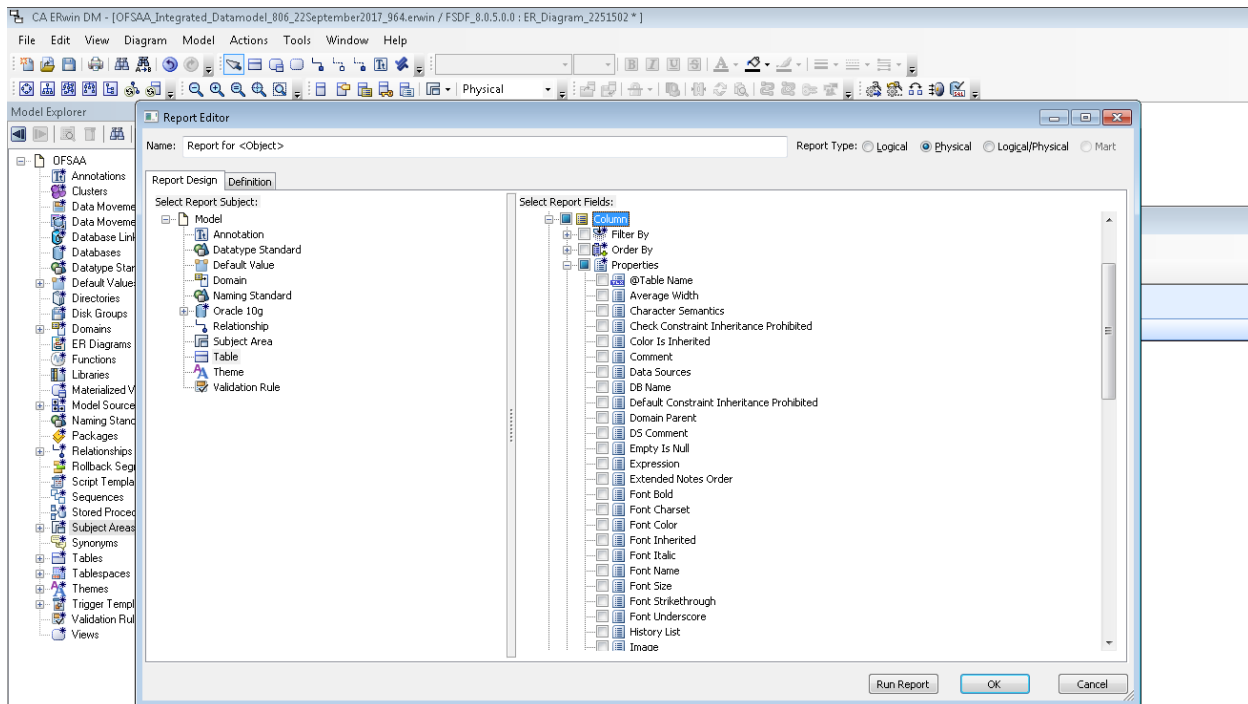
3. Select the Report Fields to match the sample Data Model Report shown as follows:

Entity/Table Physical_Name	Attribute/Column Physical_Name	Physical Data Type	Null Option	Is PK	Is FK	Domain Parent
Dim_Financial_Elements_Attr	financial_elem_id	NUMBER(14)	Not Null	Yes	No	ID_NUMBER
Dim_Financial_Elements_Attr	attribute_id	NUMBER(22)	Not Null	Yes	No	Number
Dim_Financial_Elements_Attr	dim_attribute_numeric_member	NUMBER(22)	Null	No	No	Number
Dim_Financial_Elements_Attr	dim_attribute_varchar_member	VARCHAR2(30)	Null	No	No	Text_Short_Description
Dim_Financial_Elements_Attr	number_assign_value	NUMBER(22)	Null	No	No	Number
Dim_Financial_Elements_Attr	varchar_assign_value	VARCHAR2(1000)	Null	No	No	Text_Comments_Type2
Dim_Financial_Elements_Attr	date_assign_value	DATE	Null	No	No	Datetime
Dim_Financial_Elements_B	financial_elem_id	NUMBER(14)	Not Null	Yes	No	ID_NUMBER
Dim_Financial_Elements_B	financial_elem_display_code	NUMBER(14)	Not Null	No	No	Number
Dim_Financial_Elements_B	enabled_flag	VARCHAR2(1)	Not Null	No	No	Indicator
Dim_Financial_Elements_B	leaf_only_flag	VARCHAR2(1)	Null	No	No	Indicator
Dim_Financial_Elements_B	definition_language	VARCHAR2(10)	Not Null	No	No	Code_Alphanumeric_Medium
Dim_Financial_Elements_B	created_by	VARCHAR2(30)	Not Null	No	No	Text_Short_Description
Dim_Financial_Elements_B	creation_date	TIMESTAMP	Not Null	No	No	Datetime
Dim_Financial_Elements_B	last_modified_by	VARCHAR2(30)	Not Null	No	No	Text_Short_Description
Dim_Financial_Elements_B	last_modified_date	TIMESTAMP	Not Null	No	No	Datetime
Dim_Financial_Elements_B	financial_elem_code	VARCHAR2(20)	Null	No	No	VARCHAR2
Dim_Financial_Elements_Hier	hierarchy_id	NUMBER(10)	Not Null	Yes	No	Number_Medium
Dim_Financial_Elements_Hier	parent_depth_num	NUMBER(22)	Not Null	No	No	Number_Generic
Dim_Financial_Elements_Hier	parent_id	NUMBER(14)	Not Null	Yes	No	ID_NUMBER
Dim_Financial_Elements_Hier	child_depth_num	NUMBER(22)	Not Null	No	No	Number

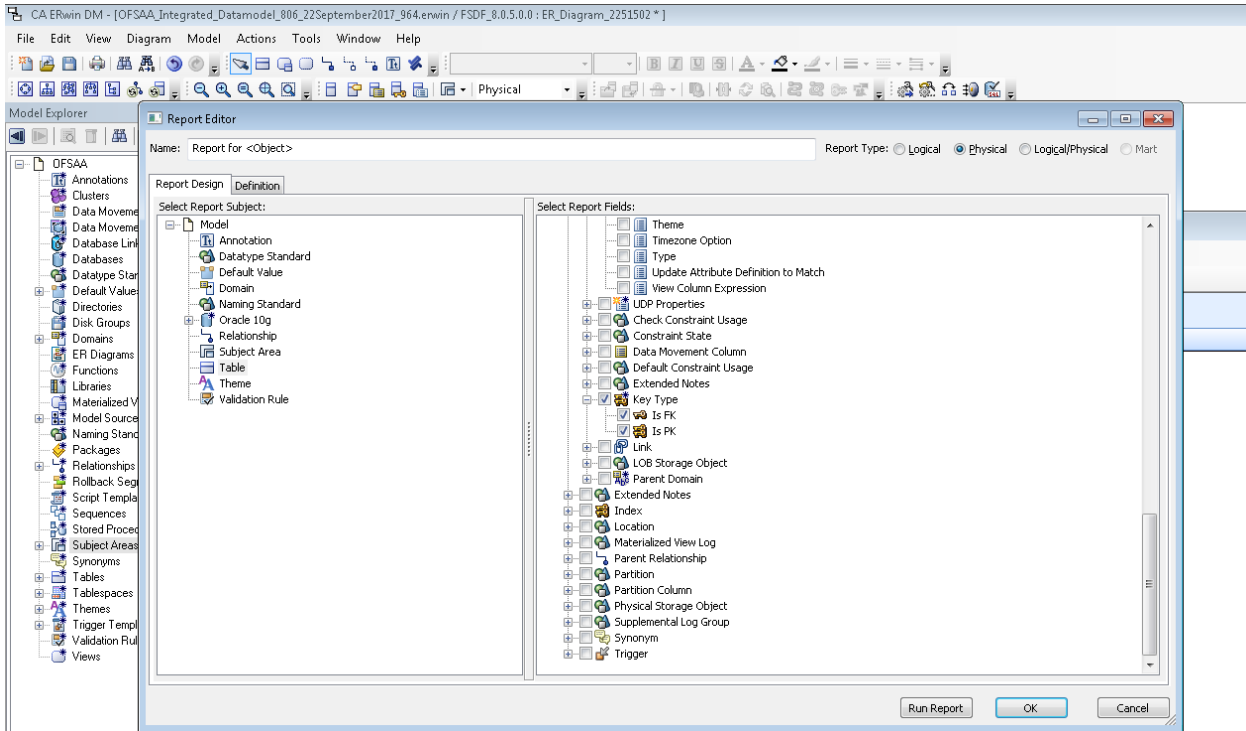
4. Under **Table --> Properties**, select **Physical Name (Entity/Table Physical_Name)**.



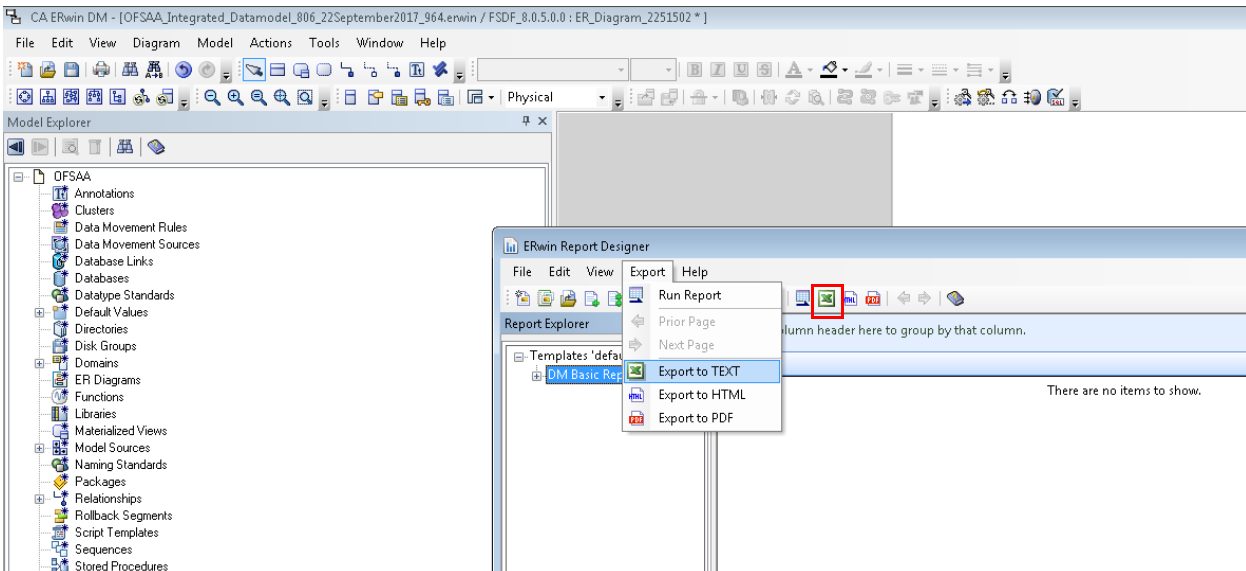
5. Under **Table --> Column**, select **Physical Name (Attribute/Column Physical_Name)**, **Physical Data Type**, **Null Option**, and **Domain Parent**.



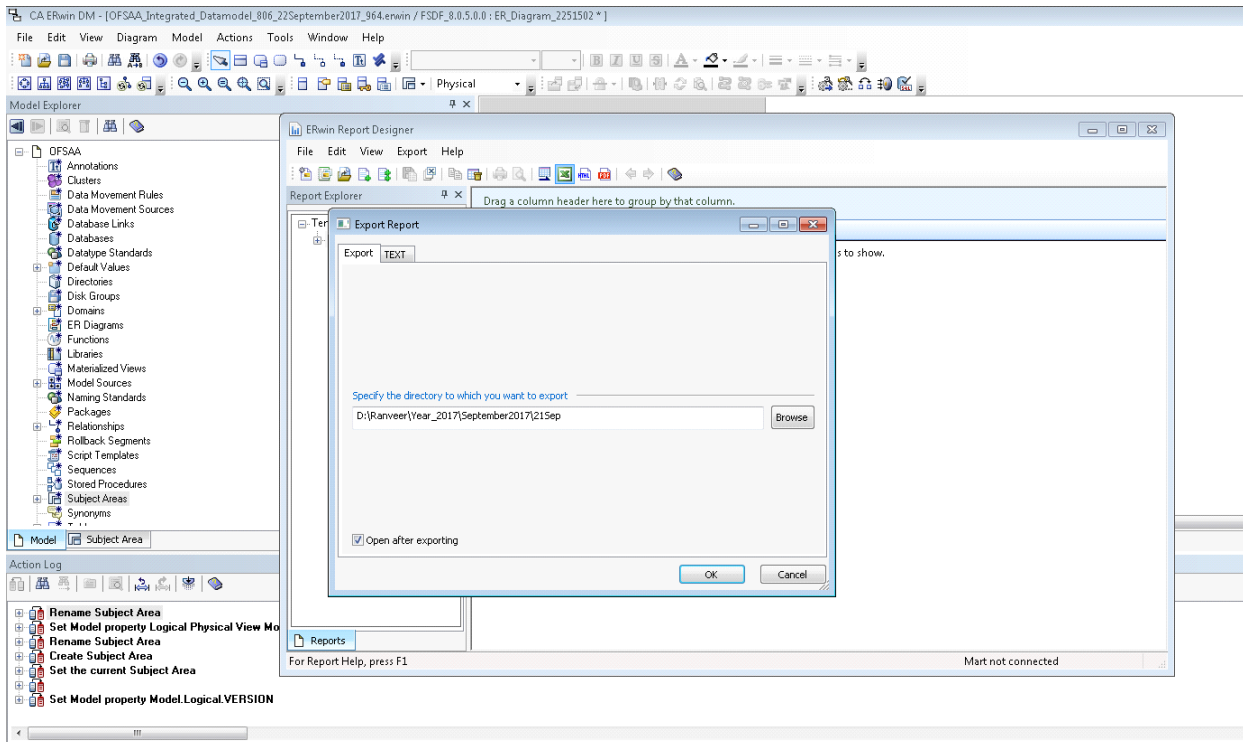
6. Under **Table** --> **Key Type**, select **Is PK** and **Is FK**. Click **OK**.



7. In the *ERwin Report Designer* window, select **Export** --> **Export to Text (Excel Format)** or the **XL** icon.



8. Enter or Browse the Export file path where you want to save the Data Model Report file.



9. The generated Data Model Report file will open and must be in the following format.

	A	B	C	D	E	F	G	H
1	Entity/Table Physical_Name	Attribute/Column Physical_Name	Physical Data Type	Null Option	Is PK	Is FK	Domain Parent	
2	Dim_Financial_Elements_Attr	financial_elem_id	NUMBER(14)	Not Null	Yes	No	ID_NUMBER	
3	Dim_Financial_Elements_Attr	attribute_id	NUMBER(22)	Not Null	Yes	No	Number	
4	Dim_Financial_Elements_Attr	dim_attribute_numeric_member	NUMBER(22)	Null	No	No	Number	
5	Dim_Financial_Elements_Attr	dim_attribute_varchar_member	VARCHAR2(30)	Null	No	No	Text_Short_Description	
6	Dim_Financial_Elements_Attr	number_assign_value	NUMBER(22)	Null	No	No	Number	
7	Dim_Financial_Elements_Attr	varchar_assign_value	VARCHAR2(1000)	Null	No	No	Text_Comments_Type2	
8	Dim_Financial_Elements_Attr	date_assign_value	DATE	Null	No	No	Datetime	
9	Dim_Financial_Elements_B	financial_elem_id	NUMBER(14)	Not Null	Yes	No	ID_NUMBER	
10	Dim_Financial_Elements_B	financial_elem_display_code	NUMBER(14)	Not Null	No	No	Number	
11	Dim_Financial_Elements_B	enabled_flag	VARCHAR2(1)	Not Null	No	No	Indicator	
12	Dim_Financial_Elements_B	leaf_only_flag	VARCHAR2(1)	Null	No	No	Indicator	
13	Dim_Financial_Elements_B	definition_language	VARCHAR2(10)	Not Null	No	No	Code_Alphanumeric_Medium	
14	Dim_Financial_Elements_B	created_by	VARCHAR2(30)	Not Null	No	No	Text_Short_Description	
15	Dim_Financial_Elements_B	creation_date	TIMESTAMP	Not Null	No	No	Datetime	
16	Dim_Financial_Elements_B	last_modified_by	VARCHAR2(30)	Not Null	No	No	Text_Short_Description	
17	Dim_Financial_Elements_B	last_modified_date	TIMESTAMP	Not Null	No	No	Datetime	
18	Dim_Financial_Elements_B	financial_elem_code	VARCHAR2(20)	Null	No	No	VARCHAR2	
19	Dim_Financial_Elements_Hier	hierarchy_id	NUMBER(10)	Not Null	Yes	No	Number_Medium	
20	Dim_Financial_Elements_Hier	parent_depth_num	NUMBER(22)	Not Null	No	No	Number_Generic	
21	Dim_Financial_Elements_Hier	parent_id	NUMBER(14)	Not Null	Yes	No	ID_NUMBER	
22	Dim_Financial_Elements_Hier	child_depth_num	NUMBER(22)	Not Null	No	No	Number	

Note: Ensure that the file is saved in **.xlsx** format.

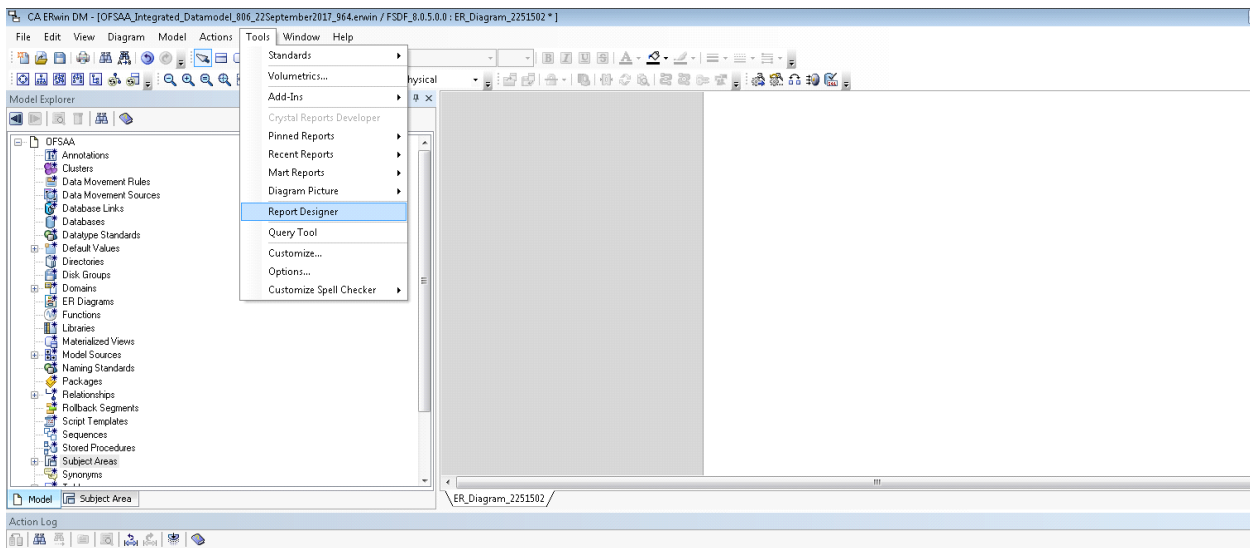
Extracting Data Model Report from ERwin

OFSDF is a collection of data model artifacts delivered as ERwin files or can be extracted as **.XLS** file from ERwin Data Modeling tool. OFSDF hence requires a license of the ERwin Data Modeling tool.

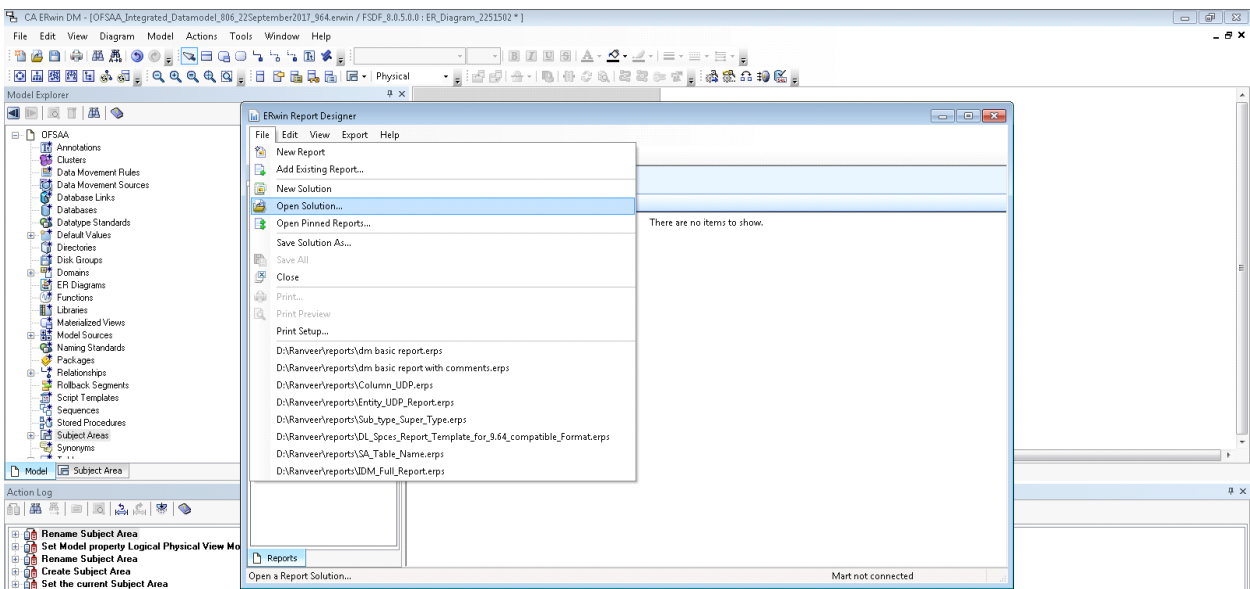
ERwin is the current and only supported modeling tool to view and edit the model. Currently, the minimum versions of ERwin supported are 9.5 and 9.64.

Perform the following steps to extract the Data Model Report form ERwin Data Modeling tool:

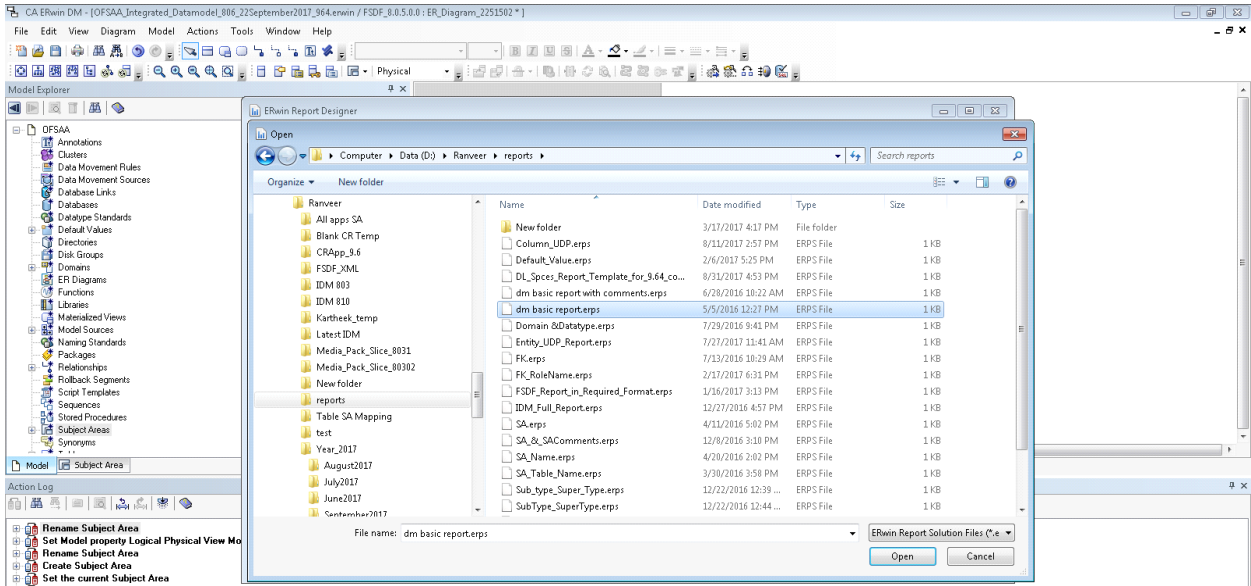
1. Open the Erwin Data Modeling tool.
2. Select **Tools --> Report Designer** from the Menu bar.



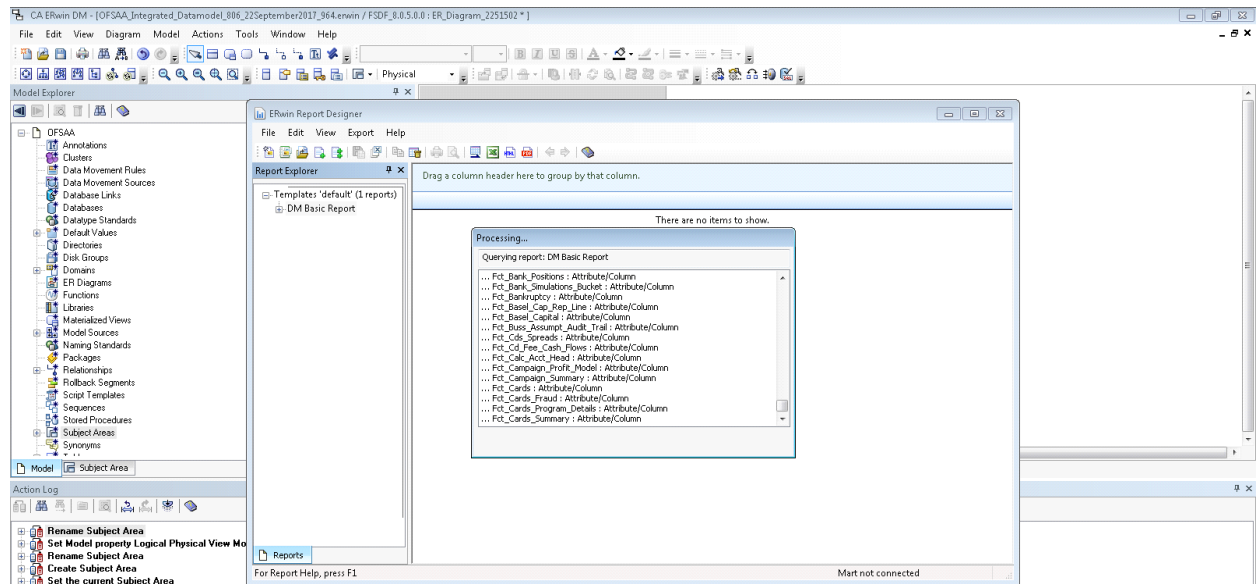
3. In the *ERwin Report Designer* window, select **File --> Open Solution** (to extract the **.erps** report file).



4. Browse the path of the .erps file, select the file and click **Open**.



5. The existing Data Model Report file is extracted and processed to export the file in XL format.

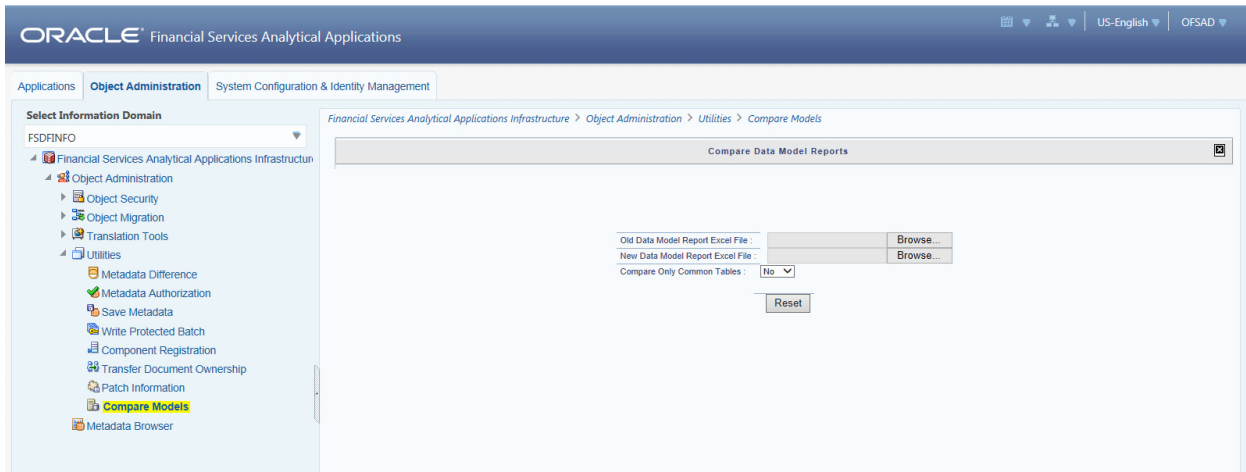


6. The generated Data Model Report file in XL format is opened. Save the file in **.xlsx** format.

Comparing Data Model Reports

Perform the following steps to extract the Data Model Report form ERwin Data Modeling tool:

1. Open the OFSA Application with your login credentials.
2. In the **Object Administration** tab, Select **Object Administration --> Utilities --> Compare Models**.



3. Browse the **Old Data Model Report Excel File** and **New Data Model Report Excel File** respectively. The **Compare** button is displayed.

Note: The **Old Data Model Report** and **New Data Model Report** should be in same format and file extension should be **.xlsx**

Note: Click **Reset** if you want to select a different Data Model Report.

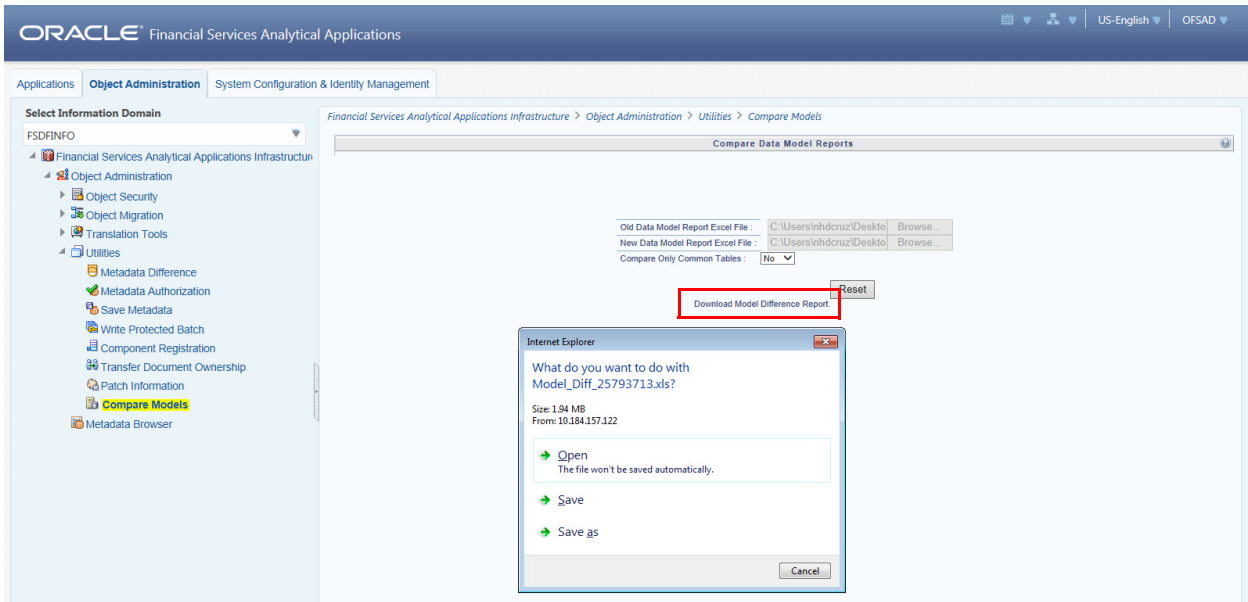
4. The **Compare Only Common Tables** option is “No” by default, select “Yes” only if required.

Note: The Compare Only Common Tables option “Yes” is used when comparing Data Model Report of different products. The Report fields will be different for each product and only common fields are required to compare for investigation.

5. Click **Compare**. The Data Model Difference Report will get generated.



6. Click **Download Model Difference Report** link and **Save** the report.



Naming Conventions Used in OFSDF Data Model

This Appendix chapter explains the various naming conventions used in OFSDF Logical Data Model and Physical Data Model. In addition, the domains in PDM and LDM are also listed with their descriptions.

This appendix covers the following topic:

- Naming Conventions Used in OFSDF LDM and PDM

Naming Conventions Used in OFSDF LDM and PDM

ERwin Data Model consists of Logical and Physical data structures for each model file.

The following section explains the various naming conventions used in Oracle Financial Services Data Foundation (OFSDF) Logical Data Model (LDM) and Physical Data Model (PDM).

- OFSDF Logical Data Model Naming Conventions
- OFSDF Physical Data Model Naming Conventions
- Domains (PDM and LDM)

OFSDF Logical Data Model Naming Conventions

ERwin modeler tool allows two views namely Logical view and Physical view for each model file. Accordingly, the OFSDF LDM file can be viewed in logical view mode and physical view mode. Different naming conventions and standards are applied to the two views of the OFSDF LDM.

Note: The physical view of the OFSDF LDM is not the same as the OFSDF PDM. The PDM is a distinct model and is shipped as a separate ERwin File.

- LDM Logical View Mode Entity Naming Conventions
- LDM Physical View Mode Table Naming Conventions

LDM Logical View Mode Entity Naming Conventions

Entities in the logical view are named in such a way that the names themselves convey the functional meaning of the entity. The first letter of each word in an entity name is capitalized (e.g. "Application Stage"). Entity names also contain common suffixes for the classes of entities listed in the table below:

Attribute names also contain common set of suffixes for the classes of attributes as listed in table below:

Table 21. Entity Naming Conventions

Class of Entity	Suffix	Example
Type	Type	Asset-Liability Type
Group	Group	Product Group
Code	Code	Collection Status Code
History	History	Account Balance History
Associative entities	Map	Customer Account Map

Table 22. Attribute Name Suffixes

Class of Attribute	Suffix	Required Values (If any)
Unique Identifier	ID	
Reference Identifiers	Reference ID	
Code	Code	
Number	Number	
Name	Name	
Description	Description	
Boolean Values	Flag	Y or N
Indicator	Indicator	More than one value ('A', 'B', 'C' etc)
Percentage	Percent	
Rate	Rate	
Amount	Amount	
Balance	Balance	
Term	Term	
Frequency	Frequency	
Unit	Unit	
Record created by		Created By
Record modified by		Modified By
Record created on		Created On
Record modified on		Modified On
Record validity start date		From Date
Record validity end date		End Date

LDM Physical View Mode Table Naming Conventions

In the physical view of the OFSDF LDM follows a different naming convention for entities and attributes in the model, and these reflect naming standards that are more readily acceptable to table and column naming constraints of common database systems.

The entity names in the physical view mode of the LDM are capitalized and each entity follows the following naming pattern:

- FSW_<Subject Area>_<Table Name>
- 'FSW' is a constant prefix.
- <Subject Area> is abbreviated form of the entity's primary subject area.

- <Table Name> is a meaningful name for the entity with no embedded spaces between words. Table Name may be abbreviated.

Table 23. Naming Conventions

Logical View Name	Physical View Name	Description
Account Address	FSW_ACCT_ACCOUNT_AD DRESS	Account Address entity belonging primarily to Account (ACCT) subject area.
Letter Of Credit	FSW_CON_LETTER_OF_CR EDIT	Letter of Credit entity belongs primarily to the Contracts (CON) subject area.

OFSDF Physical Data Model Naming Conventions

The OFSDF PDM consists of tables grouped into two distinct areas namely the Staging Area and Results Area. The Staging Area consists of tables for data sourcing and the Results Area consists of the star schemas / datamarts for reporting and BI.

ERwin modeler tool allows two views namely Logical view and Physical view for each model file. Accordingly, the OFSDF PDM (Physical Data Model) file can be viewed in logical view mode and physical view mode. Different naming conventions and standards are applied to the two views of the OFSDF Physical Data Model.

- PDM Logical View Mode Naming Conventions
- PDM Physical View Mode Naming Conventions

PDM Logical View Mode Naming Conventions

In the logical view model, OFSDF PDM model tables and columns have descriptive names that readily convey the meaning and use of the element. In the logical view, names of tables and columns can have more than one word with the first letter of each word capitalized. Staging Area structures and Results Area structures have different name prefixes and suffixes as outlined in the table below:

Table 24. Prefixes and Suffixes

PDM Area		Prefix	Suffix	Example
Staging (Data Sourcing)	All Tables	Stage		Stage Internet Transactions
	Transaction Tables		Transactions	Stage Card Settlement Transactions

Table 24. Prefixes and Suffixes

PDM Area		Prefix	Suffix	Example
	Master Tables		Master	Stage Product Master
Results (Datamart)	All Fact Tables	Fact		Fact Process Cashflow
	All Dimension Tables		Dimension	Account Dimension

PDM Physical View Mode Naming Conventions

In the physical view model, OFSDF PDM tables and columns may have abbreviated words joined by underscore character to form more meaningful and descriptive names.

Table names in the physical view are capitalized

Table 25. Physical view

PDM Area		Prefix	Suffix	Example
Staging (Data Sourcing)	All Tables	STG_		STG_LOAN_CO NTRACTS
	Interface tables for data from MDM and core banking systems		_INTF	STG_ORG_UNI T_ATTR_INTF
	Transaction Tables		_TXNS	STG_CARDS_SE TTLEMENT_TX NS
	Master Tables		_MASTER	STG_CUSTOME R_MASTER
Results (Datamart)	All Fact Tables	FCT_		FCT_ACCOUNT _SUMMARY
	All Dimension Tables	DIM_		DIM_ DIM_PRODUCT

The table below lists the prefix and/or suffix used for columns names in the physical view of the OFSDF PDM. The prefix or suffix depends on the class and data type of the column.

Column name prefix to indicate column datatype:

Table 26. Column Name Prefix

Column Data Type	Prefix
Varchar	v_
Number	n_
Date	d_
Flag	f_

Column name suffix for common classes of columns:

Table 27. Column Name Suffix

Column Class	Suffix
Method	_method
Percentage	_pct
Rate	_rate
Balance	_bal/_balance
Amount	_amt/_amount
Term	_term
Type	_type
Frequency	_freq

In addition, frequently occurring keywords in column names may be abbreviated as shown in table below:

Table 28. Abbreviations

Name	Abbreviated Form	Name	Abbreviated Form
Accrual	accr	Local Currency	lcy
Account	acct	Line Of Business	lob
Accounting Currency	acy	Maximum	max
Address	addr	Minimum	min
Adjustment	adj	Mortgage	mort
Advance	adv	Message	msg
Amount	amt	Multiplier	mult
Application	app	Number	num
Average	avg	Over Draft	od
Balance	bal	Option	opt
Business	bus	Origination	org
Currency	ccy	Percent	pct
Consolidation	cons	Payment	pmt
Customer	cust	Prepayment	ppmt

Table 28. Abbreviations

Name	Abbreviated Form	Name	Abbreviated Form
Description	desc	Product	prod
Dimension	dim	Source	src
Detail	dtl	Status	stat
Earnings at Risk	ear	Statistics	stats
End Of Period	eop	Temporary	temp
Error	err	Total	tot
Flag	flg	Transaction	txn
Frequency	freq	Value at Risk	var
Future	fut	Value	val
Forex	fx		
Generation	gen		
General Ledger	gl		
Hierarchy	hier		
History	hist		

Domains (PDM and LDM)

Domains are Logical data types that are attached to each column within the model. The following table lists the domains and their descriptions.

Table 29. Domains and their Descriptions

Serial No	Domain Name	Domain Description
1	Date	DATE
2	Timestamp	TIMESTAMP
3	Number	NUMBER(10)
4	Amount	NUMBER(22,3)
5	Code	NUMBER(5)
6	Flag	CHAR(1)
7	Frequency	NUMBER(5)
8	ID	VARCHAR2(25)
9	Percent	NUMBER(10,6)
10	Percent_Long	NUMBER(15,11)
11	Phone_Fax_Number	NUMBER(15)
12	Rate	NUMBER(10,6)
13	Term	NUMBER(5)
14	Alphanumeric_Code	VARCHAR2(10)
15	Name	VARCHAR2(60)
16	Currency_Code	VARCHAR2(3)
17	Short_Description	VARCHAR2(60)

Table 29. Domains and their Descriptions

Serial No	Domain Name	Domain Description
18	Description	VARCHAR2(255)
19	Account_Number	VARCHAR2(25)
20	System_Identifier	NUMBER(20)
21	Long_Description	VARCHAR2(4000)

Rate and Percentage

Data in the columns associated with the below mentioned domains must be provided as a counting number (a whole number, which must not begin from 0).

- LONG_RATE
- Rate
- RATE
- Short_Rate
- RATE_LONG
- Number_Percentage
- Percent
- Percent_Long

For example: If the interest rate is 8.9, then FSDF considers 8.9 as the value in the column instead of 0.089, because 0.089 is not valid as interest rate value.

Custom Reporting Line Codes

Custom values can be added in the DIM_REP_LINE table. DIM_REP_LINE custom range is between 10000000001 and 100002000001 for custom values for the N_REP_LINE_CD sequence.

This Appendix provides information about How to Define a Batch in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topic:

- Batch Definition

Batch Definition

Create a batch from the OFSAAI *Batch Maintenance* screen as follows:

Procedure

1. From the OFSAAI **Home** menu, navigate to **Operations > Batch Maintenance**.
2. In the *Batch Maintenance* window, Select '+' button from the *Batch Name* tool bar.

The *New Batch Definition* window is displayed.

3. Enter the Batch details as tabulated.

Table 30. Batch Details

Field	Description
Batch Name	The Batch Name is auto generated by the system. You can edit to specify a Batch name based on the following conditions: <ul style="list-style-type: none"> • The Batch Name should be unique across the Information Domain. • The Batch Name must be alphanumeric and should not start with a number. • The Batch Name should not exceed 41 characters in length. • The Batch Name should not contain special characters "." and "-".
Batch Description	Enter a description for the Batch based on the Batch Name.
Duplicate Batch	(Optional) Select the check box to create a new Batch by duplicating the existing Batch details. On selection, the Batch ID field is enabled.
Batch ID (If duplicate Batch is selected)	It is mandatory to specify the Batch ID if Duplicate Batch option is selected. Select the required Batch ID from the list.
Sequential Batch	Select the check box if the Batch has to be created sequentially based on the task specified. For example, if there are 3 tasks defined in a Batch, task 3 should have precedence as task 2, and task 2 should have precedence as task 1.

-
4. Click **Save** to save the Batch definition details.

The new Batch definition details are displayed in the *Batch Name* section of *Batch Maintenance* window with the specified **Batch ID**.

Template to Generate Data Dictionary and Download Specification for ERwin 9.x

OFSAA data models have been designed and released on 7.x version of ERwin. However, if there is a requirement to upgrade to the latest version of ERwin (9.x series), the existing procedure to generate DL specification using the published report templates fails in the 9.x version. This is because of architectural changes between the two ERwin versions. This chapter provides information about the procedure to generate data dictionary and download specification for ERwin 9.x, for all OFSAA data models qualified on FSDF 8.0.6.

Prerequisite

Upgrade model to 9.x series before initiating generation of the DL specification.

Procedure

Download the excel based template "[OFSAA Data Model Dictionary Template.xls](#)" and follow the steps mentioned below. This template is compatible with all the OFSAA data models that are qualified with FSDF version 8.0.6.0.0 and ERwin version 9.x.

1. Ensure that the Erwin Data Modeler version is ERwin 9 or later.
2. Ensure that macros are enabled in this excel before execution.
3. Open the OFSAA data model in ERwin 9 or a later version before executing this template.
4. Click the "UDP selection Menu" button to launch UDP selection form.
5. Select the appropriate application UDP's (optional).
6. Click "Generate Report" button.
7. On successful execution, the following message is displayed "Report generation is Complete".
8. The download specification and the data dictionary report is located in "OFSAA_Download_Spec" worksheet.

Note :

To generate report for two application UDP's such as "BASEL_III_USA_ADVNCND" and "OR" , you need to manually remove the extra characters from the UDP name in the data model else the execution will throw an error. The UDP names will be fixed in the upcoming release of FSDF.



Oracle Financial Services Data Foundation User Guide

Release 8.0.6.0.0

Oracle Corporation

World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065

U.S.A.

Worldwide Inquiries:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

www.oracle.com/financial_services/

Copyright © 2018 Oracle Financial Services Software Limited. All rights reserved.

No part of this work may be reproduced, stored in a retrieval system, adopted or transmitted in any form or by any means, electronic, mechanical, photographic, graphic, optic recording or otherwise, translated in any language or computer language, without the prior written permission of Oracle Financial Services Software Limited.

Due care has been taken to make this User Guide and accompanying software package as accurate as possible. However, Oracle Financial Services Software Limited makes no representation or warranties with respect to the contents hereof and shall not be responsible for any loss or damage caused to the user by the direct or indirect use of this User Guide and the accompanying Software System. Furthermore, Oracle Financial Services Software Limited reserves the right to alter, modify or otherwise change in any manner the content hereof, without obligation of Oracle Financial Services Software Limited to notify any person of such revision or changes.

All company and product names are trademarks of the respective companies with which they are associated.

