Oracle Financial Services Data Foundation Application Pack

User Guide

Release 8.1.2.0.0
F51012-01
November 2023

Oracle Financial Services Data Foundation Application Pack User Guide, Release 8.1.2.0.0

F51012-01

Primary Authors: (primary author), (primary author)

Contributing Authors: (contributing author), (contributing author)

Contributors: (contributor), (contributor)

Contents

12 Big Data in OFSDF

13 Account Dimension Table

14 Customer Dimension Table

15 Loading Multiple Load Runs in OFSAA

16 Restatement Support Feature in OFSDF

17 Exchange Rates Tables

28 Credit Score Model and Probability of Default Model

29 GL to Management Reporting

39 Key Business Data Use Cases

40 Data Quality Rules Execution

1
Preface

1.1 Preface

1.1.1 What is New in this Release of OFSDF Application Pack

This section lists new features and changes in the Oracle Financial Services Data Foundation (OFSDF) Application Pack.

1.1.1.1 New Features

This section lists the new features described in this manual.

Feature Name	Feature Description
External Engine	Created a process to source the OFSAA External Engine Data into OFSDF in the absence of OFSAA Applications.
Sourcing Account Balances	When you need to add a new Amount and Balance related columns in the Stage tables and they are not available in the Stage PP tables, then you can add those Amount and Balance related columns in the Stage Account Balances table for dynamic sourcing. In the Account Balances tables, the Balance attributes are added in the form of row-level entries.
Sourcing Special Terms and Covenants	The special terms and covenants are precautionary measures put in place to handle the financial risks.
User Interface for Managing Slowly Changing Dimension	This is an existing AAI capability, now also packaged within the Data Foundation. It allows user to view, create and edit SCD metadata.

1.1.1.2 Deprecated Features

There are no deprecated features in this release.

1.1.1.3 Desupported Features

There are no desupported features in this release.

1.1.2 Audience

This document is the user guide and reference guide for the Oracle Financial Services Data Foundation (OFSDF) application pack Release 8.1.2.0.0. This document is intended for System Administrator and all other 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.

1.1.3 Access to Oracle Support

Oracle customers can access electronic support through My Oracle Support (MOS). For more 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.

1.1.4 Related Information Sources

Oracle strives to keep this and all other related documents updated regularly. Visit the OHC Documentation Library and My Oracle Support web pages to download the latest document version available. The list of related documents is as follows:

The following are the OFSDF Application Pack Release 8.1.x.x.x documents:

- Oracle Financial Services Data Foundation Application Pack Installation and Configuration Guide Release 8.1.2.0.0
- Oracle Financial Services Data Foundation Application Pack Release Notes 8.1.2.0.0
- Oracle Financial Services Analytical Applications (OFSAA) Data Model Extension Guidelines Document Release 8.1.x
- Oracle Financial Services Analytical Applications (OFSAA) Data Model Naming Standards Guide Release 8.1.x
- Oracle Financial Services Analytical Applications (OFSAA) Data Model Document Generation Release 8.1.x
- Oracle Financial Services Data Foundation Application Pack Cloning Reference Guide Release 8.1.x
- Oracle Financial Services Data Foundation Application Pack Security Guide Release 8.1.x
- Oracle Financial Services Data Foundation Application Pack Data Protection Implementation Guide Release 8.1.x

The following are the OFS AAAI Application Pack Release 8.1.x.x.x documents:

- Oracle Financial Services Advanced Analytical Applications Infrastructure Installation Guide Release 8.1.2.0.0
- Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0
- Oracle Financial Services Advanced Analytical Applications Infrastructure Environment Check Utility Guide Release 8.1.x
- Oracle Financial Services Advanced Analytical Applications Infrastructure Administration and Configuration Guide Release 8.1.x
- Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0

- Oracle Financial Services Analytical Applications (OFSAA) Metadata Browser User Guide Release 8.1.0.0.0
- Oracle Financial Services Analytical Applications Technology Matrix Release 8.1.2.0.0
- Oracle Financial Services Analytical Applications (OFSAA) Licensing Information User Manual Release 8.1.2.0.0
- Oracle Financial Services Analytical Applications Data Model Utilities User Guide

2
Introduction to OFSDF

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 to run OFSDF.

2.1 Overview

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 enables Financial Services institutions to develop, deploy, and operate analytical solutions covering key functional areas in Financial Services, including:

- Enterprise Risk Management
- Enterprise Performance Management
- Customer Insight
- 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.

2.2 Components of OFSDF

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

Table 2-1 The Components of OFSDF and their description

Criteria	Description
Financial Services Analytical Warehouse Data Model	This is a physical data model that supports data sourcing and reporting related to key analytical use cases in the Financial Services industry. Analytical Warehouse Data Model or Warehouse Model is a physical data model. Warehouse Model is readily deployable. It consists of database object definitions, and additional supporting scripts. Warehouse Model is classified into two distinct sets of tables based on purpose: Staging Model: This model facilitates data sourcing from the internal operational systems of the banks such as Lending Systems, Trading Systems, Collateral Management Systems, and Master Data Management Systems. Reporting Model: This model facilitates the storage of outputs from analytical tools, applications, and engines in a manner that is conducive to BI reporting. The Warehouse Model is typically deployed into production through a set of management tools called the Oracle Financial Services Advanced Analytical Application Infrastructure (OFSAAAI). AAI is a separate product, and is a prerequisite for OFSDF. For more information on AAI, see Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0
Supporting scripts	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.

2.3 Relationship to Oracle Financial Services Analytical Applications

OFSDF is 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.

- 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 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

2.4 OFSDF Prerequisite Components and Tools

The key prerequisites for running OFSDF are listed as follows:

Table 2-2 Key prerequisites to run OFSDF

Component	Provider	Purpose
Oracle Financial Services Analytical Applications Infrastructure version 8.1.2.0.0	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 19c	Oracle	OFSDF is certified on Oracle Database releases 19c.
erwin Data Modeler application 2019R1 or a higher version	Computer Associates (CA)	erwin is a Data Modeler 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.

3
Other OFSDF Result Tables

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.

3.1 Background

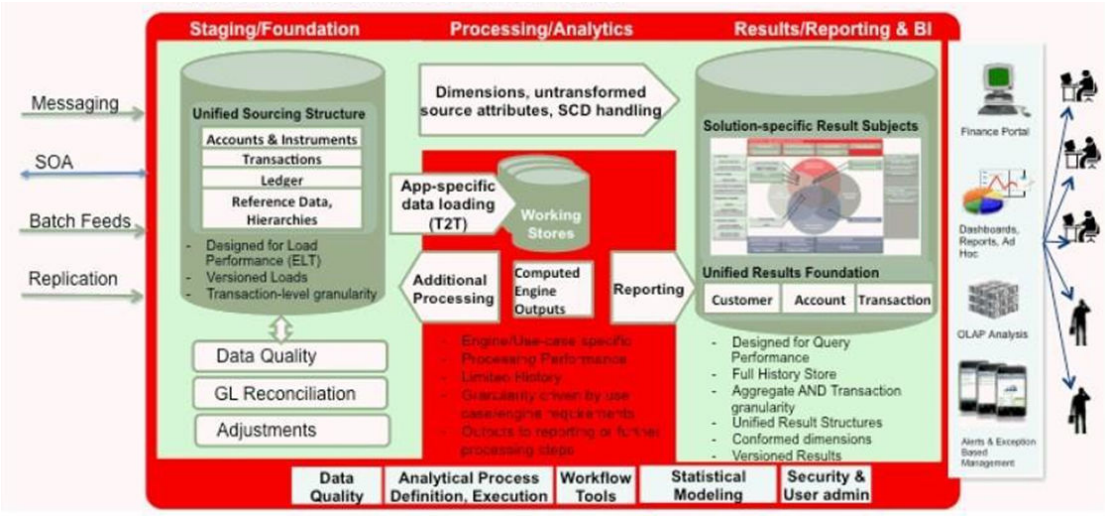
It is important to note that OFSDF architecture differs from 'traditional' data warehouse architecture in key ways.

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.

3.2 OFSDF Architecture

The following diagram depicts the OFSDF functional architecture.

Figure 3-1 Data Foundation Functional Architecture diagram



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.

- 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.

- 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:

- 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.
- 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.
- 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
- 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.

> **Note:**
>
> 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.

3.3 Differences from Traditional Warehouse Architecture

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

Table 3-1 Differences between the traditional Data Warehouse architecture and 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 file system-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. 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.

- 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, seehttps://erwin.com/products/erwin-data-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).

- 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.
- 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
- Reporting Data Model

3.4 Staging Data Model

3.4.1 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:

- 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 3-2 Download Specifications sample

COLUMN DOMAIN	TABLE NAME	COLUMN NAME	COLUMN DESCRIPTION	COLUMN DATATYPE	LRM	FTP	PFT	BASEL_II_USA_SEC	CIRCA	MRMM	ICAAP
Code_Alphanumeric_Short_T	Stg_Accounting_Head	v_account_head_identifie	A unique identifier for a Income Accou	VARCHAR2(6)	DL-NO	DL-NO	DL-NO	DL-NO	DL-NO	NO	DL-NO
DATE	Stg_Accounting_Head	fic_mis_date	This column stores the date as on which the snapshot of source data extracted for processing.	DATE	DL-NO	DL-NO	DL-NO	DL-NO	DL-NO	NO	DL-NO
Code_Alphanumeric_Long	Stg_Accounting_Head	v_branch_code	The code of the branch to which the ac	VARCHAR2(20)	DL-NO	DL-NO	DL-NO	DL-NO	DL-NO	NO	DL-NO
Code_Alphanumeric_Long	Stg_Accounting_Head	v_leg_rep_code	Stores the code of the Booking Transit which is the lowest level in the Legal Reporting Hierarchy.	VARCHAR2(20)	DL-NO	DL-NO	DL-NO	DL-NO	DL-NO	NO	DL-NO
Code_Alphanumeric_Long	Stg_Accounting_Head	v_lob_code	Line of business (LOB) is a functional u	VARCHAR2(20)	DL-NO	DL-NO	DL-NO	DL-NO	DL-NO	NO	DL-NO
Amount	Stg_Accounting_Head	n_account_head_amt	The amount corresponding to this Inco	NUMBER(22,3)	DL-NO	DL-NO	DL-NO	DL-NO	DL-NO	NO	DL-NO
Code_Alphanumeric_Long	Stg_Accounting_Head	v_gl_code	This column stores the code of the GL account that stores the customer accounts balance. Dim_GL_Account is the equivalent dimension table for this column.	VARCHAR2(20)	DL-NO	DL-NO	DL-NO	DL-NO	DL-NO	NO	DL-NO
DATE	Stg_Accounting_Head	fic_mis_date	This column stores the date as on which the snapshot of source data extracted for processing.	DATE	DL-NO	DL-NO	DL-NO	DL-NO	DL-NO	NO	DL-NO
Text_Medium_Description	Stg_Accounting_Head	v_gl_account_head_descr	The description of general ledger acco	VARCHAR2(60)	DL-NO	DL-NO	DL-NO	DL-NO	DL-NO	NO	DL-NO
Code_Alphanumeric_Short_T	Stg_Accounting_Head	v_gl_account_head_type	Identifies the GL head type	VARCHAR2(6)	DL-NO	DL-NO	DL-NO	DL-NO	DL-NO	NO	DL-NO
Code_Alphanumeric_Long_Ty	Stg_Account_Cash_Flo	v_account_number	This column stores the unique identifi	VARCHAR2(50)	DL-NO	DL-NO	DL-NO	DL-NO	DL-NO	NO	DL-NO
DATE	Stg_Account_Cash_Flo	fic_mis_date	This column stores the date as on which the snapshot of source data extracted for processing.	DATE	DL-NO	DL-NO	DL-NO	DL-NO	DL-NO	NO	DL-NO
Number_Short	Stg_Account_Cash_Flo	n_cash_flow_sequence	Numeric value indicating the location	NUMBER(5)	DL-NO	DL-NO	DL-NO	DL-NO	DL-NO	NO	DL-NO

- 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.

- 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)

> **Note:**
>
> 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.

3.4.2 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:

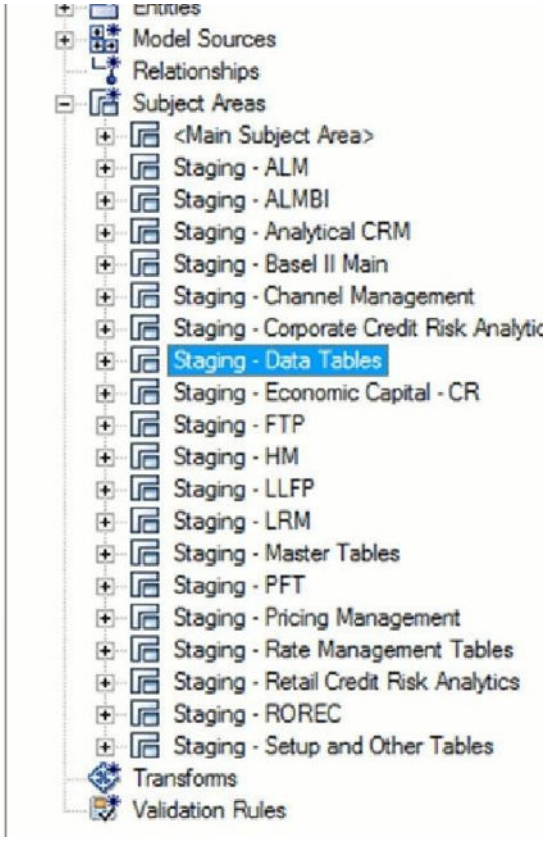
- Business Data: This set of tables captures the actual business events and the resulting state of a bank from those business events. The OLTP systems (or Transactional Systems) capture this information resulting from the execution of the bank's different business processes. Broadly, this information 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.
- 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 given transaction 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.

3.4.2.1 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 Modeler application, and click on Subject Areas as shown below. Right click and switch to the 'Staging-Data Tables' subject area.

Figure 3-3 OFSDF Staging Area Data Tables



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:

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

Figure 3-4 The key groups of business data tables in the data model

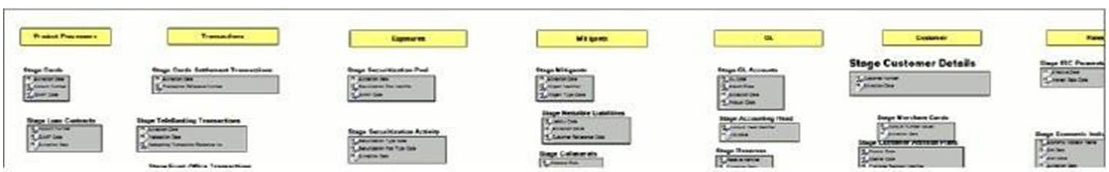


Table 3-2 Key Groups of Business Data Tables and their purpose in the Staging Data Mode

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 con- tract 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.

3.4.2.2 Product Processors

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

Table 3-3 Classification of the Product Processors

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

Table 3-3 (Cont.) Classification of the Product Processors

Product Category	Entity Name
Derivatives	Stage Credit Derivatives
	Stage Futures Contract
	Stage Option Contracts
	Stage Swaps Contracts
	Stage Forwards Contract

For detailed mapping information, see the Banking Product to Product Processor Mappingdocument 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.

3.4.2.3 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 Modeler application, and open up the subject areas menu on the left hand side, as shown below.

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

Figure 3-5 OFSDF Staging Area 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.

3.5 Reporting Data Model

3.6 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:

- 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.

- 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.

- 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.

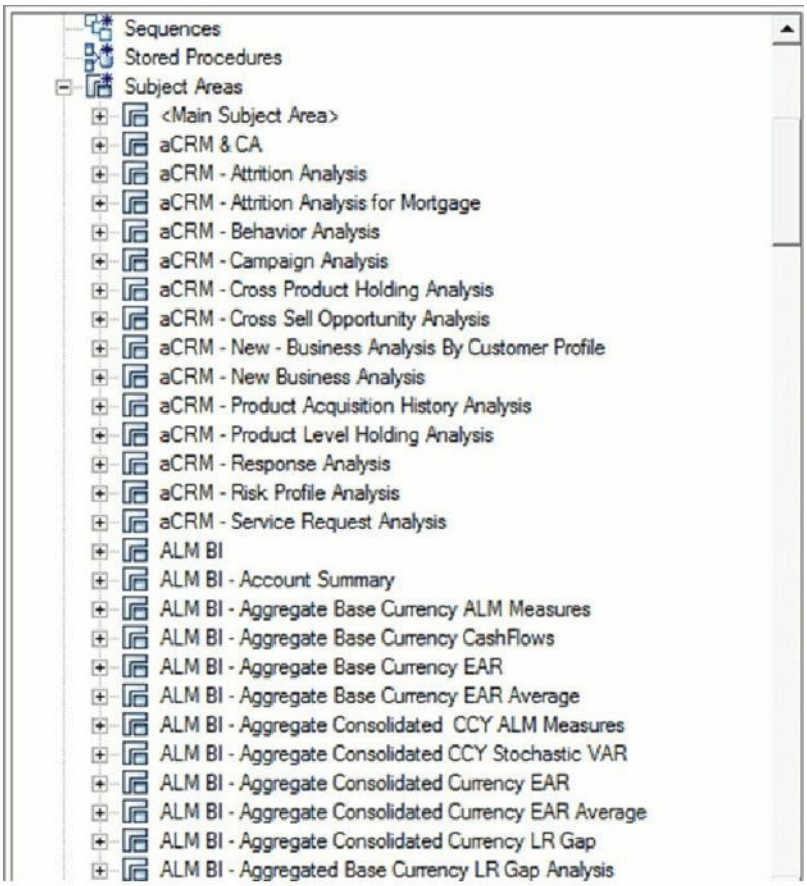
Increasingly, the trend is for cross-functional analytics and reporting. Majority of emerging needs relate to the analytical problems at the intersection of the distinct areas of Risk, Performance, Customer Insight, and Compliance.

3.7 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 Modeler application as show below.

Figure 3-6 Reporting Data Model 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 3-4 Reporting solutions supported by OFSDF at the detailed level

Name	Definition
ALM	Subject Areas corresponding to Asset Liability Management (ALM).

Table 3-4 (Cont.) Reporting solutions supported by OFSDF at the detailed level

Name	Definition
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.

The list of solutions supported by OFSDF is as follows.

Table 3-5 Table 7: Solutions supported by OFSDF at the higher-level

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 Oracle Financial Services Analytical Applications (OFSAA) Data Model Document Generation Release 8.1.x, which details how to extract the data dictionary from erwin section.

4
Catalog Viewer

A Business Term is a functional keyword that represents a unique functional aspect of the Financial Artefacts (Entities and Attributes). There is a business meaning to each Financial Artefact, and therefore, a business name is created in the form of a Business Term. Then the Business Term is used to find and fit into a purpose of different contexts. Therefore, Business Terms are expressions of the Participants/Actors, their Activities, and requirements conveyed in common business.

For each Business Term, there can be one or more usage contexts. The Glossary consists of the information such as description, usage, sample values, and usage examples of the Business Terms.

Catalog Viewer enables the users to search Business Terms in various Glossaries through the User Interface. You can use the Glossary dropdown to select a Glossary and view only specific Business Terms that belongs to that Glossary. It also has a global search feature which can be used to search Business Terms, Business Terms Description and External Mapping details. The Business Terms are also segregated into different usage categories and subject areas. The user interface has a detailed pop-up page for displaying these attributes, mappings and usage.

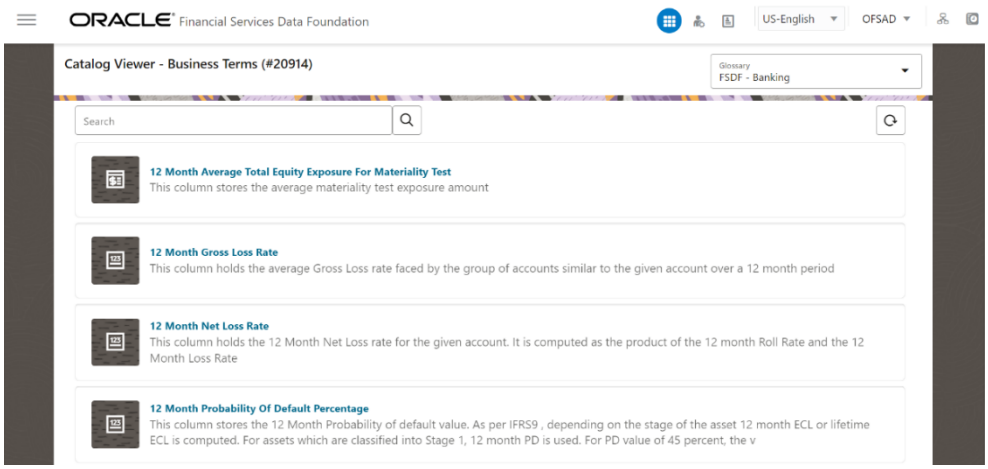
To View the Glossary in FSDF Application

The Catalog Viewer User Interface provides the list of Glossaries available in system. This list appears as a drop down on the top right corner of the page. The Business Terms will be loaded for the default Glossary selected on load of the page.

This section provides the procedure to view the Glossary details of BIAN/FSDF such as Business Terms, Business Term Identifier and so on in the OFSDF Application.

1. After logging into the OFSAAI Applications page, navigate to **Application for** Financial Services Data Foundation, select Financial Services Data Foundation, and then select Catalog Viewer.

Figure 4-1 Catalog Viewer- Business Terms Summary Page



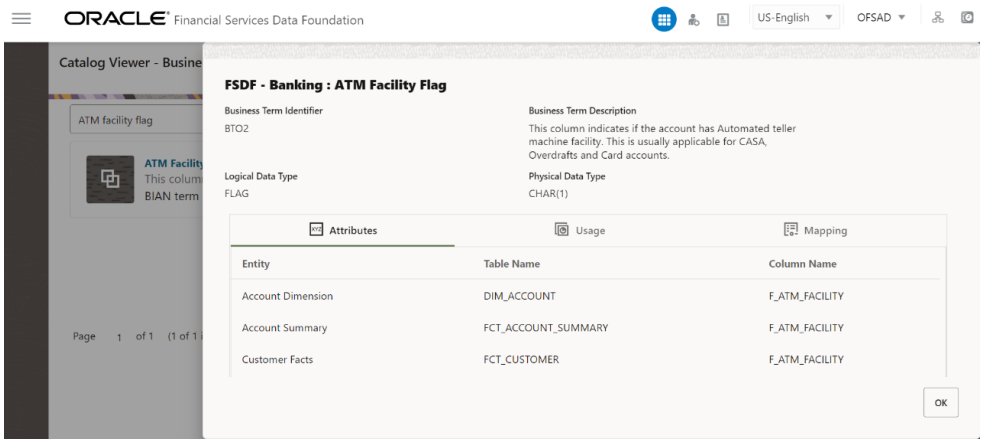
2. Select the **Glossary** from the dropdown list and in the **Search Pane**, enter the Business Term or the Business Term Identifier that you wish to view. Click the **Search** button or the **Enter** key to view the search results.

> **Note:**
>
> You must enter alphanumeric characters without special characters ? _ % . \ / during the search or you can perform a partial search for the corresponding search results to be displayed.
> You can use the Reload button to reset the search and reload the page again.
>
> You can use the Per Page textbox to set the number of records to display as search result per page. Enter any number between 1 and 100 in the per page textbox to view the corresponding records in the search result page.

The **Load Next** Items button displays the next 100 records available and the **Load Previous Items** button displays previous 100 records available in the search. Pagination objects such as page number links, next, previous, first and last links can be used for navigating from one page to another within the 100 records list.

3. Search for **ATM Facility Flag** using the search pane.Click on the **ATM Facility Flag** link and the Business Term details associated with the Glossary is displayed.

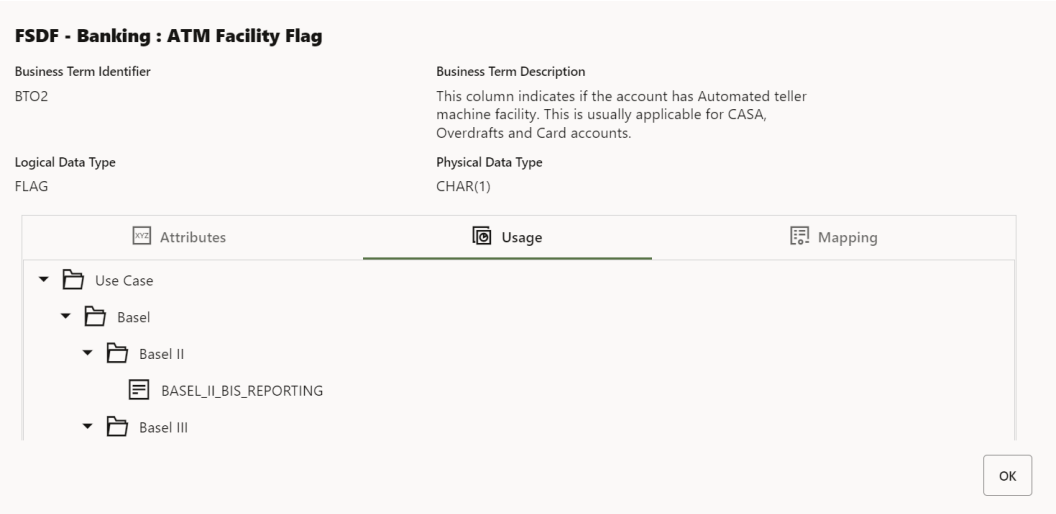
Figure 4-2 FSDF Banking ATM Facility Flag - Attribute page



The Business Terms details such as **Business Term Identifier**, **Business Term Description**, **Logical Data Type** and **Physical Data Type** are displayed. Click on Attributes tab to view the data model objects related to each Business Term. The attributes listed are generally available in the data model of the system such as **Entity**, **Table Name** and the **Column name** associated with the Business Term.

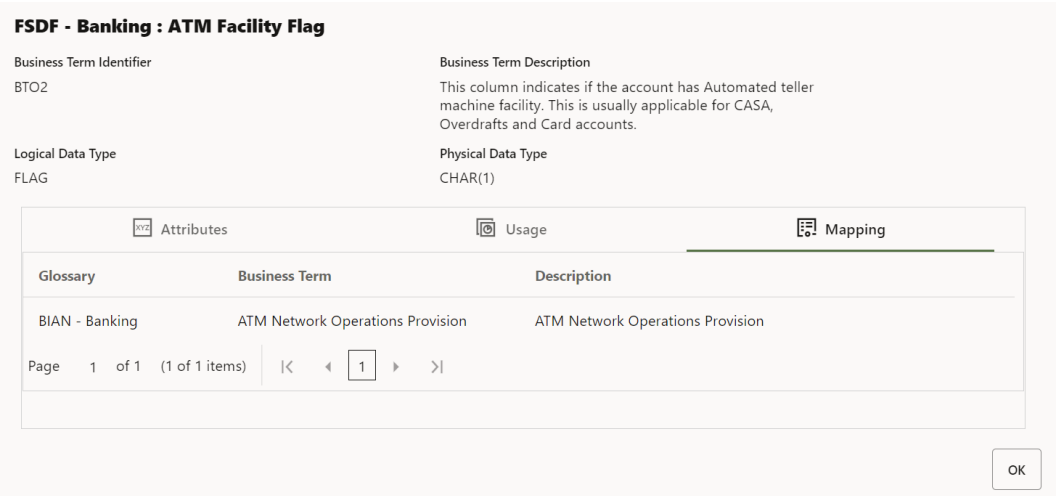
4. Click **Usage** tab to view the Subject Area related to the Business Term. It contains a tree structure of the subject area and its hierarchy.

Figure 4-3 FSDF Banking ATM Facility Flag - Usage page



5. Click **Mapping** tab to view the relationship of the Business Term with other Glossary Business Terms with details such as **Glossary**, **Business Term** and **Description**. The mappings listed is the mapping details between selected Glossary and other available Glossary.

Figure 4-4 FSDF Banking ATM Facility Flag - Mapping page



5
Dimension Loading Process

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

5.1 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.

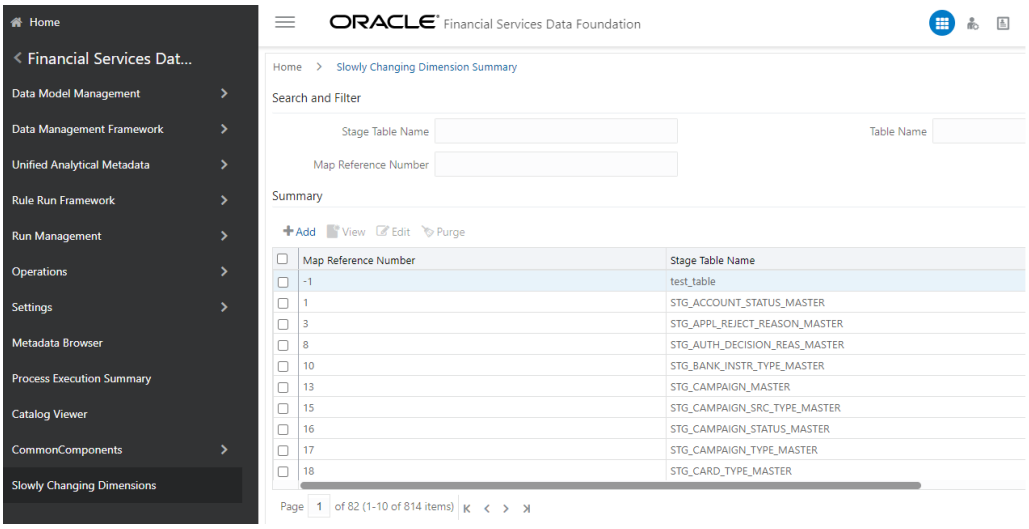
5.2 Overview of Seeded Dimensions

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.

5.3 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.

Figure 5-1 Slowly Changing Dimensions



For more information, see Slowly Changing Dimensions (SCD) section on OFS Analytical Applications Infrastructure User Guide.

5.3.1 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 5-1 Type 1 SCD methodology example after the execution during a processing period

N_PRODUCT_SKEY	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 the following values represented by an example.

Record Change Example:

Table 5-2 Type 1 SCD methodology example after the execution and record change during a new processing period

N_PRODUCT_SKEY	V_PRODUCT_NAME	D_START_DATE	D_END_DATE	F_LATEST_RECORD_INDICATOR
1	Personal Loan	6/30/2010	12/31/9999	Y

5.3.2 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 5-3 Type 2 SCD methodology example after the execution during a processing period

N_PRODUCT_SKEY	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'.

5.3.3 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-bestpractices-datawarehouse-whi-129686.pdf

- Oracle® Warehouse Builder Data Modeling, ETL, and Data Quality Guide at

  http://docs.oracle.com/cd/E11882_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

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 Technical Metadata file.

5.4 Prerequisites

- The SCD executable should be present under <installation home>ficdb/bin. The file name is scd.
- The user executing the SCD component should have execute rights on the file mentioned as prerequisite in point 2.
- 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.

5.5 Tables Used by the SCD Component

The database tables used by the SCD component are:

- SYS_TBL_MASTER
- SYS_STG_JOIN_MASTER

5.5.1 SYS_TBL_MASTER

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

Table 5-4 Rows populated for the seeded dimensions in the SYS_TBL_MASTER table

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.

Table 5-4 (Cont.) Rows populated for the seeded dimensions in the SYS_TBL_MASTER table

Column Name	Data Type	Column Description
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 5-5 Sample data in the rows populated for the Line of Business dimension in the SYS_TBL_MASTER table

Parameter	Value
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.

5.5.2 SYS_STG_JOIN_MASTER

The solution installer will populate this table for the seeded dimensions. Table for Seeded Dimension:

Table 5-6 Rows populated for the seeded dimensions in the SYS_STG_JOIN_MASTER table

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:

- PK – Primary Dimension Value (may be multiple for a given ?Mapping Reference Number")
- SK – Surrogate Key
- DA – Dimensional Attribute (may be multiple for a given ?Mapping Reference Number")
- SD – Start Date
- ED – End Date
- LRI – Latest Record Indicator (Current Flag)
- CSK – Current Surrogate Key
- PSK – Previous Surrogate Key
- SS – Source Key
- LUD – Last Updated Date / Time
- LUB – Last Updated By
- NN – Not Null

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

Table 5-7 Sample data in the rows populated for the Line of Business dimension in the SYS_STG_JOIN_MASTER table

Parameter	Value
MAP_REF_NUM	6
COL_NM	V_LOB_CODE
COL_TYP	PK
STG_COL_NM	V_LOB_CODE
SCD_TYP_ID	
PRTY_LOOKUP_REQD_FLG	N
COL_DATATYPE	VARCHAR
COL_FORMAT	61

> **Note:**
>
> For any new dimension added, the column details will have to be inserted to this table manually.

DIM_<dimensionname>_V – The database view which SCD uses as the source. Example: Dim_Bands_V

These views come as part of install for the dimensions seeded with the application.

> **Note:**
>
> For any new dimension added, a view will have to be created similar to DIM_BANDS_V.

DIM_<dimensionname> – Output table to which SCD writes the dimension data. A sequence should be added for every user-defined dimension.

> **Note:**
>
> Example:
> Create sequence SEQ_DIM_<DIM> minvalue 1 maxvalue 999999999999999999999999999 increment by 1.

5.6 Guidelines for Configuring User Defined (Custom) SCDs

All positive numbering series of Map Reference Numbers ranging from 1 to 999 are reserved for OFSAA Out Of Box Application usage. For custom SCDs, you must use negative numbering series of Map Reference Numbers ranging from -2 to -999.

5.7 Executing the SCD Component

(Required) <Enter a short description here.>

To execute the SCD component from Operations module of OFSAAI, create a batch according to the following steps:

> **Note:**
>
> For a more comprehensive coverage of configuration and execution of a batch, see Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0

1. From the Home menu, select Operations, then select Batch Maintenance.
2. Click New Batch (?+? symbol in Batch Name container) and enter the Batch Name and Description.
3. Click Save.
4. Select the Batch you created in the earlier step by clicking the check box in the Batch Name container.
5. Click New Task (?+? symbol in Task Details container).
6. Enter the Task ID and Description.
7. Select Run Executable, from the Component ID list.
8. 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>

   > **Note:**
   >
   > 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.

> **Note:**
>
> 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.

5.8 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 Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.

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.

5.9 Different Methods of Loading or Executing the Dimensions

This section provides information about different methods of loading or executing the Dimension tables.

5.9.1 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.

5.9.2 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.

5.9.3 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 5-8 The Dimension IDs used for the key Dimensions

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_STRUC- TURE	STG_LEGAL_ENTITY_**_INTF

5.9.4 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 User Guide.

5.9.5 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 User Guide.

5.9.6 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.

5.9.7 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 5-9 Merge Hint and Session Enable Statement details

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.

6
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.

6.1 Overview of the 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.

6.2 Overview of the 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 6-1 The out of the box user-specific Dimensions to Standard Dimensions reclassification available in OFSDF

User Specific Dimension	User Specific Dimension	Standard Dimension	Standard Dimension
DIM_BALANCE_CATEGO RY	Balance Category	DIM_STD_BALANCE_CAT EGORY	Standard Balance Category
DIM_CREDIT_LINE_PURP OSE	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
DIM_LOB	Line of Business	DIM_STANDARD_LOB	Standard Line of Business
DIM_MITIGANT_TYPE	Mitigant Type	DIM_STD_MITIGANT_TYP E	Standard Mitigant Type
DIM_PARTY_TYPE	Party Type	DIM_STANDARD_PARTY_ TYPE	Standard Party Type
DIM_PRODUCT	Product	DIM_STANDARD_PRODU CT_TYPE	Standard Product Type

Table 6-1 (Cont.) The out of the box user-specific Dimensions to Standard Dimensions reclassification available in OFSDF

User Specific Dimension	User Specific Dimension	Standard Dimension	Standard Dimension
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_REASO NS	Write Off Reasons	DIM_STD_WRITE_OFF_RE ASONS	Standard Write Off Reasons
DIM_RECOVERY_TYPE	Recovery Type	DIM_STD_RECOVERY_TY PE	Standard Recovery Type

6.3 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.

6.4 Overview of the Mappers for the 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

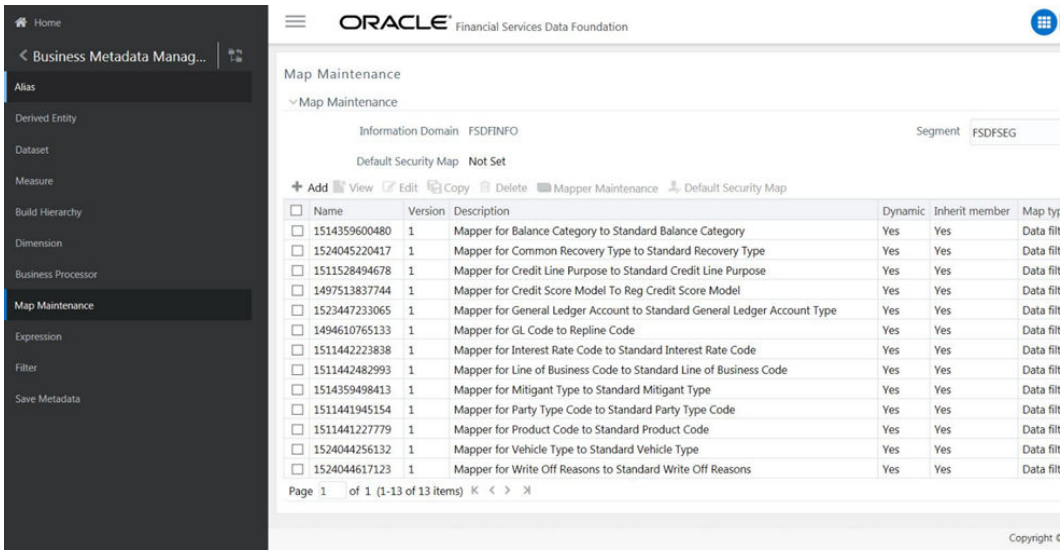
· MAP_WRTOFF_STD_WRTOFF_REASN - Mapper for Write Off Reasons to Standard Write Off Reasons

6.5 Maintenance of the Mapper for the Reclassification of Standard Dimensions

Mapper can be maintained under OFSAAI.

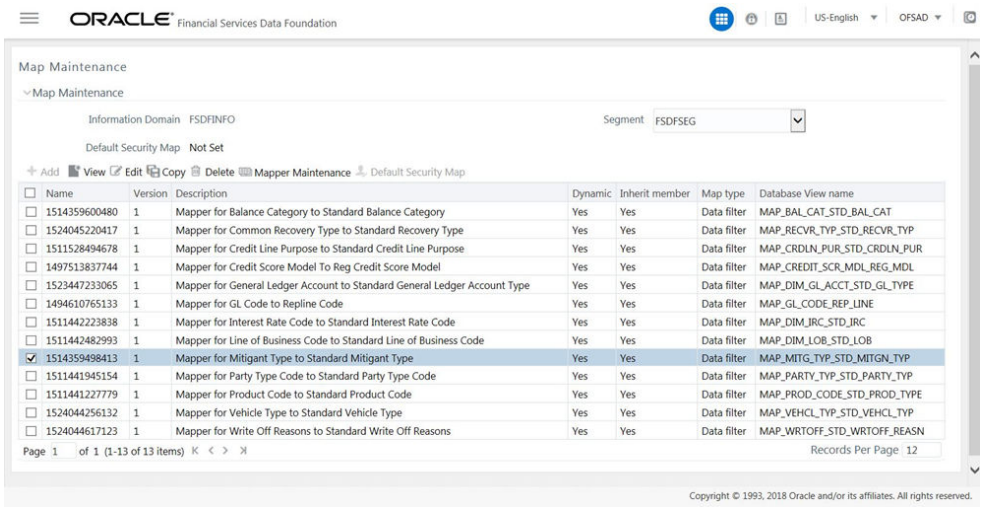
1. Navigate to OFSAAI > Financial Services Data Foundation > Unified Analytical Metadata > Business Metedata Management > Map Maintenance.

Figure 6-1 Navigate to the Map Maintenance page



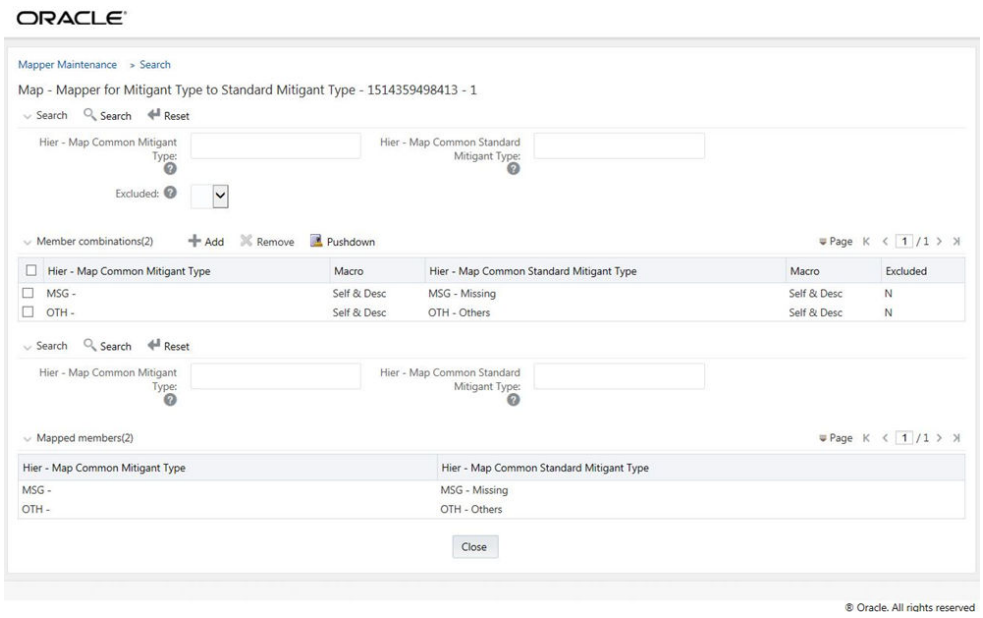
2. For illustration, selected the Mapper for Mitigant Type to Standard Mitigant Type. Click Mapper Maintenance.

Figure 6-2 Select the Mapper Maintenance for a Mapper



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.

Figure 6-3 Mapper Maintenance page



6.5.1 Prerequisites for Mapper Maintenance

Load all the required user specific dimensions using SCD.

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

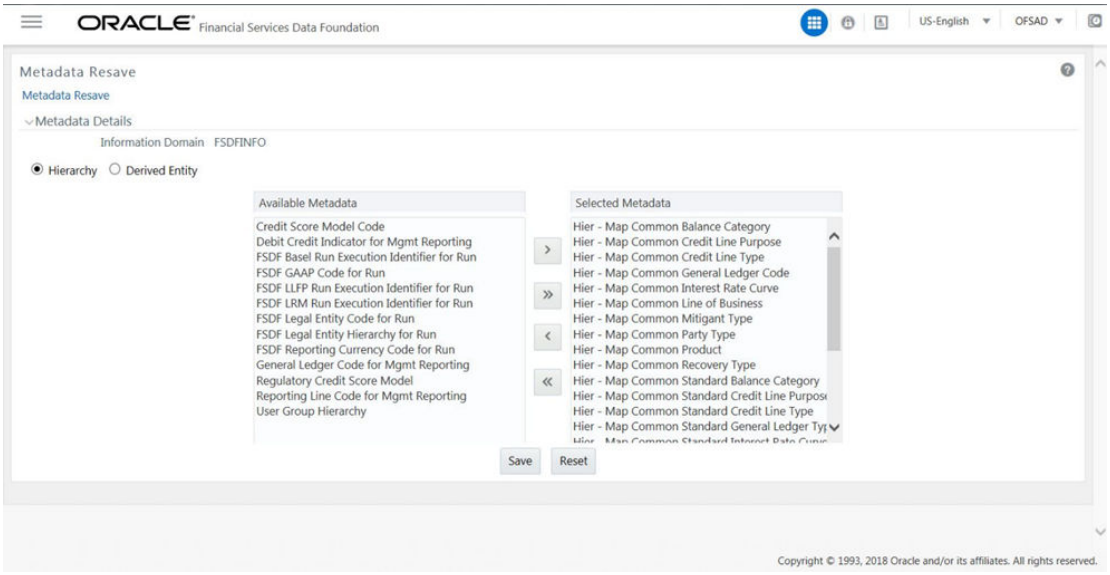
· HCMDF021 - Hier - Map Common Write Off Reasons

· HCMDF022 - Hier - Map Common Standard Write Off Reasons

· HCMDF023 - Hier - Map Common Recovery Type

· HCMDF024 - Hier - Map Common Standard Recovery Type

Figure 6-4 Metadata Resave for Hierarchies



6.5.2 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.

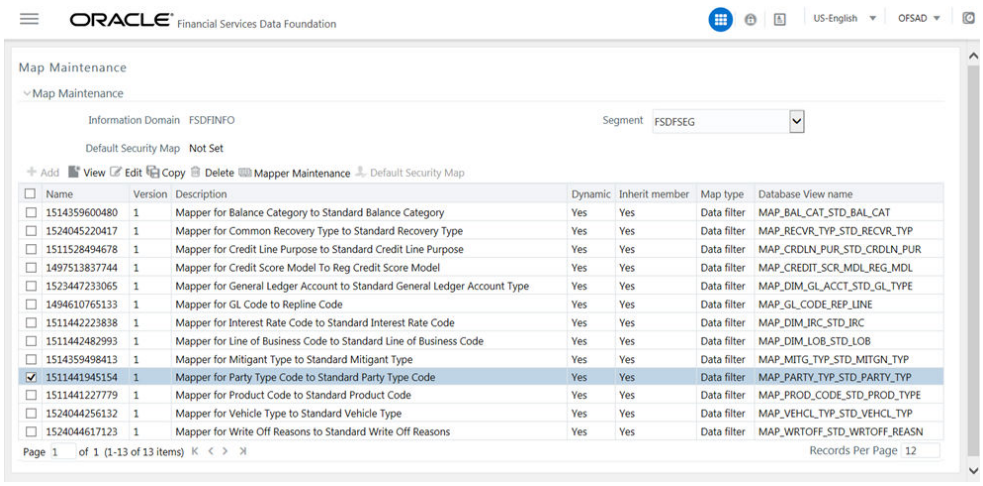
· One to One Mapping: You can map one Mitigant Type data model to one Standard Mitigant Type data model using the Mapper Maintenance screen. Here, you must select one value in Mitigant Type data model and one value in Standard Mitigant Type data model.

· 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:

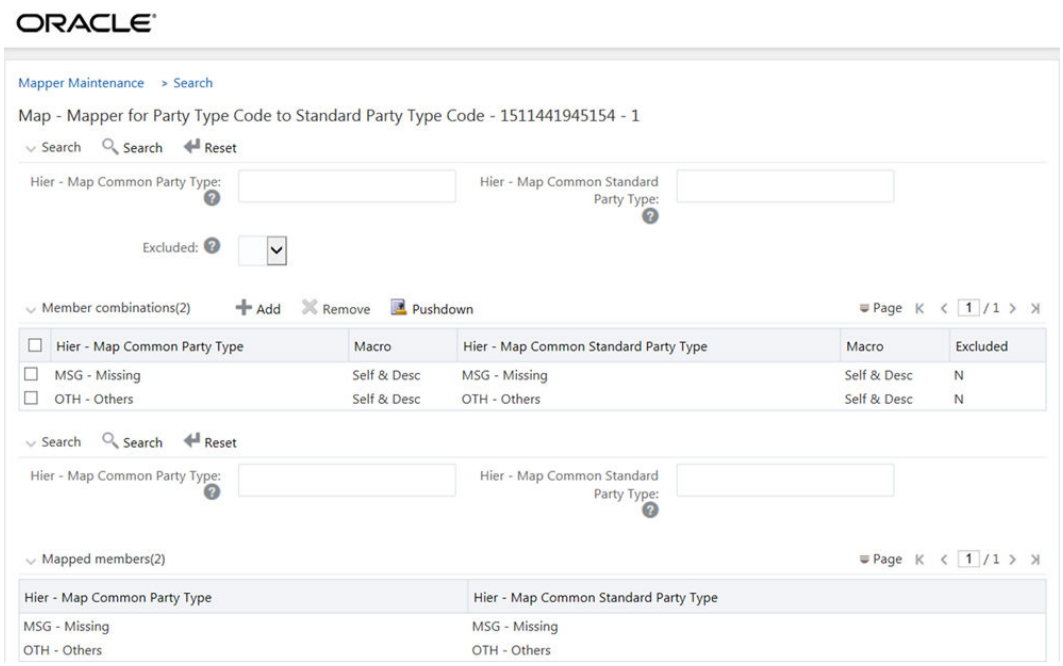
1. Navigate to OFSAAI > Financial Services Data Foundation > Unified Analytical Metadata > Business Metedata Management > Map Maintenance.
2. Click Add to create a new map or select an existing Map. For illustration, an existing Mapper for Party Type Code to Standard Party Type Code value is selected. Click Mapper Maintenance.

Figure 6-5 Select an existing Mapper for One to One or Many to One mapping



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**.

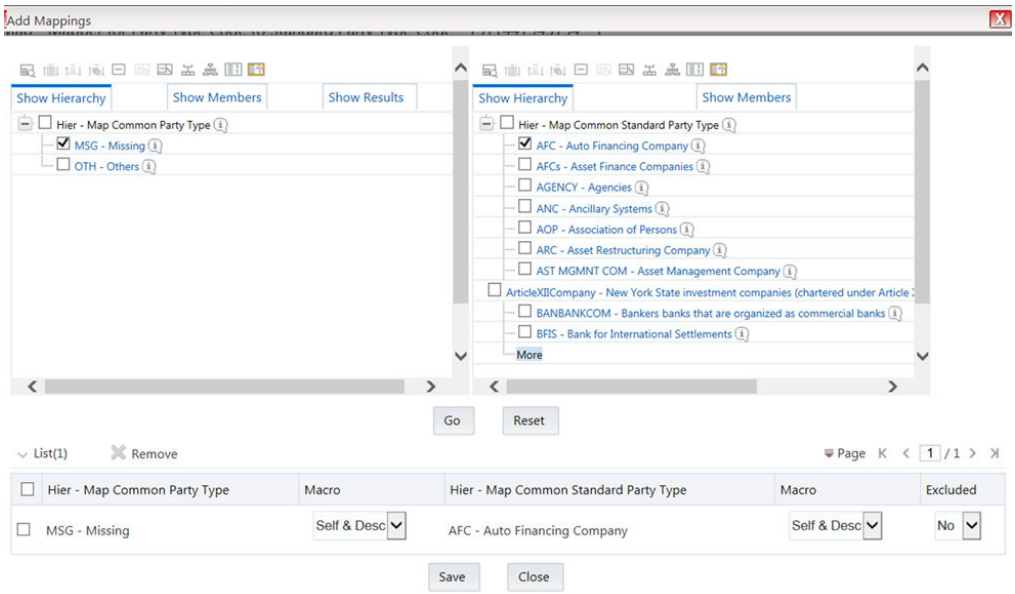
Figure 6-6 Mapper Maintenance page of the selected Mapper



4. The Add Mappings page is displayed. 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.

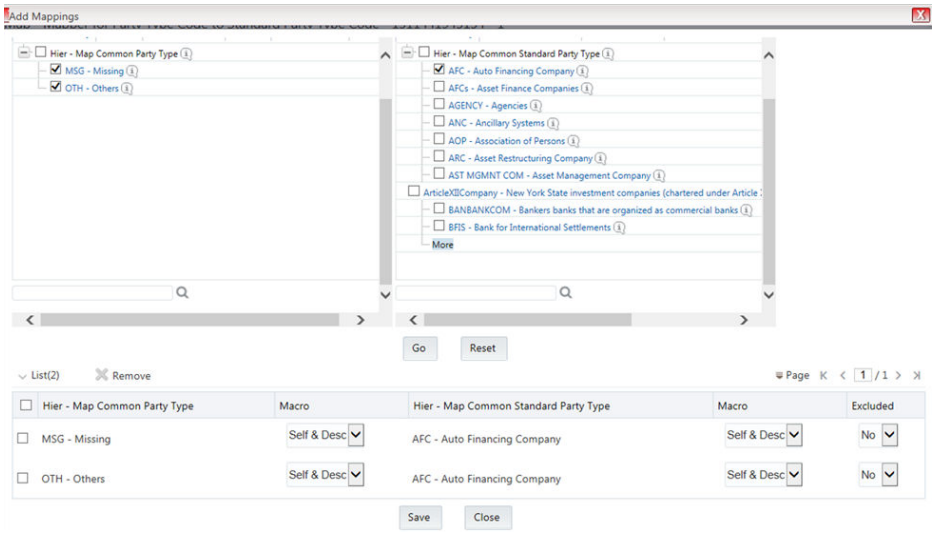
Figure 6-7 Add Mappings page with the one-to-one mapping



- 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.

Figure 6-8 Add Mappings page with the many-to-one mapping



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.

6.6 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.

Table 6-2 The list and details of the Mapper physical tables

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

Table 6-2 (Cont.) The list and details of the Mapper physical tables

Physical Table	V_MAP_ID
MAP_RECVR_TYP_STD_RECVR_TYP	1524045220417
MAP_VEHCL_TYP_STD_VEHCL_TYP	1524044256132
MAP_WRTOFF_STD_WRTOFF_REASN	1524044617123

6.7 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.

7

Legal Entity Consolidation

This chapter provides information about the Legal Entity consolidation in the Oracle Financial Services Data Foundation application.

7.1 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. Process Modelling Framework selects this process by default.

Run Parameters Assignment:

BASEL Accord mentions about different approaches for calculating RWA. Process Modelling 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 Process Modelling Framework, the parameters are populated based on the process defined.

7.2 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.

7.3 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).

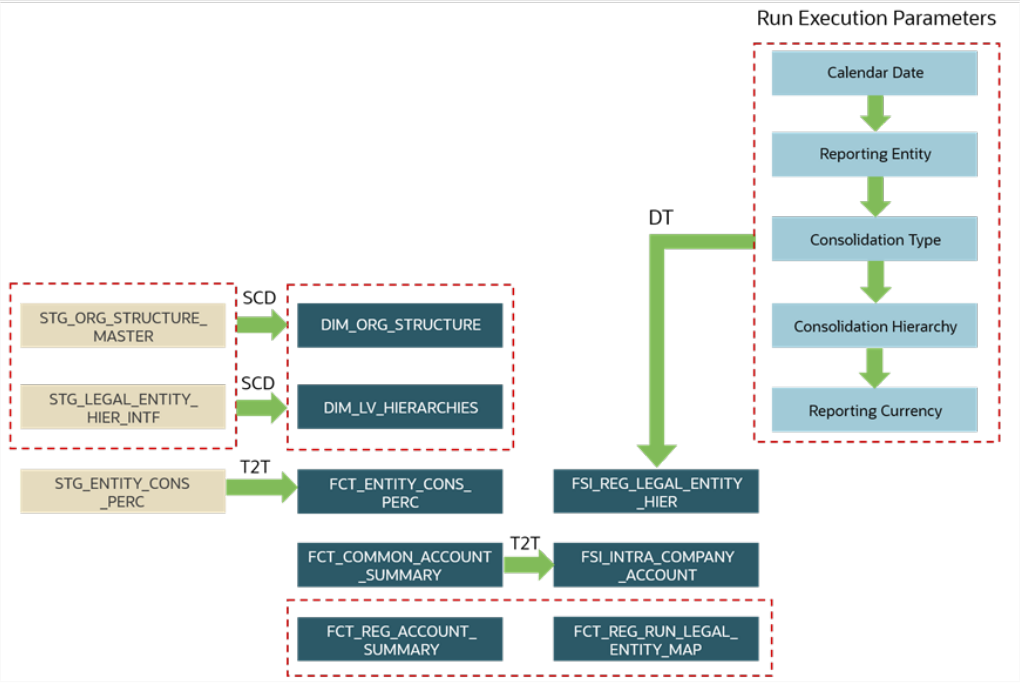
7.4 Consolidation Activities

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.

7.5 Legal Entity Consolidation Data Flow

This section provides data flow information in the Legal Entity Consolidation process.

Figure 7-1 Legal Entity Consolidation Data Flow



8

Executing Run through Process Modelling Framework in OFSDF

Process Modeling Framework (PMF) is a design and execution framework that enables the Process Pipeline developers to implement various Pipelines modeled by the Business Analysts. The Process Pipeline developers use the framework to orchestrate the Business Pipelines and the Run Pipelines within OFSAA and to design the artifacts that participate in the Pipelines to complete their implementation.

This chapter provides information about the usage of the Process Modeling Framework (PMF) feature in the Oracle Financial Services Data Foundation application.

> **Note:**
>
> For detailed information about the Process Modeling Framework (PMF) feature in OFSAA, see the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

8.1 Overview

After OFSDF is installed successfully, the following out-of-the-box standard Run pipelines (PMF processes) are available in the PMF:

- Financial Services Data Foundation Sourced Run: This Run pipeline (Process) loads all non-Run enabled tables in OFSDF.
- Financial Services Data Foundation Execution Run: This Run pipeline (Process) loads all Run enabled tables in OFSDF.

To load all non-Run enabled tables in OFSDF using the out-of-the-box Financial Services Data Foundation Sourced Run process, or to load all Run enabled tables in OFSDF using the out-of-the-box Financial Services Data Foundation Execution Run process, perform the steps in the sections Designing a Run Pipeline and Executing the Run Pipeline.

8.2 Process Modeling Framework Roles

To grant access to the PMF functionality, assign the following PMF Roles to the user.

Table 8-1 The list of PMF Roles and the corresponding users

Role Code	Role Name
WFMWRITE	Manage Workflow Monitor
WFADMINACC	Process Admin User
WFDELACC	Process Delegation User
WFACC	Workflow Access

Table 8-1 (Cont.) The list of PMF Roles and the corresponding users

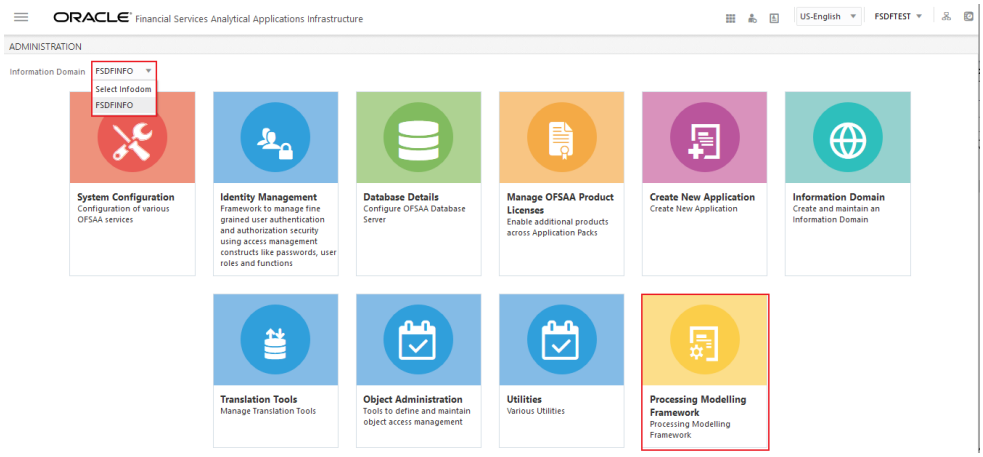
Role Code	Role Name
WFADV	Workflow Advanced
WFAUTH	Workflow Authorize
WFDELGADM	Workflow Delegation Admin
WFMACC	Workflow Monitor Access
WFREAD	Workflow Read
WFWRITE	Workflow Write

8.3 Accessing the Process Modeling Framework

To access the PMF, follow these steps:

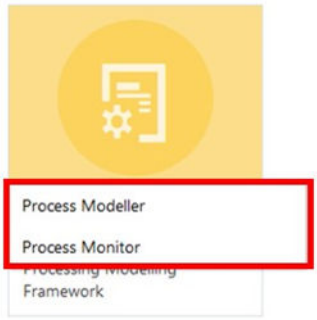
1. Log in to the OFSDF application and select Administrator .
2. In the Information Domain drop-down list, select the required FSDF information domain. Then click the Process Modeling Framework tile.

Figure 8-1 Select the OFSDF Infodom and then select the Process Modelling Framework tile



3. A submenu is displayed with the following menu items:
   - To access the Process Modeller, select Process Modeller.
   - To monitor currently running processes, select Process Monitor.

Figure 8-2 Select the Process Modeller or Process Monitor option in PMF submenu



8.4 Using the Process Modeling Framework in OFSDF

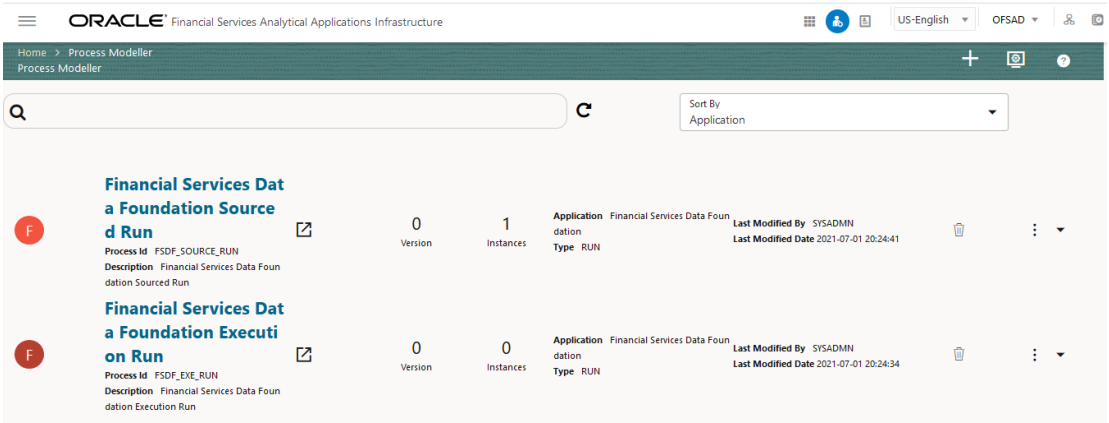
The Process Modeling Framework consists of these features:

- Process Modeller
- Process Monitor

8.4.1 Process Modeller

The Process Modeller is used to create and modify types of Pipelines, test the Process flow, and execute the Run.

Figure 8-3 Process Modeller page



The Process Modeller page displays the existing Business Process Pipelines and Run Pipelines with the details such as Process ID, Process Name, Process Description, Version, Instance, Application, and Last Modified.

You can perform the following tasks from the Process Modeller window:

- Click to create a new Pipeline.

- Click the Process Name link to launch and edit the Process Flow.
- Click to delete a Pipeline.
- Click to view the following submenu:
  - Click **View** to see the process flow.
  - Click **Copy** to create a new Pipeline with the same process flow.
  - Click **Process Flow Monitor** to monitor the Pipeline.
  - Click **Test Process Flow** to test or check whether the Business Pipeline you designed works as expected.
  - Click **Execute Run** to execute a Run Pipeline.
  - Click **Export Process** to export a Process.

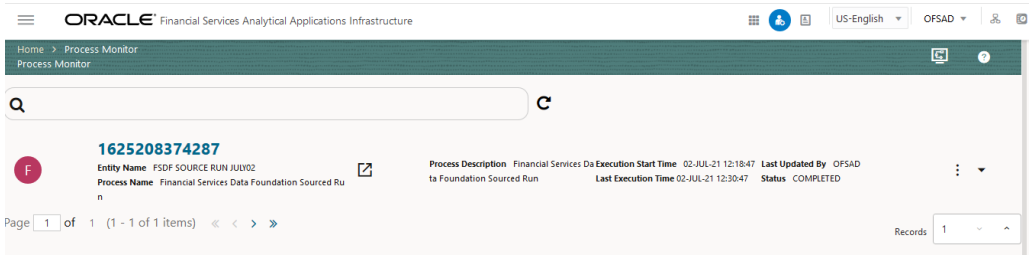
> **Note:**
>
> The export process is performed using the Object Migration feature.

- Use the Search grid to search for a specific Pipeline by providing a keyword from the Process ID, Process Name, or Process Description and then click **Search** icon. Click **Reset** to reset the Search fields.
- You can sort the Pipelines based on Process ID, Process Name, or Application. Click the **Sort by** drop-down and select the required attribute to sort.
- You can use the Filter Pipeline field to filter pipelines based on the pipeline type. For example, to view only the Run Pipelines, remove Process from the Filter Pipeline field.
- Click  to launch Process in a new window.
- Click **Process Monitor** to launch the Process Monitor page.

8.4.2 Process Monitor

The Process Monitor is used to supervise the current stage of the Process for different instances. After integration with an application, the workflow can be invoked. After invoking, the workflow goes through all the stages defined. The Process Monitor displays all the stages finished, the current stage, and future stages. Your user group must be mapped to the function role WFMACC (Workflow Monitor Access) to access the Process Monitor window. For the list of the PMF Roles, see the Process Modeling Framework Roles section.

Figure 8-4 Process Monitor Page



8.5 Designing a Pipeline in OFSDF

Business pipelines are defined in OFSAA to design and execute a sequence of tasks, which are either OFSAA tasks or external tasks, to derive a well-defined outcome. This flow is defined by using various OFSAA artifacts from the component toolbar.

Use the Process Modeler for the following tasks:

- Designing a Run Pipeline
- Executing the Run Pipeline
- Verifying the Execution Logs

> **Note:**
>
> For illustration, a Run Pipeline design for FSDF Sourced Run is depicted in this section.
>
> For information about designing a Business Pipeline, see the Orchestration of a Business Pipeline section in the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

8.5.1 Designing a Run Pipeline

Visual representation of the Run is enabled through PMF by the construction of a Run Pipeline. Various OFSAA widgets that enable the construction of Run Pipeline are available in the Component toolbar. For detailed information about the Components in PMF, see the Components for Designing Your Process Flow section in the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0

8.5.1.1 Creating a new Run Pipeline

To create a new Run Pipeline in the Process Modeller, follow this procedure:

1. Navigate to the Process Modeller page. For details, see the Accessing the Process Modeling Framework in OFSDF section.
2. In the Process Modeller page, click **Add** icon.
3. The Process Details page is displayed. Enter or select the required values for each field.

Figure 8-5 Process Details page

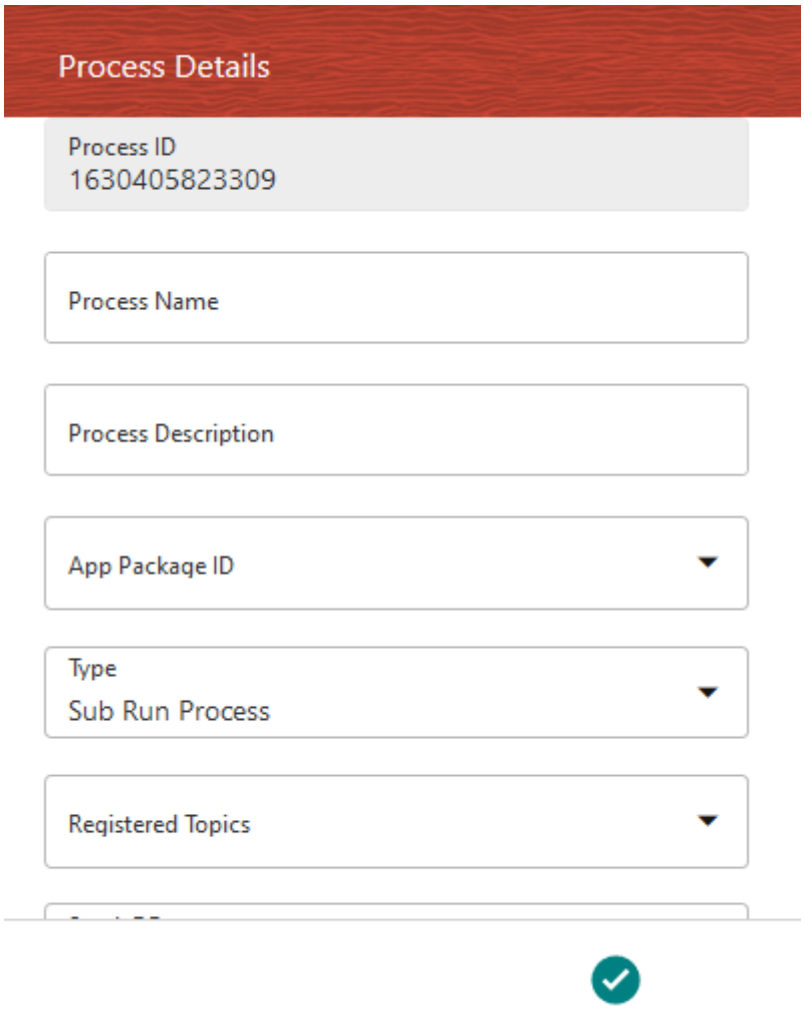


Table 8-2 Process Details page fields and descriptions

Field Name	Description and instruction
Process ID	Enter a unique and easily recognizable value in the Process ID field.
Process Name	Enter a unique and easily recognizable value in the Process Name field.
Process Description	Enter a unique and easily recognizable value in the Process Description field.
App Package ID	Select the required application package in which the process must be created from the drop-down list.
Type: • Business Process Pipeline • Run Pipeline • Stream Pipeline	Select the required process type from the Type drop-down list.

Table 8-2 (Cont.) Process Details page fields and descriptions

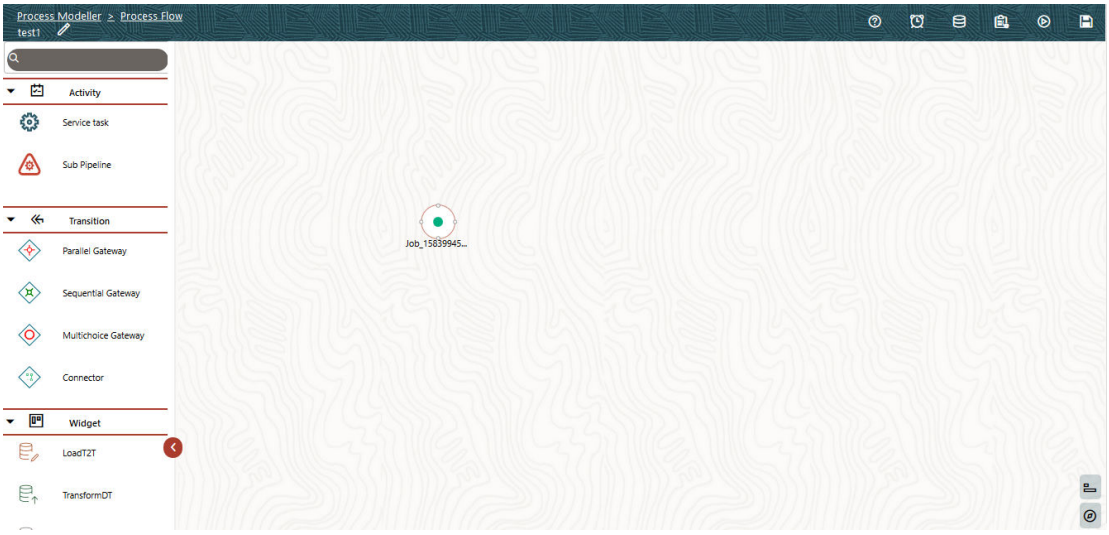
Field Name	Description and instruction
Registered Topics	Note: This functionality is currently not applicable to OFSDF.
Spark DB	Enable the Spark DB option if you are executing the Run for Hive metadata.
Infodom	Select the information domain from the Infodom drop-down list in which the Run Pipeline must be created. The list displays all the infodoms mapped to the applications configured in your OFSAA instance.
Tag	Enter the text that can be used as search keywords or helpful in defining a relationship between two or more Pipelines.

4. To save the details and launch the Process Flow designing window, click **Save & Launch**.

8.5.1.2 Designing a Process Flow for a Run Pipeline

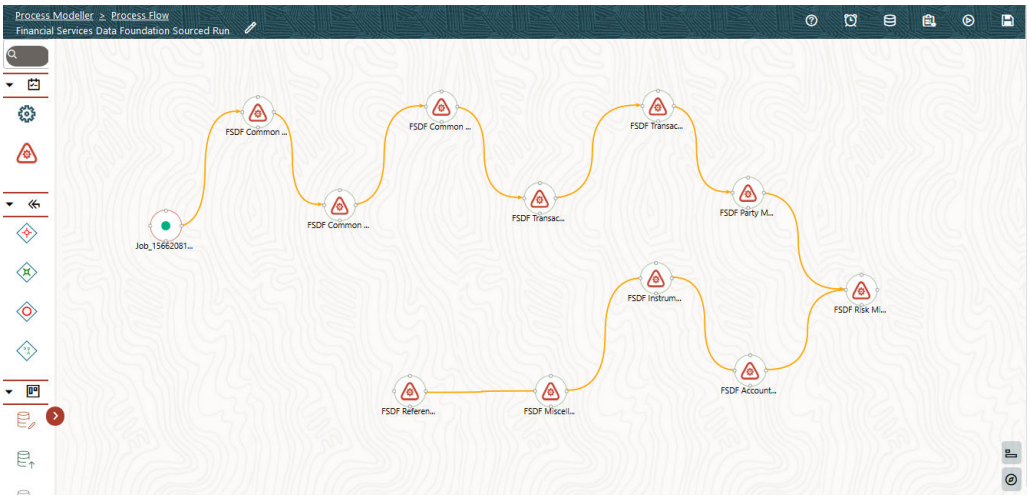
After you click Save & Launch in the Process Details page, the Process Flow page is displayed. The Process Flow tab contains a floating toolbar and a drawing canvas. Use the drawing canvas to design the process flow with the Tools, Activities, and OFSAA Widgets components available in the floating toolbar.

Figure 8-6 Process Flow page



You can design the process flow diagrams for all three types of processes (Business Process Pipeline, Run Pipeline, and Stream Pipeline). This is an example of a process flow diagram for a Run Pipeline (for FSDF Sourced Run).

Figure 8-7 Process Flow diagram for a Run Pipeline



To design a process flow diagram for a Run Pipeline, see the Run Pipeline section, and to use various components available in the Process Flow tab, see the Components for Designing Your Process Flow section in the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

8.5.2 Executing the Run Pipeline

After a process is designed and defined in the process flow diagram, you must assign values to the Run parameters, and execute the Run. You can execute a Run Pipeline from the UI or using a command-line utility called `wfExecExternal.sh`.

This section includes the following topics that describe the Run Pipeline execution from the UI:

- Selecting the Run Parameters and Executing the Run
- Verifying the Run Execution
- Aborting, Resuming, or Rerunning the Run Pipeline Process
- Verifying the Execution Logs

> **Note:**
>
> For information about executing the Run Pipeline using a command-line utility, see the section Using Command Line Utility in the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

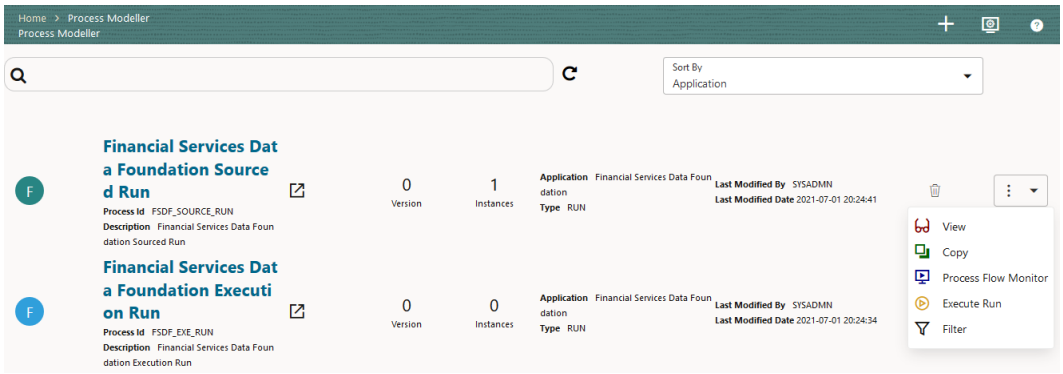
8.5.2.1 Selecting the Run Parameters and Executing the Run

After designing and saving the process flow diagram, the Process is listed in the Process Modeller page.

To select the Run parameters and execute the Run, follow this procedure:

1. In the **Process Modeller** page, click  corresponding to the Run Pipeline that must be executed.

Figure 8-8 Select Execute Run in the Run Pipeline



2. When you click **Execute Run**, the Execution dialog is displayed. Select the Execution Type as **Without Parameters** or **With Parameters**.

Figure 8-9 Execution Without Parameters

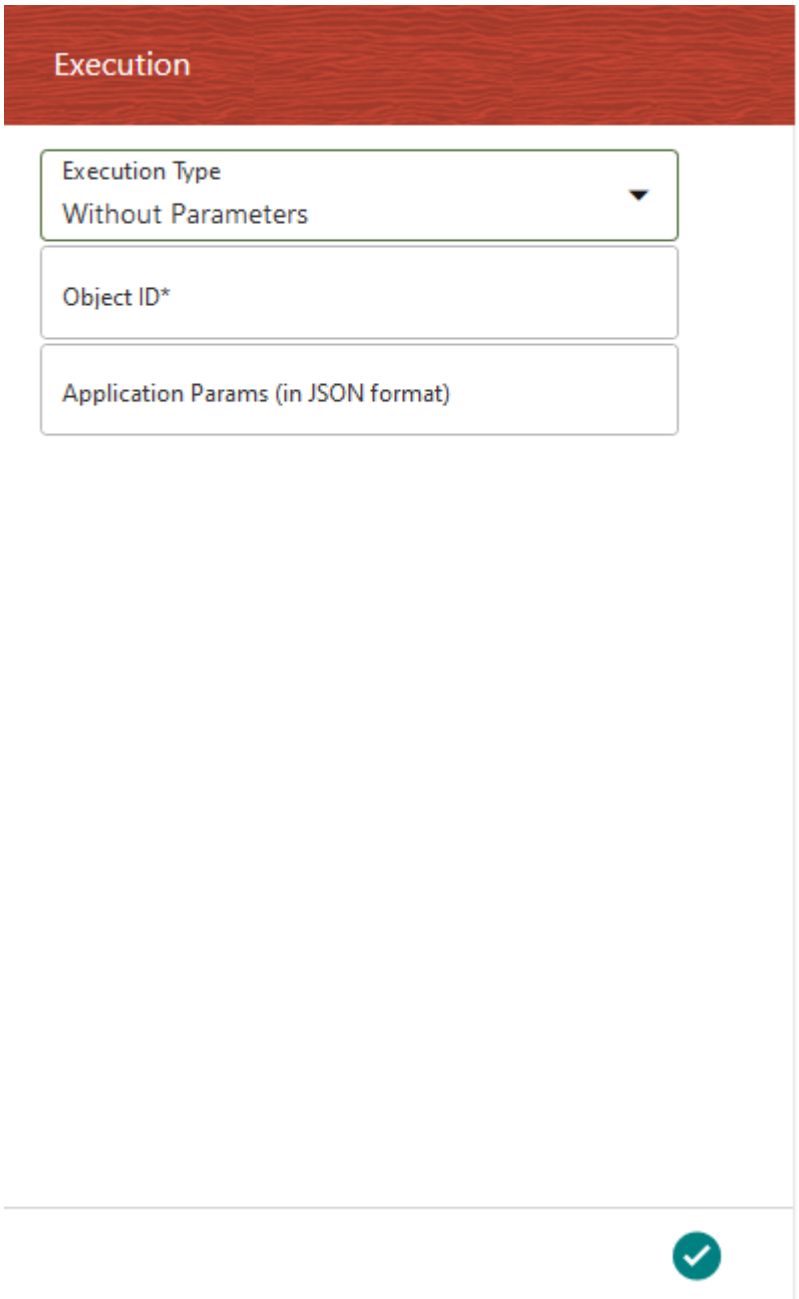


Figure 8-10 Execution With Parameters

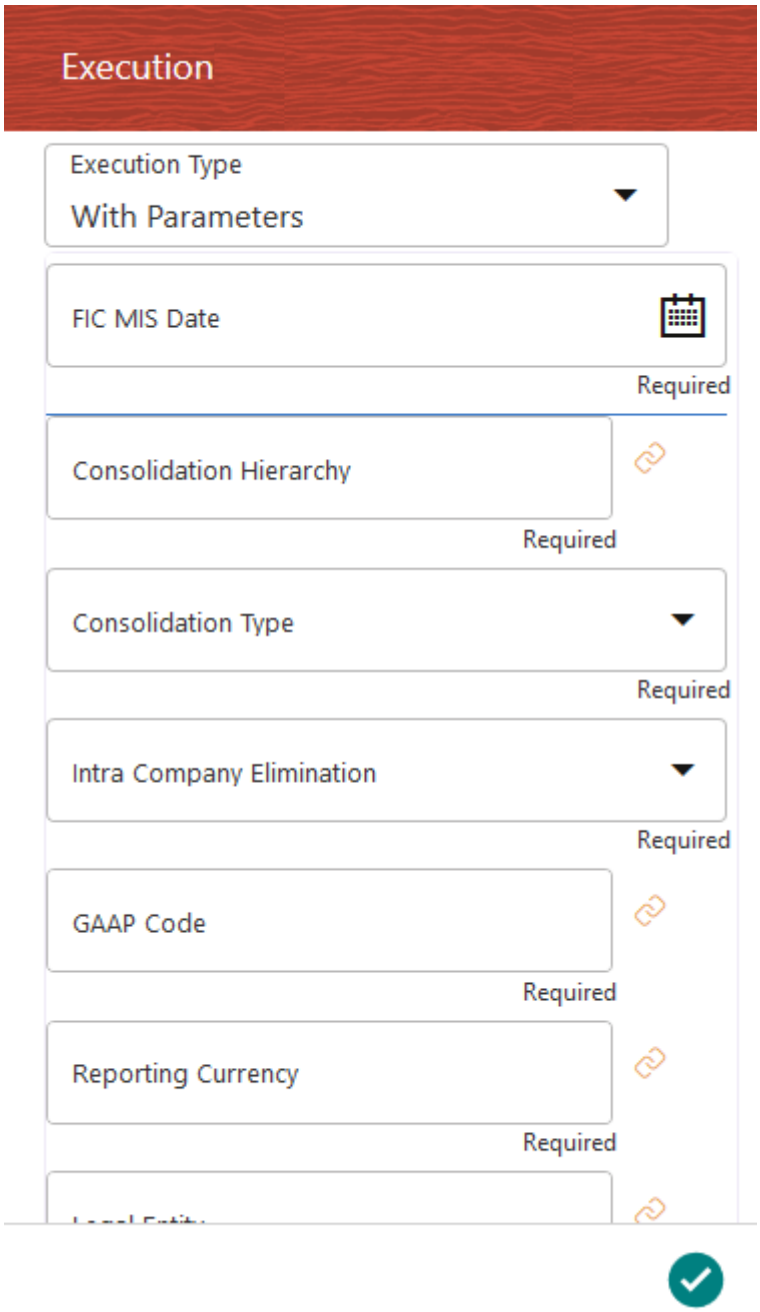


Table 8-3 Execution page fields and descriptions

Field Name	Description or Instruction
Object ID	Enter the Object ID, which is the entity ID used to identify if a workflow needs to be started from the beginning of the current stage.
Application Params (in JSON format)	Enter the Application Run execution parameters in the JSON format.
FIC MIS Date	Select the extraction date.

Table 8-3 (Cont.) Execution page fields and descriptions

Field Name	Description or Instruction
Consolidation Hierarchy	Enter the Legal Entity Hierarchy used for the consolidated run. This parameter is not required for the Solo Run.
Consolidation Type	Select the Consolidation Type of legal entities on a solo or consolidation basis. In a Solo Run, only the selected legal entity is used. In a Consolidated Run, along with the selected legal entity, all its child legal entities are also used.
Intra Company Elimination	Select the Intra Company Elimination type to eliminate (YES) or skip the elimination (NO) of Intra Company Accounts during a Consolidated Run.
GAAP Code	Enter the required accounting standard.
Reporting Currency	Enter the Reporting Currency Code used to calculate the amount during the data population in the target table.
Legal Entity	Select the Legal Entity Code to identify the legal entity used for the Run.
Run Execution Description	Enter a longer description of the Run.

3. When you click **OK**, the Run execution begins. The **Select Run Params** page closes.

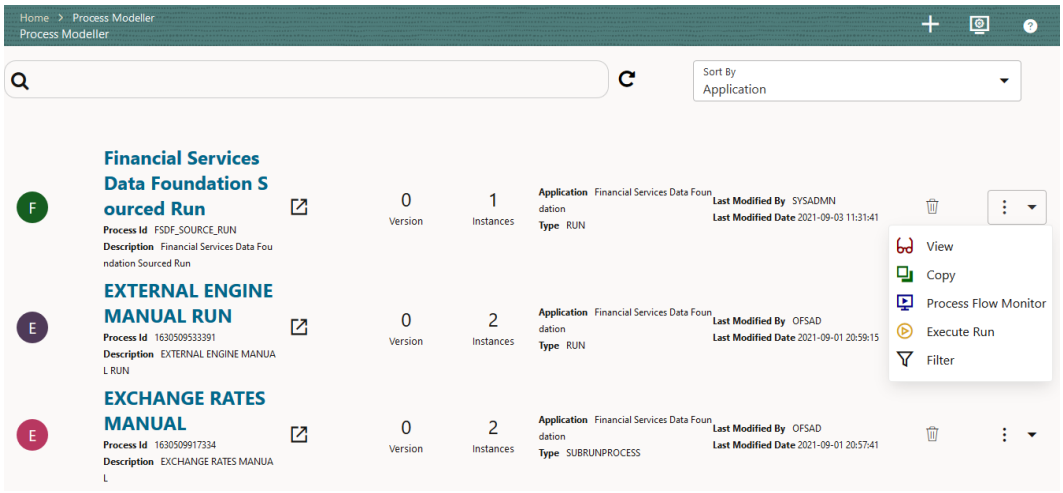
> **Note:**
>
> The execution of the Run Pipeline is triggered using the selected FIC MIS DATE. The Run SKey is generated and inserted into the DIM_RUN table. For the Run SKey generated, the corresponding user-selected Run parameters are inserted into the RUN_EXE_PARAMETERS table.

8.5.2.2 Verifying the Run Execution

After selecting the Run parameters and beginning the Run execution, verify the progress of the Run. To verify the Run execution progress, follow this procedure:

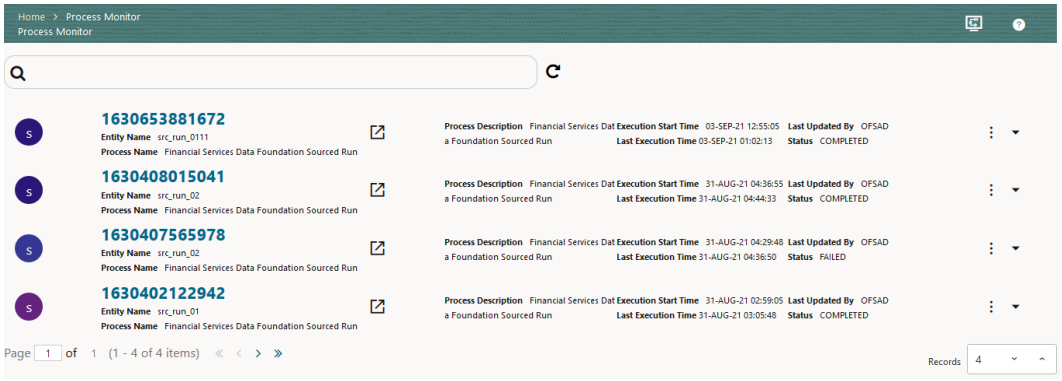
1. In the **Process Modeller** page, click  corresponding to the Run Pipeline that must be verified. Click Process Flow Monitor.

Figure 8-11 Select Process Flow Monitor in the Run Pipeline



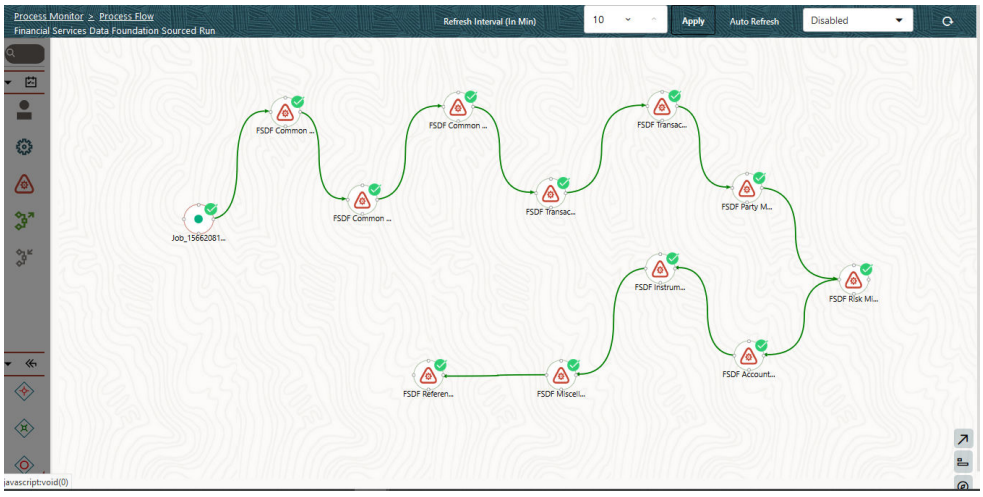
2. The **Process Monitor** page is displayed. You can see the generated process flow ID, the Run execution timestamp, and the status of the Run execution. To verify the Run execution status at the Pipeline level, click the corresponding process flow ID.

Figure 8-12 Process Monitor page with the Process Flow ID



3. The Process Flow diagram page is displayed. The  icon at each Sub Pipeline indicates that the Run execution is successful.

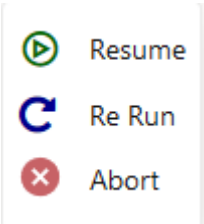
Figure 8-13 Process Flow diagram page with the indication of successful Run execution



8.5.2.3 Aborting, Resuming, or Rerunning the Run Pipeline Process

In the Process Monitor page, click corresponding to the Run Pipeline process that must be aborted, resumed, or rerun. Then select Abort, Resume, or Re-Run.

Figure 8-14 Process Monitor Menu



To abort, resume, or rerun the Run Pipeline Process, see the Abort Run Pipeline, Resume Run Pipeline, and Re-run Run Pipeline sections respectively in the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

8.5.3 Verifying the Execution Logs

You can access the execution logs to verify the details of the Run. To verify the execution logs, see the Viewing Activity Execution Logs and Viewing Execution Log for Widgets sections in the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

8.6 Creating Filter using DIM_DATA_ORIGIN

Filters using DIM_DATA_ORIGIN are not present in T2Ts. Therefore, follow this execution procedure to introduce the filters using DIM_DATA_ORIGIN:

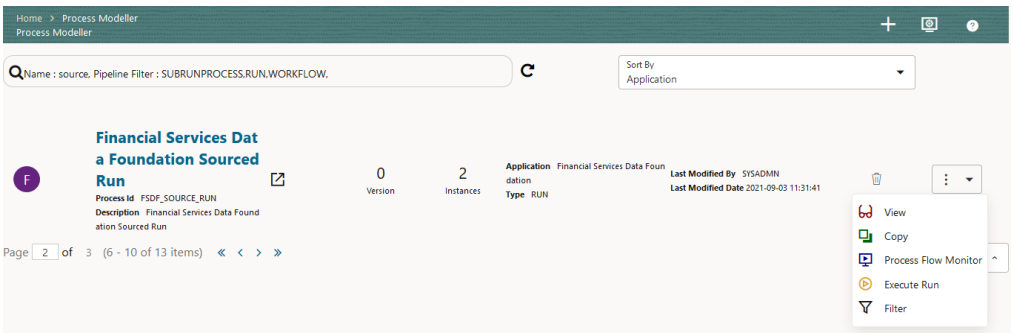
1. Download the post-script file SOR_postscript_update.sql from the My Oracle Support and execute it in the CONFIG schema.

> **Note:**
>
> Oracle recommends executing the post-script after resaving the Hierarchy if any.

2. Follow these steps to create a Data Origin Filter:
Create a DIM_DATA_ORIGIN Hierarchy for the attribute key as N_DATA_SOURCE_CD, and add Level 0 as V_DATA_SOURCE_CODE.

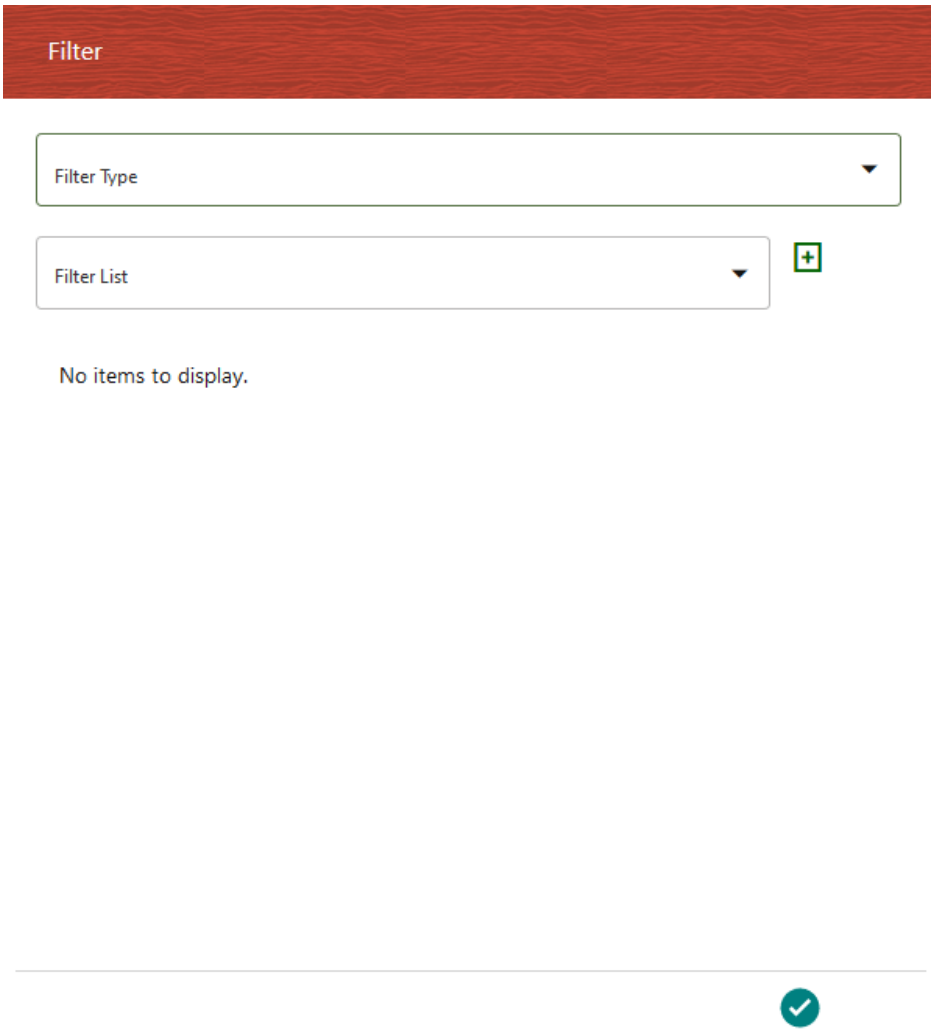
   a. In the **Process Modeller** page, in the **Financial Services Data Foundation Sourced Run** pipeline, click to view the menu, and select **Filter**.

Figure 8-15 Select Filter in the menu of the Financial Services Data Foundation Sourced Run pipeline



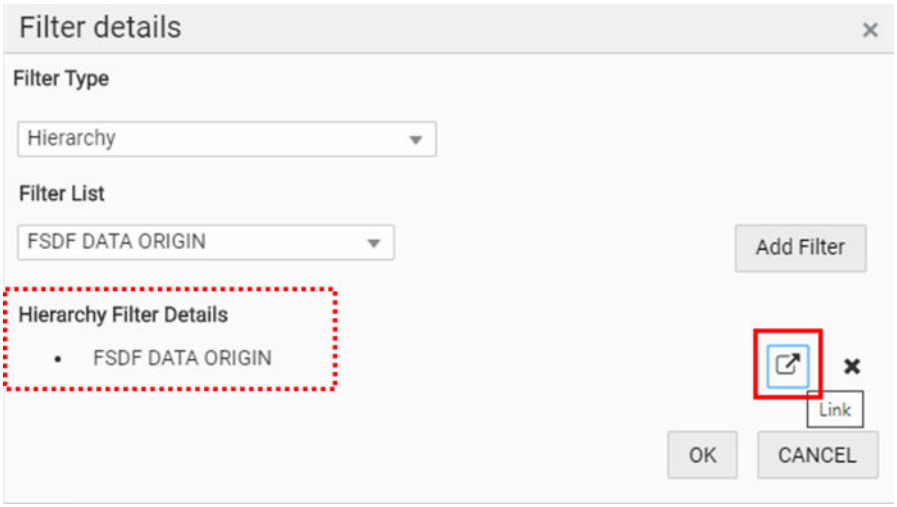
   b. The Filter dialog is displayed. Select the Filter Type as Hierarchy and select the name of the Hierarchy created in the Filter List (in this illustration, it is FSDF DATA ORIGIN). Then click Add icon.

Figure 8-16 Added Filter is listed in the Filter Details section



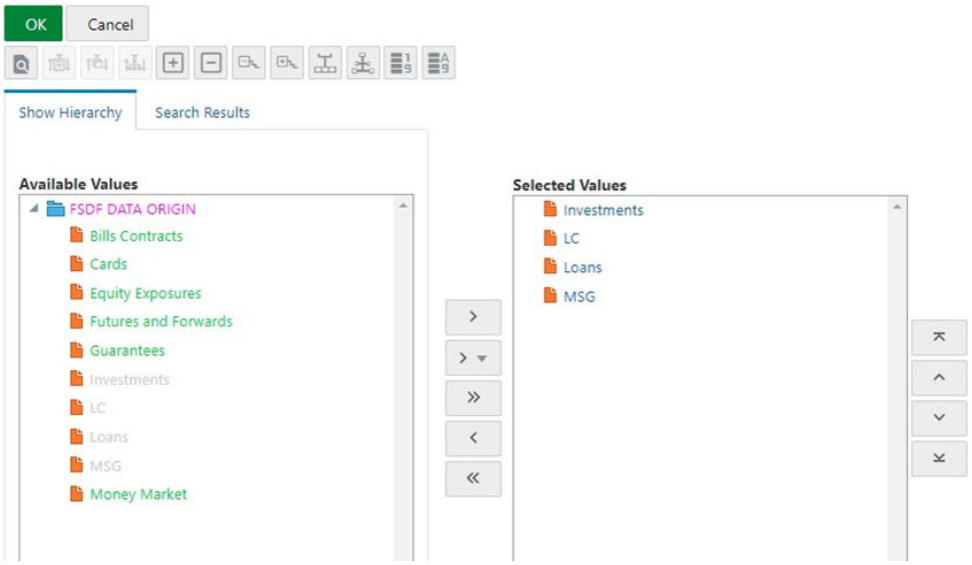
c. This filter is listed in Hierarchy Filter Details. Click the pop-out page.

Figure 8-17 Filter Details page with the Hierarchy Filter Details



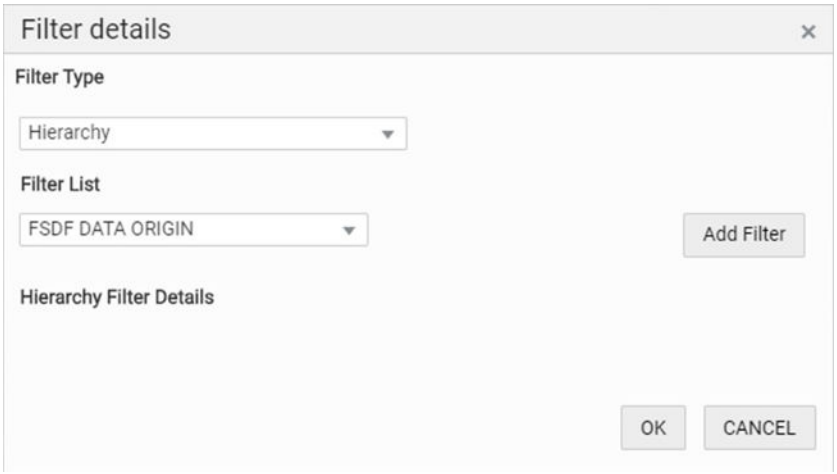
d. The Hierarchy page is displayed with the LOVs (lost of values). In the FSDF DATA ORIGIN folder, select and move all the required LOVs from the Available Values section to the Selected Values section. Then click **OK**.

Figure 8-18 Select the LOVs in the Hierarchy page for the Hierarchy Filter



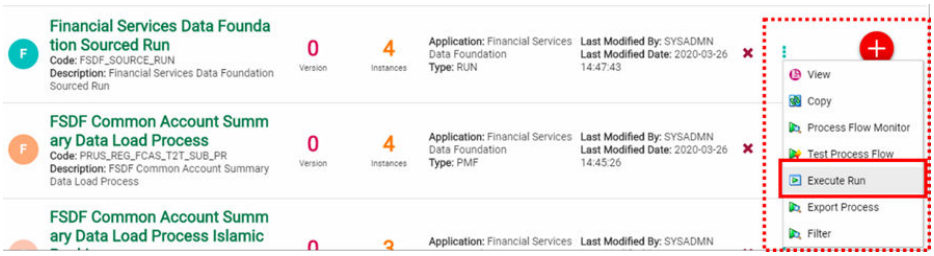
e. On the Filter details page, click **OK** to save the Hierarchy Filter details.

Figure 8-19 Save the Hierarchy Filter details



f. In the **Process Modeller** page, click **Execute Run** to execute the PMF Sourced Run.

Figure 8-20 Execute the PMF Sourced Run



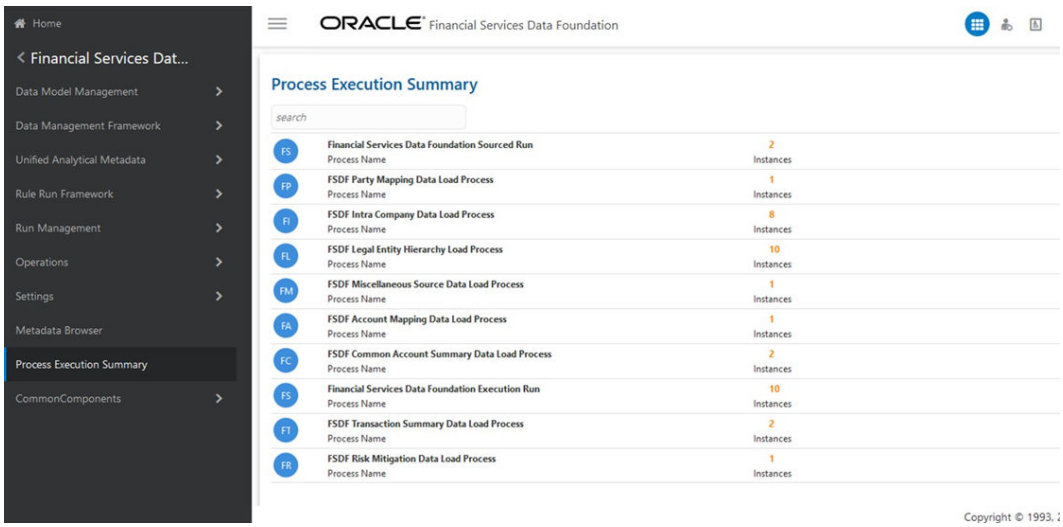
9
Process Execution Summary

The Process Execution Summary provides information on all the instances of any Run executed in the PMF (Process Modeling Framework). The Process Execution Summary page displays all the Runs executed in the PMF for the corresponding application along with the count of the Run instances.

In the **Process Execution Summary** page, to view the details of a Run executed, follow these steps:

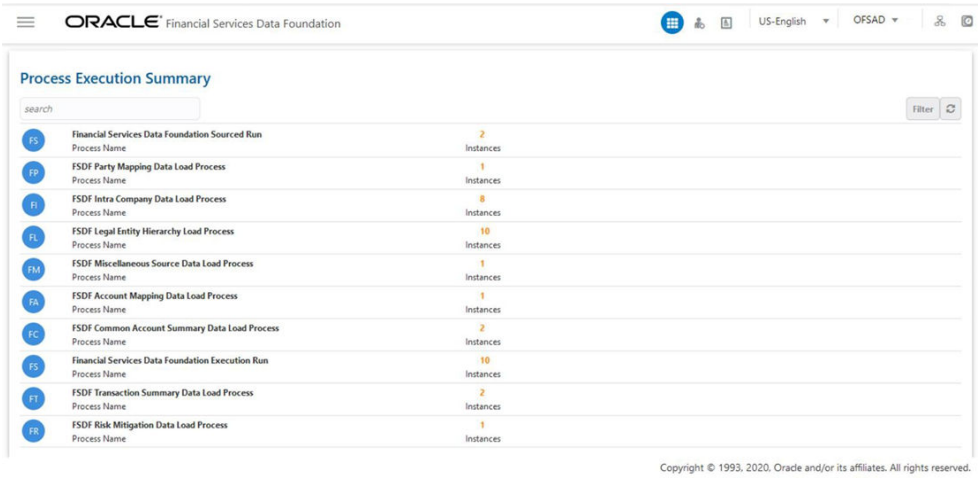
1. From the **OFSDF Home**, select the **Financial Services Data Foundation** tile, select **Financial Services Data Foundation**, and then select **Process Execution Summary**.

Figure 9-1 Navigation to the Process Execution Summary page



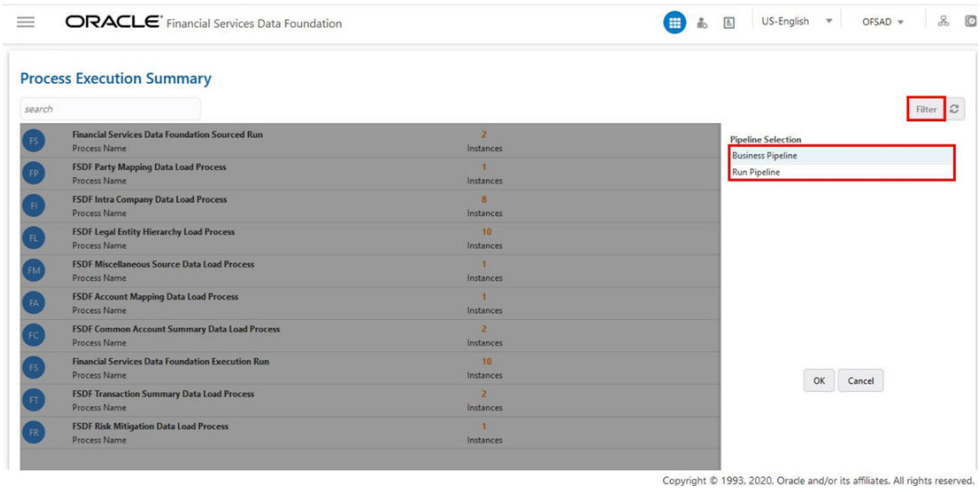
2. The **Process Execution Summary** page is displayed. This page lists all the Processes and Runs executed.

Figure 9-2 Process Execution Summary page



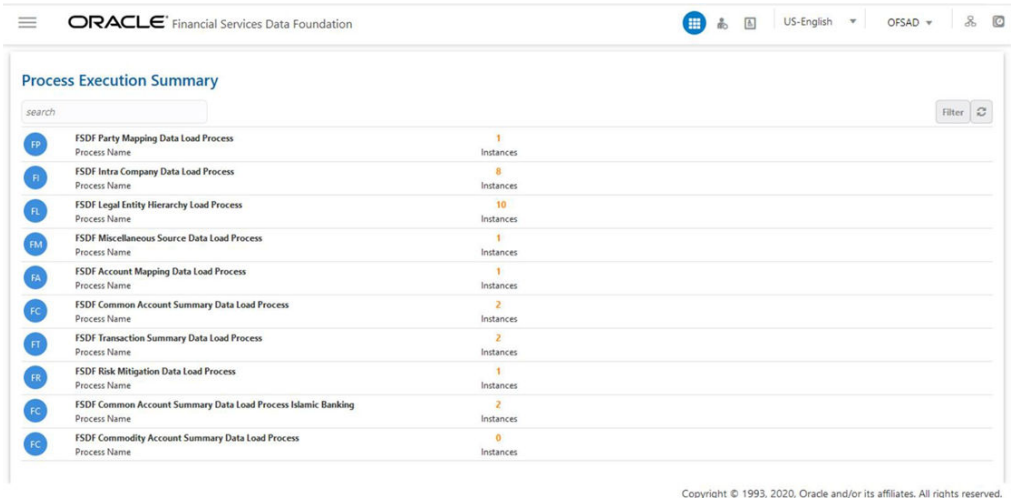
3. Click **Filter** to filter the process list by Pipeline type. You can filter either by the Business Pipeline type or the Run Pipeline type.

Figure 9-3 Filter the process list by Pipeline type



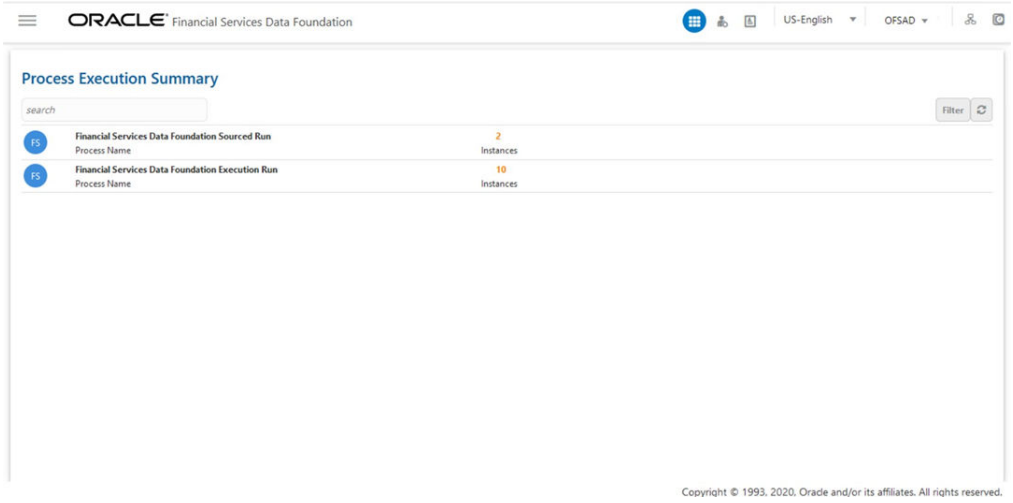
- Select the Business Pipeline type filter and click OK. The list of executed Processes in the Business Pipeline is displayed.

Figure 9-4 Business Pipeline type filter selection



- Select the Run Pipeline type filter and click OK. The list of executed Processes in the Run Pipeline is displayed (lists the types of Runs executed).

Figure 9-5 Run Pipeline type filter selection

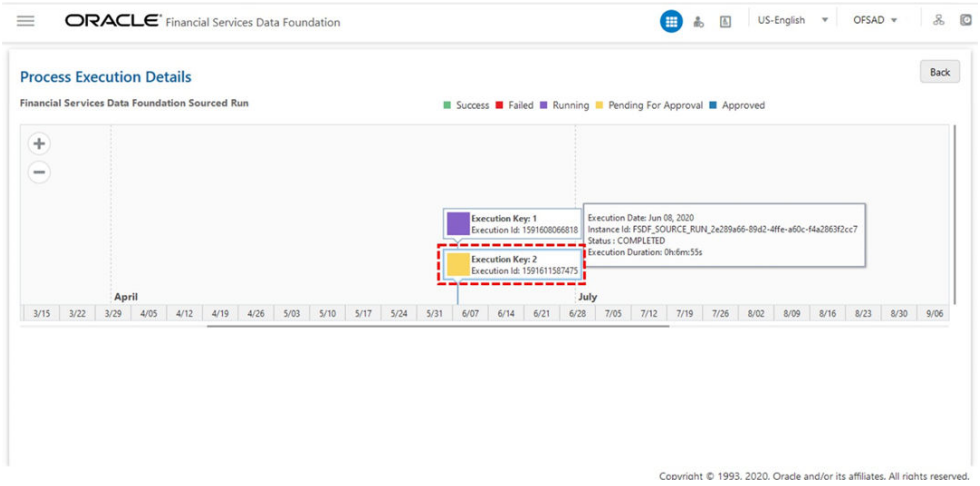


4. In the Process Execution Summary page, on the required process, hover the cursor and click  .

Figure 9-6 View the Process

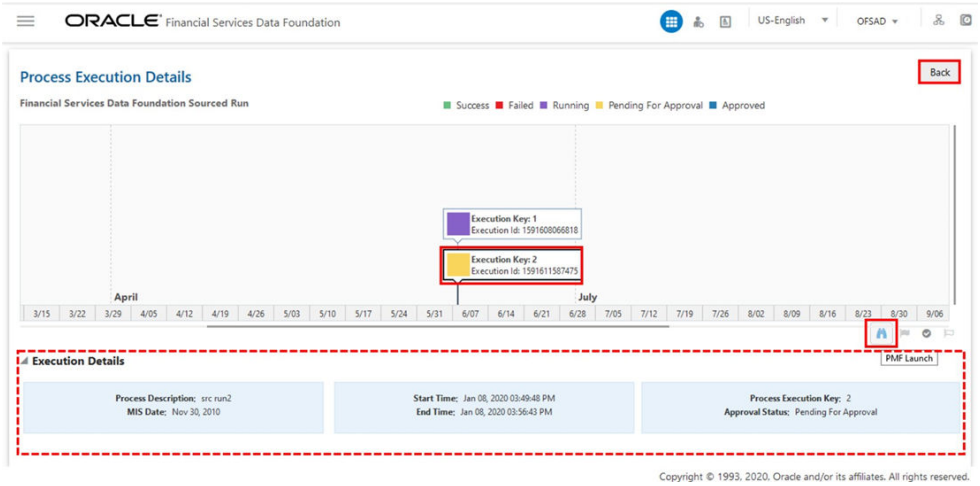
5. The **Process Execution Details** page is displayed. This page lists all the Run instances in the timeline. Hover the cursor on the process execution instance to see the instance details.

Figure 9-7 Process Execution Details page



6. Click a process execution instance, to see its process execution details. To launch the **Process Monitor** page for a required instance of the Run, click PMF Launch  . To return to the **Process Execution Summary** page, click **Back**.

Figure 9-8 Process execution details of an instance



After all the executions are completed in PMF, you can use the Process Execution Summary page to verify the execution status with the timeline.

For information about the Process Modeller and Process Monitor features in OFSDF, see the Process Modeller and Process Monitor sections respectively.

10
Run Execution from Command Line

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

10.1 Run Execution from the Command Line

To Run Execution from the Command Line, follow these steps:

1. Navigate to the $FIC_HOME/ficdb/conf directory.

   (Optional) <Enter a step example.>
2. Enter the details for the following fields in the <RUN_NAME>.properties.template file:
   - FSDF Execution Run - FSDF_EXE_RUN.properties.template
   - FSDF Sourced Run - FSDF_SOURCED_RUN.properties.template details for the <RUN_NAME>.properties.template file:
3. Navigate to $FIC_HOME/ficdb/bin directory.
4. Rename the .properties.template file to .properties. Changed file names are as follows:
   a. (Optional) <Enter a substep.>
   b. (Optional) <Enter a substep.>

11
External Engine

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

11.1 Why is External Engine Required

OFSDF has an Integrated Data Model. Application-specific Outputs or Results Tables are populated by the respective Applications when an OFSAA Application along with OFSDF is installed. Application-specific outputs are consumed by other Applications and for the Regulatory Reporting purposes. However, when an OFSAA Application is not installed with OFSDF, then there is no data in the respective Application-specific Result Tables, which are required for some other installed application to work.

In such a scenario, the Customer was sourcing the data from the Application-specific External Analytical Engine (for example, from OFS IFRS) into the Result Tables directly. However, there are elements such as Surrogate Keys and Amounts represented in different currencies, which are not available directly in the Source System and are derived or computed using OFSDF specific logic. Therefore, the Customer used to do the following:

1. Extract the data from the External Analytical System.

2. Write an ETL to lookup the OFSDF corresponding Dimension Tables to get the required Surrogate Keys, do the currency conversion for the Amount attributes wherever required, and then populate the data set in the Application-specific Result Tables.

3. Update the ETL for each release to accommodate the new changes of the Result Tables.

11.2 About the External Engine

Oracle provides Interface Tables to source the External Engine Outputs as POI (Processing Output Integration) Tables and related Metadata (T2Ts) to handle all the ETL Activities and load the data into the Application-specific Result Tables in OFSDF. As a result, the need of rebuilding of OFSAA specific complex Data Transformation logics, constant maintenances, and data anomalies during currency conversions are eliminated.

POI Tables form the Input Layer, using which the customer can source the data.

As a part of the External Engine, the following provisions are made:

· A separate POI Subject Area is created to hold all the POI Tables.

· POI Table Names are prefixed with POI_.

· T2Ts are provided to load the Application-specific Result Tables from corresponding POI Tables. These T2Ts are loaded as Common Metadata.

· The FSDF_EXCHNG_RATE Sub Process is introduced in the EXTERNAL ENGINE RUN.

NOTE

The External Engine T2Ts are not a part of the default PMF in OFSDF. Customers can include them as required.

The structure of these POI Tables is similar to that of the Application-specific Fact Tables, however, in the place of Surrogate Key, the POI Tables contain the Business Key. The following similar changes are made:

· All the Flag, Indicator, Amount, Rate, Percentage, Date, Code, and ID Columns are a part of the POI Table and these Columns are loaded as-is from the POI Tables to the Fact Tables.

· The Code Columns are provided in the POI tables corresponding to the Fact Table Surrogate Keys.

· Base Amount Columns corresponding to the RCY (Reporting Currency) and LCY (Local Currency) Columns of the Fact Table are added to the POI Table. Currency Conversion is a part of the T2T based on the populated Exchange Rates. Currency Code Column is added to the POI Table.

· POI Table granularity is same as that of the Application-specific Fact Tables along with the FIC_MIS_DATE and LOAD_RUN_ID.

11.3 POI Tables and External Engine T2Ts

The following POI (Processing Output Integration) Staging Tables are created in the OFSDF in the respective Application-specific Subject Areas to source the External Engine Data from the listed Applications to OFSDF:

Table 26: POI Tables, T2T Names, and the corresponding OFSAA Application Name

POI Table Name (Source)	Target Table Name	T2T Name	OFSAA Application Name
POI_IFRS_CREDIT_LINE_DETAILS	FCT_IFRS_CREDIT_LINE_DETAILS	T2T_FCT_IFRS_CREDIT_LINE_DETAILS	OFS IFRS
POI_IFRS_MITIGANTS_SUMMARY	FCT_IFRS_MITIGANTS_SUMMARY	T2T_FCT_IFRS_MITIGANTS_SUMMARY	OFS IFRS
POI_IFRS_PLACED_COLLATERAL	FCT_IFRS_PLACED_COLLATERAL	T2T_FCT_IFRS_PLACED_COLLATERAL	OFS IFRS
POI_LLFP_CRE_FACILITY_SUMMARY	FCTI_LLFP_CRE_FACILITY_SUMMARY	T2T_FCT_LLFP_CRE_FACILITY_SUMMARY	OFS LLFP
POI_COHORT_LOAN_LOSS_DETAILS	FCT_COHORT_LOAN_LOSS_DETAILS	T2T_FCT_COHORT_LOAN_LOSS_DETAILS	OFS LLFP
POI_LRM_MITIGANTS_SUMMARY	FCT_LRM_MITIGANTS_SUMMARY	T2T_FCT_LRM_MITIGANTS_SUMMARY	OFS LRM
POI_LRM_PLACED_COLLATERAL	FCT_LRM_PLACED_COLLATERAL	T2T_FCT_LRM_PLACED_COLLATERAL	OFS LRM
POI_HEDGE_DETAILS	FCT_REG_HEDGE_DETAILS	T2T_FCT_REG_HEDGE_DETAILS	OFS HM

NOTE

The External Engine T2Ts are not a part of the default PMF in OFSDF. Customers can include T2Ts in PMF as required. To include T2Ts in PMF, the EXTERNAL_ENGINE_RUN Batch, containing the External Engine related POI Tables and Fact Tables, need to be tailored to a PMF process. Therefore, execute the pmf_EXTERNAL_ENGINE_LOAD_RUN.sql Script.

POI SCD Batch Name is ##INFODOM##_EXTERNAL_ENGINE_SCD. The Script Files execution order is as follows:

1. poi_batch_master.sql

2. poi_batch_task_master.sql

3. poi_batch_parameter_master.sql

4. poi_batch_task_precedence_master.sql

For more information, see the Configure and Use the External Engine POI Tables section in the Oracle Financial Services Data Foundation Application Pack Installation and Configuration Guide Release 8.1.2.0.0.

11.4 How to Configure and Use the POI Tables

For information about configuring and using the POI Tables, see the Configure and Use the External Engine POI Tables section in the Oracle Financial Services Data Foundation Application Pack Installation and Configuration Guide Release 8.1.2.0.0.

12
Big Data in OFSDF

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

12.1 Overview

OFSDF 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 Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.

12.2 OFSDF Big Data Architecture

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

· Staging and Results on Hive

· Staging on Hive and Results on RDBMS

12.2.1 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.

12.2.2 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 Big Data SQL (For the software version information, see the Oracle Financial Services Analytical Applications Technology Matrix Release 8.1.2.0.0). Oracle Big Data 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. Refer to the below architecture diagram for the representation of this deployment process.

12.2.3 Workarounds in the Staging and Results on Hive

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

12.2.3.1 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

12.2.3.2 Workarounds

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

· 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.

Workarounds:

§ 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.

· 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

· 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.

· In Hive, the exchange rates population for cross currencies is not being derived using exchange rates values for a base currency.

Workaround:

Source the corresponding Exchange Rate value.

· 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 "<Atomic_Schema_Name>.FSI_CREATE_SEQUENCE", line 6 Workaround:

This error can be ignored.

12.3 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).

12.3.1 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 OFSDF HIVE 8.1.2.0.0.

12.3.2 Run Enabled T2Ts

To execute FSDF_SOURCED_RUN and FSDF_EXE_RUN, use the Rules Run Framework. For more information, see the Rules Run Framework section in the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.

12.3.3 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 OFSDF HIVE 8.1.2.0.0.

12.3.4 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).

12.3.5 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-195

SCD-196

SCD-205

SCD-208

SCD-465

SCD-762

SCD-761

SCD-332

SCD-126

SCD-127

SCD-128

SCD-129

SCD-270

SCD-132

> **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.

12.3.6 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_FCT_PARTY_EMPLOYMENT_DETAILS
- T2T_FCT_PARTY_ROLE_MAP
- T2T_FCT_SERV_ACCT_CREDIT_SCORE_DTL
- T2T_FCT_ACCT_CREDIT_SCORE_DETAILS
- T2T_FCT_CAP_INSTR_DETL
- T2T_FCT_SECURITIZATION_TRANCHE
- T2T_FCT_REF_CCY_SWAPS_RATE
- T2T_FCT_REF_EQ_CMDTY_VOL
- T2T_FCT_REF_EQ_DIV_RATE
- T2T_FCT_REF_FOREX_RATE
- T2T_FCT_REF_FOREX_VOL
- T2T_FCT_REF_FRA_SWAPS_RATE
- T2T_FCT_REF_INFLATION_RATE
- T2T_FCT_REF_INTEREST_RATE
- T2T_FCT_REF_MKT_CREDIT_RATE
- T2T_FCT_REF_OPTION_VOL
- fn_Pop_Reg_LE_Hier
- T2T_FSI_EXCHANGE_RATES_FRWD
- T2T_FCT_FIXED_ASSETS
- T2T_FCT_CREDIT_PARTCPN_TRNCH_DETL
- T2T_FCT_HEDGE_PORTFL_SET_ACCT_MAP
- T2T_FCT_SPEND_OBLIGATIONS
- T2T_FCT_PAYMENTS_SUMMARY
- T2T_FCT_MERCHANT_BANKING
- T2T_FCT_LOANS_SERVICED
- T2T_FCT_LITIGATION_DETAILS

- T2T_FCT_FUND_CIS_COMPOSITION
- T2T_FCT_ACCT_WRITE_OFF_DETAILS
- T2T_FCT_ACCT_RECOVERY_DETAILS
- T2T_FCT_CARDS_SUMMARY
- T2T_FDB_STG_TD_CONTRACTS
- T2T_FDB_STG_CASA
- T2T_FDB_STG_BORROWINGS
- T2T_FFSIS_STG_TRUSTS
- T2T_FFSIS_STG_MUTUAL_FUNDS
- T2T_FFSIS_STG_RETIREMENT_ACCOUNTS
- T2T_FFSIS_STG_TD_CONTRACTS
- T2T_FFSIS_STG_MANAGED_INV_ADV
- T2T_FFSIS_STG_CUSTODIAL_ACCOUNTS
- T2T_FFSIS_STG_INVESTMENTS
- T2T_FFSIS_STG_CASA
- T2T_FLAS_STG_OD_ACCOUNTS
- T2T_FLAS_STG_LOAN_CONTRACTS
- T2T_FLAS_STG_LEASES_CONTRACTS
- T2T_FCT_MORT_SERV_RIGHTS
- T2T_FCT_NET_EXPOSURES
- T2T_FCT_PARTY_SHR_HLD_PERCENT
- T2T_FSCAS_STG_CREDIT_DERIVATIVES
- T2T_FSCAS_SWAP_CONTRACTS
- T2T_FSCAS_FORWARDS
- T2T_FSCAS_FUTURES
- T2T_FSCAS_MM
- T2T_FSCAS_STG_INVESTMENTS

> **Note:**
>
> The integration T2Ts with the OFSAA applications such as BASEL, LLFP, and LRS are not supported in Hive.

12.4 Executing Run through Rules Run Framework for Hive

To load data in OFSDF Hive, use the Rules Run Framework. For more information, see the Rules Run Framework section in the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.

13
Account Dimension Table

This chapter provides information about Account Dimension table loading process in the Oracle Financial Services Data Foundation application.

13.1 Overview of Account Dimension

The SCD population in DIM_ACCOUNT table generates individual numeric Surrogate Keys for every account number with an additional leg skey. 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

13.2 Overview of Account Dimension Population

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

Table 13-1 Table to Table Seeded Definitions

Map Reference Number	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
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

Table 13-1 (Cont.) Table to Table Seeded Definitions

Map Reference Number	Source View Name
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
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
494	STG_CREDIT_PARTCPN_DETAILS_V
188	STG_ANNUITY_CONTRACTS_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.

13.3 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 13-2 Configuration for the Account Dimension SCD GAAP Codes

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.

13.3.1 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'.

13.3.2 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 more information, see the section Loading Multiple Load Runs in OFSAA.

14

Customer Dimension Table

This chapter provides information about Customer Dimension table loading process in the Oracle Financial Services Data Foundation application.

14.1 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.

14.2 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.

14.2.1 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’.

14.3 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.

14.3.1 Prerequisites

STG_CUSTOMER_MASTER should be loaded with all records which are required.

14.4 Execution of Customer Dimension Population Batch

Topics:

14.5 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.

To execute the batch, follow these steps:

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.

14.5.1 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.

14.6 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.

To execute the batch, follow these steps:

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.

14.6.1 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 Population using Stage Party Master Entity

Customer Dimension Population using Stage Customer Master Entity

15
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.

Topics:

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.

15.1 Objective

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).

1. Users need to adjust and reload data (either full or partial) for the current date.
2. Users need to adjust and reload data (either full or partial) for any of past dates.
3. 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.

OFSDF 8.1.2.0.0 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.

15.2 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

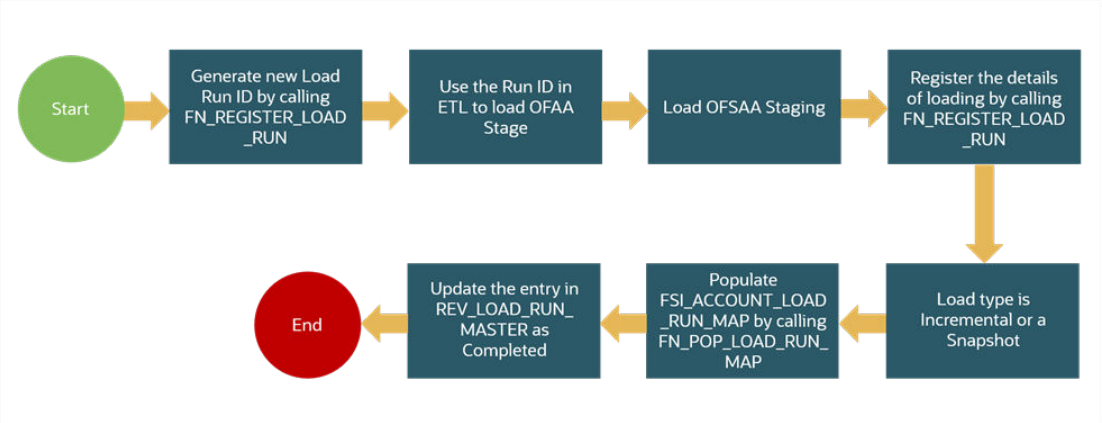
OFSAA data model includes load run identifier as part of the primary key for a set of staging tables. This enables data to be stored for multiple load runs for any date. OFSAA data model also has a table to maintain master information about load run and can be used for identifying or filtering load run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

15.3 Implementation

Before loading data into the staging table, generate a Load Run Identifier to stamp the records from the source. These records can be a complete snapshot or can be partial

or 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.

Figure 15-1 Generate the Load Run Identifier



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

Table 15-1 Parameters of FN_REGISTER_LOAD_RUN

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

Table 15-2 Column values for LOAD_RUN_NAME and LOAD_PURPOSE

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 01 _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 or Extraction date. You can use this load identifier to load either one or more staging tables.

15.4 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

15.4.1 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:

Table 30: Complete Snapshot Load example

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

Table 31: Populate Load Run Map Example

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:

Table 15-3 Records in the FSI_ACCOUNT_LOAD_RUN_MAP table

FIC_MIS_DATE	V_ACCOUNT_NUMBER	V_GAAP_CODE	N_LOAD_RUN_ID	F_LATEST_LOAD_RUN_FLAG
1-Jan-15	LOAN1000	USGAAP	1	N
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.

Table 15-4 Table 33: The STG_LOAN_CONTRACTS table after the loading

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:

Table 15-5 The REV_LOAD_RUN_MASTER table after the second function call

LOAD_RUN_ID	MIS_DATE	LOAD_TYPE	LOAD_PURP OSE	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 customize 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 customized 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.

15.4.2 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 identifer.

Table 15-6 Incremental Snapshot Load example

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

Table 15-7 Populate Load Run Map example

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:

Table 15-8 Records in the FSI_ACCOUNT_LOAD_RUN_MAP table

FIC_MIS_DATE	V_ACCOUNT_NUMBER	V_GAAP_CODE	N_LOAD_RUN_ID	F_LATEST_LOAD_RUN_FL AG
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

15.5 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.

15.5.1 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

Table 15-9 Register Load Run Details

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
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

Table 15-10 Population of the REV_LOAD_RUN_DETAILS table

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

15.5.2 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 ;

15.6 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.

15.6.1 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 Conidtion 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.

15.6.2 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 or fire the Run after the above step.

16

Restatement Support Feature in OFSDF

This chapter provides information about the Restatement Support feature in the Oracle Financial Services Data Foundation application explicitly for the Start Date/End Date attributes.

Topics:

> **Note:**
>
> The Restatement Support feature is a part of the OFSDF application. This chapter is applicable only if you have followed the post-installation procedure to enable the Restatement Support feature.

The Restatement Support is applicable only for the RDBMS based OFSDF application.

When a regulatory body asks a bank or financial institution to revise, make a correction, and refile the report for a prior reporting period, the Restatement Support feature can be used to correct and refile the report. In the Restatement Support feature, using data versioning property, the bank or financial institution can make the required modifications or add a few more data records to the existing report or delete the records from an existing report. The Start Date/End Date attributes of dimensions support the backdated Run execution. Default T2Ts are configured to use Latest Record Indicator to select the appropriate dimensional attributes irrespective of an execution date. Therefore, Start Date/End Date attributes need to be used while rerunning the report for a prior date. A backdated Run execution for Restatement Support is similar to a regular Run execution in OFSDF.

16.1 Features Supporting Restatement

The existing features in OFSDF that support the functioning of the Restatement Support feature are as follows:

16.1.1 Data Versioning

The Data Versioning in OFSDF is a feature, where the new snapshot of a report is saved after any modification and the report is rerun for a particular prior date. Therefore, you obtain the access to all the saved versions of a report.

The Data Versioning supported in OFSDF is in the following modes:

Data Versioning in Seeded Data

Data Versioning in Mappers

Incremental Data Load

16.1.1.1 Data Versioning in Seeded Data

Seeded dimensions are of SCD Type 2 in nature. For the Data Versioning in Seeded Data, the updates to the Seeded Data are constrained and the new record/codes for the same are introduced.

16.1.1.2 Data Versioning in Mappers

Data Versioning supports different versions of the Mapper data. For the Data Versioning in Mappers, the following actions must be performed:

· History tables for each mapper tables are introduced to store the Mapper data versioning for each MIS date.

· For given MIS date during the Run execution, the mapper history tables are populated from the available Mapper tables that are maintained through the UI and then the Fact table data is populated.

· For a backdated execution, if the Mapper data is already available for that prior date, the generation of the history Mappers is restricted.

· Mappers enhanced Fact T2Ts use the history Mappers as the base instead of the MIS date.

16.1.1.3 Incremental Data Load

The existing OFSDF incremental data load support can be used to correct the Fact table data for a prior date.

16.1.2 Data Flow

The Data Flow supported in OFSDF is in the following modes:

- The right version of dimensional attributes are considered in the data flow based on the Start Date/End Date attributes.
- In the Joins with Dimensions, the T2Ts are enhanced to utilize Start Date/End Date attributes instead of the Latest Record Indicator.
- The Rules are modified to look-up the right version of the Dimension data in the source or target node selection during the Rule execution.
- The right processing logic is used that is applicable for the prior date.
- An MIS date for which the Restatement based Run is executed, the Mapper data from the Mapper history table is picked up for that Run.

16.2 Prerequisites

The prerequisites required for the Restatement Support feature are:

- In the SCDs, there must be no overlapping between the Start Date and End Date attributes.
- The Dimensions with no available data flow, custom methods must be used to populate the data. You must ensure that versioning is maintained for those Dimensions.
- You must enhance the T2T metadata to look-up the appropriate version of the Dimension data for all the Dimensions in the list.
- OFSAA does not maintain any versioning for the Seeded Data. The Seeded Data look-up continues to be on the LRI (Latest Record Indicator). For maintaining any custom Seeded data, a similar approach must be followed.
- You must handle any customized objects used as an alternative to the mentioned impacted objects.

16.3 Use Case for Restatement in OFSDF

This section consists of a use case for Restatement in OFSDF.

Consider that a Bank or financial institution has missed to report some accounts or transactions when reporting to the regulator. Then the regulator has asked the Bank or financial institution to resubmit the report with complete details. To address this requirement, the Bank or financial institution must load the missing entries or reload the complete data set for past dates. Then rerun the reports for that prior date. The Restatement process helps in loading or reloading data into the respective Fact tables by looking-up right version of the Surrogate keys that is applicable for the prior MIS date. After the data is corrected, the Bank or financial institution can rerun the report and resubmit it.

16.4 Assumptions for the Implementation of Restatement

The assumptions for the implementation of the Restatement Support feature are:

Applicable only on those MIS dates that are versioned after the date on which this feature is enabled.

17
Exchange Rates Tables

This chapter contains information about the Exchange Rates tables in the Oracle Financial Services Data Foundation application.

Topics:

The Exchange Rates table stores the list of all exchange rates for all types of currency. This is a standalone table.

The purpose of the Exchange Rate table is to offer the value of one currency in relation to another currency.

17.1 Handling Alternate Currency

In downloadable format, the customers supply Exchange Rates. The applications may require the information in a different format. The current Exchange Rates population is 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.

17.2 Business Use Case Indicating the Requirement of two Exchange Rates Tables

All the balance sheet computations are done based on the average price or the closing price. For illustration, assume that a bank has branches in multiple countries and therefore, multiple exchange rates scenarios exist. Then the bank must decide to choose the closing price of exchange rate.

Therefore, the bank computes the balance sheet based on the entity in each country. Assume that the Bank is operative in four different countries. If the legal entity is based on country A, then A category of prices is used for balance sheet computation, and if the legal entity is based on country B, then B category of prices is used, and C category of prices based on country C, and D category of prices based on country D.

To achieve this, for a single day, bank captures multiple exchange rates because of different countries. For CurrencyA to CurrencyB, the bank must capture three pairs of exchange rates. OFSDF handles this type of scenario using the Rate Data Source Code, which is based on the data source. The bank chooses the time zone. The bank creates different data sources such as country AB, country AC, and country AD. Mapping exists from Legal Entity to each data source. If the Legal Entity is of country B, then the bank uses AB as the data source and all the prices of data source AB is used to compute all the transactions. This is the business use case that resulted in the requirement of two Exchange Rates tables in OFSDF.

The existing Stage Exchange Rates table cannot be modified due to the presence of the PK column. The existing customers also do not need two Stage Exchange Rates tables.

17.3 Overview of the Exchange Rates Tables

OFSDF has two Exchange Rates Staging tables. They are:

- STG_EXCHANGE_RATE_HIST
- STG_FORWARD_EXCHG_RATES

Both Spot and Forward FX rates. Sport rates will be loaded with Tenor 0.

The Exchange Rate table FSI_EXCHANGE_RATES is loaded from a View table VW_FSI_RATE_TRIANGULATION, where the VW_FSI_RATE_TRIANGULATION table is created on top of the Exchange Rates Stage tables through the T2T process.

Prior to the OFSDF 8.1.0.0.0 release, T2T to load data from STG_EXCHANGE_RATE_HIST was only provided for exchange rates. . Now T2T for loading data from STG_FORWARD_EXCHG_RATES also provided. There are two different data loading categories in OFSDF for the Exchange Rates tables as follows:

Data loading method supported for the Exchange Rates table from the OFSDF 8.1.0.0.0 release

Data loading method supported for the Exchange Rates table for the OFSDF 8.0.9.0.0 and earlier versions

17.3.1 Data loading method supported from the OFSDF 8.1.0.0.0 release

The existing customers may follow the new data loading method for the Exchange Rates tables. However, the new customers must follow this new data loading method for the Exchange Rates tables. In the new method, the data loads from the View table VW_FSI_RATE_TRIANGULATION into the Stage table STG_FORWARD_EXCHG_RATES.

> **Note:**
>
> This is the only data loading method available in the v8.1.0.0.0 and onward releases for the Exchange Rates tables.

17.3.2 Data loading method supported for the OFSDF 8.0.9.0.0 and earlier versions

Only the existing customers may follow the previous data loading methods for the Exchange Rates tables.

> **Note:**
>
> The method of data loading into the STG_EXCHANGE_RATE_HIST table is no more supported in the OFSDF v8.1.0.0.0 and onward versions for the Exchange Rates tables.

In an integrated environment, there can be a scenario, where the customer is using two applications and each of the applications refer to the STG_FORWARD_EXCHG_RATES table or the STG_EXCHANGE_RATE_HIST table. This scenario may result in the duplicate data loads into the STG_FORWARD_EXCHG_RATES and STG_EXCHANGE_RATE_HIST tables. For this scenario, these are the recommendations:

- For the data load, the STG_EXCHANGE_RATE_HIST table supersedes the STG_FORWARD_EXCHG_RATES table.
- The implementation team ensures that the T2T T2T_FSI_EXCHANGE_RATES is used for the STG_EXCHANGE_RATE_HIST table data load.
- The STG_EXCHANGE_RATE_HIST table loads the FSI_EXCHANGE_RATES table.

17.4 About Exchange Rates T2Ts (Result Table)

This section provides information about the existing Exchange Rates T2Ts.

17.4.1 About Exchange Rates T2T for the STG_EXCHANGE_RATE_HIST table

Exchange Rates T2T for the STG_EXCHANGE_RATE_HIST table and its description are as follows.

Table 17-1 Exchange Rates T2T for the STG_EXCHANGE_RATE_HIST table and its Description

T2T Name	T2T Description
T2T_FSI_EXCHANGE_RATES	This T2T stores history of the exchange rates between two currencies sourced through the STG_EXCHANGE_RATE_HIST table.

The mapping details for the Exchange Rates T2T are as follows.

> **Note:**
>
> Ensure to verify and load data into the STG_EXCHANGE_RATE_HIST table using the Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework.

Table 17-2 Mapping details for the Exchange Rates T2T

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
VW_FSI_RATE_ TRI ANGULATION	FSI Rate Triangulation View	FSI_EXCHANGE _ RATES	FSI Exchange Rates	T2T_FSI_EXCHA NGE_ RATES

17.4.2 About Exchange Rates T2T for the STG_FORWARD_EXCHG_RATES table

T2T_FSI_EXCHANGE_RATES_FRWD is added in the OFSDF 8.1.0.0.0 release. The Exchange Rates T2T for the STG_FORWARD_EXCHG_RATES table and its description are as follows.

Table 41: Exchange Rates T2T for the STG_FORWARD_EXCHG_RATES table and its Description

T2T Name	T2T Description
T2T_FSI_EXCHANGE_RATES_FRWD	This T2T stores history of the exchange rates between two currencies sourced through the STG_FORWARD_EXCHG_RATES table.

The mapping details for the Exchange Rates T2T are as follows.

> **Note:**
>
> Only this T2T is part of the OOB Runchart.

Ensure to verify and load data into the STG_FORWARD_EXCHG_RATES table using the Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework.

Table 42: Mapping details for the Exchange Rates T2T

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
VW_FSI_RATE_ TRI ANGULATION	FSI Rate Triangulation View	FSI_EXCHANGE _ RATES	FSI Exchange Rates	T2T_FSI_EXCHA NGE_ RATES_FRWD

17.5 Multiple Execution of T2Ts using the Runchart

Use the Process ID FSDF_SOURCED_RUN to populate data in the FSI_EXCHANGE_RATES table with the Run SKey defaulted to -1.

> **Note:**
>
> Use these exchange rates to populate data in the FSDF Result tables that do not contain any Run SKey.
> Use the FSDF Source Run to populate the previously mentioned Results tables (that use the exchange rates), which were populated against the Run SKey -1.

The T2T is also a part of the seeded Financial Services Data Foundation Execution Run. When executing the Run, the Run SKey is auto-generated and stamped against each record. The FSDF Result tables with valid Run SKey in their Primary Key are a part of this Run. The exchange rates populated against each Run SKey are used to populate the Results Area of the same Run SKey.

> **Note:**
>
> The Exchange Rates T2Ts are a part of the Run Chart twice:
>
> The first is through a batch. This inserts the Run SKey as -1 in the Exchange Rates table.
>
> The second is through a Run. This inserts the Run SKey as an actual Run SKey in the Exchange Rates table.
>
> For example, the FCT_COMMON_ACCOUNT_SUMMARY table is not a Run enabled table. Therefore, the join is with -1. However, the FCT_REG_ACCOUNT_SUMMARY table is a Run enabled table. Therefore, the join is with the actual Run SKey.

All the T2Ts that are a part of the Process ID FSDF_EXE_RUN, use the exchange rates with the actual Run SKey, and the other T2Ts use the Run SKey -1.

17.6 Populating Exchange Rates T2T Result Tables

Execute Runs through Process Modelling Framework. The T2T is part of the Financial Services Data Foundation Sourced Run. Therefore, execute the process through the Financial Services Data Foundation Execution Run. For more information, see the section Executing Run through Process Modelling Framework in OFSDF.

> **Note:**
>
> When executing the Run, the Run SKey is auto-generated and stamped against each record.

For more information about Process Modelling Framework, see Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

18
Account Summary Tables

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

18.1 Overview of the 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 (OFSAA) 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.

18.2 Overview of the 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 18-1 Account Summary T2T Definitions

Source Table Name	T2T Name
STG_ANNUITY_CONTRACTS	T2T_STG_ANNUITY_CONTRACTS_CAS
STG_BILLS_CONTRACTS	T2T_STG_BILLS_CAS
STG_BORROWING_COMMITMENTS	T2T_STG_BORROWING_COMMITMENT S_CAS
STG_BORROWINGS	T2T_STG_BORROWINGS_CAS
STG_CARDS	T2T_STG_CARDS_CAS
STG_CASA	T2T_STG_CASA_CAS
STG_COMMITMENT_CONTRACTS	T2T_STG_COMMITMENT_CONTRACTS_ CAS

Table 18-1 (Cont.) Account Summary T2T Definitions

Source Table Name	T2T Name
STG_CORRESPONDENT_ACCOUNT	T2T_STG_CORRESPONDENT_ACCOUNT_CAS
STG_CREDIT_DERIVATIVES	T2T_STG_CREDIT_DERIVATIVES_CAS
STG_FUTURES	T2T_STG_FUTURES_CAS
STG_FX_CONTRACTS	T2T_STG_FX_CONTRACTS_CAS
STG_GUARANTEES	T2T_STG_GUARANTEES_CAS
STG_IJARAH	T2T_STG_IJARAH_CAS
STG_INVESTMENTS	T2T_STG_INVESTMENTS_CAS
STG_ISTISNA	T2T_STG_ISTISNA_CAS
STG_LC_CONTRACTS	T2T_STG_LC_CAS
STG_LEASES_CONTRACTS	T2T_STG_LEASES_CONTRACTS_CAS
STG_LOAN_CONTRACTS	T2T_STG_LOANS_CAS
STG_MM_CONTRACTS	T2T_STG_MM_CAS
STG_MUDARABAH	T2T_STG_MUDARABAH_CAS
STG_MURABAHAH	T2T_STG_MURABAHAH_CAS
STG_MUSHARAKAH	T2T_STG_MUSHARAKAH_CAS
STG_MUTUAL_FUNDS	T2T_STG_MUTUAL_FUNDS_CAS
STG_OD_ACCOUNTS	T2T_STG_OD_CAS
STG_OPTION_CONTRACTS	T2T_STG_OPTIONS_CAS
STG_REPO_CONTRACTS	T2T_STG_REPO_CONTRACTS_CAS
STG_RETIREMENT_ACCOUNTS	T2T_STG_RETIREMENT_ACCOUNTS_C AS
STG_SALAM	T2T_STG_SALAM_CAS
STG_SUKUK	T2T_STG_SUKUK_CAS
STG_SWAPS_CONTRACTS	T2T_STG_SWAPS_CONTRACTS_CAS
STG_TD_CONTRACTS	T2T_STG_TD_CONTRACTS_CAS
STG_TRUSTS	T2T_STG_TRUSTS_CAS
STG_COMMODITIES	T2T_STG_COMMODITIES_CAS
STG_CUSTODIAL_ACCOUNTS	T2T_STG_CUSTODIAL_ACCOUNTS_CAS
STG_MANAGED_INV_ADV	T2T_STG_MANAGED_INV_ADV_CAS
STG_PREPAID_CARDS	T2T_STG_PREPAID_CARDS_CAS
STG_TRADING_ACCOUNT	T2T_STG_TRADING_ACCOUNT_CAS

18.3 Executing the Account Summary T2Ts using PMF

Fact Common Account Summary table has to be loaded prior loading any of the other Account Summary tables.

> **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.

18.3.1 Execution through PMF

The T2T is a part of Financial Services Data Foundation Sourced Run. The process can be executed through the out-of-the-box standard Run, which is Financial Services Data Foundation Execution Run. For more information, see the section Executing Run through Process Modelling Framework in OFSDF.

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

Table 18-2 GAAP code configuration to execute the Account Summary T2T

V_COMPONENT_ CODE	V_COMPONENT_DESC	V_COMPONENT_VALUE
DEFAULT_GAAP	DEFAULT_GAAP	USGAAP

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

18.4 Checking the Execution Status

The status of execution can be monitored in the Process Monitor page.

> **Note:**
>
> For a more comprehensive coverage of configuration and execution of a process, see the section Executing Run through Process Modelling Framework in OFSDF and the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

Access the execution log on the application server in the following directory:

ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA

The file name contains the batch execution ID. The error log table in the atomic schema is as follows: FCT_COMMON_ACCOUNT_SUMMARY$

18.5 Retrieving the Account Summary T2T Definitions

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

19

Customer Summary Tables

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

19.1 Overview of the 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.

19.2 Prerequisites

Following is the list 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_SUMMAY is loaded from the respective T2T processes.

19.3 Executing the Customer Summary T2T using PMF

The T2T is a part of Financial Services Data Foundation Sourced Run. The process can be executed through the out-of-the-box standard Run, which is Financial Services Data Foundation Execution Run. For more information, see the section Executing Run through Process Modelling Framework in OFSDF.

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

19.3.1 Error Messages

The following most common error message may be logged in the T2T log file directory ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA:

Unique Constraint Violation

This error occurs after there is an attempt to re-load or load the existing records for the already executed AS_OF_DATE.

20
Fact Transaction Summary Tables

This chapter provides information about Fact Transaction Summary tables 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.

20.1 Fact Transaction Summary T2Ts

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

Table 45: Fact Transaction Summary T2T Definitions

Source Table Name	T2T Name
STG_ANNUITY_TXNS	T2T_STG_ANNUITY_TXNS_FTS
STG_BILL_CONTRACTS_TXNS	T2T_STG_BILL_CONTRACTS_TXNS_FTS
STG_BORROWING_COMMITMENT_TXNS	T2T_STG_BORROWING_COMMITMENT_TXNS_FTS
STG_BORROWINGS_TXNS	T2T_STG_BORROWINGS_TXNS_FTS
STG_CARDS_PAYMENT_TXNS	T2T_STG_CARDS_PAYMENT_TXNS_FTS
STG_CARDS_SETTLEMENT_TXNS	T2T_STG_CARDS_SETTLEMENT_TXNS_FTS
STG_CASA_TXNS	T2T_STG_CASA_TXNS_FTS
STG_COMMITMENT_CONTRACT_TXNS	T2T_STG_COMMITMENT_CONTRACT_TXNS_FTS
STG_CORRESPONDENT_ACCT_TXNS	T2T_STG_CORRESPONDENT_ACCT_TXNS_FTS
STG_CREDIT_DERIVATIVES_TXNS	T2T_STG_CREDIT_DERIVATIVES_TXNS_FTS
STG_FOREX_TXNS	T2T_STG_FOREX_TXNS_FTS
STG_FUTURES_TXNS	T2T_STG_FUTURES_TXNS_FTS
STG_GUARANTEES_TXNS	T2T_STG_GUARANTEES_TXNS_FTS
STG_IJARAH_TXNS	T2T_STG_IJARAH_TXNS_FTS
STG_INVESTMENT_TXNS	T2T_STG_INVESTMENT_TXNS_FTS
STG_ISTISNA_TXNS	T2T_STG_ISTISNA_TXNS_FTS
STG_LC_TXNS	T2T_STG_LC_TXNS_FTS
STG_LEASES_TXNS	T2T_STG_LEASES_TXNS_FTS
STG_LOAN_CONTRACT_TXNS	T2T_STG_LOAN_CONTRACT_TXNS_FTS
STG_MM_TXNS	T2T_STG_MM_TXNS_FTS
STG_MUDARABAH_TXNS	T2T_STG_MUDARABAH_TXNS_FTS
STG_MURABAHAH_TXNS	T2T_STG_MURABAHAH_TXNS_FTS
STG_MUSHARAKAH_TXNS	T2T_STG_MUSHARAKAH_TXNS_FTS
STG_MUTUAL_FUNDS_TXNS	T2T_STG_MUTUAL_FUNDS_TXNS_FTS

Source Table Name	T2T Name
STG_OD_ACCOUNTS_TXNS	T2T_STG_OD_ACCOUNTS_TXNS_FTS
STG_OPTION_CONTRACTS_TXNS	T2T_STG_OPTION_CONTRACTS_TXNS_FTS
STG_RETIREMENT_ACCOUNTS_TXNS	T2T_STG_RETIREMENT_ACCOUNTS_TXNS_FTS
STG_SALAM_TXNS	T2T_STG_SALAM_TXNS_FTS
STG_SUKUK_TXNS	T2T_STG_SUKUK_TXNS_FTS
STG_SWAP_ACCOUNT_TXNS	T2T_STG_SWAP_ACCOUNT_TXNS_FTS
STG_TERMDEPOSITS_TXNS	T2T_STG_TERMDEPOSITS_TXNS_FTS
STG_TRADING_ACCOUNT_TXNS	T2T_STG_TRADING_ACCOUNT_TXNS_FTS
STG_TRUSTS_TXNS	T2T_STG_TRUSTS_TXNS_FTS
STG_COMMODITIES_TXNS	T2T_STG_COMMODITIES_TXNS_FTS
STG_CUSTODIAN_ACCOUNT_TXNS	T2T_STG_CUSTODIAN_ACCOUNT_TXNS_FTS
STG_PREPAID_CARDS_TXNS	T2T_STG_PREPAID_CARDS_TXNS_FTS
STG_REPO_TRANSACTIONS	T2T_STG_REPO_TRANSACTIONS_FTS
STG_FORWARDS_TXNS	T2T_STG_FORWARDS_TXNS_FTS

20.2 Executing the Fact Transaction Summary T2Ts using PMF

The T2T is a part of Financial Services Data Foundation Sourced Run. The process can be executed through the out-of-the-box standard Run, which is Financial Services Data Foundation Execution Run. For more information, see the section Executing Run through Process Modelling Framework in OFSDF.

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

21
Loan Account Summary Tables

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

21.1 Overview of the 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.

21.2 Overview of the 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 21-1 Loan Account Summary T2T Definitions

Source Table Name	T2T Name
STG_LEASES_CONTRACTS	T2T_FLAS_STG_LEASES_CONTRACTS
STG_LOAN_CONTRACTS	T2T_FLAS_STG_LOAN_CONTRACTS
STG_OD_ACCOUNTS	T2T_FLAS_STG_OD_ACCOUNTS

21.3 Executing the Loan Account Summary T2Ts using PMF

The T2T is a part of Financial Services Data Foundation Sourced Run. The process can be executed through the out-of-the-box standard Run, which is Financial Services Data Foundation Execution Run. For more information, see the section Executing Run through Process Modelling Framework in OFSDF.

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

21.3.1 Error Messages

The following most common error message may be logged in the T2T log file directory ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA:

Unique Constraint Violation

This error occurs after there is an attempt to re?load or load the existing records for the already executed AS_OF_DATE

21.4 Checking the Execution Status

The status of execution can be monitored in the Process Monitor page.

> **Note:**
>
> For a more comprehensive coverage of configuration and execution of a process, see the section Executing Run through Process Modelling Framework in OFSDF and the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

Access the execution log on the application server in the following directory:

```
ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA
```

The file name contains the batch execution ID. The error log table in the atomic schema is as follows: `FCT_LOAN_ACCOUNT_SUMMARY$`.

21.5 Retrieving the Loan Account Summary T2T Definitions

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

22
Deposit Borrowings Summary Tables

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

22.1 Overview of the 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.

22.2 Overview of the 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 22-1 Deposit Borrowings Summary T2T Definitions

Source Table Name	T2T Definition Name
STG_BORROWINGS	T2T_FDB_STG_BORROWINGS
STG_CASA	T2T_FDB_STG_CASA
STG_TD_CONTRACTS	T2T_FDB_STG_TD_CONTRACTS

22.3 Executing the Deposit Borrowings Summary T2Ts using PMF

The T2T is a part of Financial Services Data Foundation Sourced Run. The process can be executed through the out-of-the-box standard Run, which is Financial Services Data Foundation Execution Run. For more information, see the section Executing Run through Process Modelling Framework in OFSDF.

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

22.3.1 Error Messages

The following most common error message may be logged in the T2T log file directory `ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA`:

```
Unique Constraint Violation
```

This error occurs after there is an attempt to re?load or load the existing records for the already executed AS_OF_DATE.

22.4 Checking the Execution Status

The status of execution can be monitored in the Process Monitor page.

> **Note:**
>
> For a more comprehensive coverage of configuration and execution of a process, see the section Executing Run through Process Modelling Framework in OFSDF and the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

Access the execution log on the application server in the following directory:

```
ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA
```

The file name contains the batch execution ID. The error log table in the atomic schema is as follows:

```
FCT_DEPOSITS_BORROWINGS$
```

22.5 Retrieving the Deposit Borrowings Summary T2T Definitions

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

23
Cards Summary Tables

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

23.1 Overview of the 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.

23.2 Overview of the 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 23-1 Cards Summary T2T Definitions

Source Table Name	T2T Definition Name
STG_CARDS, STG_CARDS_MASTER	T2T_FCT_CARDS_SUMMARY
STG_CARD_ACCT_MAPPING	T2T_FCT_CARD_ACCT_MAPP ING
STG_CARDS_BALANCE_SUMMARY	T2T_FCT_CARDS_BALANCE_ SUMMARY

23.3 Executing the Cards Summary T2Ts using PMF

The T2T is a part of Financial Services Data Foundation Sourced Run. The process can be executed through the out-of-the-box standard Run, which is Financial Services Data Foundation Execution Run. For more information, see the section Executing Run through Process Modelling Framework in OFSDF.

- 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.

23.3.1 Error Messages

The following most common error message may be logged in the T2T log file directory `ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA`:

```
Unique Constraint Violation
```

This error occurs after there is an attempt to re?load or load the existing records for the already executed AS_OF_DATE.

23.4 Checking the Execution Status

The status of execution can be monitored in the Process Monitor page.

> **Note:**
>
> For a more comprehensive coverage of configuration and execution of a process, see the section Executing Run through Process Modelling Framework in OFSDF and the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

Access the execution log on the application server in the following directory:

```
ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA
```

The file name contains the batch execution ID. The error log table in the atomic schema is as follows:

- FCT_CARDS_SUMMARY$
- FCT_CARD_ACCT_MAPPING$
- FCT_CARDS_BALANCE_SUMMARY$

23.5 Retrieving the Cards Summary T2T Definitions

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

24
Mitigants Results Tables

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.

24.1 Overview of the 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

24.2 Overview of the Mitigants Results Population

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

24.3 Executing the Mitigants Results T2Ts using PMF

This section provides the Mitigants Results T2T details.

Table 24-1 Mitigants Results T2T Definitions

Source Table Name	Target Table Name	T2T Name
STG_MITIGANTS	FCT_MITIGANTS	T2T_FCT_MITIGANTS
STG_ACCOUNT_MITIGANT_MAP	FCT_ACCOUNT_MITIGANT_MAP	T2T_FCT_ACCOUNT_MITIGANT_MAP
STG_LOAN_SERVICED_MITIGANT_MAP	FCT_LOAN_SERVICED_MITIGANT_MAP	T2T_FCT_LOAN_SERVICED_MITIGANT_MAP

The T2Ts are a part of Financial Services Data Foundation Sourced Run. The process can be executed through the out-of-the-box standard Run, which is Financial Services Data Foundation Execution Run. For more information, see the section Executing Run through Process Modelling Framework in OFSDF.

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

24.3.1 Error Messages

The following most common error message may be logged in the T2T log file directory ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA:

```
Unique Constraint Violation
```

This error occurs after there is an attempt to re?load or load the existing records for the already executed AS_OF_DATE.

24.4 Checking the Execution Status

The status of execution can be monitored in the Process Monitor page.

> **Note:**
>
> For a more comprehensive coverage of configuration and execution of a process, see the section Executing Run through Process Modelling Framework in OFSDF and the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

Access the execution log on the application server in the following directory:

```
ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA
```

The file name contains the batch execution ID. The error log table in the atomic schema is as follows:

- FCT_MITIGANTS$
- FCT_ACCOUNT_MITIGANT_MAP$
- FCT_LOAN_SERVICED_MITIGANT_MAP$

24.5 Retrieving the Mitigants Results T2T Definitions

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

25 Fiduciary Services Investment Summary Tables

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

25.1 Overview of the Fiduciary Services Investment Summary Tables

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, Mututal 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.

25.2 Overview of the 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 25-1 Fiduciary Services Investment Summary T2T Definitions

Source Table Name	T2T Definition Name
STG_CASA	T2T_FFSIS_STG_CASA
STG_CUSTODIAL_ACCOUNTS	T2T_FFSIS_STG_CUSTODIAL_ACCOUNTS
STG_INVESTMENTS	T2T_FFSIS_STG_INVESTMENTS
STG_MANAGED_INV_ADV	T2T_FFSIS_STG_MANAGED_INV_ADV
STG_MUTUAL_FUNDS	T2T_FFSIS_STG_MUTUAL_FUNDS
STG_RETIREMENT_ACCOUNTS	T2T_FFSIS_STG_RETIREMENT_ACCOUNTS
STG_TD_CONTRACTS	T2T_FFSIS_STG_TD_CONTRACTS

Table 25-1 (Cont.) Fiduciary Services Investment Summary T2T Definitions

Source Table Name	T2T Definition Name
STG_TRUSTS	T2T_FFSIS_STG_TRUSTS

25.3 Executing the Fiduciary Services Investment Summary T2Ts using PMF

The T2Ts are a part of Financial Services Data Foundation Sourced Run. The process can be executed through the out-of-the-box standard Run, which is Financial Services Data Foundation Execution Run. For more information, see the section Executing Run through Process Modelling Framework in OFSDF.

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

25.3.1 Error Messages

The following most common error message may be logged in the T2T log file directory `ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA`:

```
Unique Constraint Violation
```

This error occurs after there is an attempt to re?load or load the existing records for the already executed AS_OF_DATE.

25.4 Checking the Execution Status

The status of execution can be monitored in the Process Monitor page.

> **Note:**
>
> For a more comprehensive coverage of configuration and execution of a process, see the section Executing Run through Process Modelling Framework in OFSDF and the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

Access the execution log on the application server in the following directory:

```
ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA
```

The file name contains the batch execution ID. The error log table in the atomic schema is as follows:

```
FCT_FIDUCIARY_SERV_INVST_SUMM$
```

25.5 Retrieving the Fiduciary Services Investment Summary T2T Definitions

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

26
Party Attributes

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.

26.1 Overview of the 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
- FCT_PARTY_DETAILS

26.2 Overview of the Party Attributes Results Population

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

Table 26-1 Party Attributes Result T2T Definitions

Source Table Name	Target Table Name	T2T Definition Name
STG_PARTY_ADDRESS_MAP	FCT_PARTY_ADDRESS_MAP	T2T_FCT_PARTY_ADDRESS_MAP
STG_PARTY_EMAIL_MAP	FCT_PARTY_EMAIL_MAP	T2T_FCT_PARTY_EMAIL_MAP
STG_PARTY_PHONE_MAP	FCT_PARTY_PHONE_MAP	T2T_FCT_PARTY_PHONE_MAP
STG_PARTY_FINANCIALS	FCT_PARTY_FINANCIAL_DETAIL	T2T_FCT_PARTY_FINANCIAL_DETAIL
STG_PARTY_FINANCIALS	FCT_PARTY_FINANCIALS	T2T_FCT_PARTY_FINANCIALS
STG_PARTY_RATING_DETAILS	FCT_PARTY_RATING_DETAILS	T2T_FCT_PARTY_RATING_DETAILS
STG_ACCOUNT_RATING_DETAILS	FCT_ACCOUNT_RATING_DETAILS	T2T_FCT_ACCOUNT_RATING_DETAILS
STG_CREDIT_SCORE_DETAILS	FCT_CREDIT_SCORE_DETAILS	T2T_FCT_CREDIT_SCORE_DETAILS
STG_PARTY_DETAILS	FCT_PARTY_DETAILS	T2T_FCT_PARTY_DETAILS

26.3 Executing the Party Attribute T2Ts using PMF

The T2Ts are a part of Financial Services Data Foundation Sourced Run. The process can be executed through the out-of-the-box standard Run, which is Financial Services Data Foundation Execution Run. For more information, see the section Executing Run through Process Modelling Framework in OFSDF.

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

26.4 Error Messages

The following most common error message may be logged in the T2T log file directory ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA:

Unique Constraint Violation

This error occurs after there is an attempt to re?load or load the existing records for the already executed AS_OF_DATE.

26.5 Checking the Execution Status

The status of execution can be monitored in the Process Monitor page.

> **Note:**
>
> For a more comprehensive coverage of configuration and execution of a process, see the section Executing Run through Process Modelling Framework in OFSDF and the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

Access the execution log on the application server in the following directory:

ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA

The file name contains the batch execution ID. The error log table in the atomic schema is as follows:

- 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$
- FCT_PARTY_DETAILS$

26.6 Retrieving the Party Attributes Result T2T Definitions

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

27
Placed Collateral Tables

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.

27.1 Overview of the Placed Collateral Tables

Placed Collateral 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

27.2 Overview of the Placed Collateral Population

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

Table 27-1 Placed Collateral T2T Definitions

Source Table Name	Target Table Name	T2T Definition Name
STG_PLACED_COLLATERAL	FCT_PLACED_COLLATERAL	T2T_FCT_PLACED_COLLATERAL
STG_ACCT_PLACED_COLL_MAP	FCT_ACCT_PLACED_COLL_MAP	T2T_FCT_ACCT_PLACED_COLL_MAP

27.3 Executing the Placed Collateral T2Ts using PMF

The T2Ts are a part of Financial Services Data Foundation Sourced Run. The process can be executed through the out-of-the-box standard Run, which is Financial Services Data Foundation Execution Run. For more information, see the section Executing Run through Process Modelling Framework in OFSDF.

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

27.3.1 Error Messages

The following most common error message may be logged in the T2T log file directory ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA:

```
Unique Constraint Violation
```

This error occurs after there is an attempt to re?load or load the existing records for the already executed AS_OF_DATE.

27.4 Checking the Execution Status

The status of execution can be monitored in the Process Monitor page.

> **Note:**
>
> For a more comprehensive coverage of configuration and execution of a process, see the section Executing Run through Process Modelling Framework in OFSDF and the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

Access the execution log on the application server in the following directory:

```
ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA
```

The file name contains the batch execution ID. The error log table in the atomic schema is as follows:

- FCT_PLACED_COLLATERAL$
- FCT_ACCT_PLACED_COLL_MAP$

27.5 Retrieving the Placed Collateral Result T2T Definitions

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

28

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.

28.1 Overview of the 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

28.2 Overview of the Credit Score Model and Probability of Default Model Population

Table 28-1 Credit Score Model and Probability of Default Model T2T Definitions

Source Table Name	Target Table Name	T2T Name
STG_ACCT_CREDIT_SCORE_DETAILS	FCT_ACCT_CREDIT_SCORE_DETAILS	T2T_FCT_ACCT_CREDIT_SCORE_DET AILS
STG_SERV_ACCT_CREDIT_SCORE_DTL	FCT_SERV_ACCT_CREDIT_SCORE_DTL	T2T_FCT_SERV_ACCT_CREDIT_SCORE_DTL
STG_PARTY_PD_DETAILS	FCT_PARTY_PD_DETAILS	T2T_FCT_PARTY_PD_DETAILS
STG_INSTRUMENT_PD_DETAILS	FCT_INSTRUMENT_PD_DETAILS	T2T_FCT_INSTRUMENT_PD_DETAILS

28.2.1 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-1need 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.

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

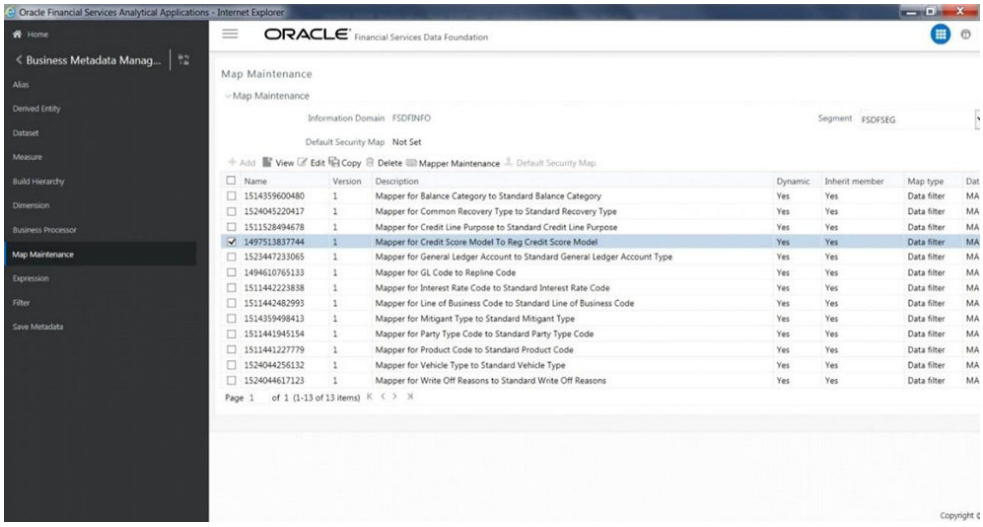
Credit Score Model dimension table is a slowly changing dimension (SCD), which has user-specific values for available credit score models. Regulatory Credit Score Model dimension table is a seeded dimension, which has the reporting specific values for regulatory reporting requirements. There is one column 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.

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

Mapper can be maintained under OFSAAI.

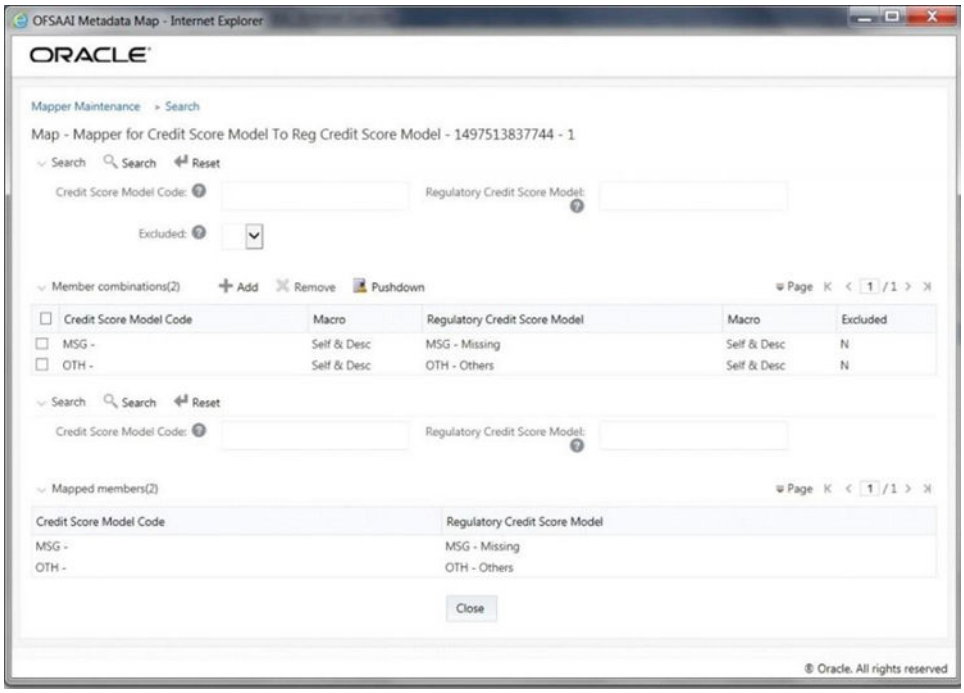
1. Navigate to OFSAAI, select Financial Services data Foundation, select Unified Analytical Metadata, select Business Metedata Management, select Map Maintenance. Select Mapper for Credit Score Model to Regulatory Credit Score Model. Click the Mapper Maintenance icon.

Figure 28-1 Navigate to the Map Maintenance page and select the Mapper



2. The Mapper Maintenance page is displayed. 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.

Figure 28-2 Mapper Maintenance page



28.4.1 Prerequisites for Mapper Maintenance

Load Credit Score Model Dimension using SCD.

Resave the following hierarchies:

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

28.4.2 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 are duplicated. The possible combinations for Credit Score Model to Regulatory Credit Score Model mapping are One to One and Many to One mappings.

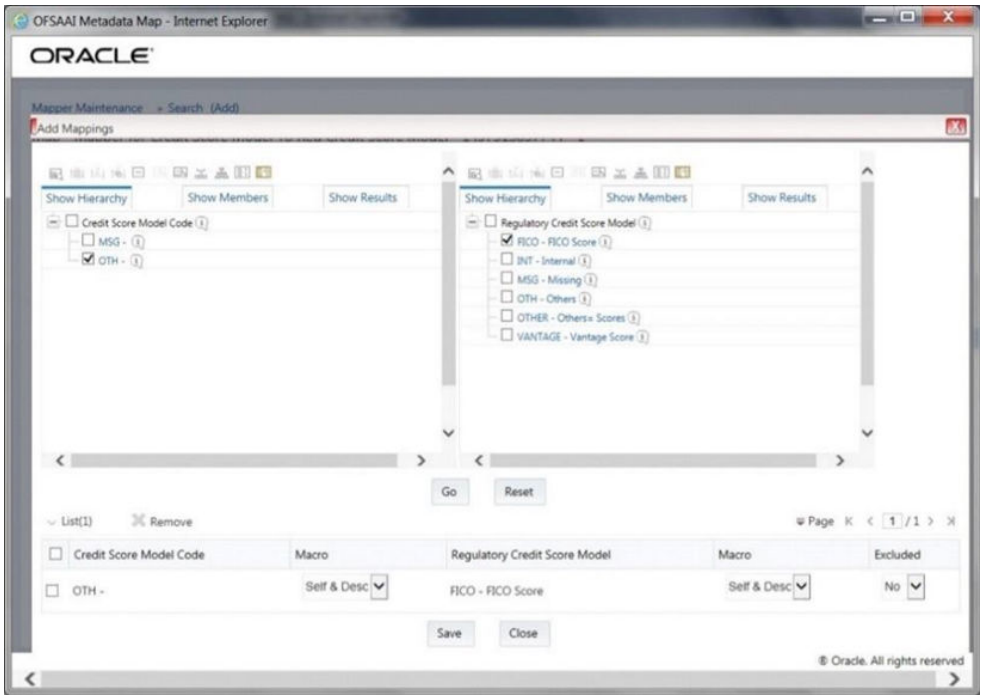
28.4.2.1 One to One Mapping

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 when loading the T2T.

To perform One-to-One mapping, follow these steps:

1. In the Mapper Maintenance page, click Add.
2. The Add Mappings page is displayed.

Figure 28-3 Add Mappings page for the One-to-One Mapping



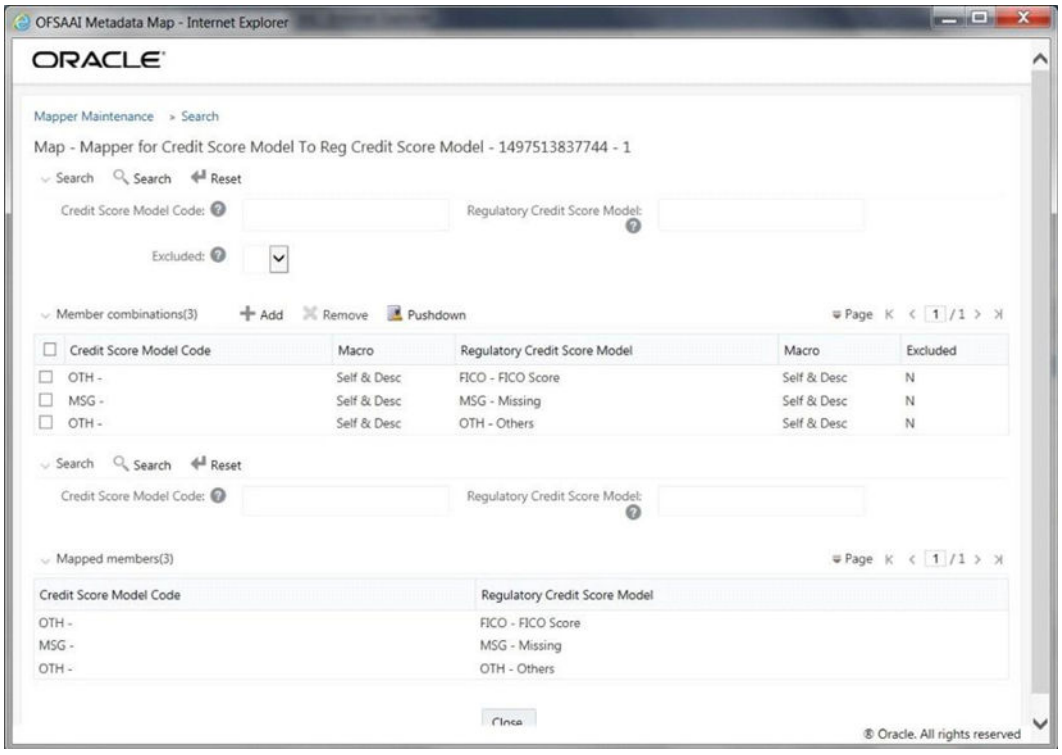
In this example, OTH - Others is mapped to FICO - FICO Score. To map, click Go and then to save the mapping, click Save.

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

Figure 28-4 Acknowledegment to confirm saving the added mappings

4. The mapped members are displayed in the Mapper Maintenance page.

Figure 28-5 Added mappings listed in the Mapper Maintenance page



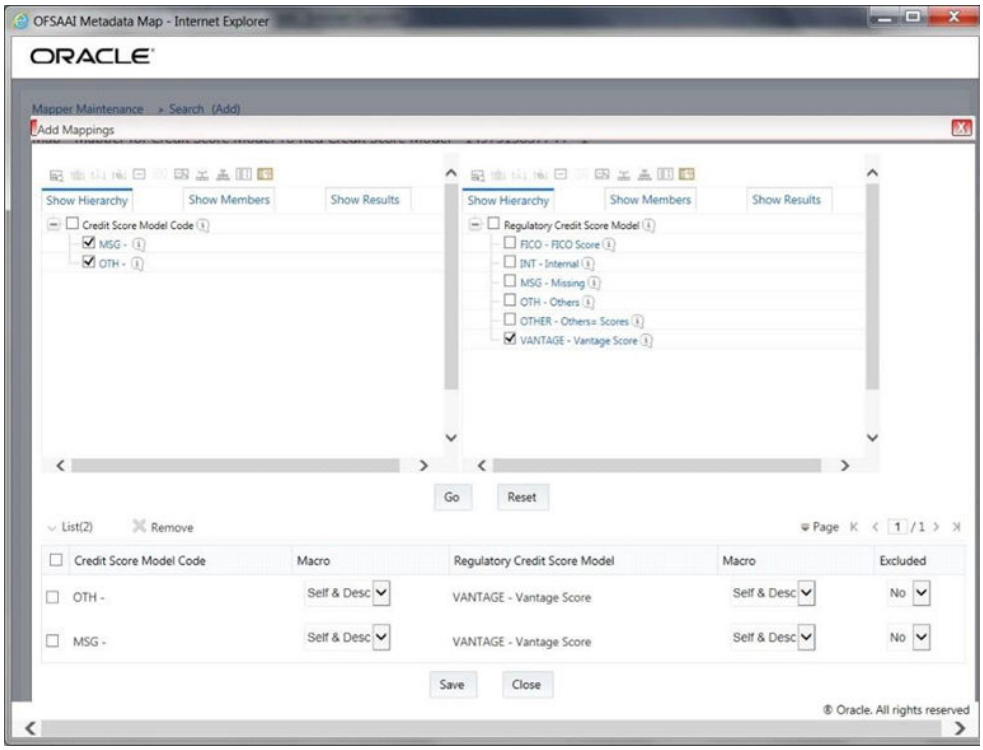
28.4.2.2 Many to One Mapping

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.

To perform Many-to-One mapping, follow these steps:

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

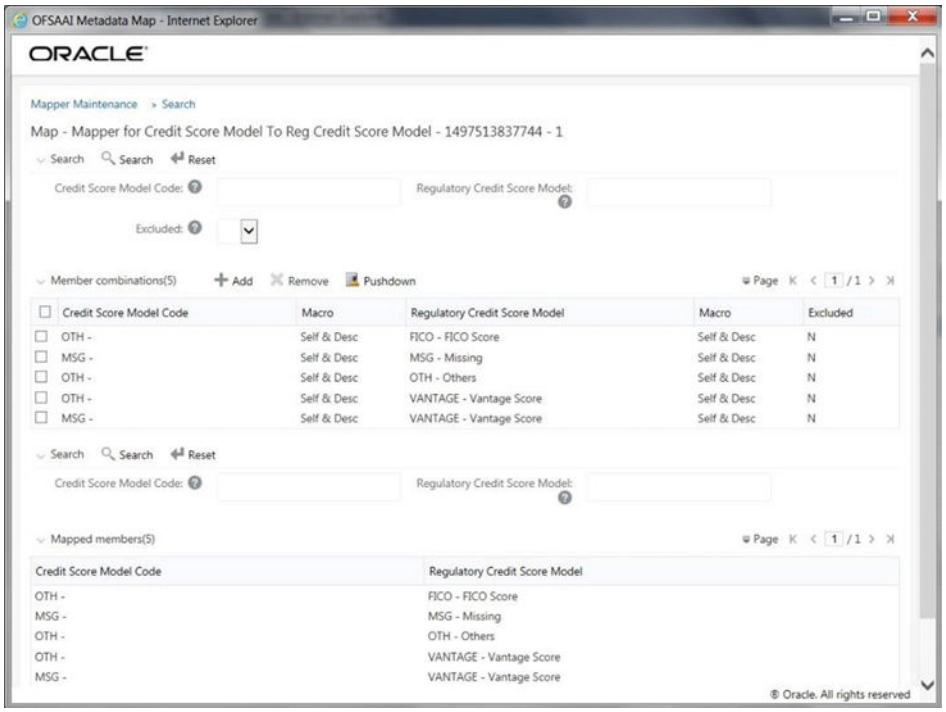
Figure 28-6 Add Mappings page for the Many-to-One Mapping



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

3. The mapped members are displayed in the Mapper Maintenance page.

Figure 28-7 Added mappings listed in the Mapper Maintenance page



28.5 Loading the 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.

Figure 28-8 Loading the Mapper Maintenance from backend

28.6 Executing the Credit Score Model and Probability of Default Model T2Ts using PMF

The T2Ts are a part of Financial Services Data Foundation Sourced Run. The process can be executed through the out-of-the-box standard Run, which is Financial Services Data Foundation Execution Run. For more information, see the section Executing Run through Process Modelling Framework in OFSDF.

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

28.6.1 Error Messages

The following most common error message may be logged in the T2T log file directory `ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA`:

```
Unique Constraint Violation
```

This error occurs after there is an attempt to re?load or load the existing records for the already executed AS_OF_DATE.

28.7 Checking the Execution Status

The status of execution can be monitored in the Process Monitor page.

> **Note:**
>
> For a more comprehensive coverage of configuration and execution of a process, see the section Executing Run through Process Modelling Framework in OFSDF and the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

Access the execution log on the application server in the following directory:

```
ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA
```

The file name contains the batch execution ID. The error log table in the atomic schema is as follows:

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

28.8 Retrieving the Credit Score Model and Probability of Default Model 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.

29
GL to Management Reporting

This chapter provides information about GL to Management Reporting tables in the Oracle Financial Services Data Foundation application.

29.1 Overview of the GL to Management Reporting Result Table

To load the GL to Management Reporting Result Table, which is the Fact Management Reporting table, from the respective Stage table, use the Table-to-Table (T2T) component and the Map Maintenance component of the Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework. The Result table that stores GL to Management Reporting data is:

- FCT_MGMT_REPORTING

29.2 Populating the GL to Management Reporting Result Table

To load data into the target table, the Table-to-Table seeded definition is as follows.

Table 29-1 GL to Management Reporting Result T2T Definition

Source Table Name	Mapper Table Name	Target Table Name	T2T Name
STG_GL_DATA	MAP_GL_CODE_REP_LINE	FCT_MGMT_REPORTING	T2T_FCT_MGMT_REPORTING_STG_GL_DATA

29.3 Overview of the Mapper for GL to Management Reporting Result Table

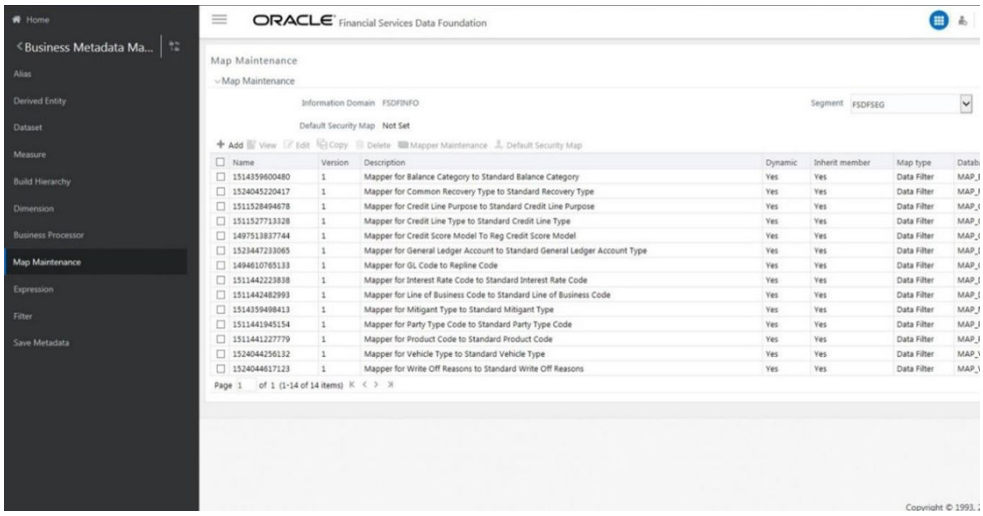
The Fact Management Reporting table uses reporting line codes, which are OFSAA specific values that represent each line item of reporting requirements. On the other hand, the Stage GL Data table uses the General Ledger Codes. These General Ledger Codes are user specific values that represent each General Ledger Code of the user data records. Map GL Codes to Reporting Line Codes and then load the Fact Management Reporting target table. You can maintain these mappings through the Map Maintenance component of OFSAAI.

29.4 Maintaining the Mappers for GL to Management Reporting Result Table

To maintain Mappers through the Map Maintenance component of OFSAAI:

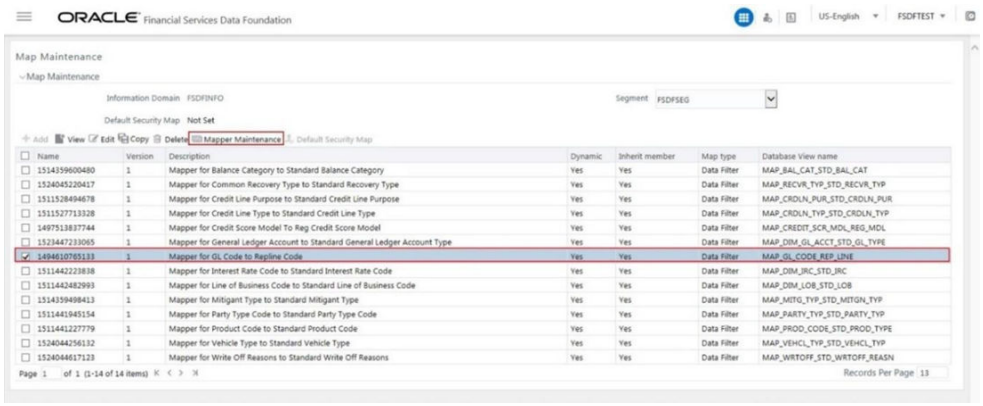
1. Navigate to OFSAAI, select Financial Services data Foundation, select Unified Analytical Metadata, select Business Metedata Management, select Map Maintenance. The Map Maintenance page is displayed.

Figure 29-1 Navigate to the Map Maintenance page



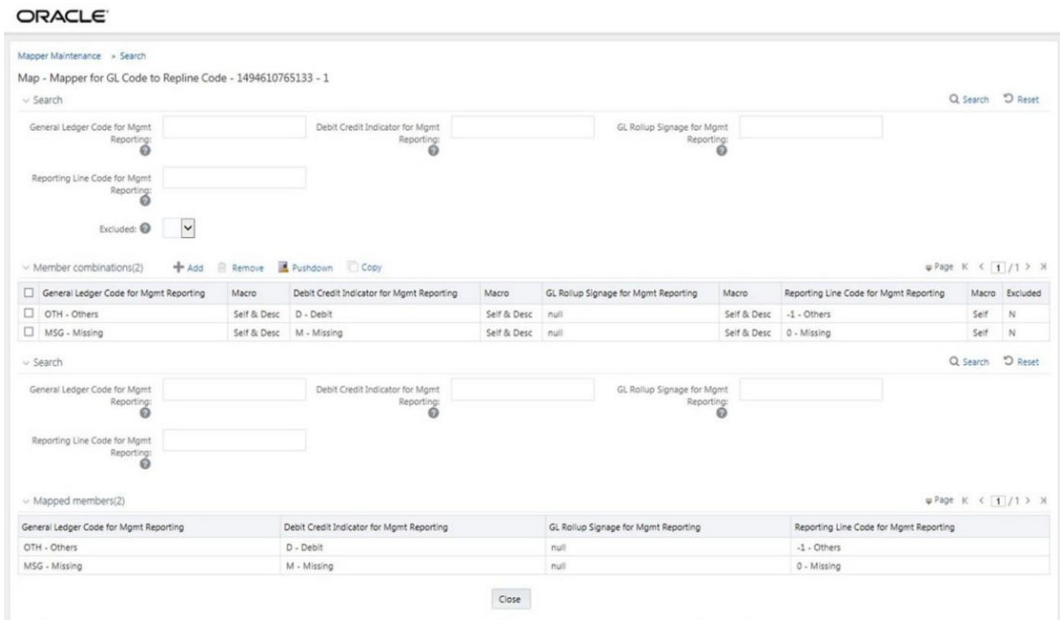
2. Select the mapper record Mapper for GL Code to Repline Code and then click the Mapper Maintenance icon.

Figure 29-2 Select the Mapper in the Map Maintenance page



3. The Mapper Maintenance page is displayed. The GL codes OTH and MSG are out-of-the-box mappings. You can maintain the remaining mappings according to the user specific values.

Figure 29-3 Mapper Maintenance page



29.4.1 Prerequisites for Mapper Maintenance

Perform these prerequisite steps for maintaining Mappers:

1. Through the SCD process, load the GL Dimension table.
2. In the Metadata Resave section, resave these hierarchies:
   - HMGMT001 - General Ledger Code for Mgmt Reporting
   - HMGMT002 - Debit Credit Indicator for Mgmt Reporting
   - HMGMT003 - GL Rollup Signage for Mgmt Reporting
   - HMGMT004 - Reporting Line Code for Mgmt Reporting

29.4.2 Possible Mapping Combinations

There are four types of mapping combinations. They are:

Mapping combinations at child hierarchy level

Mapping combinations at parent and child hierarchy level

Mapping combinations at parent hierarchy level without Descendants

Mapping combinations at parent hierarchy level by removing one or more Descendants

29.4.2.1 Mapping Combinations at Child Hierarchy Level

The mapping combinations for the General Ledger Code for Mgmt Reporting hierarchy at child member level (individual nodes) are as follows:

- One-to-One mapping with or without Debit Credit Indicator
- Many-to-One mapping with or without Debit Credit Indicator

- Many-to-Many mapping with or without Debit Credit Indicator

The procedures for mapping combinations are as follows.

29.4.2.1.1 One-to-One mapping with or without Debit Credit Indicator

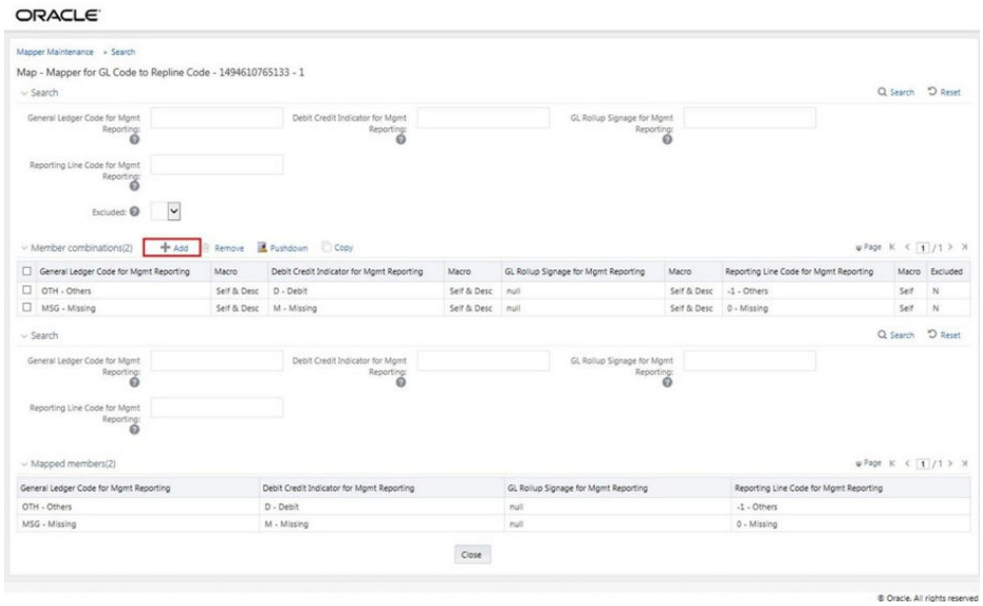
To map one GL Code to one Reporting Line Code, use the Mapper Maintenance utility. In this utility, select one member in the General Ledger Code for Mgmt Reporting hierarchy, select one member in the Debit Credit Indicator hierarchy with the value Debit or Credit or Missing, and select one member in the Reporting Line Code hierarchy. As a result, during the T2T process, the corresponding data from the Stage GL Data table for the selected GL Code loads into the Fact Management Reporting table.

> **Note:**
>
> The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:
>
> - The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
> - The value must be D, when you map GL Code to the Debit member.
> - The value must be C, when you map GL Code to the Credit member.

To perform One-to-One mapping with or without Debit Credit Indicator, follow these steps:

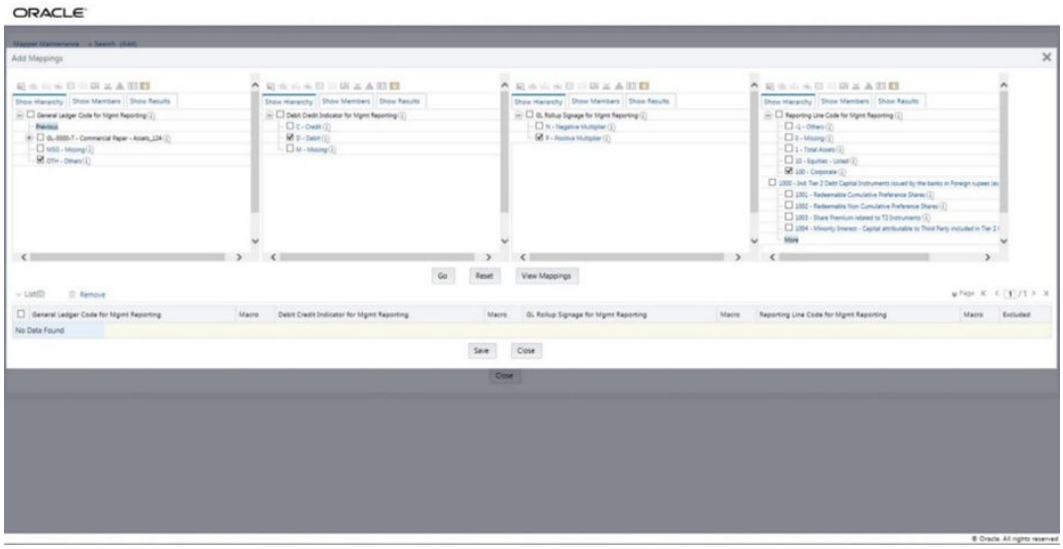
1. In the Mapper Maintenance page, in the Member combinations section, click Add.

Figure 29-4 Select Add in the Mapper Maintenance page for the One-to-One mapping at child hierarchy level



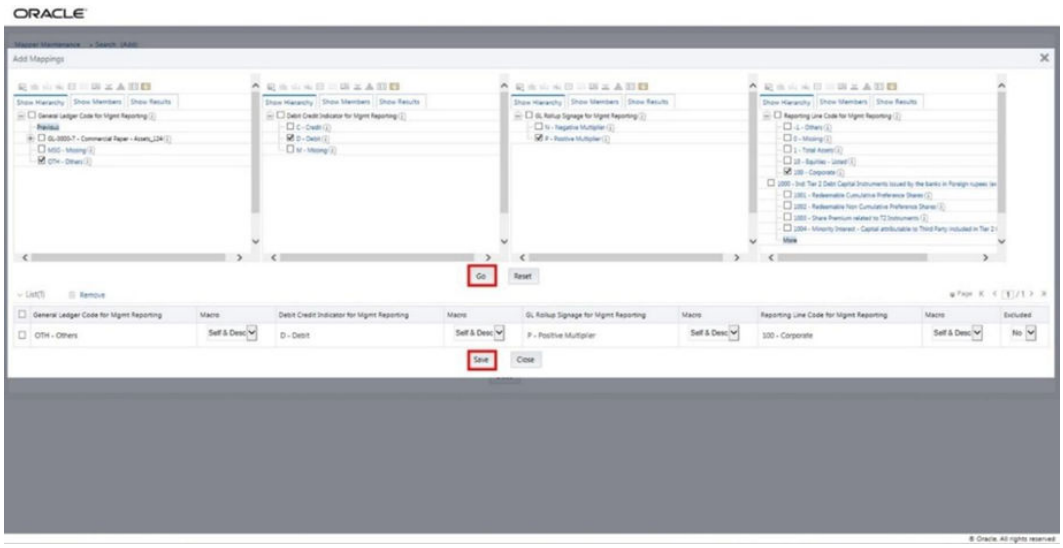
2. The Add Mappings page is displayed. For illustration, select the member of the General Ledger Code for Mgmt Reporting hierarchy OTH - Others, and the member of the Debit Credit Indicator for Mgmt Reporting hierarchy D - Debit to map to the member of the Reporting Line Code for Mgmt Reporting hierarchy 100 - Corporate with the member of the GL Rollup Signage for Mgmt Reporting hierarchy P - Positive Multiplier.

Figure 29-5 Add Mappings page for the One-to-One mapping at child hierarchy level



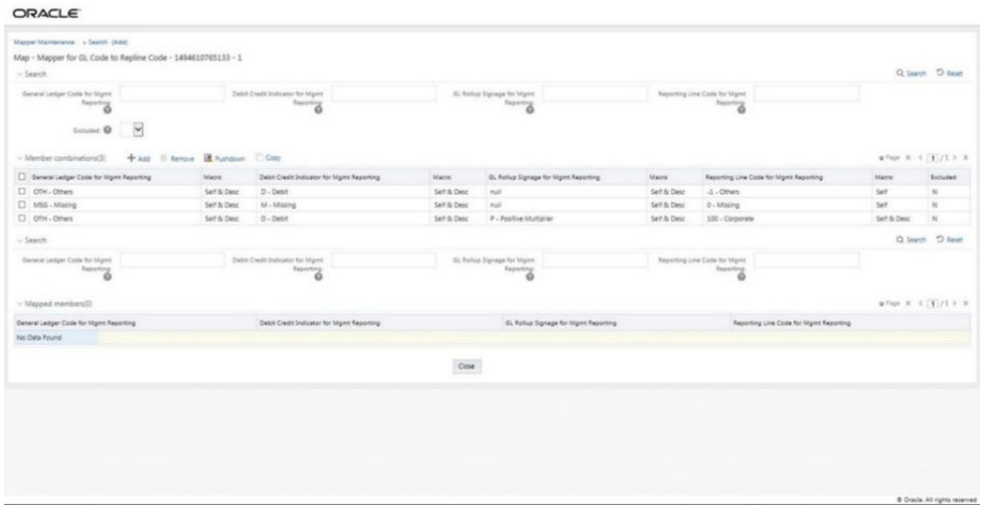
3. To map the members, click Go. The list of mapped members appear at the bottom. To save the mappings, click **Save**.

Figure 29-6 Map the members and save the mappings



4. An acknowledgement message is displayed. To confirm saving the mapping, click **Yes**.
5. The mapped member combinations are listed in the **Mapper Maintenance** page.

Figure 29-7 Added mappings listed in the Mapper Maintenance page



29.4.2.1.2 Many-to-One mapping with or without Debit Credit Indicator

To map multiple GL Codes to one Reporting Line Code, use the Mapper Maintenance utility. In this utility, select two or more members in the General Ledger Code for Mgmt Reporting hierarchy, select one member in the Debit Credit Indicator hierarchy with the value Debit or Credit or Missing, and select one member in the Reporting Line Code hierarchy. As a result, during the T2T process, the corresponding data from the Stage GL Data table, for the selected GL Codes, are aggregated and then loaded into the Fact Management Reporting table.

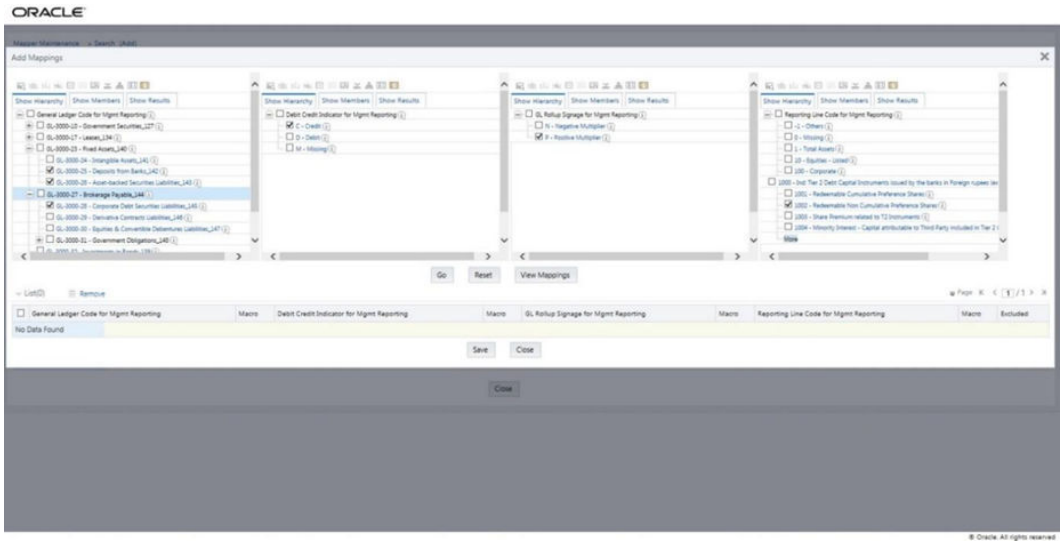
> **Note:**
>
> In the T2T process, in the Stage GL Data table, the GL Code aggregation takes place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:
>
> - The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
> - The value must be D, when you map GL Code to the Debit member.
> - The value must be C, when you map GL Code to the Credit member.

To perform Many-to-One mapping with or without Debit Credit Indicator, follow these steps:

1. In the **Mapper Maintenance** page, in the Member combinations section, click **Add**.
2. The **Add Mappings** page is displayed. For illustration, select the child members of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-25 - Deposits from Banks_142, GL-3000-26 – Asset-backed Securities Liabilities_143 and GL-3000-28 - Corporate Debt Securities Liabilities_145, and the member of the Debit Credit Indicator for Mgmt Reporting hierarchy, C - Credit to map to the Reporting Line Code hierarchy member 1002 - Redeemable Non Cumulative

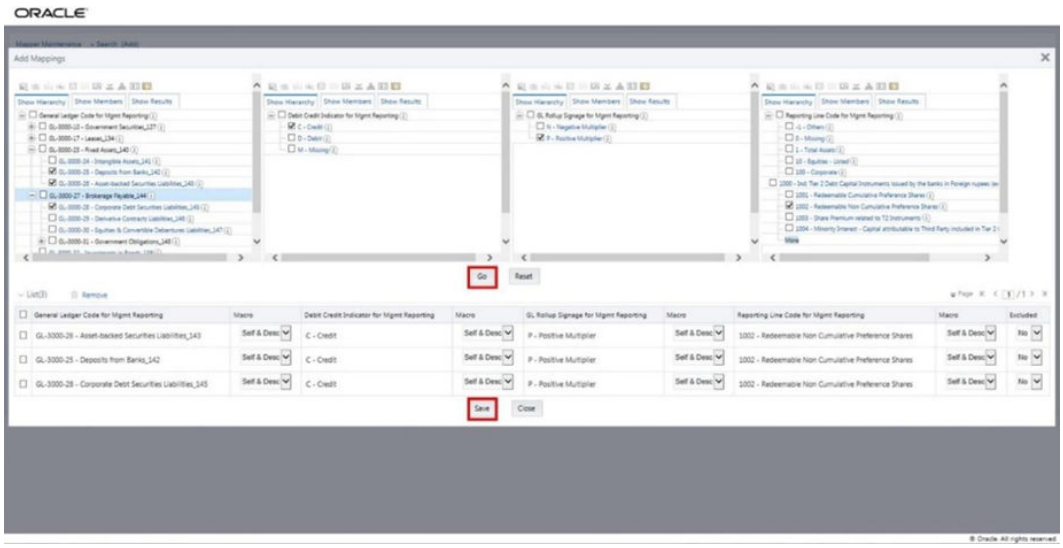
Preference Shares with the GL Rollup Signage hierarchy member P - Positive Multiplier.

Figure 29-8 Add Mappings page for the Many-to-One mapping at child hierarchy level



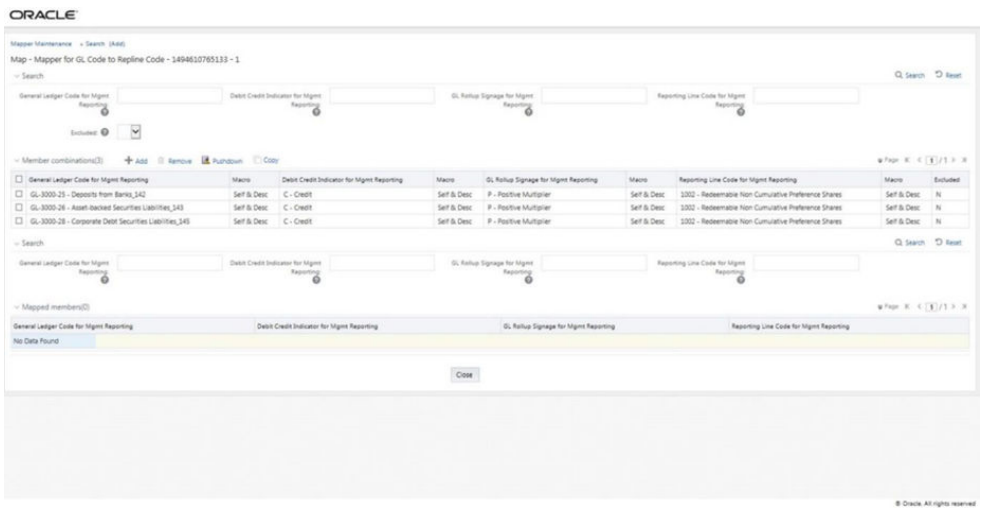
3. To map the members, click Go. The list of mapped members appear at the bottom. To save the mappings, click Save.

Figure 29-9 Map the members and save the mappings



4. An acknowledgement message is displayed. To confirm saving the mappings, click **Yes**.
5. The mapped member combinations are listed in the **Mapper Maintenance** page.

Figure 29-10 Added mappings listed in the Mapper Maintenance page



29.4.2.1.3 Many-to-One mapping with or without Debit Credit Indicator

To map multiple GL Codes to one Reporting Line Code, use the Mapper Maintenance utility. In this utility, select two or more members in the General Ledger Code for Mgmt Reporting hierarchy, select one member in the Debit Credit Indicator hierarchy with the value Debit or Credit or Missing, and select one member in the Reporting Line Code hierarchy. As a result, during the T2T process, the corresponding data from the Stage GL Data table, for the selected GL Codes, are aggregated and then loaded into the Fact Management Reporting table.

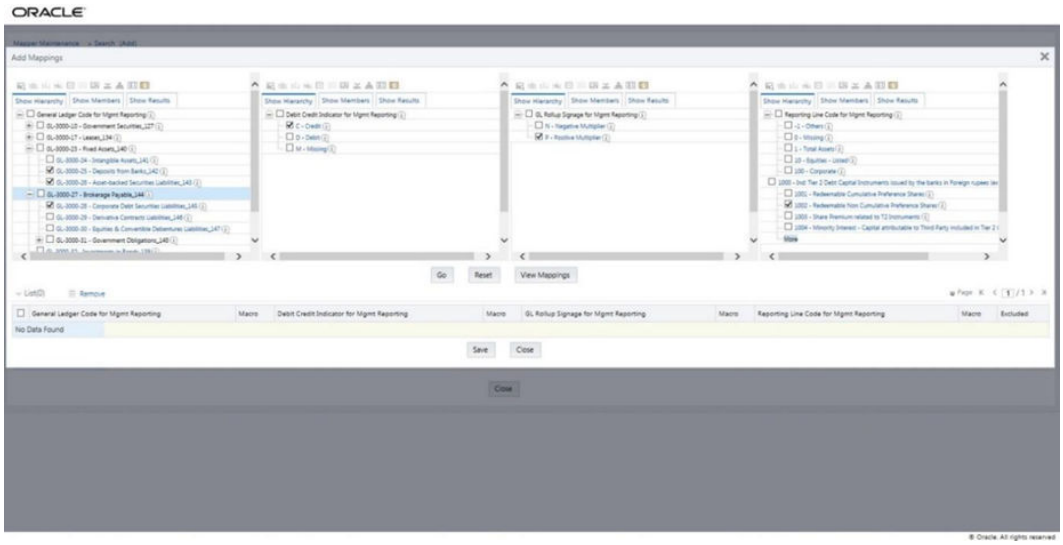
> **Note:**
>
> In the T2T process, in the Stage GL Data table, the GL Code aggregation takes place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:
>
> - The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
> - The value must be D, when you map GL Code to the Debit member.
> - The value must be C, when you map GL Code to the Credit member.

To perform Many-to-One mapping with or without Debit Credit Indicator, follow these steps:

1. In the **Mapper Maintenance** page, in the Member combinations section, click **Add**.
2. The **Add Mappings** page is displayed. For illustration, select the child members of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-25 - Deposits from Banks_142, GL-3000-26 – Asset-backed Securities Liabilities_143 and GL-3000-28 - Corporate Debt Securities Liabilities_145, and the member of the Debit Credit Indicator for Mgmt Reporting hierarchy, C - Credit to map to the Reporting Line Code hierarchy member 1002 - Redeemable Non Cumulative

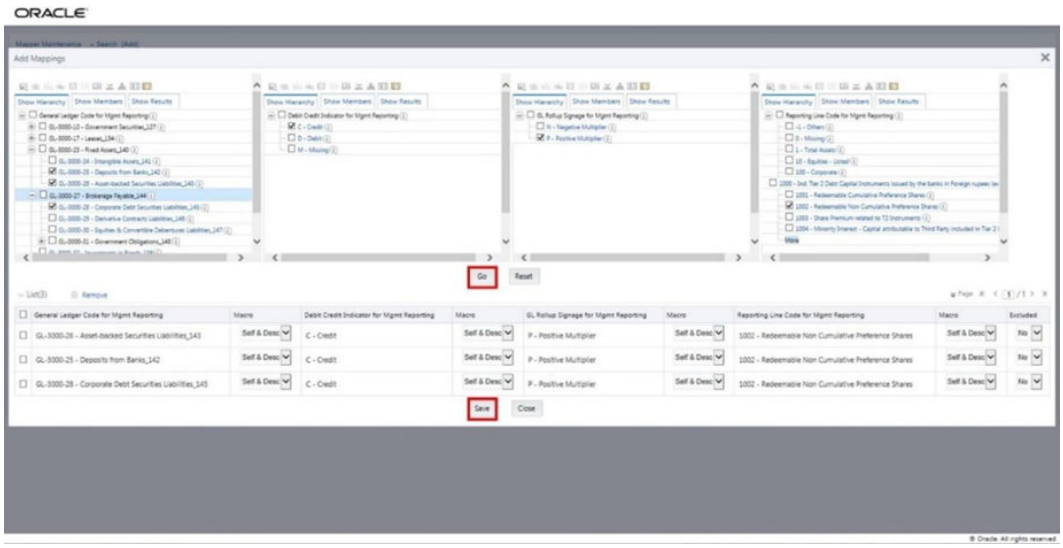
Preference Shares with the GL Rollup Signage hierarchy member P - Positive Multiplier.

Figure 29-11 Add Mappings page for the Many-to-One mapping at child hierarchy level



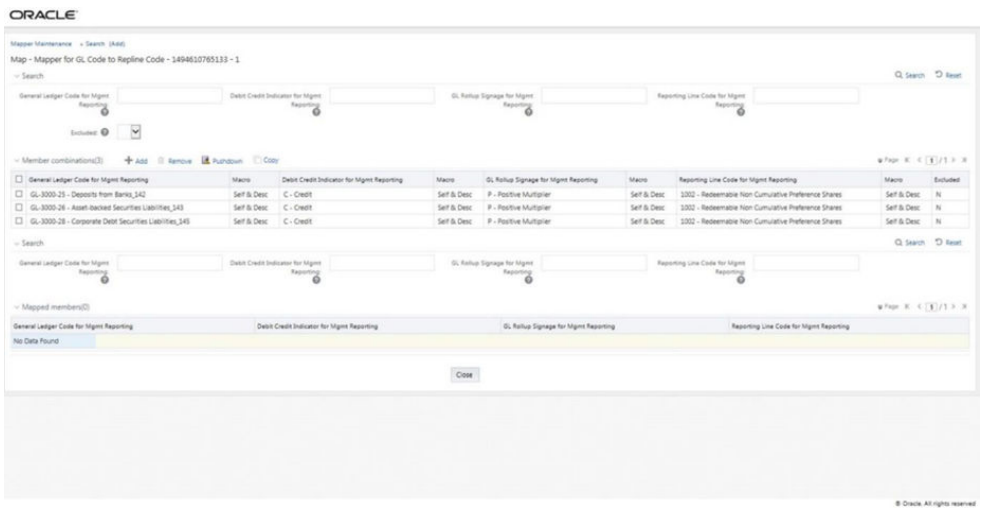
3. To map the members, click **Go**. The list of mapped members appear at the bottom. To save the mappings, click **Save**.

Figure 29-12 Map the members and save the mappings



4. An acknowledgement message is displayed. To confirm saving the mappings, click **Yes**.
5. The mapped member combinations are listed in the **Mapper Maintenance** page.

Figure 29-13 Added mappings listed in the Mapper Maintenance page



29.4.2.2 Mapping Combinations at Parent and Child Hierarchy Level

The mapping combinations for the General Ledger Code for Mgmt Reporting hierarchy maintained as Parent-Child hierarchy are as follows:

One Parent to One Reporting Line Code mapping with or without Debit Credit Indicator

Many Parents to One Reporting Line Code mapping with or without Debit Credit Indicator

Many Parents to Many Reporting Line Codes mapping with or without Debit Credit Indicator

The procedures for mapping combinations are as follows.

29.4.2.2.1 One Parent to One Reporting Line Code mapping with or without Debit Credit Indicator

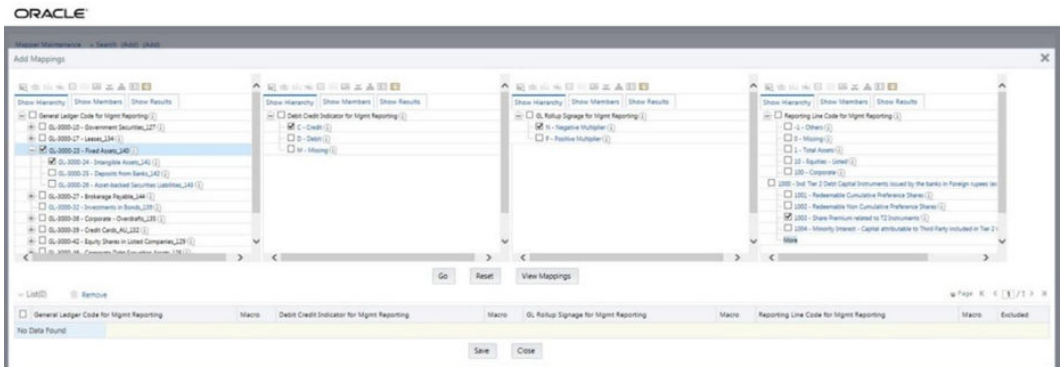
To map one parent GL Code to one Reporting Line Code, use the Mapper Maintenance utility. In this utility, select one parent GL Code, select one member in the Debit Credit Indicator hierarchy with the value Debit or Credit or Missing, and select one member in the Reporting Line Code hierarchy. As a result, during the T2T process, the corresponding data from the Stage GL Data table, for the selected parent GL Codes and their descendants, are aggregated and then loaded into the Fact Management Reporting table.

> **Note:**
>
> In the T2T process, in the Stage GL Data table, the aggregation of parent GL Codes and their descendants take place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:
>
> - The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
> - The value must be D, when you map GL Code to the Debit member.
> - The value must be C, when you map GL Code to the Credit member.

To perform One Parent to One Reporting Line Code mapping with or without Debit Credit Indicator, follow these steps:

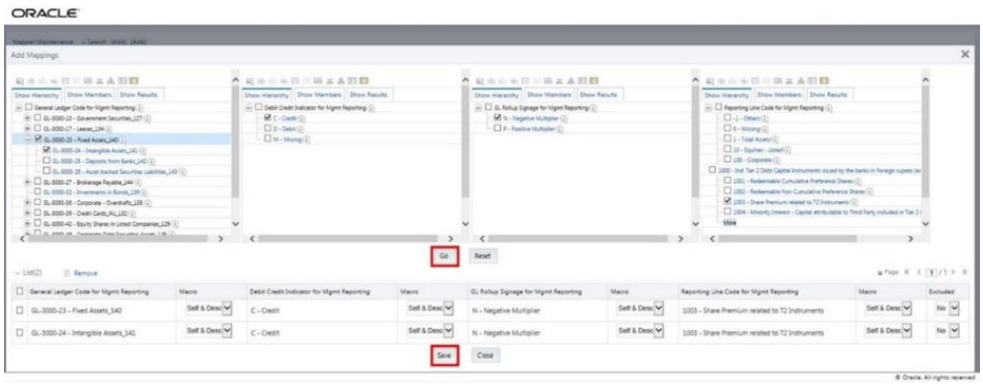
1. In the Mapper Maintenance page, in the Member combinations section, click **Add**.
2. The **Add Mappings** page is displayed. For illustration, select one parent member of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-23 – Fixed Assets_140 and its child member GL-3000-24 – Intangible Assets_141, and the member of the Debit Credit Indicator for Mgmt Reporting hierarchy C - Credit to map to the member of the Reporting Line Code for Mgmt Reporting hierarchy 1003 – Share Premium related to T2 Instruments with the member of the GL Rollup Signage for Mgmt Reporting hierarchy **N - Negative Multiplier**.

Figure 29-14 Add Mappings page for the one parent to one Reporting Line Code mapping at parent and child hierarchy level



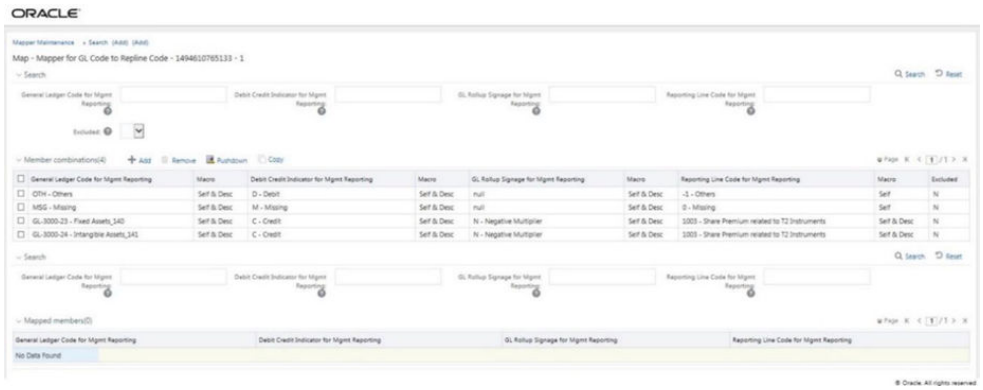
3. To map the members, click **Go**. The list of mapped members appear at the bottom. To save the mappings, click **Save**.

Figure 29-15 Map the members and save the mappings



4. An acknowledgement message is displayed. To confirm saving the mappings, click **Yes**.
5. The mapped member combinations are listed in the **Mapper Maintenance** page.

Figure 29-16 Added mappings listed in the Mapper Maintenance page



29.4.2.2.2 Many Parents to One Reporting Line Code with or without Debit Credit Indicator

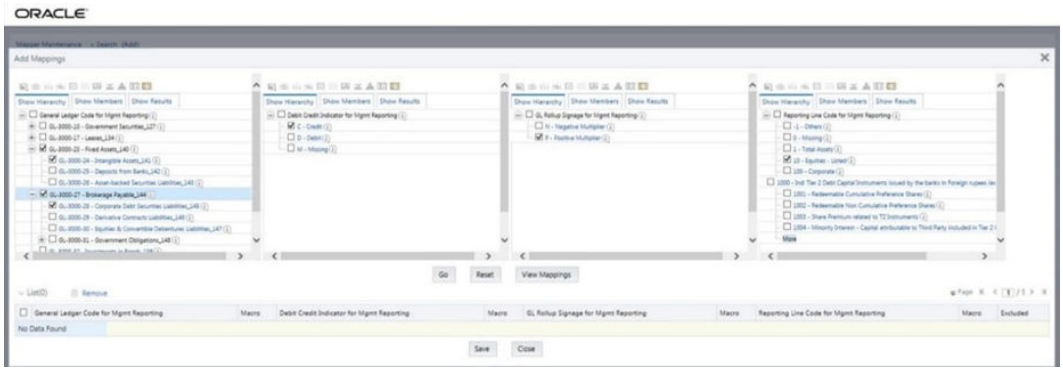
To map multiple parent GL Codes to one Reporting Line Code, use the Mapper Maintenance utility. In this utility, select two or more parent GL Codes, select one member in the Debit Credit Indicator hierarchy with the value Debit or Credit or Missing, and select one member in the Reporting Line Code hierarchy. As a result, during the T2T process, the corresponding data from the Stage GL Data table, for the selected parent GL Codes and their descendants, are aggregated and then loaded into the Fact Management Reporting table.

> **Note:**
>
> In the T2T process, in the Stage GL Data table, the aggregation of parent GL Codes and their descendants take place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:
>
> - The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
> - The value must be D, when you map GL Code to the Debit member.
> - The value must be C, when you map GL Code to the Credit member.

To perform Many Parents to One Reporting Line Code mapping with or without Debit Credit Indicator, follow these steps:

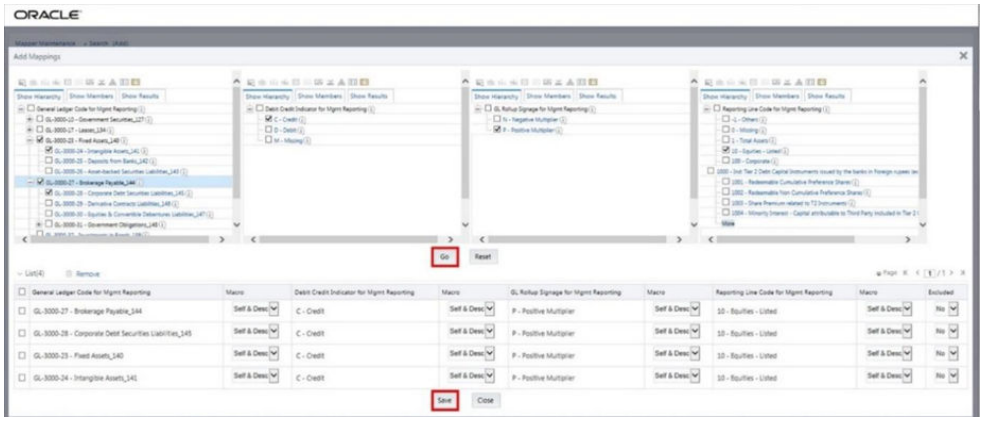
1. In the **Mapper Maintenance** page, in the Member combinations section, click **Add**.
2. The Add Mappings pop-up page appears. For illustration, select parent members of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-23 – Fixed Assets_140 and GL-3000-27 – Brokerage Payable_144, and their child members GL-3000-24 – Intangible Assets_141 and GL- 3000-28 – Corporate Debt Securities Liabilities_145 respectively, and the member of the Debit Credit Indicator for Mgmt Reporting hierarchy C - Credit to map to the member of the Reporting Line Code for Mgmt Reporting hierarchy 10 – Equities - Listed with the member of the GL Rollup Signage for Mgmt Reporting hierarchy P - Positive Multiplier.

Figure 29-17 Add Mappings page for the many parents to one Reporting Line Code mapping at parent and child hierarchy level



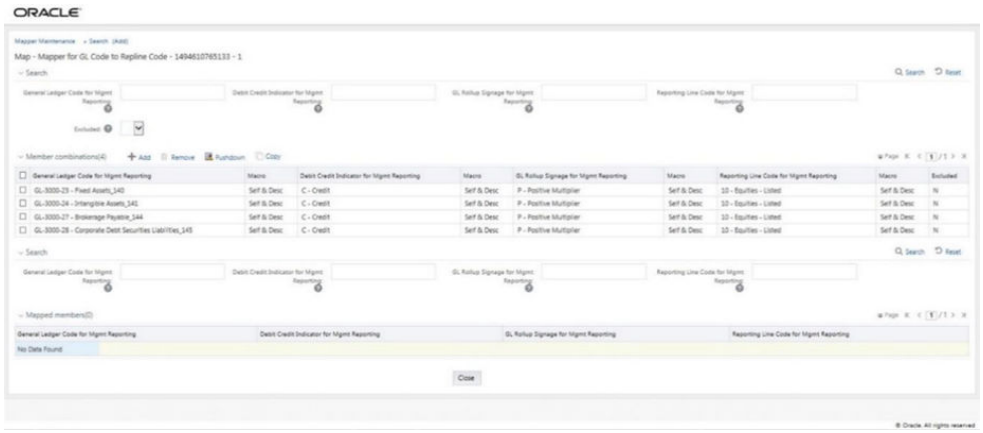
3. To map the members, click **Go**. The list of mapped members appear at the bottom. To save the mappings, click **Save**.

Figure 29-18 Map the members and save the mappings



4. An acknowledgement message is displayed. To confirm saving the mappings, click **Yes**.
5. The mapped member combinations are listed in the **Mapper Maintenance** page.

Figure 29-19 Added mappings listed in the Mapper Maintenance page



29.4.2.2.3 Many Parents to Many Reporting Line Codes with or without Debit Credit Indicator

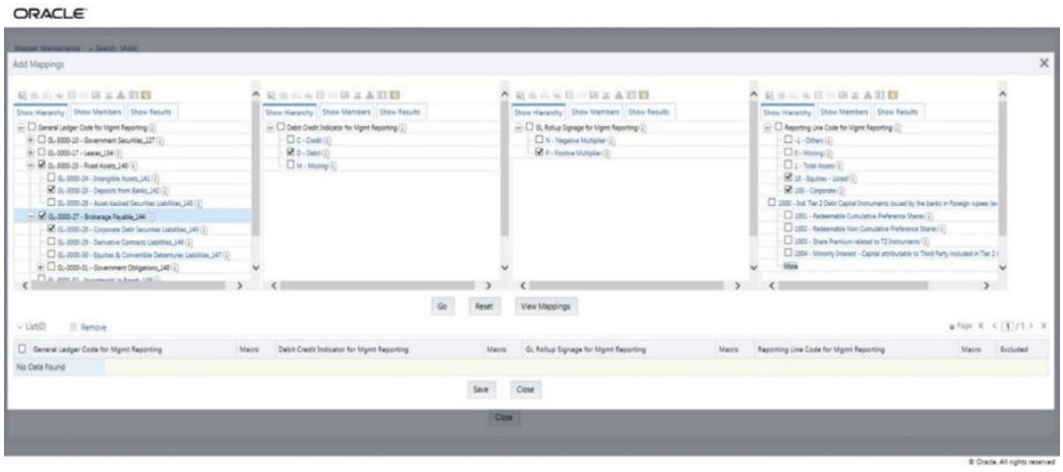
To map multiple parent GL Codes to multiple Reporting Line Codes, use the Mapper Maintenance utility. In this utility, select two or more parent GL Codes, select one member in the Debit Credit Indicator hierarchy with the value Debit or Credit or Missing, and select two or more members in the Reporting Line Code hierarchy. As a result, during the T2T process, the corresponding data from the Stage GL Data table, for the selected parent GL Codes and their descendants, are aggregated and then loaded into the Fact Management Reporting table.

> **Note:**
>
> In the T2T process, in the Stage GL Data table, the aggregation of parent GL Codes and their descendants take place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:
>
> - The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
> - The value must be D, when you map GL Code to the Debit member.
> - The value must be C, when you map GL Code to the Credit member.

To perform Many Parents to Many Reporting Line Codes mapping with or without Debit Credit Indicator, follow these steps:

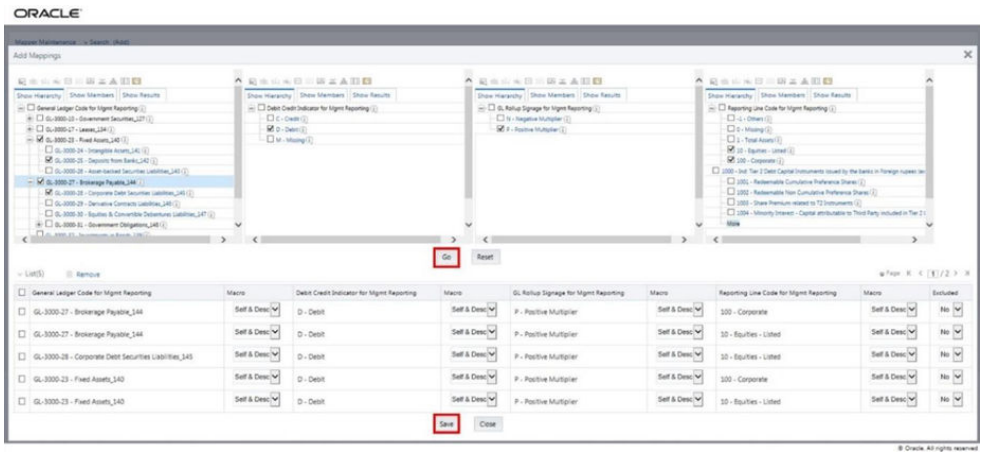
1. In the **Mapper Maintenance** page, in the Member combinations section, click **Add**.
2. The Add Mappings page is displayed. For illustration, select parent members of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-23 – Fixed Assets_140 and GL-3000-27 – Brokerage Payable_144, and their child members GL-3000-24 – Intangible Assets_141 and GL- 3000-28 – Corporate Debt Securities Liabilities_145 respectively, and the member of the Debit Credit Indicator for Mgmt Reporting hierarchy C - Credit to map to the members of the Reporting Line Code for Mgmt Reporting hierarchy 10 – Equities - Listed and 100 – Corporate with the member of the GL Rollup Signage for Mgmt Reporting hierarchy P - Positive Multiplier.

Figure 29-20 Add Mappings page for the many parents to many Reporting Line Codes mapping at parent and child hierarchy level



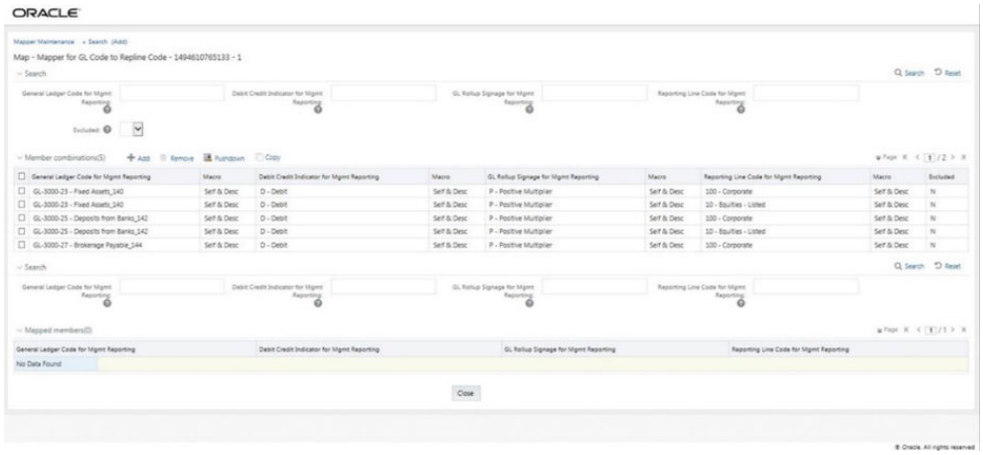
3. To map the members, click **Go**. The list of mapped members appear at the bottom. To save the mappings, click **Save**.

Figure 29-21 Map the members and save the mappings



4. An acknowledgement message is displayed. To confirm saving the mappings, click **Yes**.
5. The mapped member combinations are listed in the Mapper Maintenance page.

Figure 29-22 Added mappings listed in the Mapper Maintenance page



29.4.2.3 Mapping Combinations at Parent Hierarchy Level without Descendants

This section explains about the mapping combinations for the General Ledger Code for Mgmt Reporting hierarchy at Parent level hierarchy without Descendants (child members).

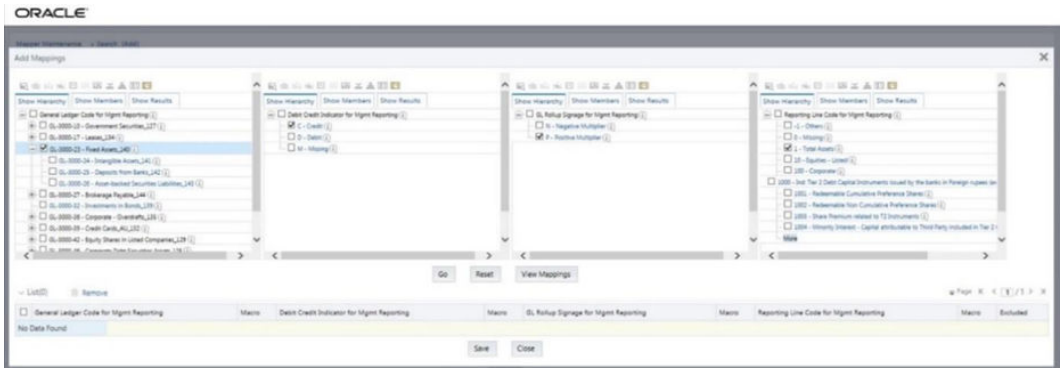
To map one parent GL Code without Descendants to one or more Reporting Line Codes, use the Mapper Maintenance utility. In this utility, select one parent GL Code, select one member in the Debit Credit Indicator hierarchy with the value Debit or Credit or Missing, and select one or more members in the Reporting Line Code hierarchy. Then for the GL hierarchy, select Self in the Macro column. As a result, during the T2T process, the corresponding data from the Stage GL Data table, for the selected parent GL Code, is aggregated and then loaded into the Fact Management Reporting table.

> **Note:**
>
> In the T2T process, in the Stage GL Data table, the aggregation of parent GL Code takes place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:
>
> - The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
> - The value must be D, when you map GL Code to the Debit member.
> - The value must be C, when you map GL Code to the Credit member.

To perform One Parent to One or Many Reporting Line Codes mappings without Descendants, follow these steps:

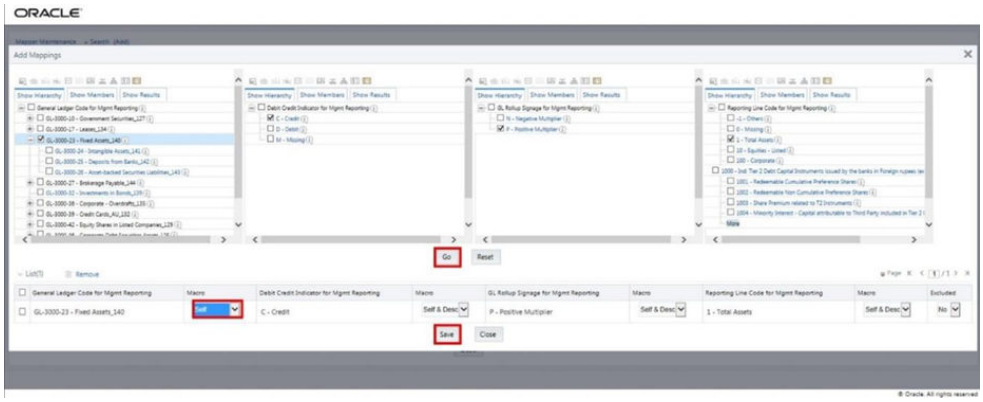
1. In the **Mapper Maintenance** page, in the Member combinations section, click **Add**.
2. The **Add Mappings** page is displayed. For illustration, select one parent member of the General Ledger Code for Mgmt Reporting hierarchy GL-3000-23 – Fixed Assets_140, and the member of the Debit Credit Indicator for Mgmt Reporting hierarchy C - Credit to map to the member of the Reporting Line Code for Mgmt Reporting hierarchy 1 – Total Assets with the member of the GL Rollup Signage for Mgmt Reporting hierarchy P - Positive Multiplier.

Figure 29-23 Add Mappings page for the one parent to one or many Reporting Line Codes mapping without Descendants



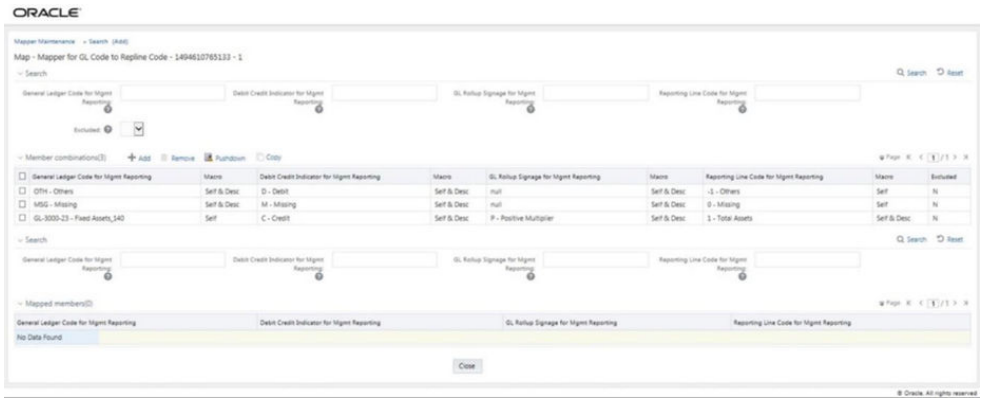
3. To map the members, click **Go**. The list of mapped member appears at the bottom. To exclude the Descendants of the GL Code from the mapping, in the Macro column, select **Self**. To save the mappings, click **Save**.

Figure 29-24 Map the members and save the mappings by excluding Descendants



4. An acknowledgement message is displayed. To confirm saving the mappings, click **Yes**.
5. The mapped member combinations are listed in the **Mapper Maintenance** page with the Macro value of GL Code as **Self**.

Figure 29-25 Added mappings listed in the Mapper Maintenance page



29.4.2.4 Mapping Combinations at Parent Hierarchy Level by Removing one or more Descendants

This section explains about the mapping combinations for the General Ledger Code for Mgmt Reporting hierarchy at Parent level hierarchy by removing one or more Descendants (child members).

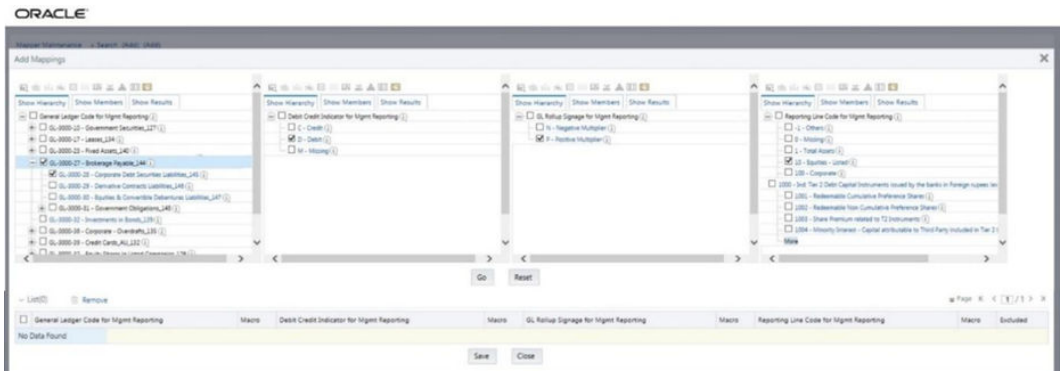
To map one parent GL Code to one or more Reporting Line Codes by removing one or more Descendants, use the Mapper Maintenance utility. In this utility, select one parent GL Code with one or more Descendants, select one member in the Debit Credit Indicator hierarchy with the value Debit or Credit or Missing, and select one or more members in the Reporting Line Code hierarchy. Then for the GL hierarchy, in the Macro column, select Self & Descendants and in the Excluded column, select either Yes to exclude or No to include the Descendants from mapping. As a result, during the T2T process, the corresponding data from the Stage GL Data table, for the selected parent GL Code and its descendants, is aggregated and then loaded into the Fact Management Reporting table.

> **Note:**
>
> In the T2T process, in the Stage GL Data table, the aggregation of parent GL Code and its descendants take place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:
>
> - The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
> - The value must be D, when you map GL Code to the Debit member.
> - The value must be C, when you map GL Code to the Credit member.

To perform One Parent to One or Many Reporting Line Codes mapping by removing one or more GL Code Descendants, follow these steps:

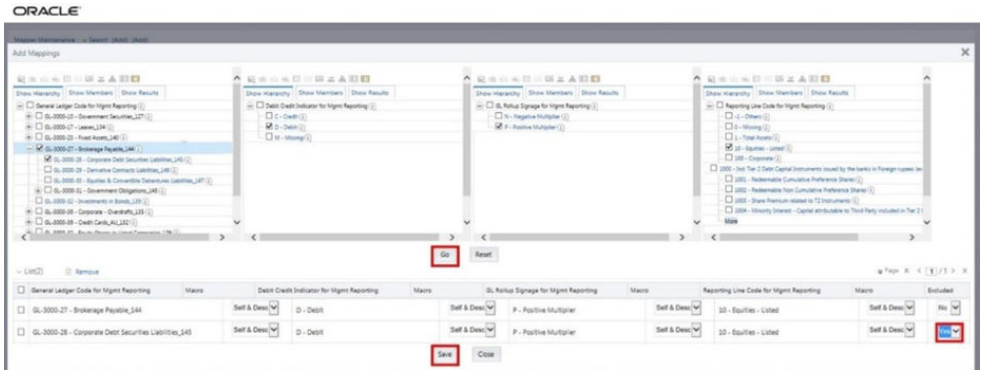
1. In the Mapper Maintenance page, in the Member combinations section, click Add.
2. The Add Mappings page is displayed. For illustration, select one parent member of the General Ledger Code for Mgmt Reporting hierarchy GL-3000-27 – Brokerage Payable_144 and its child member (Descendant) GL-3000-28 – Corporate Debt Securities Liabilities_145, and the member of the Debit Credit Indicator for Mgmt Reporting hierarchy D - Debit to map to the member of the Reporting Line Code for Mgmt Reporting hierarchy 10 – Equities - Listed with the member of the GL Rollup Signage for Mgmt Reporting hierarchy P - Positive Multiplier.

Figure 29-26 Add Mappings page for the mapping combinations at parent hierarchy level by removing one or more Descendants



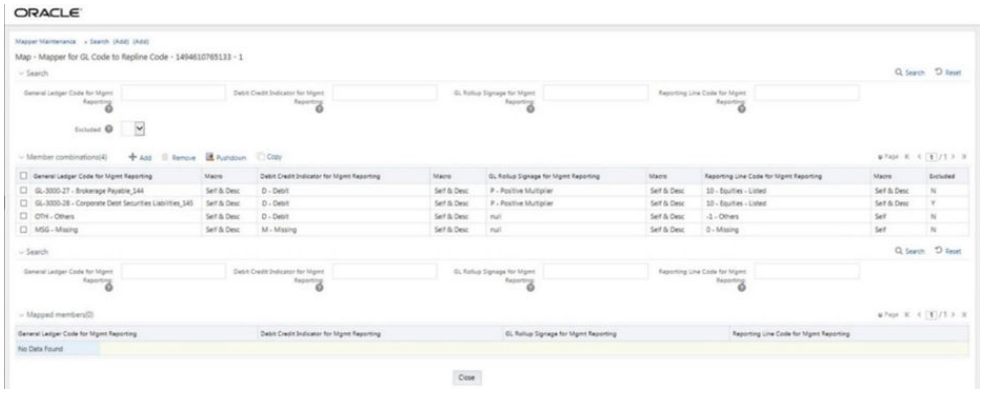
3. To map the members, click Go. The list of mapped members appear at the bottom. To exclude the Descendant of the GL Code from the mapping, in the Macro column, select Self & Descendants, and in the Excluded column, select Yes. To save the mappings, click **Save**.

Figure 29-27 Map the members and save the mappings by excluding Descendants



4. An acknowledgement message is displayed. To confirm saving the mappings, click Yes.
5. The mapped member combinations are listed in the Mapper Maintenance page with the Macro value of GL Code as **Self**.

Figure 29-28 Mapped member combinations are listed in the Mapper Maintenance page

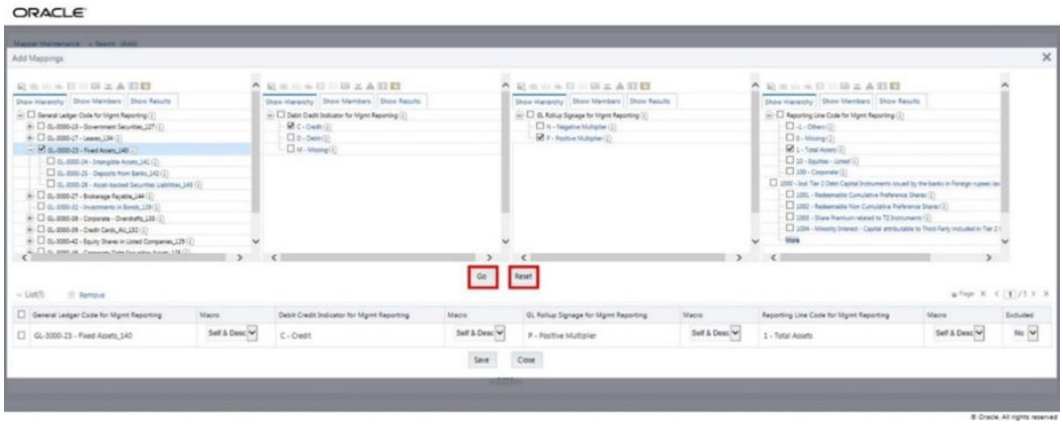


29.4.3 Performing Multiple Sets of Mapping Combinations

To perform mapping more than one time in the Add Mappings page, use the Reset functionality. Follow these steps:

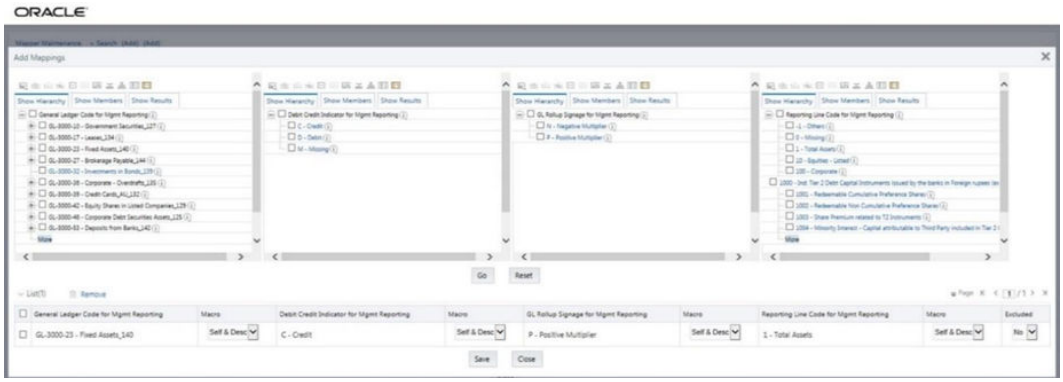
1. In the **Mapper Maintenance** page, in the Member combinations section, click **Add**.
2. The **Add Mappings** page is displayed. For illustration, select one parent member of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-23 – Fixed Assets_140, and the member of the Debit Credit Indicator for Mgmt Reporting hierarchy C - Credit to map to the member of the Reporting Line Code for Mgmt Reporting hierarchy 1 – Total Assets with the member of the GL Rollup Signage for Mgmt Reporting hierarchy P - Positive Multiplier. To map the members, click Go. The list of mapped members appear at the bottom. To initiate mapping of another set of member combinations, click **Reset**.

Figure 29-29 Add Mappings page for adding mappings for the first set of members and select Reset



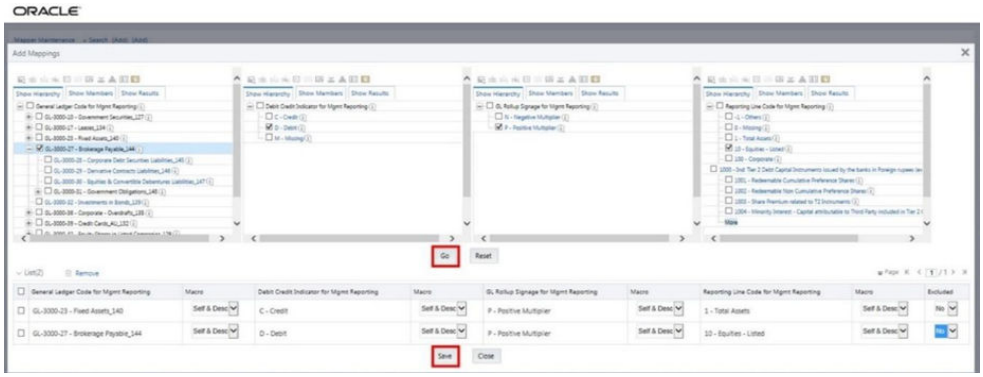
3. The mapping selections clear.

Figure 29-30 Mapping selections clear up



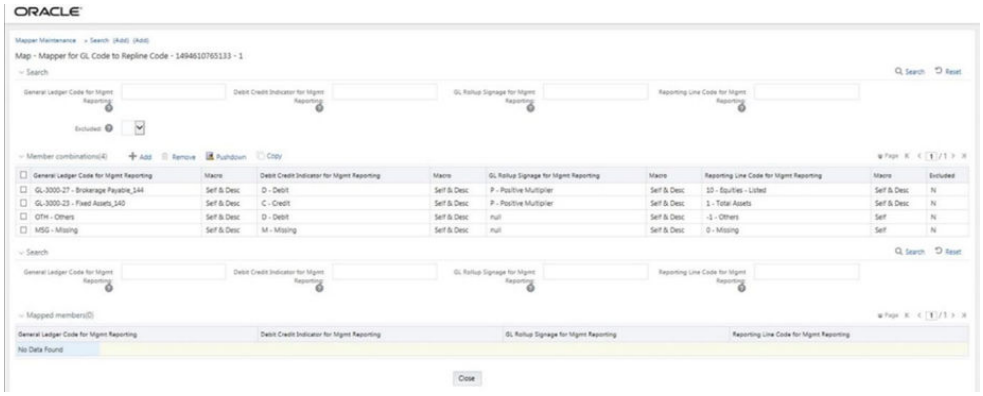
4. For illustration, select parent member of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-27 – Brokerage Payable_144, and the member of the Debit Credit Indicator for Mgmt Reporting hierarchy D - Debit to map to the member of the Reporting Line Code for Mgmt Reporting hierarchy 10 – Equities - Listed with the member of the GL Rollup Signage for Mgmt Reporting hierarchy P - Positive Multiplier. To map this set of member combinations, click Go. To save the mappings, click **Save**.

Figure 29-31 Add Mappings page for adding mappings for the next set of members



5. An acknowledgement message is displayed. To confirm saving the mappings, click **Yes**.
6. Both set of mapped member combinations are listed in the Mapper Maintenance page.

Figure 29-32 Mapped member combinations are listed in the Mapper Maintenance page



29.5 Loading the Mapper Maintenance from Backend

In this illustration, load the MAP_GL_CODE_REP_LINE table in the Atomic schema with the V_MAP_ID value as 1494610765133, and load these column values:

- 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 = GL Rollup Signage.
- V_MEMBER_4 = Reporting Line Code (values from DIM_REP_LINE.N_REP_LINE_CD).

Figure 29-33 Loading the Mapper Maintenance from backend

> **Note:**
>
> These values are real business key columns and not display codes.

29.6 Executing the GL to Management Reporting T2T using PMF

The T2T is a part of Financial Services Data Foundation Sourced Run. The process can be executed through the out-of-the-box standard Run, which is Financial Services Data Foundation Execution Run. For more information, see the section Executing Run through Process Modelling Framework in OFSDF.

> **Note:**
>
> When executing the process in PMF, RUNSkey auto-generates and stamps against each record.

29.6.1 Error Messages

The following most common error message may be logged in the T2T log file directory `ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA`:

```
Unique Constraint Violation
```

This error occurs after there is an attempt to re?load or load the existing records for the already executed AS_OF_DATE.

29.7 Checking the Execution Status for GL to Management Reporting Result Table

Monitor status of execution can be monitored in the Process Monitor page.

> **Note:**
>
> For a more comprehensive coverage of configuration and execution of a process, see the section Executing Run through Process Modelling Framework in OFSDF and the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

Access the execution log on the application server in the following directory:

`ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA`

The file name contains the batch execution ID. The error log table in the atomic schema is as follows:

- FCT_MGMT_REPORTING$

29.8 Retrieving the GL to Management Reporting Result T2T Definitions

To retrieve the T2T definitions in Excel document format for reference, go to the metadata browser of the Unified Metadata Manager (UMM) component of OFSAAI.

30
Other OFSDF Result Tables

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.

30.1 Overview of the 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_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_TRD_ACCOUNT_TXN_SUMMARY
- FCT_SERV_LN_CUST_RELATIONSHIP
- FCT_ACCT_CUST_RELATIONSHIP
- FCT_CORPORATE_ACTIONS

- FCT_TRADE_EXECUTION
- FCT_SECURITIZATION_POOL
- FCT_SHARE_HOLDING_DETAILS
- DIM_CONSENT_PURPOSE
- DIM_CREDIT_REASON
- DIM_COMMODITY_GRADE
- DIM_SUB_NETTING_AGREEMENT
- DIM_LEGAL_PROCEDING_STATUS

30.2 Overview of the Other Results Population

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

Table 30-1 Other Result T2T Definitions

Source Table Name	Target Table Name	T2T Name
STG_ASSETS_SOLD	FCT_ACCOUNT_POSTION_PAIR	T2T_FCT_ASSETS_SOLD
STG_ENTITY_PARENT_DETAILS	FCT_ENTITY_PARENT_INFO	T2T_FCT_ENTITY_PARENT_INFO
STG_ISSUED_INSTR_POSITIONS	FCT_ISSUED_INSTR_POSITIONS	T2T_FCT_ISSUED_INSTR_POSITIONS
STG_CUSTOMER_IDENTIFCTN_DOC	FCT_CUST_IDENTIFICATION_DOC	T2T_FCT_CUST_IDENTIFICATION_DOC
FCT_COMMON_ACCOUNT_SUMMARY	FCT_INTRA_COMPANY_ACCT_SUMMARY	T2T_FCT_INTRA_COMPANY_ACCT_SUMMARY
STG_LOANS_SERVICED	FCT_ACCT_RECOVERY_DETAILS	T2T_FCT_LOANS_SERVICED
STG_FIXED_ASSETS_DETAILS	FCT_ACCT_WRITE_OFF_DETAILS	T2T_FCT_FIXED_ASSETS
STG_CREDIT_LINE_DETAILS	FCT_ASSETS_SOLD	T2T_FCT_CREDIT_LINE
STG_LEGAL_ENTITY_DETAILS	FCT_CAP_INSTR_POSITIONS	T2T_FCT_LEGAL_ENTITY_DETAILS
STG_CAP_INSTR_POSITIONS	FFCT_CAP_INSTR_TXNS	T2T_FCT_CAP_INSTR_POSITIONS
STG_CAP_INSTR_TXNS	FCT_CAP_INSTR_TXNS	T2T_FCT_CAP_INSTR_TXNS
STG_FUND_CIS_COMPOSITION	FCT_FUND_CIS_COMPOSITION	T2T_FCT_FUND_CIS_COMPOSITION
STG_MERCHANT_BANKING	FCT_MERCHANT_BANKING	T2T_FCT_MERCHANT_BANKING
STG_SPEND_OBLIGATIONS	FCT_SPEND_OBLIGATIONS	T2T_FCT_SPEND_OBLIGATIONS
STG_ACCT_WRITE_OFF_DETAILS	FCT_ACCT_WRITE_OFF_DETAILS	T2T_FCT_ACCT_WRITE_OFF_DETAILS
STG_RECOVERIES	FCT_ACCT_RECOVERY_DETAILS	T2T_FCT_ACCT_RECOVERY_DETAILS

Table 30-1 (Cont.) Other Result T2T Definitions

Source Table Name	Target Table Name	T2T Name
STG_ACCT_RECOVERY_DETAILS	FCT_RECOVERY	T2T_FCT_RECOVERY
STG_INSTR_PROPOSED_TXNS	FCT_INSTR_PROPOSED_TXNS	T2T_FCT_INSTR_PROPOSED_TXNS
STG_LITIGATION_DETAILS	FCT_LITIGATION_DETAILS	T2T_FCT_LITIGATION_DETAILS
STG_SERV_LN_CUST_RELATIONSHIP	FCT_SERV_LN_CUST_RELATIONSHIP	T2T_FCT_SERV_LN_CUST_RELATIONSHIP
STG_PARTY_ACCOUNT_ROLE_MAP	FCT_ACCT_CUST_RELATIONSHIP	T2T_FACR_STG_PARTY_ACCOUNT_ROLE_MAP
STG_ASSETS_SOLD	FCT_ACCOUNT_POSTION_PAIR	T2T_FCT_ASSETS_SOLD
STG_ENTITY_PARENT_DETAILS	FCT_ENTITY_PARENT_INFO	T2T_FCT_ENTITY_PARENT_INFO
STG_ISSUED_INSTR_POSITIONS	FCT_ISSUED_INSTR_POSITIONS	T2T_FCT_ISSUED_INSTR_POSITIONS
STG_CUSTOMER_IDENTIFCTN_DOC	FCT_CUST_IDENTIFICATION_DOC	T2T_FCT_CUST_IDENTIFICATION_DOC
FCT_COMMON_ACCOUNT_SUMMARY	FCT_INTRA_COMPANY_ACCT_SUMMARY	T2T_FCT_INTRA_COMPANY_ACCT_SUMMARY
STG_LOANS_SERVICED	FCT_ACCT_RECOVERY_DETAILS	T2T_FCT_LOANS_SERVICED
STG_FIXED_ASSETS_DETAILS	FCT_ACCT_WRITE_OFF_DETAILS	T2T_FCT_FIXED_ASSETS
STG_CREDIT_LINE_DETAILS	FCT_ASSETS_SOLD	T2T_FCT_CREDIT_LINE
STG_LEGAL_ENTITY_DETAILS	FCT_CAP_INSTR_POSITIONS	T2T_FCT_LEGAL_ENTITY_DETAILS
STG_CAP_INSTR_POSITIONS	FFCT_CAP_INSTR_TXNS	T2T_FCT_CAP_INSTR_POSITIONS
STG_CAP_INSTR_TXNS	FCT_CAP_INSTR_TXNS	T2T_FCT_CAP_INSTR_TXNS
STG_FUND_CIS_COMPOSITION	FCT_FUND_CIS_COMPOSITION	T2T_FCT_FUND_CIS_COMPOSITION
STG_MERCHANT_BANKING	FCT_MERCHANT_BANKING	T2T_FCT_MERCHANT_BANKING
STG_SPEND_OBLIGATIONS	FCT_SPEND_OBLIGATIONS	T2T_FCT_SPEND_OBLIGATIONS
STG_ACCT_WRITE_OFF_DETAILS	FCT_ACCT_WRITE_OFF_DETAILS	T2T_FCT_ACCT_WRITE_OFF_DETAILS
STG_RECOVERIES	FCT_ACCT_RECOVERY_DETAILS	T2T_FCT_ACCT_RECOVERY_DETAILS
STG_ACCT_RECOVERY_DETAILS	FCT_RECOVERY	T2T_FCT_RECOVERY
STG_INSTR_PROPOSED_TXNS	FCT_INSTR_PROPOSED_TXNS	T2T_FCT_INSTR_PROPOSED_TXNS
STG_LITIGATION_DETAILS	FCT_LITIGATION_DETAILS	T2T_FCT_LITIGATION_DETAILS

Table 30-1 (Cont.) Other Result T2T Definitions

Source Table Name	Target Table Name	T2T Name
STG_SERV_LN_CUST_RELATIONSHIP	FCT_SERV_LN_CUST_RELATIONSHIP	T2T_FCT_SERV_LN_CUST_RELATIONSHIP
STG_PARTY_ACCOUNT_ROLE_MAP	FCT_ACCT_CUST_RELATIONSHIP	T2T_FACR_STG_PARTY_ACCOUNT_ROLE_MAP
STG_CORPORATE_ACTIONS	FCT_CORPORATE_ACTIONS	T2T_FCT_CORPORATE_ACTIONS
STG_TRADE_EXECUTION	FCT_TRADE_EXECUTION	T2T_FCT_TRADE_EXECUTION
STG_SECURITIZATION_POOL	FCT_SECURITIZATION_POOL	T2T_FCT_SECURITIZATION_POOL
STG_SHARE_HOLDING_DETAILS	FCT_SHARE_HOLDING_DETAILS	T2T_FCT_SHARE_HOLDING_DETAILS

The Other Dimension Definition details are as follows.

Table 30-2 Other Dimension Table Definitions

SCD Map Reference Number	Source Table Name	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_PROCEDING_STS_MASTER	DIM_LEGAL_PROCEDING_STATUS

30.3 Executing the Other T2Ts and Dimensions

30.4 Checking the Execution Status

The status of execution can be monitored in the Process Monitor page.

> **Note:**
>
> For a more comprehensive coverage of configuration and execution of a process, see the section Executing Run through Process Modelling Framework in OFSDF and the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

The execution log for the 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.

Access the execution log on the application server in the following directory:

`ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA`

The file name contains the batch execution ID. The error log table in the atomic schema is as follows:

- 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$
- FCT_SHARE_HOLDING_DETAILS$

30.5 Retrieving the Other Result T2T Definitions

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

31

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.

31.1 Overview of the 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.

31.2 Overview of the BASEL Processing to FSDF Results Integration

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

Table 31-1 BASEL Processing to FSDF Results Integration T2T Definitions

Source Table Name	Target Table Name	T2T Name
FSI_CAP_SUB_EXPOSURES	FSI_CAP_EXPOSURES_POST_CRM	T2T_FSI_CAP_EXPOSURES_POST_CRM
FSI_PARTY_GRP_MEMBER_MAP	FCT_REG_CAP_PARTY_GRP_MMBR_MAP	T2T_FCT_REG_CAP_PARTY_GRP_MMBR_MAP
FCT_ISSUED_INSTR_POSITIONS	FCT_REG_ISSUED_INSTR_POSITIONS	T2T_FCT_REG_ISSUED_INSTR_POSITIONS
FSI_COUNTRYWISE_RISK_SUMMARY	FCT_REG_CAP_CP_CTRY_RSK_SUMARY	T2T_FCT_REG_CAP_CP_CTRY_RSK_SUMARY
FSI_ENTITY_STD_ACCT_HEAD_DTLS	FCT_REG_CAP_ENTITY_SUMMARY	T2T_FCT_REG_CAP_ENTITY_SUMMARY
FCT_MARKET_RISK_COM_CAPITAL	FCT_MR_CAPITAL_SUMMARY	T2T_FCT_MR_CAPITAL_SUMMARY_FMRCC
FCT_PARTY_GROUP_LARGE_EXPOSURE	FCT_REG_LARGE_EXP_CP_LIMITS	T2T_FCT_REG_LARGE_EXP_CP_LIMITS
FSI_CAP_EXP_MITIGANT_MAPPING	FCT_REG_ACCT_MITIGANT_MAPPING	T2T_FRAMM_NET_POOL_EXP_MITIGANT_MAP
FSI_CAP_SUB_EXPOSURES	FCT_REG_POOL_MITIGANT_MAP	T2T_FRPMM_FSI_CAP_SUB_EXPOSURES
FCT_OPS_RISK_DATA	FCT_REG_OR_CAPITAL_SUMMARY	T2T_FCT_REG_OR_CAPITAL_SUMMARY
FCT_STANDARD_ACCT_HEAD	FCT_REG_LE_CAPITAL_SUMMARY	T2T_FCT_REG_LE_CAPITAL_SUMMARY
FCT_MARKET_RISK_EQ_CAPITAL	FCT_MR_CAPITAL_SUMMARY	T2T_FCT_MR_CAPITAL_SUMMARY_FMREQC
FSI_CAP_NETTABLE_POOL	FCT_REG_CAP_POOL_SUMMARY	T2T_FCT_REG_CAP_POOL_SUMMARY
FCT_COUNTERPARTY_EXPOSURE	FCT_REG_CP_CAPITAL_SUMMARY	T2T_FCT_REG_CP_CAPITAL_SUMMARY
FCT_MARKET_RISK_IR_CAPITAL	FCT_MR_CAPITAL_SUMMARY	T2T_FCT_MR_CAPITAL_SUMMARY_FMRIRC
FSI_CAP_INVESTMENT_EXPOSURES	FCT_REG_CAP_FIXED_ASST_SUMMARY	T2T_FCT_REG_CAP_FIXED_ASST_SUMMARY
FSI_CAP_SFT_EXPOSURES	FCT_REG_CAP_ACCOUNT_SUMMARY	T2T_FRCAS_FSI_CAP_SFT_EXPOSURES
FSI_CAP_SUB_EXPOSURES	FCT_REG_ACCT_MITIGANT_MAPPING	T2T_FRAMM_FSI_CAP_SUB_EXPOSURES
FSI_CAP_BANKING_EXPOSURES	FCT_REG_CAP_ACCOUNT_SUMMARY	T2T_FRCAS_FSI_CAP_BANKING_EXPOSURES
FCT_CCP_DETAILS	FCT_REG_CP_CAPITAL_SUMMARY	T2T_FRCCS_FCT_CCP_DETAILS
FSI_FORECAST_RWA	FCT_FORECAST_REG_CAP_SUMMARY	T2T_FCT_FORECAST_REG_CAP_SUMMARY

Table 31-1 (Cont.) BASEL Processing to FSDF Results Integration T2T Definitions

Source Table Name	Target Table Name	T2T Name
FSI_CAP_INVESTMENT_EXPOSURES	FCT_REG_CAP_ACCOUNT_SUMMARY	T2T_FRCAS_FSI_CAP_INVESTMENT_EXPOSURES
FSI_PLACED_COLLATERAL	FCT_REG_CAP_PLCD_COLL_SUMMARY	T2T_FCT_REG_CAP_PLCD_COLL_SUMMARY
FSI_CAP_DERIVATIVES	FCT_REG_CAP_ACCOUNT_SUMMARY	T2T_FRCAS_FSI_CAP_DERIVATIVES
FCT_SECURITIZATION_POOL	FCT_REG_SEC_POOL_SUMMARY	T2T_FCT_REG_SEC_POOL_SUMMARY
FCT_MARKET_RISK_CAPITAL	FCT_MARKET_RISK_REPORTING	MKT_RISK_REPORTING_POP_IR
FSI_CAP_BANKING_EXPOSURES	FCT_REG_CAP_ASSET_SOLD_SUMMARY	T2T_FCT_REG_CAP_ASSET_SOLD_SUMMARY
FSI_CAP_BANKING_EXPOSURES	FCT_REG_CAP_CREDIT_LINE_SUMMRY	T2T_FCT_REG_CAP_CREDIT_LINE_SUMMRY
FCT_MR_VAR_SUMMARY_DATA	FCT_MR_VAR_PORTFOLIO_SUMMARY	T2T_FCT_MR_VAR_PORTFOLIO_SUMMARY
FCT_MARKET_RISK_EXPOSURES	FCT_REG_MARKET_RISK_EXPOSURES	T2T_FCT_REG_MARKET_RISK_EXPOSURES
FCT_MARKET_RISK_FOREX_CAPITAL	FCT_MR_CAPITAL_SUMMARY	T2T_FCT_MR_CAPITAL_SUMMARY_FMRFRXC
FCT_MR_VAR_SUMMARY_DATA	FCT_MR_VAR_SUMMARY	T2T_FCT_MR_VAR_SUMMARY
FSI_CAP_MITIGANTS	FCT_MITIGANT_REG_CAPITAL	T2T_FMRC_FSI_CAP_MITIGANTS
DIM_STD_MITIGANT_TYPE	FCT_MITIGANT_REG_CAPITAL	T2T_FMRC_UNCOV_FSI_CAP_SUB_EXPOSURES
FSI_CAP_ACCT_ASSET_CLASS	FCT_REG_CAP_ACCT_ASSET_CLASS	T2T_FSI_CAP_ACCT_ASSET_CLASS_FCT_REG_CAP_ASSET_CLASS

31.3 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:

- 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.

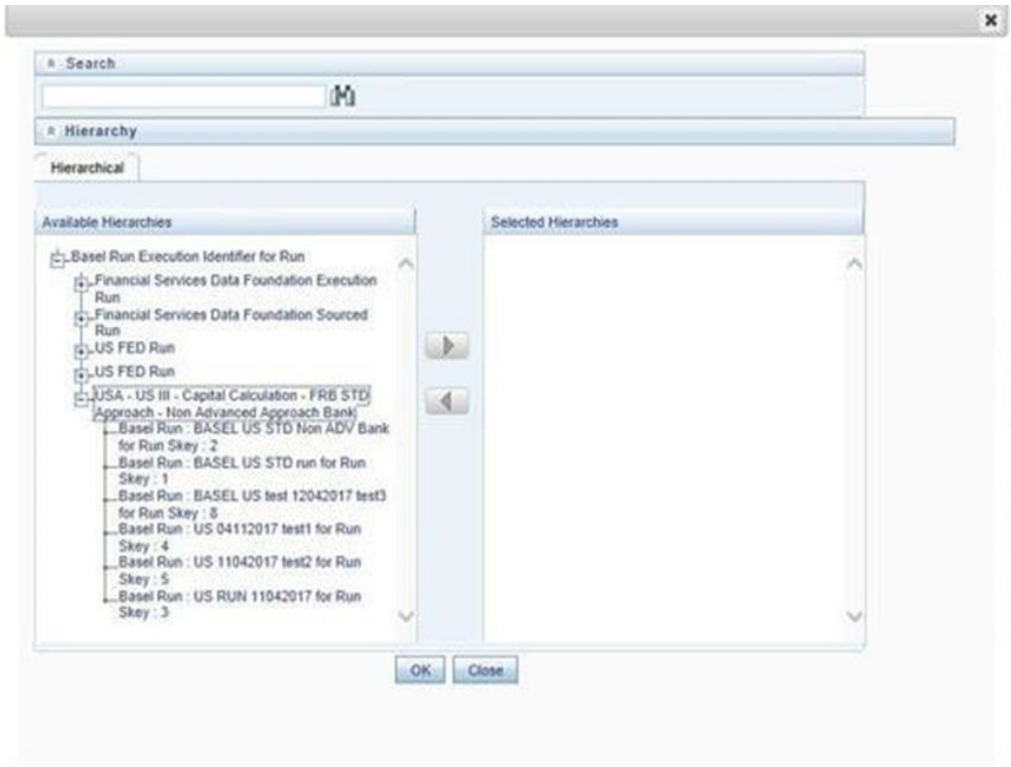
> **Note:**
>
> For BASEL-FSDF Integration Run, please use the FSDF Process Modelling Framework (PMF) screen as the Request Report Flag, Override Report Flag, and Approve Report Flag options are not available in the BASEL PMF to enable the Reporting Flag.

- 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 the PMF page. 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:

  1. Login to Oracle Financial Services Analytical Applications interface with your credentials.
  2. Navigate to the Process Modeller page. For navigation, see the section Executing Run through Process Modelling Framework in OFSDF.
  3. Select **Execute Run**.
  4. The Select Run Params page is displayed.
  5. Enter the Run Execution Description. The BASEL Run Execution Identifier and FIC MIS Date is auto-populated from the BASEL Run report used.
  6. Click **OK**.

  > **Note:**
  >
  > Resave Hierarchy HFSDF004 (FSDF BASEL Run Execution Identifier for Run) after BASEL execution for getting values in this BASEL Run Execution Identifier.

Figure 31-1 Integration of BASEL and FSDF using approved BASEL Run Execution ID in FSDF Run



7. Select only one BASEL Run from the Available Hierarchies for the execution and click **OK**.

31.3.1 Execute 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.

31.3.2 Execute the T2Ts using PMF

The T2T is a part of Financial Services Data Foundation Sourced Run. The process can be executed through the out-of-the-box standard Run, which is Financial Services Data Foundation Execution Run. For more information, see the section Executing Run through Process Modelling Framework in OFSDF.

- 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
- T2T_FCT_SHARE_HOLDING_DETAILS

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

31.3.3 Error Messages

The following most common error message may be logged in the T2T log file directory `ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA`: and in the directory `ftpshare/logs/<Run_Date>/FSDFINFO/RUN EXECUTABLE` for the Dimension tables:

```
Unique Constraint Violation
```

This error occurs after there is an attempt to re?load or load the existing records for the already executed AS_OF_DATE.

31.4 Checking the Execution Status

The status of execution can be monitored in the Process Monitor page.

> **Note:**
>
> For a more comprehensive coverage of configuration and execution of a process, see the section Executing Run through Process Modelling Framework in OFSDF and the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

Access the execution log on the application server in the following directory:

```
ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA
```

The file name contains the batch execution ID. The error log table in the atomic schema is as follows:

- 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$

31.5 Retrieving the BASEL Processing to FSDF Results Integration T2T Definitions

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

32
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.

32.1 Overview of the 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 aforementioned T2Ts. These are optional T2Ts that will be deployed only when OFS_IFRS_PACK is installed.

32.2 Overview of the LLFP Processing to FSDF Results Integration

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

Table 32-1 LLFP Processing to FSDF Results Integration T2T Definition

Source Table Name	Target Table Name	T2T Name
FCT_ACCOUNT_DETAILS	FCT_LLFP_ACCOUNT_SUMMARY	T2T_FCT_LLFP_ACCOUNT_SUMMARY

32.3 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.

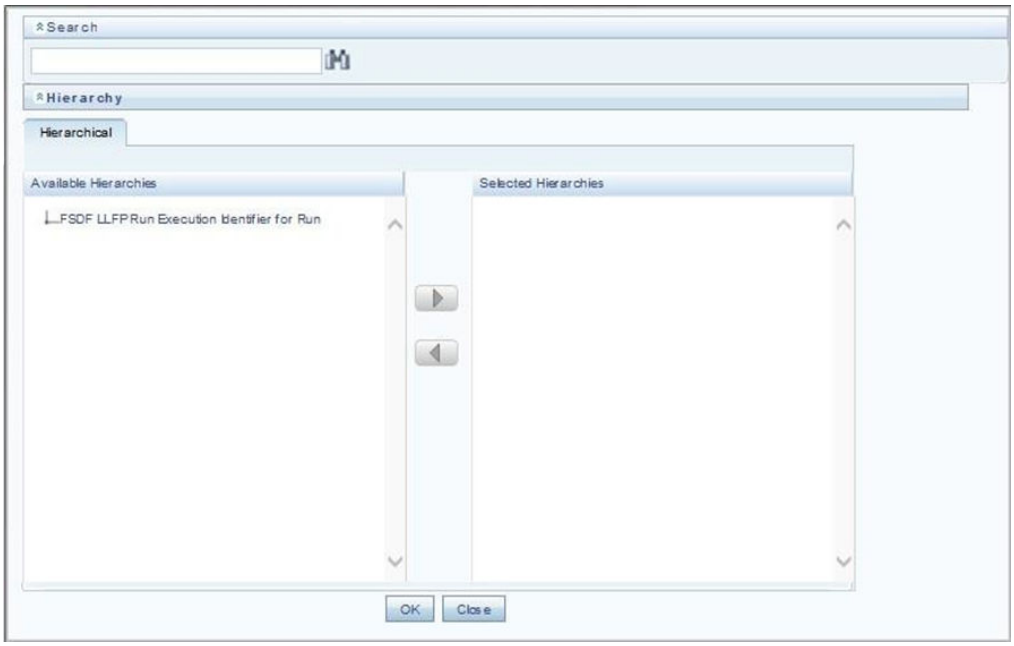
The processes inside Run should be ordered as LLFP first, then FSDF, and finally the LLFP FSDF Integration process. In this Run, the LLFP processing area and the FSDF results area tables must have the same Run SKEY across all tables.

> **Note:**
>
> For LLFP-FSDF Integration Run, please use the FSDF PMF screen as the Request Report Flag, Override Report Flag, and Approve Report Flag options are not available in the LLFP PMF 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 the PMF 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 the **Process Modeller** page. For navigation, see the section Executing Run through Process Modelling Framework in OFSDF.
   c. Select **Execute Run**.
   d. The Select Run Params window is displayed.
   e. Enter the Run Execution Description. The LLFP Run Execution Identifier and FIC MIS Date is auto-populated from the LLFP Run report used.
   f. Click **OK**.

   > **Note:**
   >
   > Resave Hierarchy HFSDF007 (FSDF LLFP Run Execution Identifier for Run) after LLFP execution for getting values in this LLFP Run Execution Identifier.

Figure 32-1 Integration of LLFP and FSDF using approved LLFP Run Execution ID in FSDF Run



g. Select only one LLFP Run from the Available Heirarchies for the execution and click **OK**.

32.4 Checking the Execution Status

The status of execution can be monitored in the Process Monitor page.

> **Note:**
>
> For a more comprehensive coverage of configuration and execution of a process, see the section Executing Run through Process Modelling Framework in OFSDF and the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

Access the execution log on the application server in the following directory:

```
ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA
```

The file name contains the batch execution ID. The error log table in the atomic schema is as follows:

- FCT_LLFP_ACCOUNT_SUMMARY$

32.5 Retrieving the LLFP Processing to FSDF Results Integration T2T Definition

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

33

LRS Pack Processing to FSDF Results Integration

This chapter provides information about LRS pack (Liquidity Risk Solutions pack, which was formerly Liquidity Risk Management) Processing to FSDF Results Integration in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section. The LRS pack consists of the following two application IDs:

- LRM_LCR (for Liquidity Coverage Ratio)
- LRM_DIC (for Deposit Insurance)

33.1 Overview of the LRS Pack Processing to FSDF Results Integration Tables

As part of LRS pack processing to FSDF results integration, FSDF tables are loaded from LRS pack processing tables using Table to Table (T2T) component of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) framework. These are the Result tables that store integrated results:

- FCT_LRM_ACCOUNT_SUMMARY
- FCT_DEP_INS_ACCT_PARTY_DETL
- FCT_DEP_INS_BENEFICIARY_DTL

> **Note:**
>
> As part of LRS pack processing results to FSDF integration, FSDF packages the T2Ts mentioned in the next section. These are optional T2Ts that are deployed only when OFS_TR_PACK is installed.

33.2 Overview of the LRS Pack Processing to FSDF Results Integration

These are the Table-to-Table seeded definitions to load data into the target tables.

Table 33-1 LRS Pack Processing to FSDF Results Integration T2T Definitions

Source Table Name	Target Table Name	T2T Name
FSI_LRM_INSTRUMENT	FCT_LRM_ACCOUNT_SUMMARY	T2T_FCT_LRM_ACCOUNT_SUMMARY
FCT_DEPOSIT_INSURANCE_DETAILS	FCT_DEP_INS_ACCT_PARTY_DETL	T2T_FCT_DEP_INS_ACCT_PARTY_DETL

Table 33-1 (Cont.) LRS Pack Processing to FSDF Results Integration T2T Definitions

Source Table Name	Target Table Name	T2T Name
FCT_DEPOSIT_INSURANCE_DETAILS	FCT_DEP_INS_BENEFICIARY_DTL	T2T_FCT_DEP_INS_BENEFICIARY_DTL

33.3 Executing the LRS Pack Processing to FSDF Results Integration T2Ts

(Required) <Enter a short description here.>

For LRS Pack FSDF integration, you must have FSDF and LRS pack installed on the same INFODOM. There are two ways to integrate LRS pack and FSDF:

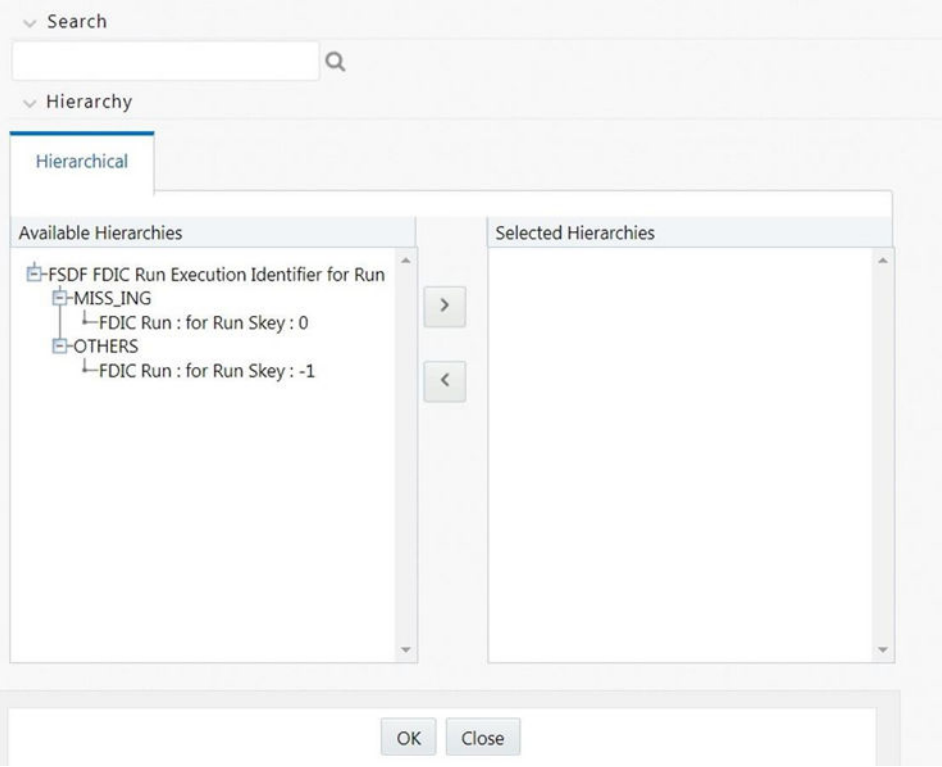
1. Creating Integrated Run at Implementation Site: During implementation, you can merge the tasks of both LRS pack and FSDF, and create an integrated Run to execute each time. The processes inside Run must be in the sequence LRS pack first, and then FSDF, and finally the LRS FSDF Integration process. In this Run, the LRS pack processing area and the FSDF results area tables must consist of the same Run SKEY across all tables.

   > **Note:**
   >
   > The Request Report Flag, Override Report Flag, and Approve Report Flag options are not available in the LRS PMF page to enable the Reporting Flag. Therefore, for LRS-FSDF Integration Run, use the FSDF PMF screen.

2. Using approved LRS Run Execution ID in FSDF Run: In this case, you can use the out-of-the-box LRS pack 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 LRS Run SKEY from the PMF screen. In this case, LRS pack processing area consists of one RUN SKEY and for the same data, FSDF consists of a different RUN SKEY in FSDF results area tables, where the data is getting reported. To generate a sample report, follow these steps:
   a. Login to Oracle Financial Services Analytical Applications interface with your credentials.
   b. Navigate to the Process Modeller page. For navigation, see the section Executing Run through Process Modelling Framework in OFSDF.
   c. Select **Execute Run**.
   d. The Select Run Params window is displayed.
   e. Enter the Run Execution Description. The LRS Run Execution Identifier and FIC MIS Date is auto-populated from the LRS Run report used.
   f. Click **OK**.

> **Note:**
>
> After LRS execution, to get values in this LRS Run Execution Identifier, Resave Hierarchy HFSDF006 (FSDF LRS Run Execution Identifier for Run).

Figure 33-1 Integration of LRS Pack and FSDF using approved LRS Pack Run Execution ID in FSDF Run



g. Select only one LRS Run from the Available Hierarchies for the execution and click **OK**.

33.4 Checking the Execution Status

The status of execution can be monitored in the Process Monitor page.

> **Note:**
>
> For a more comprehensive coverage of configuration and execution of a process, see the section Executing Run through Process Modelling Framework in OFSDF and the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

Access the execution log on the application server in the following directory:

```
ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA
```

The file name contains the batch execution ID. The error log table in the atomic schema is as follows:

- FCT_LRM_ACCOUNT_SUMMARY$
- FCT_DEP_INS_ACCT_PARTY_DETL$
- FCT_DEP_INS_BENEFICIARY_DTL$

33.5 Retrieving the LRS Pack Processing to FSDF Results Integration T2T Definitions

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

34
Metadata Browser

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

34.1 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 (OFSAA) 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.

34.2 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

34.3 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 done on regular basis to ensure that metamodel is in sync with underlying metadata.

34.4 Metadata Object to Application Map

To map all the objects the Financial Service Data Foundation application, a batch name <Infodom>_ MDB_OBJECT_APPLN_MAP has to be executed.

35
Metadata Report Utility

The Metadata Export Utility helps you to export OFSAA Metadata into csv format. This feature helps to get a view of OFSAA Metadata and its dependencies. It is a template-based approach where you create templates and select Metadata Objects that must be extracted. The extraction process is supported only for Excel Sheet. While defining the template, you are expected to have prior knowledge of the OFSAA Metadata Objects that are relevant from this application point of view.

35.1 Prerequisites

The following executions must be performed before using the Metadata Export Utility:

1. Before executing MDB Publish and Data Elements Wrapper Batch (<INFODOM>_POP_DATA_ELEMENTS_FSDF), ensure the following:
   a. Tablespace Requirement:
      i. Ensure that the USERS tablespace have a minimum of 100 GB available
      ii. Ensure that the TEMP tablespace is a minimum of 45 GB available
   b. Execute the following Gather Stat command for the mentioned tables:

```
BEGIN

DBMS_STATS.GATHER_TABLE_STATS(USER, 'TABLE_NAME');

END;
```

      i. Config Schema:
         - AAI_OBJECT_B
         - AAI_OBJECT_TL
         - AAI_DMT_DEFINITION
         - AAI_DMT_DEF_SOURCE_ENTITY
         - AAI_DMT_MAPPING_DETAILS
         - PR2_RULES_B
         - PR2_RULE_MAP
         - PR2_RULE_OBJECT
         - PR2_RULE_OBJECT_MEMBER
         - PR2_OBJECT_TL
         - PR2_OBJECT_TRACE
         - BATCH_MASTER
         - BATCH_TASK_MASTER
         - BATCH_PARAMETER_MASTER
         - METADATA_MASTER

- METADATA_ELEMENT_MASTER
- METADATA_LOCALE_MASTER
- METADATA_TYPE_MASTER
- METADATA_ATTRIBUTE_MASTER
- AAI_WF_PROCESS_B
- AAI_WF_PROCESS_TL
- AAI_WF_ACTIVITY_B
- AAI_WF_ACTIVITY_TL
- AAI_WF_APPLICATION_API_B
- AAI_WF_APPLICATION_API_TL
- AAI_WF_DATA_FIELD_B
- AAI_WF_DATA_FIELD_TL
- AAI_WF_ACTIVITY_TASK_B
- AAI_WF_ACTIVITY_TASK_TL
- AAI_WF_TRANSITION_B
- AAI_WF_TRANSITION_TL
- AAI_WF_ACTIVITY_TASK_USER_MAP
- AAI_WF_ACTUAL_PARAMS
- AAI_WF_ACTIVITY_SUBPROCESS
- AAI_WF_ATTR_COMPONENT
- AAI_WF_ATTR_COMP_VAR_MAPPING
- AAI_WF_PROCESS_DETAIL_FILTER
- AAI_WF_PROCESS_FILTER

2. Navigate to **Financial Services Data Foundation**, select **Metadata Browser**.
3. Click **Publish**: Execute the batch, INFODOM_MDB.
4. After performing an MDB Publish and executing the Data Element Wrapper Batch, ensure the following:
   a. Execute the following Gather Stat command for the mentioned tables:
   ```
   BEGIN

   DBMS_STATS.GATHER_TABLE_STATS(USER, 'TABLE_NAME');

   END;
   ```
      i. Atomic Schema:
         - FSI_DE_RUN_LINEAGE_METADATA
5. Logs: MDB logs are generated under deployed area `/Context_Name/logs/MDB_Publish_XXXX.log`
6. Data Elements Wrapper Execution : After MDB Publish is completed successfully with the message “Metadata publishing is finished.” in the `/Context_Name/logs/MDB_XXXX.log`, you must execute the Data Elements Utility with the following seeded batch to get the Data Lineage for each Metadata in OFSAA:

<INFODOM>_POP_DATA_ELEMENTS_FSDF

This execution requires adequate tablespace. Ensure that your Atomic Schema is having enough tablespace in TEMP and USERS.

Parameters used in <INFODOM>_POP_DATA_ELEMENTS_FSDF Batch

The batch can be executed in different modes according to each requirement. The following are the parameters used for executing the batch.

The default parameters used in the <INFODOM>_POP_DATA_ELEMENTS_FSDF batch are:

Table 35-1 Task1 (METADATA PARSER)

Sl. No.	Parameter	Description	List of Values	Default Value
1	P_FULL_PARSE	Full Parser Flag	Y/N	'Y'
2	P_INFODOM_NAME	Infodom Name	##INFODOM##	<Value of the Infodom where FSDF is installed>. For example: 'FSDFINFO'

35.2 Additional Configurations for User defined Metadata and Application Integration

The lineage execution will happen for the data flow which are configured FSI_DE_POP_RUN_LIST and FSI_DE_POP_RUN_RELATIONSHIP tables. Users can configure these two tables for getting the metadata lineage for user-specific runs.

Enabling Run for METADATA Parsing

Every execution for METADATA Parsing requires a minimum of one Run to be enabled in the FSI_DE_POP_RUN_LIST table in the Atomic Schema. By default, FSDF_EXE_RUN is enabled.

Table 35-2 Run Names for Metadata Parser

RUN NAME	INCLUDE RUN
FSDF_EXE_RUN	Y

Configuring prerequisite Run/ Batch

The pre-requisite data flow for FSDF EXE run is identified using records available in FSI_DE_POP_RUN_RELATIONSHIP. By default, FSDF run chart is seeded. For any additional data flow, the additional records can be inserted into this table, for the data flow to be considered during lineage execution.

Table 35-3 Run Names for Metadata Parser

RUN NAME	REL RUN NAME	PROCESSING OUTPUT FLAG
FSDF_EXE_RUN	<INFODOM>_POP_DATES_DIM	N

Table 35-3 (Cont.) Run Names for Metadata Parser

RUN NAME	REL RUN NAME	PROCESSING OUTPUT FLAG
FSDF_EXE_RUN	<INFODOM>_DIM_ACCOUNT_SCD	N
FSDF_EXE_RUN	<INFODOM>_DATA_FOUNDATION_SCD	N
FSDF_EXE_RUN	<INFODOM>_DATA_FOUNDATION_SCD_MLS	N
FSDF_EXE_RUN	<INFODOM>_POP_KEY_DIMENSION_SCD	N
FSDF_EXE_RUN	<INFODOM>_PARTY_MAPPER_SCD	N
FSDF_EXE_RUN	FSDF_MAPPER_RUN	N
FSDF_EXE_RUN	FSDF_SOURCE_RUN	N

Excluding Irrelevant Data Flows from Lineage Reports

For each Run, some of the Data Mappings can be functionally irrelevant. For these cases concerning any Run, the customer can opt for removing these Data Flow from Lineage Reports as an exclusion by inputting the same in the FSI_DE_RUN_FLOW_REMOVAL table.

Execution Types for METADATA Parsing in <INFODOM>_POP_DATA_ELEMENTS_FSDF Batch

1. Full METADATA Parsing [Default Mode] (if the P_FULL_PARSE parameter is 'Y', then the parsing happens for the entire METADATA and Run Elements for the Run(s) enabled in FSI_DE_POP_RUN_LIST table in the Atomic Schema).
2. Incremental METADATA Parsing [Optional Mode. Batch Parameter to Be Modified] (if the P_FULL_PARSE parameter is 'N', then the parsing happens for changed METADATA and Run Elements for the Run(s) enabled in FSI_DE_POP_RUN_LIST table in the Atomic Schema).
   You can edit the parameters by accessing the Batch Maintenance screen.
   a. Log in to Oracle Financial Services Analytical Applications interface with your credentials.
   b. Navigate to **Financial Services Data Foundation**, select **Operations,** and then select **Batch Maintenance.**
   c. Select Batch Name (<INFODOM>_POP_DATA_ELEMENTS_FSDF)
   d. (OPTIONAL) Select Task1 and click the Edit button. The Edit Task Definition Window is displayed.
   e. Modify the Parameter List field as applicable.

35.3 Additional Configurations for Data Transformation

By default, Data Transformations are not parsed by lineage component due to an unstructured format of SQL coding in the Data Transformation. However, the lineage component provides an option to provide the Data Transformation Mapping in a metadata form and utilize it in the lineage output.

Configuring the Data Flow name and its Mapping

The data flow name such as the Post Load Change (PLC) component name of the DT and the corresponding Target and Source tables used in the Post Load Change (PLC) Component name must be updated in the table FSI_DE_POP_DT_DATA_FLOW_MAP as shown below:

Table 35-4 Data Flow name and its Mapping

DATA_FLOW_NAME	TARGET_TABLE	TARGET_COLUMN	TARGET_MEMBER	MAP_TYPE_ID	MAP_TYPE	SOURCE_TABLE	SOURCE_COLUMN	SOURCE_MEMBER	KEY_COLUMN_FLAG
<<FUNCTIONNAME>>	STG_ACCT_CREDIT_SCORE_DETAILS	N_BEHAVIOUR_SCORE	-	1	Direct	STG_VEHICLE_DETAILS	N_NO_OF_DAYS_DRIVEN_PER_WEEK	-	-

Adding the Data Transformation in Run Relationship

The data flow name which are provided in the Datal Flow and its Mapping table, is processed by the lineage component only if the Batch for the data Transformation is added in FSI_DE_POP_RUN_RELATIONSHIP. For more information, see the Configuring prerequisite Run/ Batch.

35.4 Verifying Logs

Data Elements logs are generated in Atomic Schema under the FSI_MESSAGE_LOG table.

Table 35-5 Data Element Logs

Tasks	Batch Run ID	Indication
Task1 (METADATA Parsing)	REGISTER_ELEMENTS_<Batch_Run_ID>	Processes Metadata Parsing. The message “Completed REGISTER_ELEMENTS” indicates that the Metadata parsing is completed with Registration.

35.5 Validating Lineage Outputs

In Atomic Schema, you must verify that data is present in the following tables and ensure that the table is populated:

- FSI_DE_RUN_LINEAGE_METADATA

35.6 Utility to Generate a Lineage Report in CSV Format

This utility uses the following script to retrieve the lineage output as .csv file.

Steps to generate Lineage Report

1. Navigate to this path $FIC_HOME/utility/DataLineageUtility.

2. Provide permissions to the Shell Script.

   ```
   chmod -R 755 Data_Lineage_Extraction_Utility.sh
   ```

3. To execute the utility run the below command:

   ```
   ./Data_Lineage_Extraction_Utility.sh <<Schema_name >>
   <<Schema_password>> <<Schema_Instance>> << Lineage_Type >>
   <<MetadataName>>
   ```

   Provide the Lineage Type input as one of the following:

   - **DataMapping**- If you wish to view the T2T output.

     For example: ./Data_Lineage_Extraction_Utility.sh IUT_SCHEMA Password1234 Oracle123 DataMapping T2T_FCT_COMMON_ACCOUNT_SUMMARY

   - **TargetModel**- If you wish to view the Result area table output.

     For example: ./Data_Lineage_Extraction_Utility.sh IUT_SCHEMA Password1234 Oracle123 TargetModel FCT_COMMON_ACCOUNT_SUMMARY

   - **ExtractConnector**- If you wish to view the Result area table output.

     For example: ./Data_Lineage_Extraction_Utility.sh IUT_SCHEMA Password1234 Oracle123 ExtractConnector << CONNECTOR NAME>>

   > **Note:**
   >
   > The detailed parameter description is available in the Readme.txt in the same path $FIC_HOME/utility/DataLineageUtility.

4. After the above script is executed, the files are generated in the .csv format in the same path.

35.7 Utility to Generate a Lineage Report in Json Format

This utility uses the following script to retrieve the lineage output as a Json file.

Steps to generate Lineage Report in Json format

1. Navigate to this path $FIC_HOME/utility/DataLineageUtility.
2. Provide permissions to the Shell Script.
   ```
   chmod -R 755 Data_Lineage_JSON_Extraction_Utility.sh
   ```
3. To execute the utility run the below command:
   ./Data_Lineage_JSON_Extraction_Utility.sh <<Schema_name >> <<Schema_password>> <<Schema_Instance>> << Lineage_Type >> <<MetadataName>>

   Provide the Lineage Type input as one of the following:

   - **DataMapping**- If you wish to view the T2T output.

     For example: ./Data_Lineage_JSON_Extraction_Utility.sh IUT_SCHEMA Password1234 Oracle123 DataMapping T2T_FCT_COMMON_ACCOUNT_SUMMARY

   - **TargetModel**- If you wish to view the Result area table output.

For example: ./Data_Lineage_JSON_Extraction_Utility.sh IUT_SCHEMA Password1234 Oracle123 TargetModel FCT_COMMON_ACCOUNT_SUMMARY

- **ExtractConnector**- If you wish to view the Result area table output.

  For example: ./Data_Lineage_JSON_Extraction_Utility.sh IUT_SCHEMA Password1234 Oracle123 ExtractConnector << CONNECTOR NAME>>

> **Note:**
>
> The detailed parameter description is available in the Readme.txt in the same path $FIC_HOME/utility/DataLineageUtility.

4. After the above script is executed, the files are generated in the Json format in the same path.

35.8 Lineage Properties

Select the Lineage Properties required to be generated.

The following Lineage Properties (columns) are available in the Metadata Report Screen.

Table 35-6 Lineage Properties

Sl. No.	Lineage Property	Property Description
1	Metadata Code	Stores the Metadata Code of the Lineage Report generated for the given Cell ID, Cell Group ID, and Derived Entity.
2	Metadata Description	Stores the Metadata Description of the Lineage Report generated for the given Cell ID, Cell Group ID, and Derived Entity.
3	Metadata Type	Stores the Metadata Type of the Lineage Report generated for the given Cell ID, Cell Group ID, and Derived Entity.
4	Metadata Sub Code	Stores the Metadata Sub Code of the Lineage Report generated for the given Cell ID, Cell Group ID, Derived Entity, and Metadata Code. Metadata Sub Code represents direct Metadata (Metadata Sub Code is the same Metadata Code) or derived Metadata Code like Datasets/ Expressions.

Table 35-6 (Cont.) Lineage Properties

Sl. No.	Lineage Property	Property Description
5	Metadata Sub Description	Stores the Metadata Sub Description of the Lineage Report generated for the given Cell ID, Cell Group ID, Derived Entity, and Metadata Code. Metadata Sub Code represents direct Metadata (Metadata Sub Code is the same Metadata Code) or derived Metadata Code like Datasets/Expressions.
6	Metadata Sub Type	Stores the Metadata Sub Type of the Lineage Report generated for the given Cell ID, Cell Group ID, Derived Entity, and Metadata Code. Metadata Sub Code represents direct Metadata (Metadata Sub Code is the same Metadata Code) or derived Metadata Code like Datasets/ Expressions.
7	Result Area Table Application	Stores the Results Area Table Application of the Lineage Report generated for the given Cell ID, Cell Group ID, Derived Entity, Metadata Code, and Metadata Sub Code. The Results Area Table application is the responsible OFSAA Application to populate the table.
8	Result Area Table Type	Stores the Results Area Table Type of the Lineage Report generated for the given Cell ID, Cell Group ID, Derived Entity, Metadata Code, and Metadata Sub Code. The Results Area Table Type represents how the table is populated. For example Data Flow, Seeded Data, and so on.
9	Result Area Table	Stores the Results Area Table the Lineage Report generated for the given Cell ID, Cell Group ID, Derived Entity, Metadata Code, and Metadata Sub Code. The Results Area Table is the OFSAA data model table that populates or helps to populate the given Cell (MDRM) in the Reporting Layer.

Table 35-6 (Cont.) Lineage Properties

Sl. No.	Lineage Property	Property Description
10	Result Area Column	Stores the Results Area Column the Lineage Report generated for the given Cell ID, Cell Group ID, Derived Entity, Metadata Code, Metadata Sub Code, and Results Area Table. The Results Area Table column is the OFSAA data model column that populates or helps to populate the given Cell (MDRM) in Reporting Layer.
11	Target Metadata Operator	Stores the Target Metadata Operator of the Lineage Report generated for the given Results Area Column and Member Code embedded inside the Metadata like Business Processor, Hierarchy, or Dataset. The operator is derived after a standardization process like Reverting all <>, =, IN, NOT IN conditions to equal operator.
12	Target Metadata Member	Stores the Target Metadata Operator of the Lineage Report generated for the given Results Area Column and Member Code embedded inside the Metadata like Business Processor, Hierarchy, or Dataset. The Member Code presents its ultimate form through a standardization process like Reverting all <>, =, IN, NOT IN conditions to the equal operator and getting the respective Member Codes.
13	Reporting Run Name	Stores the Regulatory Reporting Run Name for Jurisdiction Code of Lineage Report generated.
14	Lineage Run or Batch Level1	Stores the Level1 Run Name or Batch Name of Lineage Report generated for populating the Results Area Table and Column.
15	Lineage Data Flow Name Level1	Stores the Level1 Data Flow Name of Lineage Report generated for populating the Results Area Table and Column.
16	Lineage Data Flow Type Level1	Stores the Level1 Data Flow Type of Lineage Report generated for populating the Results Area Table and Column.

Table 35-6 (Cont.) Lineage Properties

Sl. No.	Lineage Property	Property Description
17	Lineage Element Table Level1	Stores the Level1 Source Table of Lineage Report generated for populating the Results Area Table and Column.
18	Lineage Element Column Level1	Stores the Level1 Source Column of Lineage Report generated for populating the Results Area Table and Column.
19	Lineage Run or Batch Level2	Stores the Level2 Run Name or Batch Name of Lineage Report generated for populating the Level1 Source Table and Column.
20	Lineage Data Flow Name Level2	Stores the Level2 Data Flow Name of Lineage Report generated for populating the Level1 Source Table and Column.
21	Lineage Data Flow Type Level2	Stores the Level2 Data Flow Type of Lineage Report generated for populating the Level1 Source Table and Column.
22	Lineage Element Table Level2	Stores the Level2 Source Table of Lineage Report generated for populating the Level1 Source Table and Column.
23	Lineage Element Column Level2	Stores the Level2 Source Column of Lineage Report generated for populating the Level1 Source Table and Column.
24	Lineage Run or Batch Level3	Stores the Level3 Run Name or Batch Name of Lineage Report generated for populating the Level2 Source Table and Column.
25	Lineage Data Flow Name Level3	Stores the Level3 Data Flow Name of Lineage Report generated for populating the Level2 Source Table and Column.
26	Lineage Data Flow Type Level3	Stores the Level3 Data Flow Type of Lineage Report generated for populating the Level2 Source Table and Column.
27	Lineage Element Table Level3	Stores the Level3 Source Table of Lineage Report generated for populating the Level2 Source Table and Column.

Table 35-6 (Cont.) Lineage Properties

Sl. No.	Lineage Property	Property Description
28	Lineage Element Column Level3	Stores the Level3 Source Column of Lineage Report generated for populating the Level2 Source Table and Column.
29	Lineage Run or Batch Level4	Stores the Level4 Run Name or Batch Name of Lineage Report generated for populating the Level3 Source Table and Column.
30	Lineage Data Flow Name Level4	Stores the Level4 Data Flow Name of Lineage Report generated for populating the Level3 Source Table and Column.
31	Lineage Data Flow Type Level4	Stores the Level4 Data Flow Type of Lineage Report generated for populating the Level3 Source Table and Column.
32	Lineage Element Table Level4	Stores the Level4 Source Table of Lineage Report generated for populating the Level3 Source Table and Column.
33	Lineage Element Column Level4	Stores the Level4 Source Column of Lineage Report generated for populating the Level3 Source Table and Column.
34	Lineage Run or Batch Level5	Stores the Level5 Run Name or Batch Name of Lineage Report generated for populating the Level4 Source Table and Column.
35	Lineage Data Flow Name Level5	Stores the Level5 Data Flow Name of Lineage Report generated for populating the Level4 Source Table and Column.
36	Lineage Data Flow Type Level5	Stores the Level5 Data Flow Type of Lineage Report generated for populating the Level4 Source Table and Column.
37	Lineage Element Table Level5	Stores the Level5 Source Table of Lineage Report generated for populating the Level4 Source Table and Column.
38	Lineage Element Column Level5	Stores the Level5 Source Column of Lineage Report generated for populating the Level4 Source Table and Column.

Table 35-6 (Cont.) Lineage Properties

Sl. No.	Lineage Property	Property Description
39	Lineage Run or Batch Level6	Stores the Level6 Run Name or Batch Name of Lineage Report generated for populating the Level5 Source Table and Column.
40	Lineage Data Flow Name Level6	Stores the Level6 Data Flow Name of Lineage Report generated for populating the Level5 Source Table and Column.
41	Lineage Data Flow Type Level6	Stores the Level6 Data Flow Type of Lineage Report generated for populating the Level5 Source Table and Column.
42	Lineage Element Table Level6	Stores the Level6 Source Table of Lineage Report generated for populating the Level5 Source Table and Column.
43	Lineage Element Column Level6	Stores the Level6 Source Column of Lineage Report generated for populating the Level5 Source Table and Column.
44	Lineage Run or Batch Level7	Stores the Level7 Run Name or Batch Name of Lineage Report generated for populating the Level6 Source Table and Column.
45	Lineage Data Flow Name Level7	Stores the Level7 Data Flow Name of Lineage Report generated for populating the Level6 Source Table and Column.
46	Lineage Data Flow Type Level7	Stores the Level7 Data Flow Type of Lineage Report generated for populating the Level6 Source Table and Column.
47	Lineage Element Table Level7	Stores the Level7 Source Table of Lineage Report generated for populating the Level6 Source Table and Column.
48	Lineage Element Column Level7	Stores the Level7 Source Column of Lineage Report generated for populating the Level6 Source Table and Column.
49	Lineage Run or Batch Level8	Stores the Level8 Run Name or Batch Name of Lineage Report generated for populating the Level7 Source Table and Column.

Table 35-6 (Cont.) Lineage Properties

Sl. No.	Lineage Property	Property Description
50	Lineage Data Flow Name Level8	Stores the Level8 Data Flow Name of Lineage Report generated for populating the Level7 Source Table and Column.
51	Lineage Data Flow Type Level8	Stores the Level8 Data Flow Type of Lineage Report generated for populating the Level7 Source Table and Column.
52	Lineage Element Table Level8	Stores the Level8 Source Table of Lineage Report generated for populating the Level7 Source Table and Column.
53	Lineage Element Column Level8	Stores the Level8 Source Column of Lineage Report generated for populating the Level7 Source Table and Column.
54	Lineage Run or Batch Level9	Stores the Level9 Run Name or Batch Name of Lineage Report generated for populating the Level8 Source Table and Column.
55	Lineage Data Flow Name Level9	Stores the Level9 Data Flow Name of Lineage Report generated for populating the Level8 Source Table and Column.
56	Lineage Data Flow Type Level9	Stores the Level9 Data Flow Type of Lineage Report generated for populating the Level8 Source Table and Column.
57	Lineage Element Table Level9	Stores the Level9 Source Table of Lineage Report generated for populating the Level8 Source Table and Column.
58	Lineage Element Column Level9	Stores the Level9 Source Column of Lineage Report generated for populating the Level8 Source Table and Column.
59	Lineage Run or Batch Level10	Stores the Level10 Run Name or Batch Name of Lineage Report generated for populating the Level9 Source Table and Column.
60	Lineage Data Flow Name Level10	Stores the Level10 Data Flow Name of Lineage Report generated for populating the Level9 Source Table and Column.

Table 35-6 (Cont.) Lineage Properties

Sl. No.	Lineage Property	Property Description
61	Lineage Data Flow Type Level10	Stores the Level10 Data Flow Type of Lineage Report generated for populating the Level9 Source Table and Column.
62	Lineage Element Table Level10	Stores the Level10 Source Table of Lineage Report generated for populating the Level9 Source Table and Column.
63	Lineage Element Column Level10	Stores the Level10 Source Column of Lineage Report generated for populating the Level9 Source Table and Column.
64	Data Element Table Application	Stores the Ultimate Source Table Application of Lineage Report generated for populating the Results Area Table and Column. The application is responsible for sourcing the data.
65	Data Element Table Type	Stores the Ultimate Source Table Type of Lineage Report generated for populating the Results Area Table and Column. This represents the Type of the Source Table like Download, Mapper Download, Seeded Data, Run Parameters, and so on.
66	Data Element Table	Stores the Ultimate Source Table of Lineage Report generated for populating the Results Area Table and Column.
67	Data Element Column	Stores the Ultimate Source Column of Lineage Report generated for populating the Results Area Table and Column.
68	Data Element Metadata Operator	Stores the Ultimate Source Table Column Operator Code of Lineage Report generated concerning Target Metadata Operator in Results Area. This is the derived representation of the Target Metadata Operator in the Results Area.
69	Data Element Metadata Member	Stores the Ultimate Source Table Column Member Code of Lineage Report generated concerning Target Metadata Member Code in Results Area. This is the derived representation of the Target Metadata Member Code in the Results Area.

36
Time Dimension Table

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).

36.1 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:

- Database function FN_DIM_DATES
- Database procedure PROC_DIM_DATES_POPULATION that is called by the function FN_DIM_DATES mentioned earlier.

36.2 Prerequisites

The following are the prerequisites for Time dimension population.

1. All the post install steps mentioned in the Oracle Financial Services Advanced Analytical Applications Infrastructure Installation Guide Release 8.1.2.0.0 and Oracle Financial Services Data Foundation Application Pack Installation and Configuration Guide Release 8.1.2.0.0 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:
4. 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, see the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.
    - Iccserver
    - Router
    - AM Server
    - Message Server
    - OLAP Server

5. Create batches to execute the function. For more details, refer to section How to Define a Batch.

36.3 Tables Used by the Time Dimension Population Transformation

For more details on viewing the structure of earlier tables, see the Oracle Financial Services Analytical Applications (OFSAA) Data Model Document Generation Release 8.1.x or the OFSDF Data Model.

36.4 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 8.1.2.0.0 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'.

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:
   - Datastore Type - Select the appropriate datastore type from the list.
   - Datastore Name - Select the appropriate datastore name from the list.
   - IP address - Select the IP address from the list.
   - 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 do not see this in the list, contact My Oracle Support)
   - 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:**
>
> Execute this DT for each year for which data is present in the source table.

36.5 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

Access the execution log on the application server in the following directory:

`ftpshare/logs/<Run_Date>/FSDFINFO/LOAD DATA`

The file name contains the batch execution ID. The error log table in the atomic schema is as follows:

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.

37
Recommendation for Backdated Run

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.

37.1 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.

37.2 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.

37.3 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.

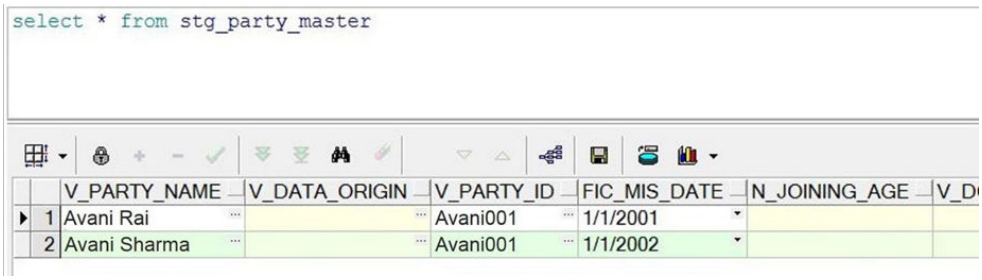
37.4 Backdated Run Execution using Latest Record Indicator (LRI) batch

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, follow these steps:

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.

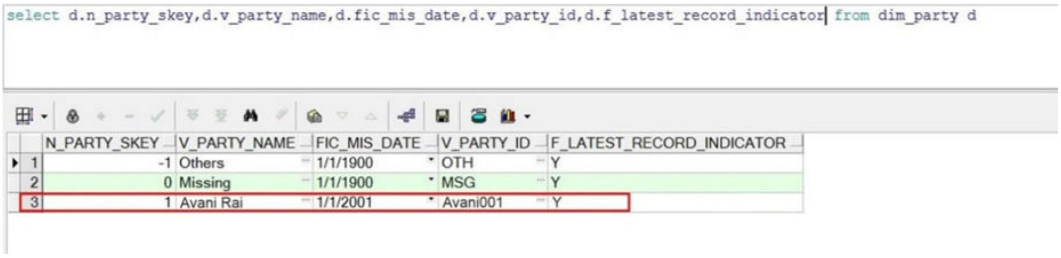
Figure 37-1 Sample of adding records for the backdated Run execution using LRI



> **Note:**
>
> Ensure that V_PARTY_ID is same in this step and the LRI batch execution step.

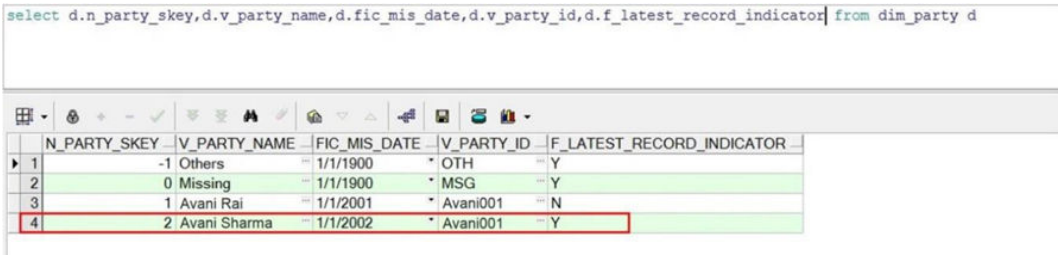
2. Execute the SCD batch to add the required records from Master table to its Dimension table with the required FIC_MIS_DATEs. For example, execute SCD batch to add records from STG_PARTY_MASTER to DIM_PARTY. The resultant data record in the DIM_PARTY is as shown below:

Figure 37-2 Sample after executing the SCD batch for first record for the backdated Run execution using LRI



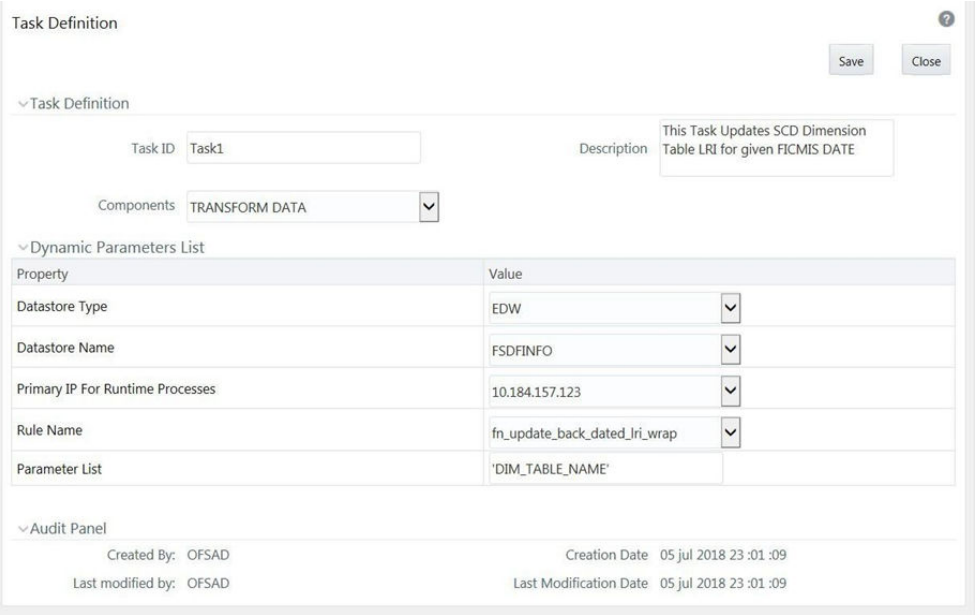
- After adding the first record to DIM_PARTY for the FIC_MIS_DATE 1/1/2001, the flag F_LATEST_RECORD_INDICATOR associated with this data record is set to Y.
- After adding the second record to DIM_PARTY for the FIC_MIS_DATE 1/1/2002, the flag F_LATEST_RECORD_INDICATOR associated with this data record is set to Y, and the flag

Figure 37-3 Sample after executing the SCD batch for the second record for the backdated Run execution



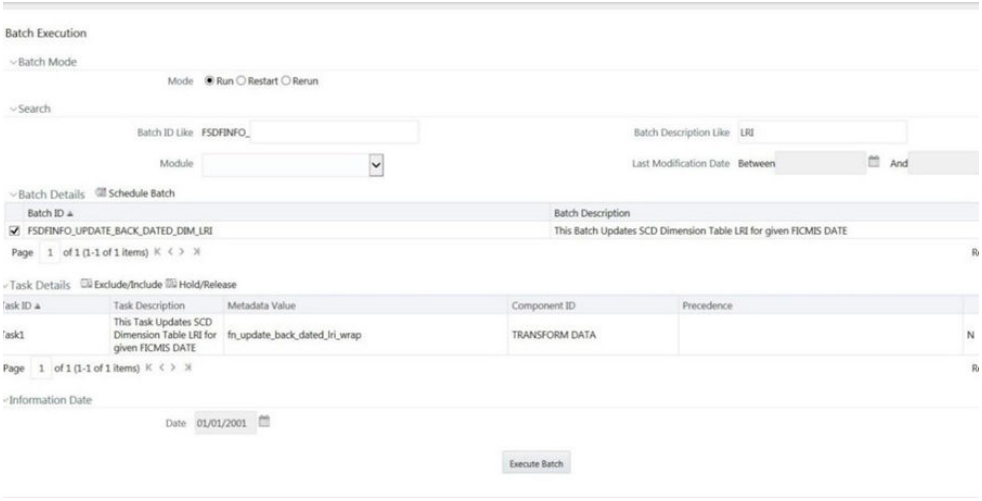
3. In **Batch Maintenance**, add a **Task** for the LRI Dimension table. In the **Parameter List** field, mention the Dimension table name ('DIM_TABLE_NAME') for which Backdated Run using LRI must be executed. In this example, 'DIM_TABLE_NAME' = 'DIM_PARTY'.

Figure 37-4 Add Task for the LRI table



4. In the Batch Execution window, execute the UPDATE_BACK_DATED_DIM_LRI batch for different FIC_MIS_DATEs.

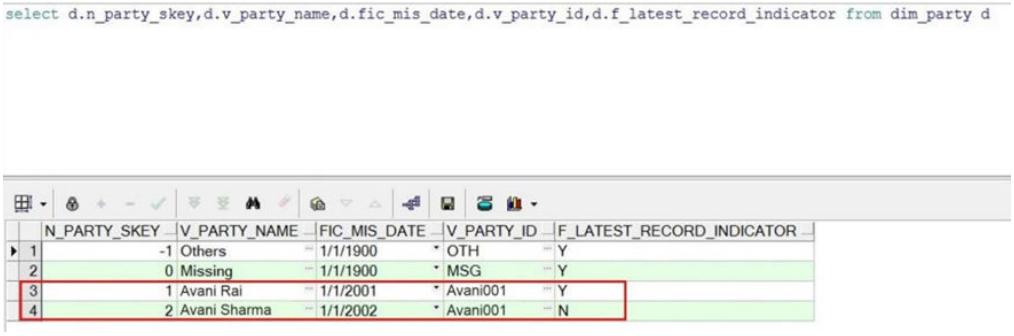
Figure 37-5 Execute the LRI batch



The following are the examples of data record after running the LRI batch with different FIC_MIS_DATEs:

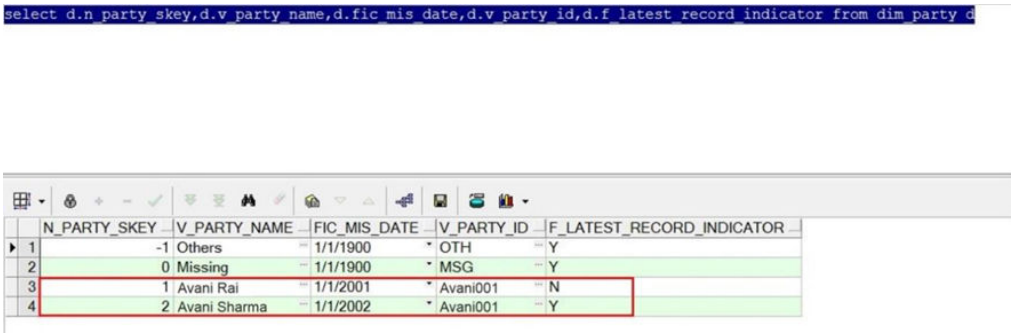
- The image depicts execution of the UPDATE_BACK_DATED_DIM_LRI batch for a backdated record for its FIC_MIS_DATE. Therefore, this record is the Latest Record. In the following image, the LRI batch is executed for the data record with historical FIC_MIS_DATE 1/1/2001. Therefore, the flag F_LATEST_RECORD_INDICATOR 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.

Figure 37-6 Sample data record after running the LRI batch with a different FIC_MIS_DATE



- 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.

Figure 37-7 Sample data record after running again LRI batch with a different FIC_MIS_DATE



5. Monitor the status of the batch in the Batch Monitor screen.

38
Using OFSDF

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 explain how the self-documenting erwin file includes the data dictionary and Download Specifications within erwin itself.

In addition, the Extending OFSDF Physical Data Model section has guidelines for customization and designing the Staging and Results Area of Physical Data Model.

38.1 Delivery Mechanism

OFSDF being a collection of data model artifacts, includes both a readily deployable model (the OFSDF Physical Data Model). The data model (Physical) is delivered as erwin files. The OFSDF hence requires a license of the erwin Data Modeler application.

erwin is the current and only supported modeling tool to view and edit the model. Currently, the minimum versions of erwin supported is 2019R1 or a higher version.

> **Note:**
>
> OFS AAI supports data model upload for data models generated using erwin 2019R1 or a higher version.

38.2 Installing OFSDF

As detailed earlier, OFSDF requires the Oracle Financial Services Analytical Application Infrastructure release to deploy and operate.

See the Oracle Financial Services Data Foundation Application Pack Installation and Configuration Guide Release 8.1.2.0.0 for step-wise instructions how to configure and install OFSDF on an OFSAAI instance.

38.3 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.

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:

- Data Dictionary
- Download Specifications

For more information on Dimension Management and AMHM, refer to the Dimension Management chapter in Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0 and Dimension Load Procedure section in Oracle Financial Services Analytical Applications Data Model Utilities User Guide.

38.4 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 Oracle Financial Services Analytical Applications (OFSAA) Data Model Document Generation Release 8.1.x, which details how to extract the data dictionary from erwin section.

38.5 Download Specifications

As detailed in the staging area section, the mapping from the Staging Data Model to use cases, called the Download Specification (My Oracle Support) 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 Oracle Financial Services Analytical Applications (OFSAA) Data Model Document Generation Release 8.1.x, which details how to extract the data dictionary from erwin section.

38.6 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. When 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.

38.6.1 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:

- 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.
- OFSDF product management to evaluate if the enhancement request is applicable more broadly to other customers and if the change should in fact is to be taken as a design requirement for subsequent releases.

> **Note:**
>
> OFS AAI supports data model upload for data models generated using erwin 2019R1 or a higher version.

38.6.2 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).

38.6.3 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

shouldbe 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.

38.6.4 Upgrading Data Model

The model upgrade process is achieved through the erwin Model Compare and Merge utility. See the Oracle Financial Services Analytical Applications (OFSAA) Data Model Extension Guidelines Document Release 8.1.x for erwin documentation for details on Menu options, process of comparing, and merging models.

39
Key Business Data Use Cases

This chapter provides information about key business data use cases for the Oracle Financial Services Data Foundation application.

39.1 Hypothecated Deposits

Hypothecated Deposit is the sum of installments paid to a deposit account, which is netted against loan account. The amount paid to the deposit account does not reduce immediately to pay off the loan installment, however, it accumulates until the total principal and interest amount of the loan is reached. The amount is paid in full and loan account is closed only after deposit balance reaches the total value of the loan. Hypothecated deposits must be netted against the related loans. Deposits, which serve as collateral for loans, are not considered hypothecated deposits.

39.1.1 Usage

Two accounts are captured when handling Hypothecated Deposits. They are:

1. Loan Account with its terms.
2. Deposit Account receiving payment, which is adjusted subject to terms, to repay loan.

These three outcomes occur when loan repayment is made using Hypothecated Deposits:

1. Debtor of the Deposit Account makes payments.
2. Loan continues to accrue interest or as per terms.
3. When deposit amount equals to loan amount, deposit proceeds are transferred to the Loan Account, and then the accounts are closed.

39.1.2 Inferences

These are inferences based on the Hypothecated Deposits usage:

- Deposit Account operates as a normal Deposit Account.
- Loan continues to operate as a normal Loan Account.
- The core banking or source system executes deposit proceeds, which is used to repay loan, as per the configured rules.
- Until the closure of loan, deposit functions as a mitigant such that, in the event of customer defaulting, the deposit balance is used to repay the loan.

39.1.3 Handling Hypothecated Deposits in OFSDF

The process of adjusting deposit proceeds to repay the Loan Contract is a part of the Deposit and Loan Transaction tables. To handle the Hypothecated Deposits in OFSDF, perform these steps:

1. You can continue to:
   a. Use the Term Deposits table to store deposits. Additionally, perform these tasks:
      i. Set the column STG_TD_CONTRACTS.V_OFFSET_ACCT_NBR to display the Loan Account Number.
      ii. Set the flag column STG_TD_CONTRACTS.F_LIEN_MARKED_IND to True or Y.
      iii. Set the flag column STG_TD_CONTRACTS.F_PRODUCT_LINKED_DEPOSIT_FLAG to True or Y.
   b. Use the Loan Contracts table to store loans.
2. Additionally, store deposit as mitigant in the Mitigant table and perform this task:
   - Set the flag column STG_MITIGANTS.F_CONSENT_TO_ADJUST_MATURITY to True or Y.

39.2 Credit Lines and Commitment Contracts

This section provides information about Credit Lines and Commitment Contracts.

39.2.1 Commitment Contract

Commitment is an assurance provided by the banker to his customer, given that the customer adheres to pre-defined conditions and provides necessary documentation. Then the customer can take up a Credit facility for an amount and/or rate.

39.2.2 Credit Line

Credit Line is the materialized result of commitment and therefore, becomes a Facility.

39.2.2.1 Differences and Similarities between Credit Lines and Commitment Contracts

The differences and similarities between Credit Lines and Commitment Contracts are explained in the following sections.

39.2.2.1.1 Differences between Credit Lines and Commitment Contracts

Table 39-1 Differences between Credit Lines and Commitment Contracts

Credit Line	Commitment Contract
Credit Line utilized is subject to interest accrual.	Does not include the concept of interest.
Subject to credit risk.	Bank makes provisions to get out, if they sense credit risk, as per conditions of the contract. There is no provision to hold capital or loan for a Commitment Contract except if the bank has no provision to back out as per terms of contract.

Table 39-1 (Cont.) Differences between Credit Lines and Commitment Contracts

Credit Line	Commitment Contract
A credit facility taken up in normal course of business.	Customer going in for a tender requires assurance on availability of Credit at a certain rate to quote a price with certain profitability. Therefore, the customer takes a Commitment Contract before they quote for the tender.
Consists of limits associated with it and limits may include one or multiple interest tiers applicable.	Commitment Contract must be mapped to a Credit Line to ensure that a customer or party level credit exposure caps are adhered. Commitment Contract can be provided to retail or SME without credit lines. It is similar to a pre-approved loan, which is valid for certain terms and date agreed. However, a fee or periodic fee must be paid to keep up the commitment.
Can be utilized by the customer under agreed conditions, without intervention of the banker.	Can be utilized by the customer after going through an abridged credit appraisal process such as providing updated income statements, KYC documents, POA etc., before any credit is extended.
Credit Line provided to a customer is summed up to understand customer or party level exposures.	Commitment Contracts roll into one of the Credit Lines, which is used to estimate total exposure to a customer.
Credit Line is factual.	Commitment Contract is only a commitment.

39.2.2.1.2 Similarities between Credit Lines and Commitment Contracts

Fees must be paid for both Credit Lines and Commitment Contracts. They must be considered for cash flow forecast.

39.3 Handling Participated/Syndicated Contracts

Participated/Syndicated contracts are applicable for Stage Loan Contracts, Stage Letter Of Credit Contracts, Stage Bill Contracts, Stage Leases Contracts product processors. Contract related details are expected in the product processors and credit facility details are in Stage Credit Facility Details. Share details of member banks are expected to be sourced to Stage Credit Participation Details table. In the product processors net shares (bank's contribution for the participated/syndicated contract) can be populated directly as a download or can be derived within OFSDF if the Gross Amounts (total amount of the participated/syndicated contract) are populated in the Product Processor tables. Account Or Contract Number of Product Processors and Syndicated Account Number of Stage Credit Participation Details are expected to be same for a Participated/syndicated contract. When the Bank is acting as Lead Bank, an additional record is expected for lead bank in Stage Credit Participation details with the Net share amounts. A Setup parameter is used to indicate whether net share amounts are provided as download or gross amounts in the product processors. If Gross amounts are provided then Net share will be calculated by netting the Gross amounts in Product Processors with the Sum of Member bank shares from Stage Credit Participation Details.

- Setup parameter PARTCPN_GLOBAL_AMT_FLAG need to be set as Y incase Gross amount is provided in the Product Processor tables and N for Net share amount. Default value is N.
- F_PARTICIPATION_FLAG flag needs to be set as Y in the Product Processors in case of Participated/Syndicated Contracts.

Following are the only attributes used for net share computation while populating Fact Common Account Summary from the respective Product Processor. Rest all attributes are expected to be populated with gross amounts.

- Stage Loan Contracts: n_sanctioned_limit, n_eop_bal, n_eop_curr_prin_bal, n_accrued_interest
- Stage Letter Of Credit Contracts: n_tot_comm_amt, n_accrued_interest, n_undrawn_amount
- Stage Bill Contracts: n_eop_bal, n_eop_curr_prin_bal, n_accrued_interest, n_undrawn_amount
- Stage Leases Contracts: n_eop_bal, n_sanctioned_limit, n_eop_curr_prin_bal, n_accrued_interest
- Stage Credit Facility Details: n_undrawn_amt, n_line_committed_amt

39.3.1 Net Share Computation Logic for Participated/Syndicated Contracts

- When PARTCPN_GLOBAL_AMT_FLAG is set to Y:
  Populate above mentioned amount of Fact Common Account Summary by subtracting total amount of the member banks associated with the same Syndicated Account Number or Syndicated Line Code of Stage Credit Participation Details from Product Processor Amounts.
- When PARTCPN_GLOBAL_AMT_FLAG is set to N:
  Populate above mentioned amount of Fact Common Account Summary with the amounts provided in Stage Credit Participation Details for Lead bank alone.

39.4 Sourcing Account Balances

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

39.4.1 Dynamic Sourcing of the Amount and Balance Columns in the Product Processors Independent of OFSDF

The listed tables are used to source the Amount and Balance attributes at the account level that are not available currently in the Product Processors (PP) tables.

Table 39-2 Account Balances Tables

Table Name	Logical Table Name	Table Type
STG_ACCOUNT_BALANCES	Stage Account Balances	Stage

Table 39-2 (Cont.) Account Balances Tables

Table Name	Logical Table Name	Table Type
FCT_ACCOUNT_BALANCES	Fact Account Balances	Result
STG_BALANCE_TYPE_MASTER	Stage Balance Type Master	Stage Master
DIM_BALANCE_TYPE	Balance Type Dimension	Dimension
DIM_STD_BALANCE_TYPE	Standard Balance Type Dimension	Seeded table for reclassification

When you need to add a new Amount and Balance related columns in the Stage tables and they are not available in the Stage PP tables, then you can add those Amount and Balance related columns in the Stage Account Balances table for dynamic sourcing. In the Account Balances tables, the Balance attributes are added in the form of row-level entries.

39.4.2 About Balance Type Dimension Table

Balance Type Dimension table name and its description are given here.

Table 39-3 Balance Type Dimension table name and its description

Logical Dimension Table Name	Dimension Table Description
Balance Type Dimension	This table stores the unique list of applicable Balance Types.

The mapping details for the Balance Type Dimension table is given here.

Table 39-4 The mapping details for the Balance Type Dimension table

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
1001	STG_BALANCE_TYPE_MASTER	Stage Balance Type Master	DIM_BALANCE_TYPE	Balance Type Dimension

39.4.3 About Standard Balance Type Dimension Table for Reclassification

This is the out of the box User Specific Dimension to Standard Dimension reclassification:

Table 39-5 The out of the box User-specific Dimension to Standard Dimension reclassification

User Specific Dimension		Standard Dimension	
DIM_BALANCE_TYPE	Balance Type Dimension	DIM_STD_BALANCE_TYPE	Standard Balance Type Dimension

39.4.4 About Account Balances T2T Population

Account Balances T2T and its description is given here.

Table 39-6 Account Balances T2T and its description

T2T Name	T2T Description
T2T_FCT_ACCOUNT_BALANCES	This T2T stores information related to the account level balances which are currently not available in the product processor tables.

The mapping details for the Account Balances T2T is given here.

Table 39-7 The mapping details for the Account Balances T2T

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_ACCOUNT_BALANCES	Stage Account Balances	FCT_ACCOUNT_BALANCES	Fact Account Balances	T2T_FCT_ACCOUNT_BALANCES

39.5 Sourcing Special Terms and Covenants

The special terms and covenants are a promise or a formal agreement between two or more parties that certain defined activities will be or will not be carried out, or that if certain conditions are met or if terms are violated, then corresponding events or actions are triggered.

Topics:

About Special Term Dimension Table

About Standard Special Term Dimension Table for Reclassification

About Special Terms and Covenants T2T Population

The special terms and covenants are precautionary measures put in place to handle the financial risks.

For special terms and covenants, use the following tables:

Table 39-8 The Account Balances Tables

Table Name	Logical Table Name	Table Type
STG_SPECIAL_TERMS_COVENANTS	Stage Special Terms Covenants	Stage
FCT_SPECIAL_TERMS_COVENANTS	Fact Special Terms Covenants	Result
STG_SPECIAL_TERM_MASTER	Stage Special Term Master	Stage Master
DIM_SPECIAL_TERM	Special Term Dimension	Dimension
DIM_STD_SPECIAL_TERM	Standard Special Term Dimension	Seeded table for reclassification

39.5.1 About Special Term Dimension Table

Special Term Dimension table name and its description are given here.

Table 39-9 Special Term Dimension table name and its description

Logical Dimension Table Name	Dimension Table Description
Special Term Dimension	This table stores the master list of special terms and covenants..

The mapping details for the Special Term Dimension table is given here.

Table 39-10 The mapping details for the Special Term Dimension table

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
1010	STG_SPECIAL_TERM_MASTER	Stage Special Term Master	DIM_SPECIAL_TERM	Special Term Dimension

39.5.2 About Standard Special Term Dimension Table for Reclassification

This is the out of the box User Specific Dimension to Standard Dimension reclassification:

Table 39-11 The out of the box User-specific Dimension to Standard Dimension reclassification

User Specific Dimension		Standard Dimension	
DIM_SPECIAL_TERM	Special Term Dimension	DIM_STD_SPECIAL_TERM	Standard Special Term Dimension

39.5.3 About Special Terms Covenants T2T Population

Account Balances T2T and its description is given here.

Table 39-12 Special Terms Covenants T2T and its description

T2T Name	T2T Description
T2T_FCT_SPECIAL_TERMS_COVENANTS	This T2T stores information such as, if the account or credit line includes any pre-conditions applicable for funding, repayment or for any other economic activates similar to change in the interest rate and so on.

The mapping details for the Special Terms Covenants T2T is given here.

Table 39-13 The mapping details for the Special Terms Covenants T2T

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_SPECIAL_TERMS_COVENANTS	Stage Special Terms Covenants	FCT_SPECIAL_TERMS_COVENANTS	Fact Special Terms Covenants	T2T_FCT_SPECIAL_TERMS_COVENANTS

40
Data Quality Rules Execution

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.

40.1 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

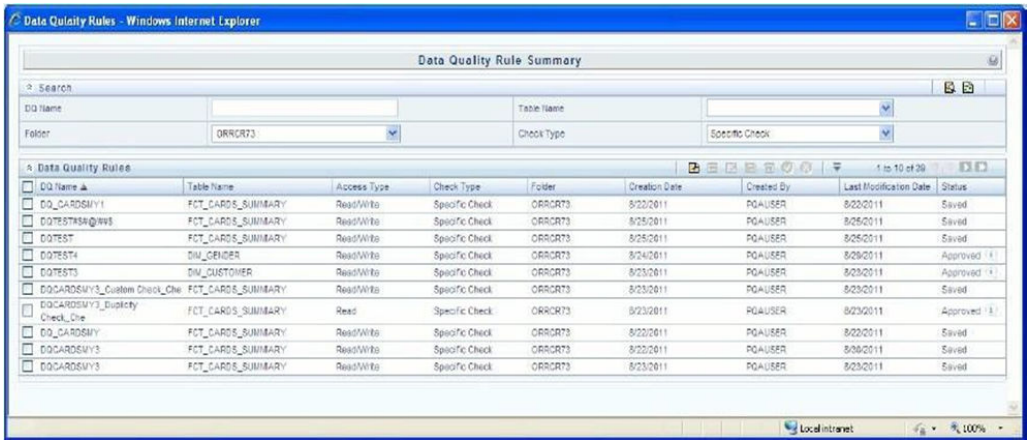
40.1.1 Data Quality Summary

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 40-1 Data Quality Rule Summary page



The Data Quality Rule Summary page 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.

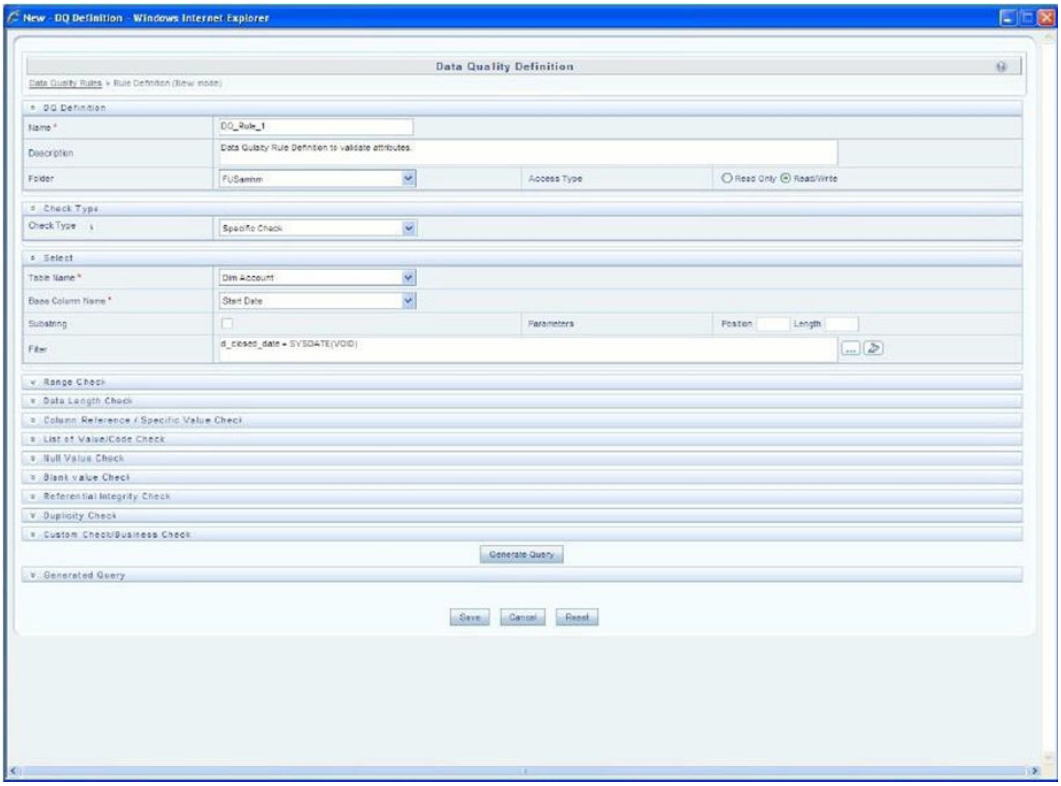
40.1.2 Create Data Quality Rule

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 page, follow these steps:

1. Click **Add** 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** page is displayed.

Figure 40-2 DQ Definition page



2. In the DQ definition section, perform the following actions:
   - 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  and define the Filter condition using the Specify Expression page.
- 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, see the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.

> **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 actions:
- 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  and define the Filter condition using the Specify Expression page.
- Click **Add** in the Condition grid. The **Specify Expression** page 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

> **Note:**
>
> 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 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 40-1 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: · 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. · If not, you are alerted with a message indicating that No Code values exists for the selected base column.
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 Severity from the drop down list.

You can also add multiple assignments by clicking Add in Assignment grid.

> **Note:**
>
> Minimum of one condition needs to be defined to save the Rule.

Click **Save**. The defined Data Quality Rule definition is displayed in the **Data Quality Rule Summary** page with the status as ?Saved?.

40.1.3 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 page, follow these steps:

- Select the check box adjacent to the required DQ Name.
- Click View from the Data Quality Rules toolbar.

The **DQ Definition** page displays the details of the selected Data Quality definition. The Audit Trail section at the bottom of DQ Definition page displays metadata information about the Data Quality Rule defined.

40.1.4 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** from the **Data Quality Rules** toolbar. The Edit button is disabled if you have selected multiple DQ Names. The **DQ Definition** page is displayed. Update the details as required. For more information, see Create Data Quality Rule section.
3. Click Save to update the changes.

40.1.5 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** page, follow these steps:

1. Select the check box adjacent to the required **DQ Name** in the list whose details are to be duplicated.
2. Click **Copy** from the Data Quality Rules tool bar. Copy button is disabled if you have selected multiple check boxes. The **DQ Definition** page 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** page with the status as ?Saved?.

40.1.6 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** page. User needs to 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** page, follow these steps:

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 actions:

- To Approve the DQ definition, click Approve. The User Comments page 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. The User Comments page 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 or 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** page. You can mouse? over i button to view the Approver comments.

40.1.7 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** page.

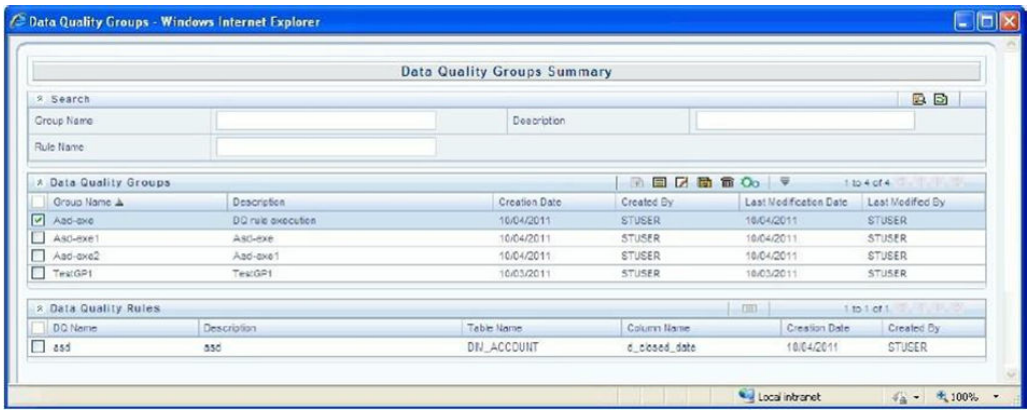
1. Select the check box adjacent to the required DQ Name whose details are to be removed.
2. Click **Delete** from the **Data Quality Rules** toolbar.
3. Click **OK** to confirm deletion.

40.2 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 40-3 Data Quality Groups Summary page



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.

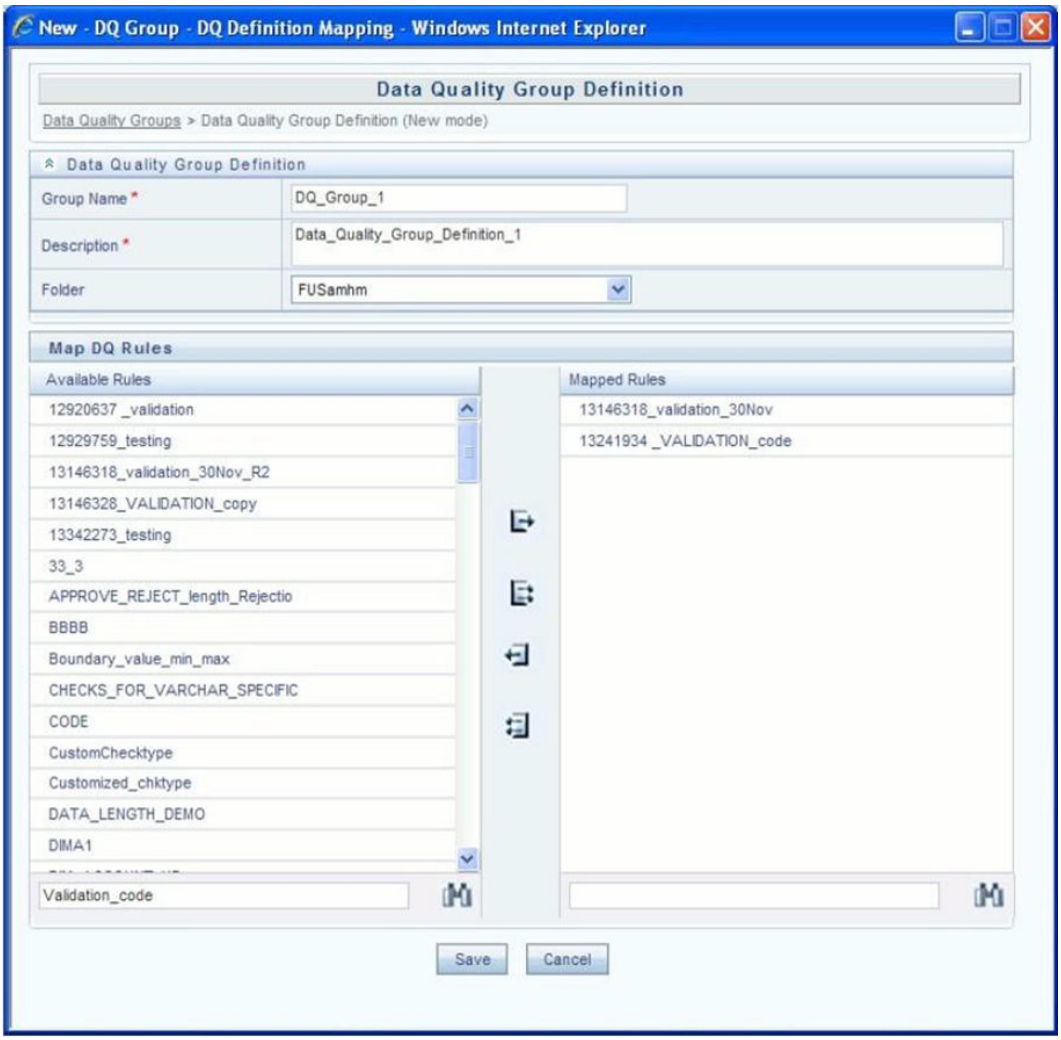
You can also make use of Search and Pagination options to search for a DQ Group definition based on Group Name, Description, or Rule Name and view the existing DQ Group definitions within the system.

40.2.1 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.

1. Click **Add** 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** page is displayed.

Figure 40-4 Data Quality Group Definition



2. In the Data Quality Group Definition section, perform the following actions:
   - 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 actions:
   - 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** page and can be executed for processing.

40.2.2 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 page. You can also execute a DQ Group in the Batch Execution page of the Operations module.

To execute a DQ Group in the Data Quality Rule Summary page, follow these steps:

1. Select the checkbox adjacent to the required **Group Name**.
2. Click **Execute** from the Data Quality Groups tool bar. Execute button is disabled if you have selected multiple check boxes. The **Group Execution** page is displayed.
3. In the **Batch Details** section, perform the following actions:
   - Select the **MIS Date** using the Calendar. MIS Date refers to the date with which the data for the execution is 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.

4. 4. 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%.
5. Click **Execute**. A confirmation message is displayed and the DQ Group is scheduled for execution. After the DQ Group is executed, you can view the details of the execution along with the log information in the **View Logs** page. For more information, see View Data Quality Group Summary Log section.

40.2.3 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** page, follow these steps:

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** toolbar. The **Data Quality Group Definition** page displays the DQ definition details.

40.2.4 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 page, follow these steps:

1. Select the check box adjacent to the required **Group Name**.
2. Click **Edit** from the **Data Quality Groups** toolbar. The **Data Quality Group Definition** page is displayed.

3. Update the details and click **Save** to update the changes.

40.2.5 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 page, follow these steps:

1. Select the check box adjacent to the required **Group Name** in the list whose details are to be duplicated.
2. Click **Copy** from the Data Quality Groups tool bar. Copy button is disabled if you have selected multiple check boxes. The **Data Quality Group Definition** page 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** page.

40.2.6 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 page, follow these steps:

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**. The **View Logs** page is displayed.
4. In the **View Logs** page, 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 in the Group Execution details tool bar to reset the selection.

40.2.7 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** page.

To delete, follow these steps:

1. Select the check box adjacent to the required **Group Name** whose details are to be removed.
2. Click **Delete** from the Data Quality Groups tool bar.
3. Click **OK** in the information dialog to confirm deletion.

40.2.8 Data Quality Rules For Staging Tables

Data Quality (DQ) Rules are framed and created based on Staging Tables. Each rule is based on specified staging table column Specific Check or table Generic Check. The rules created for each of the tables are detailed in the DQ Check Rules sheet of the Changelog Summary document of this release.

In Specific Check, a particular column is checked based on rule's predefined checks, where as in Generic Check any columns are not specified. Generic Check is useful if you have a check which is not Specific or you use IF-ELSE conditions or CASE statements.

40.2.9 Data Quality Groups for Staging Tables

To execute any DQ rule, a Group must be created and the rules for execution has to be mapped with this group. For each staging table, a group has to be created and rules for these particular table have to be mapped under this Group.

For more information regarding Group Execution, refer to the section Execute Data Quality Group.

The DQ Group Mapping sheet of the Changelog Summary document of this release displays the total groups and corresponding Rules mapped to that Group.

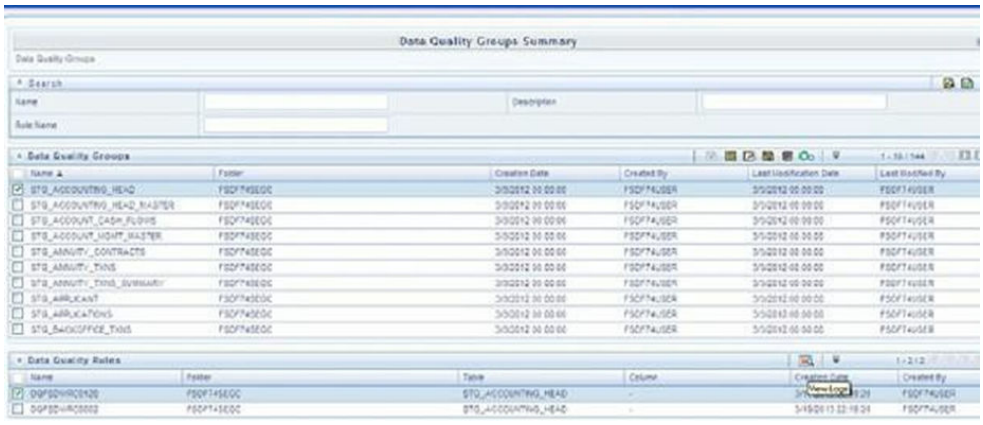
40.2.9.1 DQ Group Execution

You can execute the DQ Rules either from **Data Quality Group Summary** page or via a Batch execution.

To execute the Data Quality Groups from the **Data Quality Groups Summary** page, follow these steps:

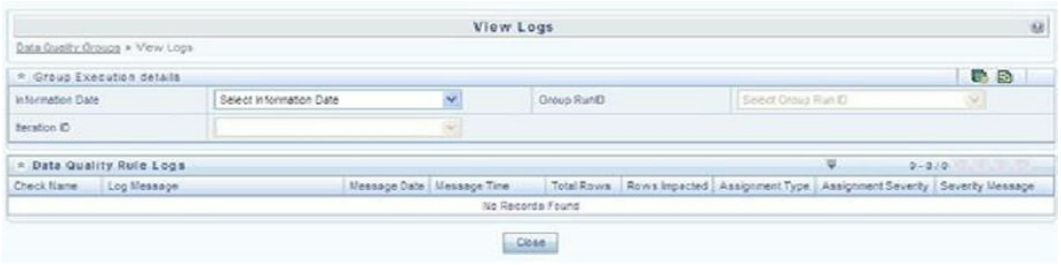
1. Select the check box adjacent to the Group, you want to execute. The corresponding DQ Rules are displayed in Data Quality Rules grid.

Figure 40-5 Data Quality Groups Summary page



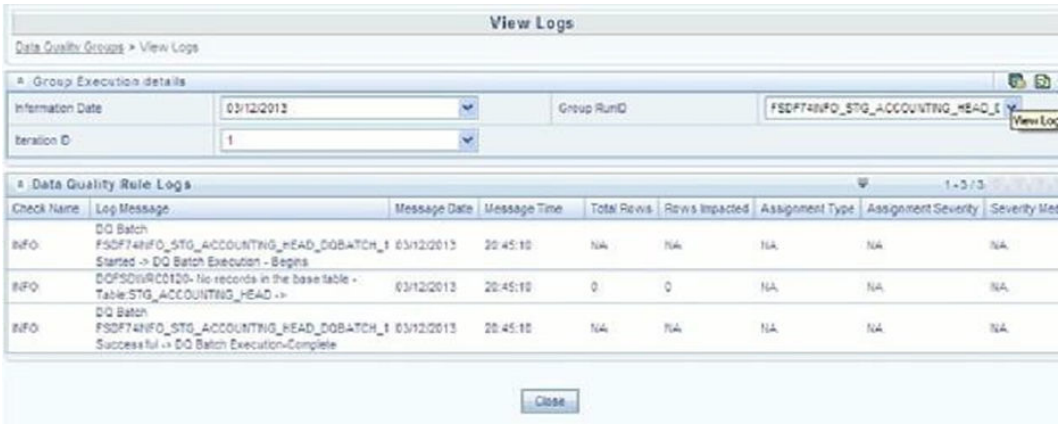
2. Click **View Logs** button. The **View Logs** page is displayed.

Figure 40-6 View Logs page



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**. The details of the selected Group Execution are displayed.

Figure 40-7 View Logs page with the details of the selected Group Execution



40.2.9.2 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 40-2 OFSDF Batches that needs to be executed

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>_COLLEC? TION_AND_RECOVERY	Data Quality batch for Collection and Recovery tables

Table 40-2 (Cont.) OFSDF Batches that needs to be executed

V_BATCH_ID	V_BATCH_DESCRIPTION
<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_AC? COUNTING	Data Quality batch for GL and Accounting group
<Infodom>_LRM	Data Quality batch for LRM tables
<Infodom>_MAR? KET_RISK	Data Quality batch for MARKET RISK tables
<Infodom>_MASTER	Data Quality batch for master tables
<Infodom>_MISCELLA? NEOUS	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>_PRO? DUCT_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>_SECURITIZA? TION	Data Quality batch for Securitization tables
<Infodom>_TRANSAC? TIONS	Data Quality batch for Transactions tables
<Infodom>_TRANSAC? TION_SUMMARY	Data Quality batch for Transaction Summary tables

The DQ Batches and corresponding groups are detailed in the RUN_CHART_SUMMARY sheet of the Runchart document of this release.

For more information regarding Batch Execution, refer to Operations module of Oracle Financial Services Analytical Applications Infrastructure User Manual.

41

Compare Data Model Reports and Generate Data Model Difference Report

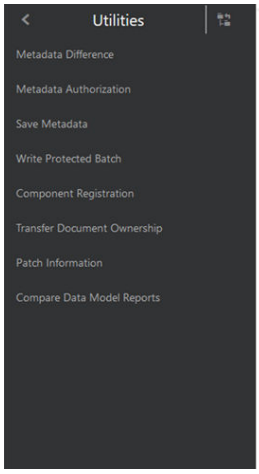
The Data Model Report can be extracted from the erwin Data Modeler application.

41.1 Extract the Data Model Report from the erwin Data Modeler application

To extract the Data Model Report from the erwin Data Modeler application, follow these steps:

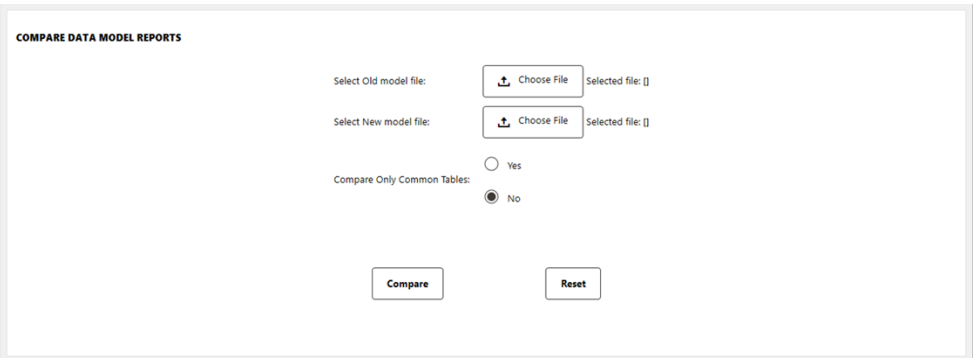
1. From **OFSAA Home**, select **Oracle Financial Services Data Foundation**, select **Common Components**, select **Utilities**, and then select **Compare Data Model Reports**.

Figure 41-1 Navigation to the Compare Data Model Reports page



2. The **Compare Data Model Reports** page is displayed.

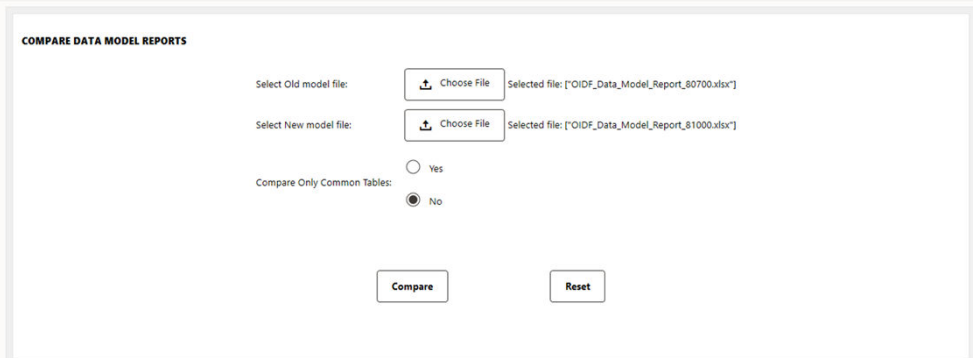
Figure 41-2 Compare Data Model Reports page



3. To browse and select the old data model report excel file, use **Choose File** associated with the Select Old model file label. To browse and select the new data model report excel file, use **Choose File** associated with the Select New model file label respectively. To select a different Data Model Report, click **Reset**.

> **Note:**
>
> The old data model report Excel file and new data model report Excel file must be in the same format and the file extension must be .xlsx.

Figure 41-3 Select the old and new data model reports



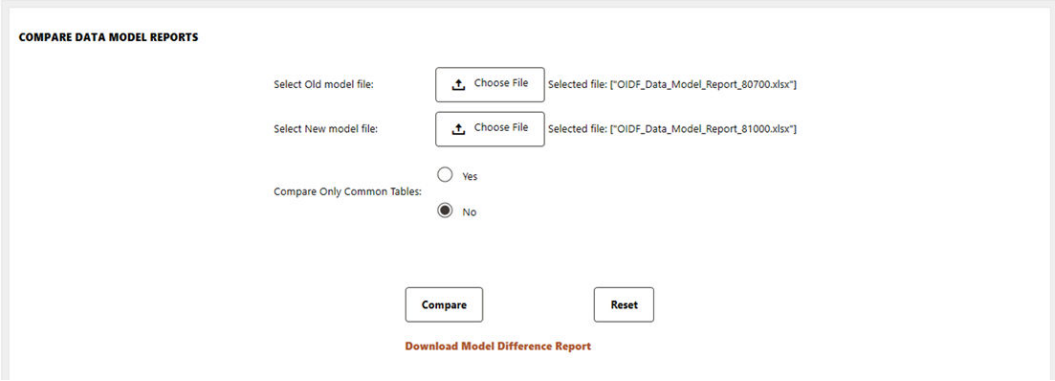
The Compare Only Common Tables option is No by default. Select Yes only if required.

> **Note:**
>
> Compare Only Common Tables with option Yes is used when comparing the Data Model Report of different OFSAA applications. The Report field is different for each OFSAA application and only common fields are required for comparison.

4. To compare the old and new data model report, and generate the Data Model Difference Report, click **Compare**. The **Download Model Difference Report** link

appears. To download the Data Model Difference Report, click the **Download Model Difference Report** link.

Figure 41-4 Generate the Data Model Difference Report



5. Save the file.

42

APPENDIX A: Naming Conventions Used in OFSDF Data Model

This Appendix chapter explains the various naming conventions used in OFSDF Physical Data Model. In addition, the domains in PDM is also listed with their descriptions.

This appendix covers the following topic:

· Naming Conventions Used in OFSDF PDM

42.1 Naming Conventions Used in OFSDF 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) Physical Data Model (PDM).

· OFSDF Physical Data Model Naming Conventions

· Domains (PDM)

42.1.1 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 Data Modeler application 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

42.1.1.1 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 following table.

Table 42-1 Prefixes and Suffixes for tables in the Logical View Mode of the Physical Data Model

PDM Area	Table Type	Prefix	Suffix	Example
Staging (Data Sourcing)	All Tables	Stage		Stage Internet Transactions
	Transaction Tables		Transactions	Stage Card Settlement Transactions
	Master Tables		Master	Stage Product Master
Results (Datamart)	All Fact Tables	Fact		Fact Process Cashflow
	All Dimension Tables		Dimension	Account Dimension

42.1.1.2 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 42-2 Prefixes and Suffixes for tables in the Physical View Mode of the Physical Data Model

PDM Area	Table Type	Prefix	Suffix	Example
Staging (Data Sourcing)	All Tables	STG_		STG_LOAN_CONTRACTS
	Interface tables for data from MDM and core banking systems		_INTF	STG_ORG_UNIT_ATTR_INTF
	Transaction Tables		_TXNS	STG_CARDS_SETTLEMENT_TXNS
	Master Tables		_MASTER	STG_CUSTOMER_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.

Table 42-3 Column Name Prefix to indicate the Column Datatype

Column Data Type	Prefix
Varchar	V_
Number	N_
Date	D_
Flag	F_

Column name suffix for common classes of columns is in the following table.

Table 42-4 Column Name Suffix to indicate the Column Class

Column Class	Suffix
Method	_METHOD
Percentage	_PCT
Rate	_RATE
Balance	_BAL or _BALANCE
Amount	_AMT or _AMOUNT
Term	_TERM
Type	_TYPE
Frequency	_FREQ

In addition, frequently occurring keywords in column names may be abbreviated as shown in the following table.

Table 42-5 Abbreviated the frequently occurring keywords in the column names

Name	Abbreviated Form
Accrual	accr
Account	acct
Accounting Currency	acy
Address	addr
Adjustment	adj
Advance	adv
Amount	amt
Application	app
Average	avg
Balance	bal
Business	bus
Currency	ccy
Consolidation	cons
Customer	cust
Description	desc
Dimension	dim
Detail	dtl
Earnings at Risk	ear
End Of Period	eop

Table 42-5 (Cont.) Abbreviated the frequently occurring keywords in the column names

Name	Abbreviated Form
Error	err
Flag	flg
Frequency	freq
Future	fut
Forex	fx
Generation	gen
General Ledger	gl
Hierarchy	hier
History	hist
Local Currency	lcy
Line Of Business	lob
Maximum	max
Minimum	min
Mortgage	mort
Message	msg
Multiplier	mult
Number	num
Over Draft	od
Option	opt
Origination	org
Percent	pct
Payment	pmt
Prepayment	Ppmt
Product	prod
Source	src
Status	stat
Statistics	stats
Temporary	temp
Total	tot
Transaction	txn
Value at Risk	var
Value	val

42.1.2 Domains (PDM)

Domains are Logical data types that are attached to each column within the model. The following table lists the domains and their descriptions.

Table 42-6 Domains and their descriptions

Domain Name	Domain Description
Date	DATE
Timestamp	TIMESTAMP

Table 42-6 (Cont.) Domains and their descriptions

Domain Name	Domain Description
Number	NUMBER(10)
Amount	NUMBER(22,3)
Code	NUMBER(5)
Flag	CHAR(1)
Frequency	NUMBER(5)
ID	VARCHAR2(25)
Percent	NUMBER(10,6)
Percent_Long	NUMBER(15,11)
Phone_Fax_Number	NUMBER(15)
Rate	NUMBER(10,6)
Term	NUMBER(5)
Alphanumeric_Code	VARCHAR2(10)
Name	VARCHAR2(60)
Currency_Code	VARCHAR2(3)
Short_Description	VARCHAR2(60)
Description	VARCHAR2(255)
Account_Number	VARCHAR2(25)
System_Identifier	NUMBER(20)
Long_Description	VARCHAR2(4000)

43

APPENDIX B: Standard Data Expectations

This section provides information about the standard data expectations in the OFSDF Application Pack.

43.1 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.

43.2 Custom Reporting Line Codes

Custom values can be added in the DIM_REP_LINE table. DIM_REP_LINE custom range maximum value is 9110001050 for the N_REP_LINE_CD sequence.

44

APPENDIX C: How to Define a Batch

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.

44.1 Batch Definition

To create a batch from the OFSAAI Batch Maintenance page, follow these steps:

1. From the OFSAAI Home menu, select Operations, select Batch Maintenance.
2. In the **Batch Maintenance** page, select ?+? button from the **Batch Name** tool bar. The **New Batch Definition** page is displayed.
3. Enter the Batch details.

Table 44-1 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: · The Batch Name must be unique across the Information Domain. · The Batch Name must be alphanumeric and should not start with a number. The Batch Name must not exceed 41 characters in length. · The Batch Name must not contain the 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 the Batch Maintenance page with the specified Batch ID.

45

APPENDIX D: Template to Generate Data Dictionary and Download Specification for erwin 2019R1 or a higher version

OFSAA data models for this release have been designed and released on the 2019R1 or a higher version of erwin. However, if there is a requirement to upgrade to the latest version of erwin (2019R1 or a higher version series), the existing procedure to generate DL specification using the published report templates fails in the lower 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 2019R1 or a higher version, for all OFSAA data models qualified on OFSDF 8.1.2.0.0.

Upgrade the data model to 2019R1 or a higher version series before initiating generation of the DL specification.

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 OFSDF version 8.1.2.0.0 and erwin 2019R1 or a higher version.

1. Ensure that the erwin Data Modeler version is erwin 2019R1 or a higher version.
2. Ensure that macros are enabled in this excel before execution.
3. Open the OFSAA Data Model in erwin 2019R1 or a higher version before executing this template.
4. Click UDP selection Menu to launch the UDP Selection form.
5. Select the correct application UDPs (optional).
6. Click Generate Report.
7. On successful execution, the following message is displayed: Report generation is Complete.
8. The download specification and the data dictionary report are available in the OFSDF Download Specifications document in OFSAA_Download_Spec worksheet.

> **Note:**
>
> To generate report for two application UDPs such as BASEL_III_USA_ADVNCD and OR , you need to manually remove the extra characters from the UDP name in the data model else the execution throws an error.

Glossary

Index