Oracle Insurance Data Foundation Application Pack

User Guide

Release 8.1.2.0.0

Oct 2023

F51284-01

ORACLE
Financial Services

ORACLE

Oracle Legal Notices

International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

Third-Party Content, Products, and Services Disclaimer

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Pre-General Availability Draft Label and Publication Date

Pre-General Availability: 2023-*MM-DD*

Pre-General Availability Draft Documentation Notice

If this document is in public or private pre-General Availability status:

This documentation is in pre-General Availability status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

Oracle Confidential Label

ORACLE CONFIDENTIAL. For authorized use only. Do not distribute to third parties.

Revenue Recognition Notice

If this document is in private pre-General Availability status:

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your pre-General Availability trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described in this document may change and remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Documentation Accessibility

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

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Document Control

Version Number	Revision Date	Change Log
1.0	December-2021	Created the OIDF Application Pack Release 8.1.2.0.0 User Guide. Added the following sections: • Product T2T (Result Table) • Appliances and Articles, and Cellphone Insurance Tables • Aviation Insurance Tables
2.0	March-2022	The metadata design information is moved to the OIDF Metadata Design Document Release 8.1.x.
3.0	April-2022	Added the Cyber Liability Insurance Tables section.
4.0	May-2023	Added new features
5.0	October-2023	• Erwin file version updated for 8.1.2.4.0 • New Features updated

Table of Contents

1 Preface

Topics:

1.1 What is New in this Release of OIDF Application Pack

This section lists new features and changes in the Oracle Insurance Data Foundation (OIDF) Application Pack.

1.1.1 New Features

This section lists the new features described in this manual.

Table 1: New features in the OIDF Application Pack and their description

Feature Name	Feature Description
Surrender Charge Schedule	Surrender Charge Schedule is applicable for different insurance products. For example, in Life Insurance Contract or Annuity Contract, based on the policy age, surrender charges are applied.
Appliances and Articles, and Cellphone Insurance Tables	The Appliances and Articles Insurance protects the contents of the building from the damages caused as agreed upon in the Policy. The Appliances and Articles Insurance is a part of the Dwelling Insurance. The Cellphone Insurance covers the cost of cellphone repairs and parts replacement as agreed upon in the Policy.
Aviation Insurance Tables	The Aviation Insurance covers the physical and operational damages to different types of aircrafts and their parts. It also covers the resultant liabilities.
Cyber Liability Insurance Tables	Cyber Liability Insurance is an insurance policy that helps protect organizations from the consequence of cyberattacks and hacking threats such as data breaches, viruses, or other forms of malicious cybersecurity activity. It covers the financial losses due to cyberattacks or other tech-related risks, as well as privacy investigations or lawsuits following an attack. Cyber Risk Quantification: Evaluation of quantitative risk factors involves more sophisticated activities like review of the IT network infrastructure and internet usage patterns. Continuous security monitoring solutions help in understanding an organization's Cyber Risk exposure
Extension of Claims Feature	T2T modified: Stage Claim Details modified.
Subrogation and Salvage	New T2T : Stage Claim Subrogation Details at the granularity level of claim and subrogation instance.

Feature Name	Feature Description
	New SCD Stage Subrogation Master and Subrogation Dimension enhancement.
Climate Risk Insurance	Added structures and related metadata to OIDF to support analytics for Climate Risk Insurance.

1.1.2 Deprecated Features

There are no deprecated features in this release.

1.1.3 Desupported Features

There are no desupported features in this release.

1.2 Audience

The Oracle Insurance Data Foundation (OIDF) Application Pack User Guide is intended for Administrators, Business User, Strategists, and Data Analyst, who are responsible for installing and maintaining the following:

- OIDF Application Pack components
- OFSAA Architecture
- UNIX Commands
- Database Concepts
- The web server or web application server

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.

1.4 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:

- OHC Documentation Library for the OIDF Application Pack Release 8.1.x.x.x product documents:
  - Oracle Insurance Data Foundation Application Pack Installation and Configuration Guide Release 8.1.2.0.0
  - Oracle Insurance Data Foundation Application Pack Release Notes 8.1.2.0.0
- OHC Documentation Library for the OIDF Application Pack Release 8.1.x.x.x other documents:

  - Oracle Insurance Data Foundation Application Pack Data Protection Implementation Guide Release 8.1.x
  - 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 Insurance Data Foundation Application Pack Security Guide Release 8.1.x
  - Oracle Insurance Data Foundation Application Pack Cloning Reference Guide Release 8.1.x
- My Oracle Support for the OIDF Application Pack Release 8.1.x.x.x technical documents:
  - *Run Chart*
  - *Changelog*
  - *SCD Metadata*
  - *T2T Metadata*
- Oracle Financial Services Analytical Applications (OFSAA) Technology Matrix Release 8.1.2.0.0
- OHC Documentation Library contains the following OFS AAAI Application Pack Release 8.1.2.0.0 product 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
- OHC Documentation Library contains the following OFSAA Release 8.1.x generic documents:
  - Oracle Financial Services Analytical Applications Infrastructure Security Guide Release 8.1.x
  - Oracle Financial Services Analytical Applications Infrastructure Cloning Reference Guide Release 8.1.x
- 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

For information about the purpose of each of the OIDF Application Pack documents, see the About OIDF Documents section.

2 Introduction to OIDF

This section explains about Oracle Insurance Data Foundation (OIDF), its components, its relationship with Oracle Financial Services Analytical Applications Infrastructure (OFSAAI), and the key prerequisites to run OIDF.

Topics:

2.1 Overview

Oracle Insurance Data Foundation (OIDF) is an analytical data warehouse platform for the Insurance industry. OIDF 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

OIDF supports a variety of insurance business segments under Direct insurance contracts namely Life Policies, Health Policies, Annuities, Property and Casualty policies, and Retirement Policies and under indirect or reinsurance business segment it supports reinsurance held as well as reinsurance issued

OIDF is a comprehensive data management platform that helps Insurance Companies to manage the analytical data life cycle from sourcing to reporting, as a consistent platform and toolset.

2.2 Components of OIDF

OIDF consists of the following components, which are explained in additional detail in the next chapter:

Table 2: OIDF Components and their description

Criteria	Description
OIDF Analytical Warehouse Data Model	• OIDF Analytical Warehouse Data Model is a Physical Data Model that supports data sourcing and reporting related to key analytical use cases in the Insurance industry. • The Warehouse Model is a Physical Data Model is readily deployable, and consists of database object definitions, and additional supporting scripts. • It is organized into two distinct sets of tables based on purpose: ▪ The Staging Model: This model facilitates data sourcing from the Insurance provider's internal operational systems such as Policy or Contract systems, investments, Claims systems, Master Data Management systems, and so on. ▪ The Reporting Model: This model facilitates the storage of outputs from analytical tools, applications, and engines in a manner that is conducive to BI reporting. • OIDF Analytical Warehouse Model is typically deployed into production via a set of management tools called the Oracle Financial Services Analytical Application Infrastructure (OFSAAI). The AAI application is a separate product and is a prerequisite for OIDF (see the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0).
Supporting Scripts	Supporting Scripts are scripts provided as part of the OIDF package for basic operations such as internal data movement between the staging and reporting areas.

2.3 Relationship to Oracle Financial Services Analytical Applications

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

- Data Model
  - OIDF Staging Model provides the complete data sourcing foundation for OFSAA applications. All use case or application-specific input data requirements are captured as part of the Staging Data Model. OIDF Staging Model is a combination of all staging models supplied with each OFSAA application.
  - OIDF Reporting Model provides a complete reporting data model common to all the OFSAA Business Intelligence (BI) applications. This includes a single set of conformed dimensions and unified fact tables used for cross-functional reporting. OIDF Reporting Model is the superset of all the BI-application specific reporting models.
  - Synchronized Releases: Staging Model and Reporting Model, which are part of an OIDF release, are updated to reflect prior application-specific releases. This means that the latest release of OIDF reflects all prior application releases across OFSAA from a data model perspective, with respect to Staging Model and Reporting Model.

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

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

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

2.4 OIDF Prerequisite Components and Tools

The key prerequisites for running the OIDF application are listed as follows:

Table 3: Prerequisites for the OIDF Application Pack

Component	Provider	Purpose
Oracle Financial Services Analytical Applications Infrastructure version 8.1.2.0.0	Oracle	OFSAAI is the platform on which the OIDF application is deployed and operated. It represents the OIDF 'runtime' environment and consists of a number of tools used to manage the data lifecycle within OIDF, from sourcing to reporting*.
Oracle Database Enterprise Edition 19c	Oracle	OIDF is certified on Oracle Database releases 11gR2 and higher.
• erwin Data Modeler application 2019R1 or a higher version for 8.1.2.3.0 and earlier versions • erwin Data Modeler application 12.1 or higher version for 8.1.2.4.0 and later versions	Computer Associates (CA)	erwin is a Data Modeler application that provides a visual environment to manage the complex enterprise data environment.

*OFSAAI, the infrastructure platform consists of its own prerequisites and supporting documentation.

3 Understanding OIDF

This section explains the background of OIDF, its functional architecture along with the differences from traditional warehouse architecture. OIDF product package consists of the Analytical Data Warehouse model. The individual components of the Physical Data Model are explained in detail.

Topics:

3.1 Background

It is important to note that OIDF architecture differs from 'traditional' data warehouse architecture.

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

For more information, see the *Understanding OIDF* Section in the OIDF Metadata Design Document Release 8.1.x.

4 Using OIDF

This section details how the OIDF models are delivered and how they can be installed and configured into the required environment. The first two sections give you an understanding of the Delivery Mechanism and OIDF 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 Data Model section has guidelines for customization and designing the Staging and Results Area of Physical Data Model.

Topics:

4.1 Delivery Mechanism

OIDF being a collection of data model artifacts includes a readily deployable model (the OIDF Physical Data Model). The data model (Physical) is delivered as erwin files. The OIDF 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 version of Erwin Data Modeler application supported is 2019R1 or a higher version.

NOTE OFS AAI supports data model upload for data models generated using erwin. For information on compatible versions, see the corresponding release of the Oracle Financial Services Analytical Applications (OFSAA) Technology Matrix.

4.2 Installing OIDF

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

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

4.3 OIDF Supporting Documentation

The preceding sections have provided an overview of the organization of the OIDF and its several component data models. Appendix A explains the naming conventions used in the OIDF data model.

The OIDF 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 OIDF can be performed within erwin, and the updated documentation is retained in the file in the form of additional metadata.

The two key detailed artifacts of OIDF documentation that can be extracted from within the erwin Data Model are as follows:

- Data Dictionary
- Download Specifications

For more information on Dimension Management and AMHM, see the *Dimension Management* section in the 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.

4.3.1 Data Dictionary

The data dictionary for OIDF 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 the erwin section.

4.3.2Download 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 OIDF 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 the erwin section.

For the information about *Extending the OIDF Physical Data Model*, see the OIDF Metadata Design Document Release 8.1.x.

5 Application of Data Domain Browser in OIDF

This section provides information about Data Domain Browser usage in the Oracle Insurance Data Foundation application.

Topics:

5.1 Required Software Licenses to Use Data Domain Browser

The software licenses that must be installed to access and use Data Domain Browser for OIDF application are:

- Oracle Financial Services Data Integration (OFS DI) Application Pack v8.1.2.0.0 Minor Release. To find the installer patch ID, see the *Preparing for Installation* section in the Oracle Financial Services Data Integration (OFS DI) Application Pack 8.1.2.0.0 Installation Guide.
- Oracle Insurance Data Foundation Application Pack v8.1.2.0.0 Minor Release installer patch ID **33024795** is available at the My Oracle Support (MOS).

5.2 Introduction to Data Domain Browser

Data Domain Browser provides Data Foundation customers a User Interface to view and query the erwin Data Model over the physical data model reports that Data Foundation provides currently. In addition, Data Domain Browser also provides all the information like physical formats along with granularity and technical details such as data type. One can view the Data Domain Browser as a logical representation of the Data Foundation. This equips User Interface with the following factors to provide a pre-built grouping of entities and data elements to enhance search by the end-user:

- To help with how to search the data model.
- To understand the entity-relationship.
- To query the data model.

The features of Data Domain Browser are to help the user:

- To view the list of entities available in the Data Foundation in a pre-grouped format.
- To filter a particular entity or a set of entities using specific filter criteria.
- To view the data elements and their properties within a chosen entity.
- To search within the data elements using specific search criteria.

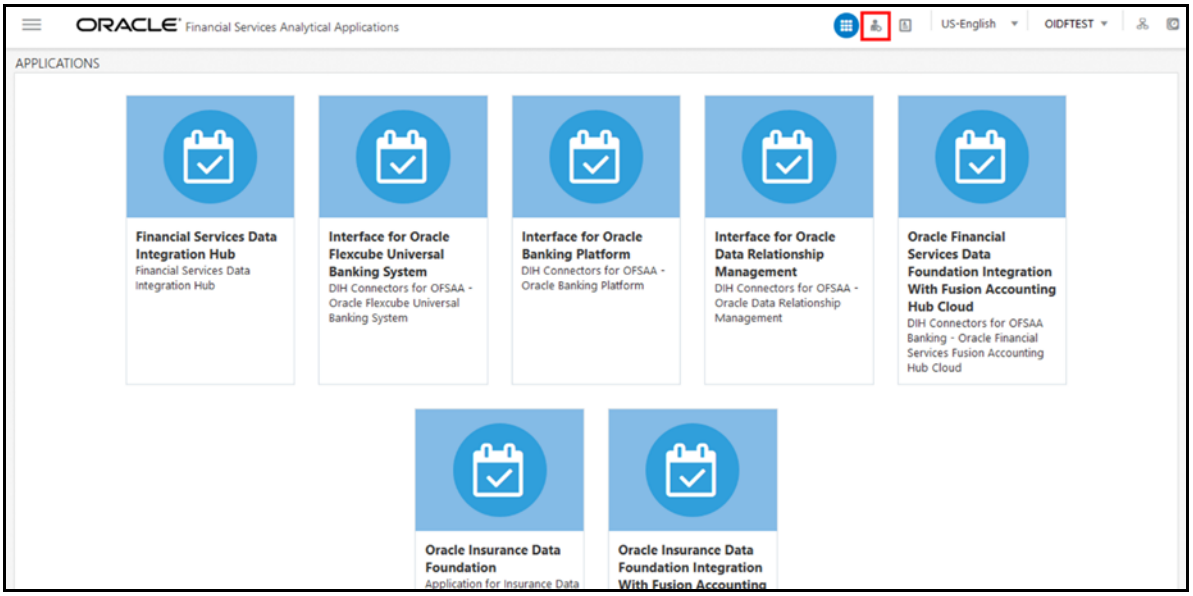
NOTE For complete information about Data Domain Browser and its User Interface, see the *Data Domain Browser* section in the OFS Data Integration Hub (DIH) User Guide Release 8.1.2.0.0.

5.3 Map the Data Domain Visualizer Role to the ETL Analyst (ETLADM) User

This section provides information about the Data Domain Visualizer access to the ETL Analyst user (with ETLADM Role Code). To map the Data Domain Visualizer access role to the ETL Analyst, follow these steps:

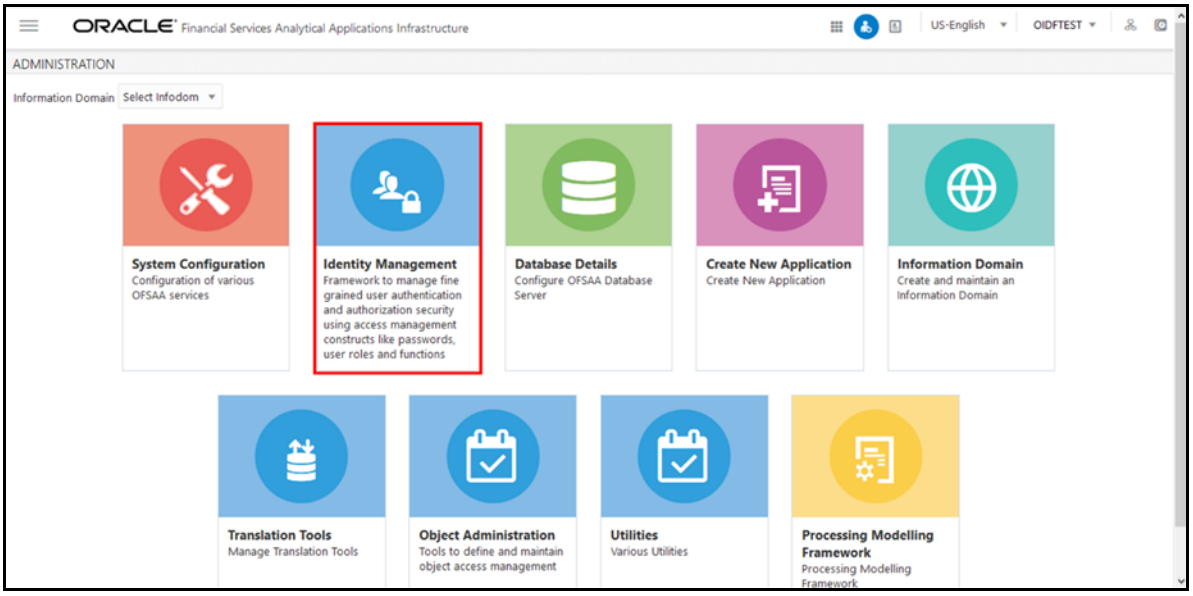
1. Log in to the OFSAA application as the System Administrator.
2. From **OFSAA Home**, click the administration icon.

Figure 1: OFSAAI landing page with OIDF and DIH applications



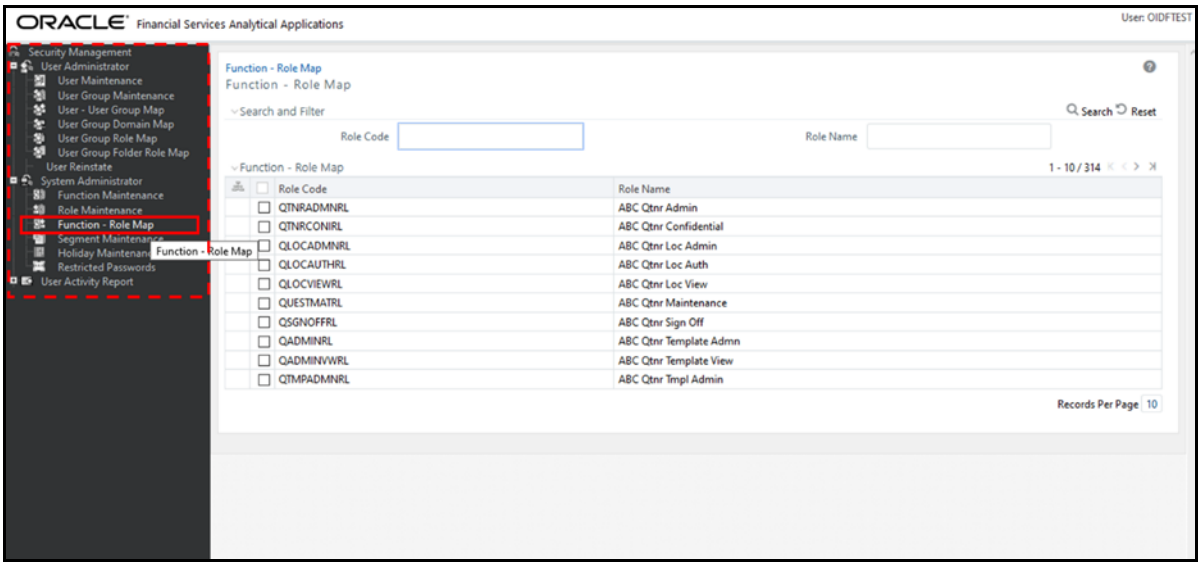
3. On the ADMINISTRATION page, select the **Identity Management** tile.

Figure 2: Select Identity Management



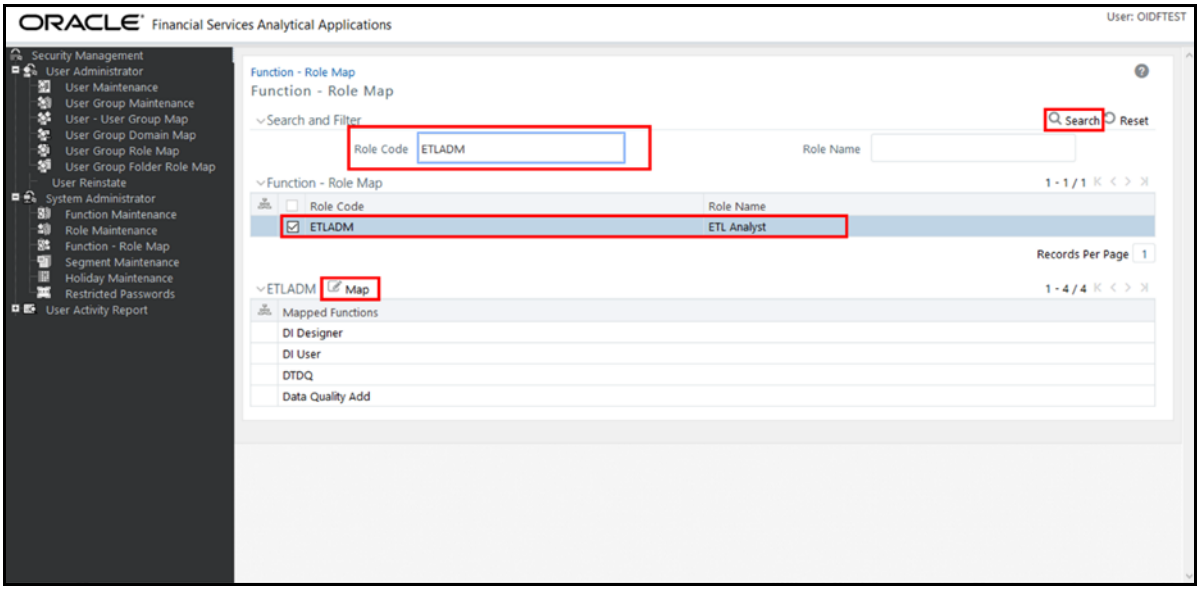
4. The **Security Management** menu is displayed on the LHS. Expand the **System Administrator** role and select **Function – Role Map** to open the **Function – Role Map** page.

Figure 3: Navigate to the Function – Role Map page



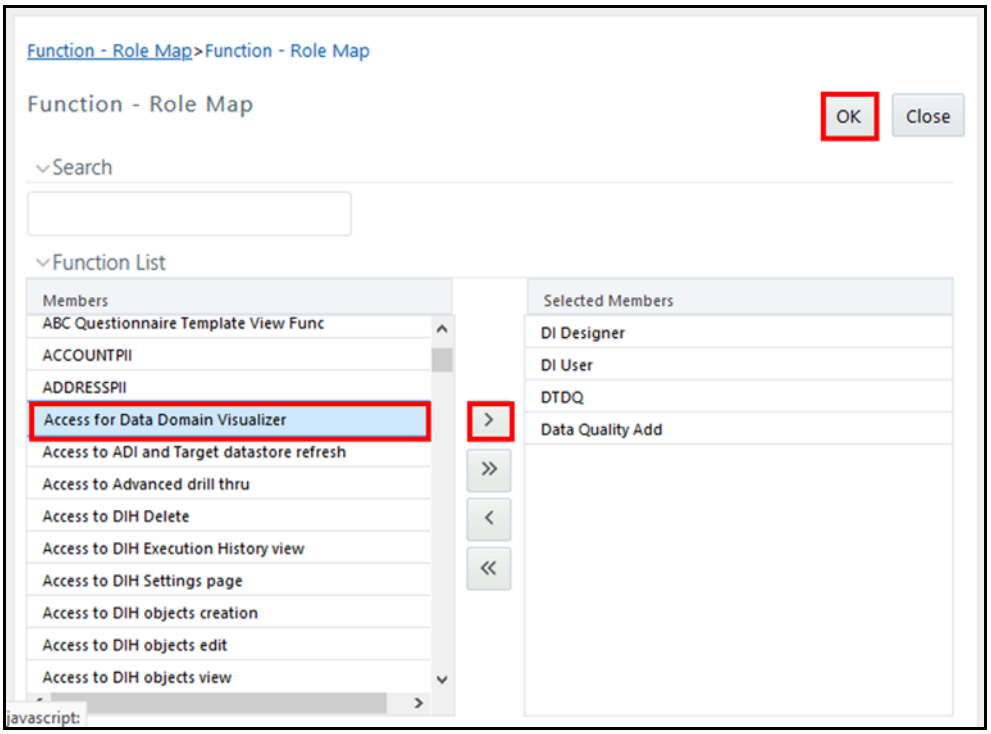
5. On the **Function – Role Map** page, in the **Role Code** field, search **ETLADM**, which is the Role Code for the ETL Analyst user. The ETLADM Role Code details are listed. Select **ETLADM**. The ETLADM mapped function details are displayed. Click the **Map** icon.

Figure 4: Search the ETLADM Role Code and select edit mapping



6. The **Function – Role Map** page opens. Search and select the **Access for Data Domain Visualizer** value in the **Members** list and move it to the **Selected Members** list. To confirm the selected mapping and save changes, click **OK**.

Figure 5: Select Access for Data Domain Visualizer for mapping



7. The following acknowledgment message is displayed when the mapping is successful: *Function – Role Map Saved Successfully*. Click **OK**.
8. On the **Function – Role Map** page, in the ETLADM mapped functions details, the newly mapped function **Access for Data Domain Visualizer** is listed.

For more information about Data Domain Browser in OIDF, see the *Application of Data Domain Browser in OIDF* Section in the OIDF Metadata Design Document Release 8.1.x.

6 Data Flow Process for Data Foundation

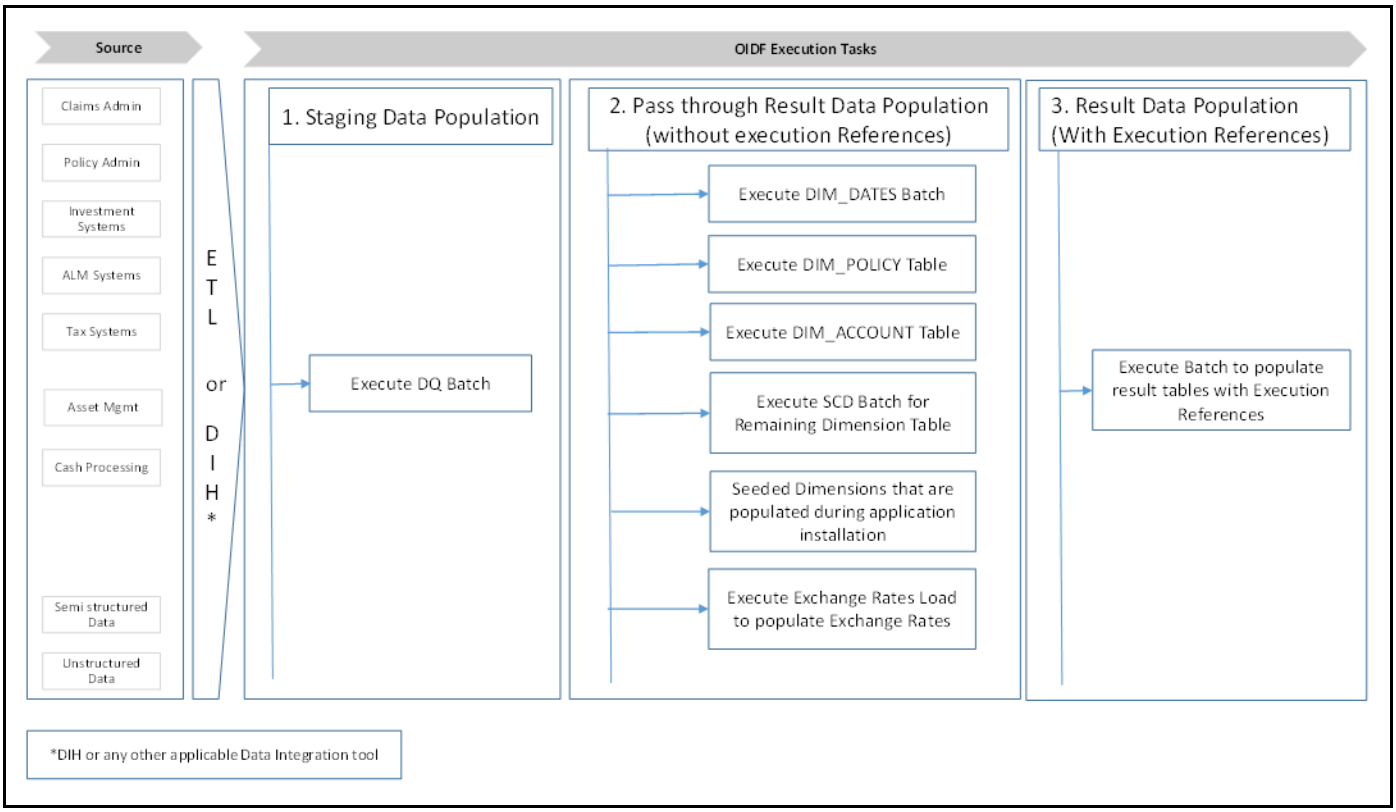
This section provides information about the data flow process for the Data Foundation application to populate the Result tables.

Topics:

6.1 Technical Flow to Populate the Result Tables

This representative data flow diagram of the Data Foundation application illustrates its data flow stages, input and output types, output usage, and the processes that populate results.

Figure 6: Representative data flow diagram of the Data Foundation application



The data flow process can be categorized into two main categories. They are:

1. Populating Stage Tables

   In this process, data is populated into the Data Foundation Stage master tables and Stage tables using these sub-processes:

   a. Run Custom ETL or DIH.

   b. Execute Data Quality batches.

2. For more information about loading Stage tables, see the section Loading OFSAA Staging Tables.

3. Populating Result Tables
4. In this process, data is populated in Dimension and Result tables using Stage tables. The sequence of sub-processes is:
   a. Dimension Loading Process:
      i. Populate the DIM_DATES table.
      ii. Seeded Dimensions that are populated during the installation process.
      iii. Populate DIM_ACCOUNT and DIM_POLICY tables.
      iv. Execute other SCD batches to populate corresponding Dimensions.
      v. Load key Dimensions using AMHM.
   b. Loading data into the Result tables using the T2T batch execution process.

These Result tables are used by other Analytical Applications to create reports. Some of the applications are listed here:

- IFRS17
- Solvency II
- Customer Insight

6.2 Execution Processes

This section provides the loading or execution process details of the Seeded Data, SCD, and T2T.

Topics:

6.2.1 Prerequisites for the Execution Processes

After OIDF application installation is successfully complete, as a prerequisite to initiate the SCD process or T2T process, follow these steps:

NOTE This procedure must be performed once only after a fresh installation of the application and also after installing an upgrade.

In this section, ensure that you execute the batch corresponding to the required Data Quality. To access the Run Name or Batch Name, and the order of execution, see the latest version of Oracle Insurance Data Foundation Application Pack Run Chart.

1. Complete the OIDF application installation and post-installation configuration procedures successfully. For detail procedures, see Oracle Insurance Data Foundation Application Pack Installation and Configuration Guide Release 8.1.2.0.0.
2. Populate Data into the Stage Master Tables.

   Populate data into all the Stage Master tables with all the required records by loading custom ETL or DIH.

> **NOTE** Data is already loaded into the OIDF Stage Master tables (see the step with *Populating Stage Tables* in the section Technical Flow to Populate the Result Tables).

3. Populate the DIM_DATES table.

   To populate the DIM_DATES table, see the section Time Dimension Transformation.
4. Execute the Data Quality (DQ) batches.

   Data that is required to execute a DQ is already loaded into the OIDF Stage tables (see the section Technical Flow to Populate the Result Tables).

   To execute the DQ batches, see the DQ Group Execution section.

6.2.2Dimension Loading Process

Use one of the following processes to load the data into the Dimension tables:

- Seeded Data
- Slowly Changing Dimension (SCD) Process
- Load Key Dimensions using AMHM

6.2.3Seeded Data

This is an automatic process wherein the Seeded data for Dimensions, which are packaged within the installer, are populated during the installation.

> **NOTE** To access the Seeded Data for Dimensions, see the latest version of the Oracle Insurance Data Foundation Application Pack Run Chart.

6.2.4 Slowly Changing Dimension (SCD) Process

A Slowly Changing Dimension (SCD) is a dimension that stores and manages both current and historical data over a period in a data warehouse. SCDs are dimensions that contain data, which changes slowly rather than changing on a time-based regular schedule.

In the SCD process, the SCD batch execution is used to populate data in the Dimension tables.

NOTE

- If you are executing the Dimension SCD Process after a fresh installation or after installing an upgrade, or when there are changes made in the Stage table data, first perform steps given in the section Prerequisites for the Execution Processes.
- Ensure you execute the batch corresponding to the required Dimension table. To access the Run Name or Batch Name, and the order of execution, see the latest version of Oracle Insurance Data Foundation Application Pack Run Chart.
- The SKeys of the Dimension tables are used to structure the Result tables (Fact tables) for populating the data. Therefore, this process serves as the prerequisite to the T2T Loading process.
- For functional information about Dimension loading, see the section About Dimension Loading Process.

To populate data in a Dimension table, follow these procedures:

1. Execute the SCD Batch of the DIM_ACCOUNT table.
2. Execute the SCD Batch of the DIM_POLICY table.
3. Execute the DIM_INSURANCE_SCD Batch.
4. Execute the <INFODOM>_DATA_FOUNDATION_SCD batch for the required Dimension table.
5. Verify Log Files and Check Error Messages.

6.2.4.1 Execute the SCD Batch of the DIM_ACCOUNT Table

To execute the SCD batch of the DIM_ACCOUNT table, perform the steps in the following section.

Topic:

- Execute the SCD Batch

6.2.4.1.1 Execute the SCD Batch

To execute the SCD batch, follow these steps:

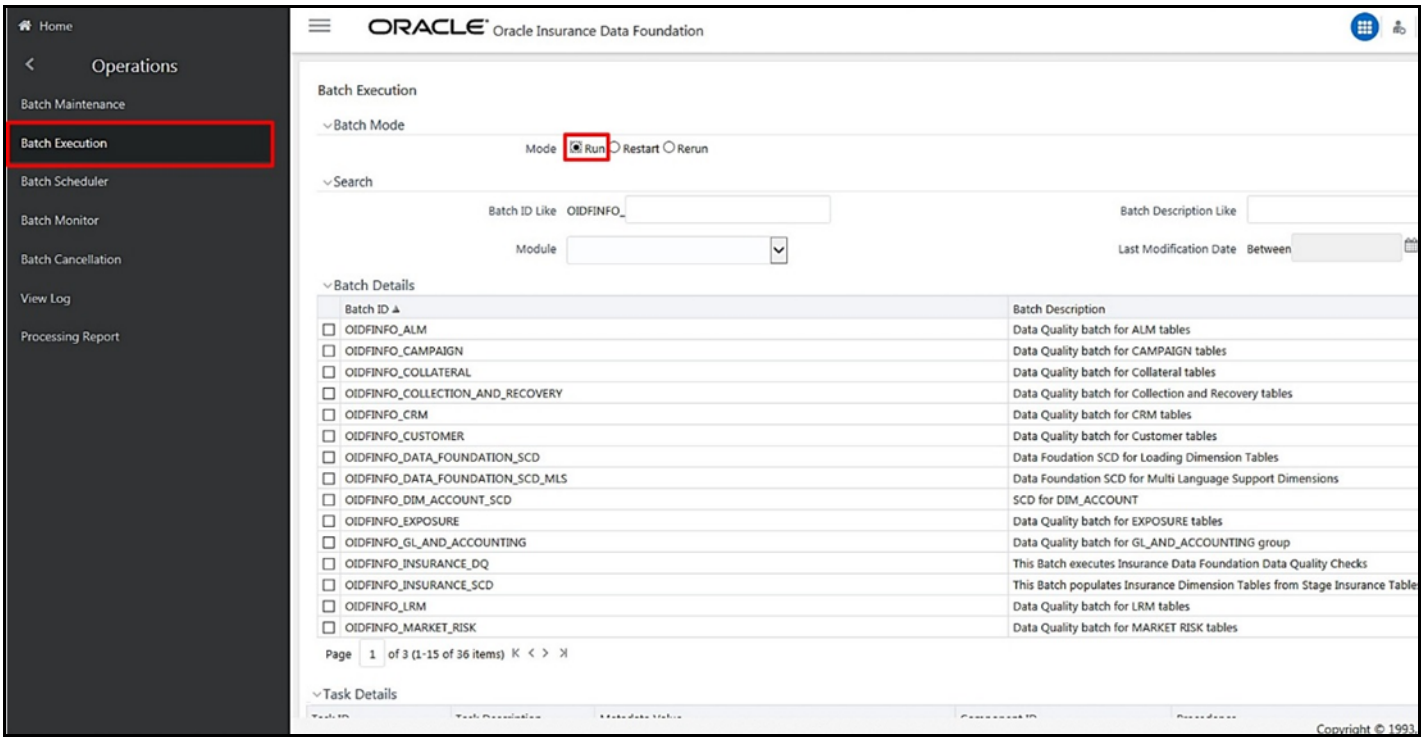
1. Navigate to the **Batch Execution** page.

   From **OFSAA Home**, select **Oracle Insurance Data Foundation**, select **Operations**, and then select **Batch Execution**.

2. Select the **Batch ID**.

   The **Batch Execution** page is displayed.

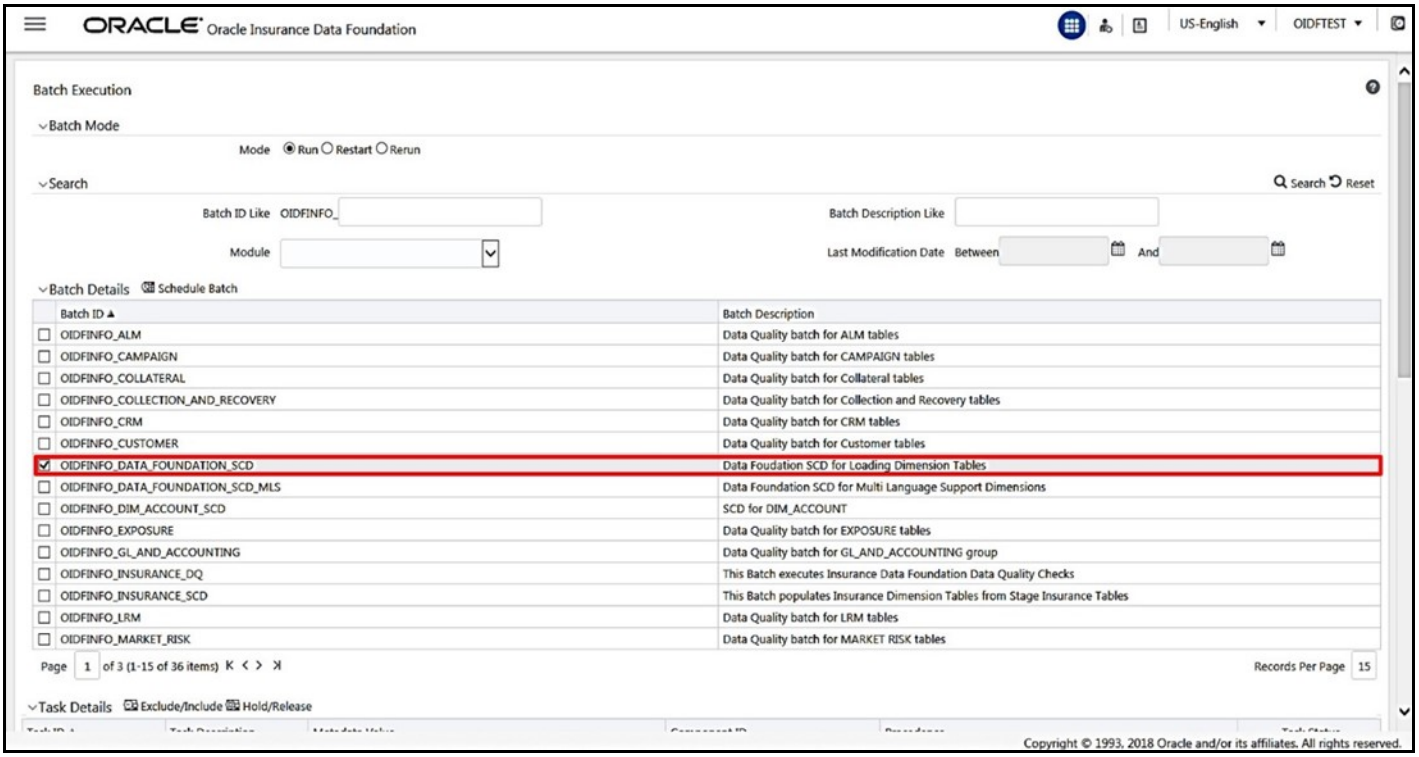
Figure 7: Navigate to the Batch Execution page



   a. In the Batch Mode section, select the **Run** option.

b. In the *Batch Details* section, search and select the required Batch ID.

Figure 8: Select the Batch ID

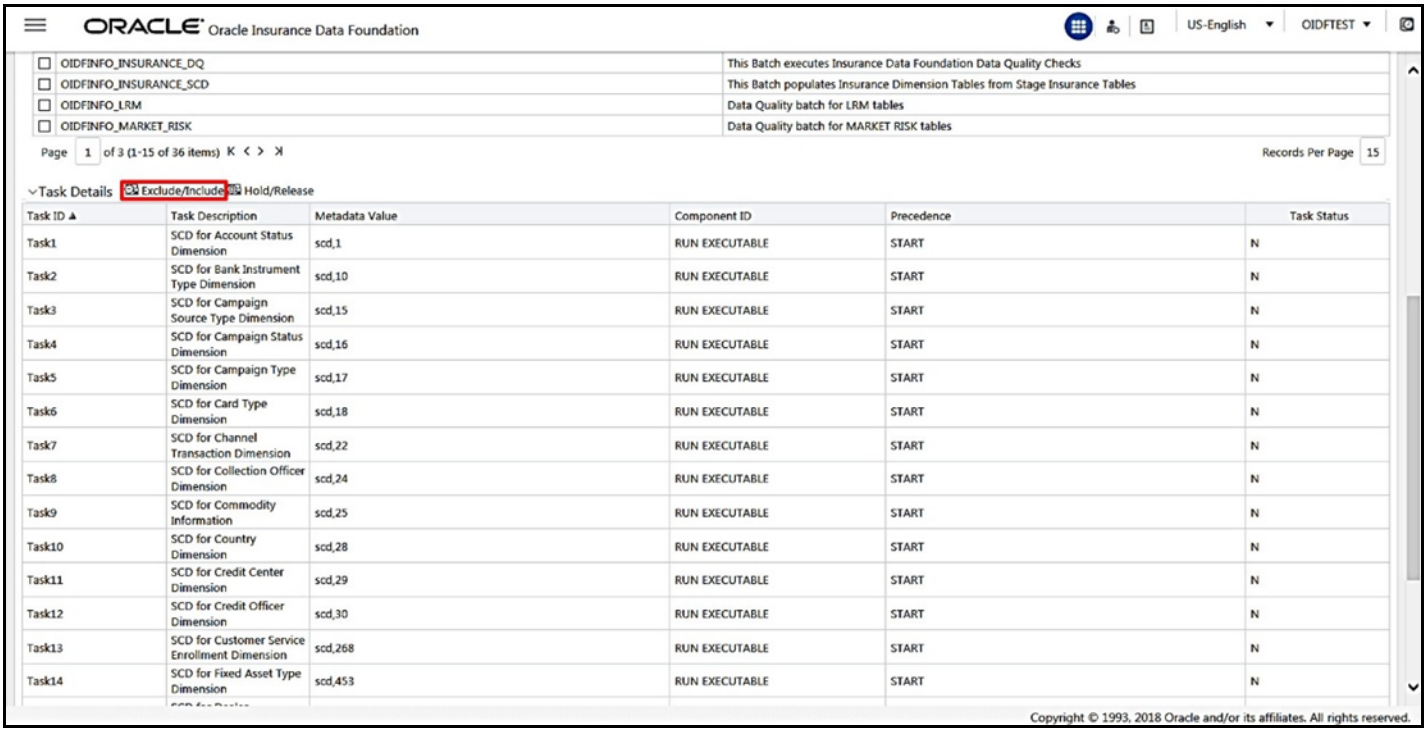


- For the DIM_ACCOUNT table, select the Batch ID `<INFODOM>_DIM_ACCOUNT_SCD`.
- For the DIM_POLICY table, select the Batch ID `<INFODOM>_INSURANCE_SCD` and include only DIM_POLICY related Stage table tasks.
- For all other Dimension tables, select the Batch ID `<INFODOM>_DATA_FOUNDATION_SCD`.

The **Task Details** section lists the tasks corresponding to the selected Batch ID. The list consists of all the Stage tables that are mapped to Dimension tables.

3. Exclude or include tasks from or into the SCD batch.

Figure 9: Select the task Exclude/Include option



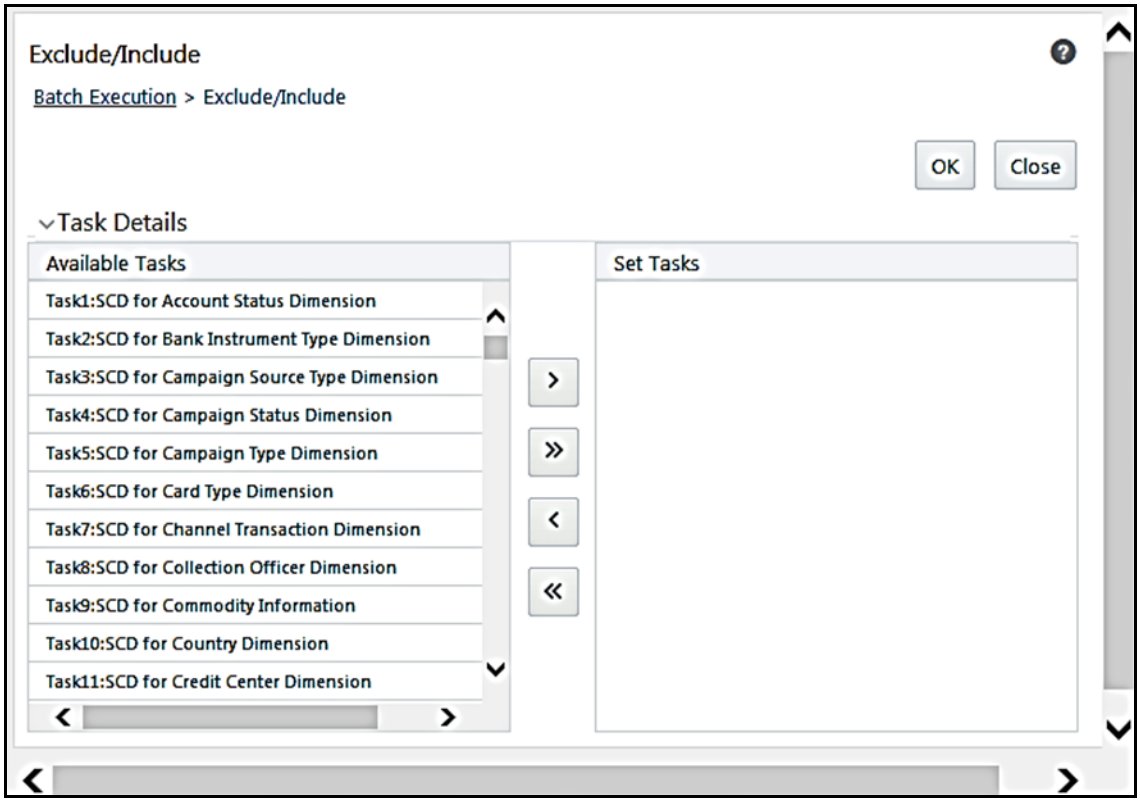
a. Make a note of the Metadata Value (SCD map reference number) and the Task ID for the required Dimension table.

> **NOTE** To populate the Dimension table from its corresponding Stage Master table, use SCD packaged in OIDF with the respective Map Reference Number.

b. To exclude or include a task from or into the required SCD batch, in the **Task Details** section, click the **Exclude/Include** icon.

c. The **Task Mapping – Exclude/Include** page is displayed.

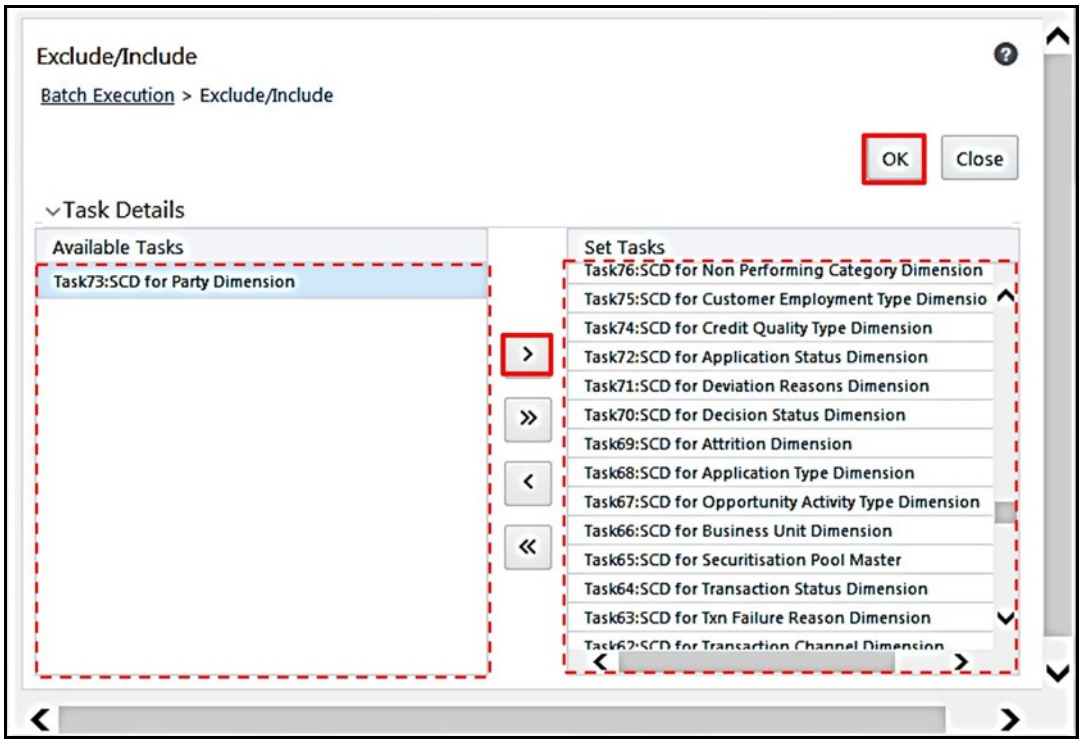
Figure 10: The task Exclude/Include page



d. Include or exclude the tasks.

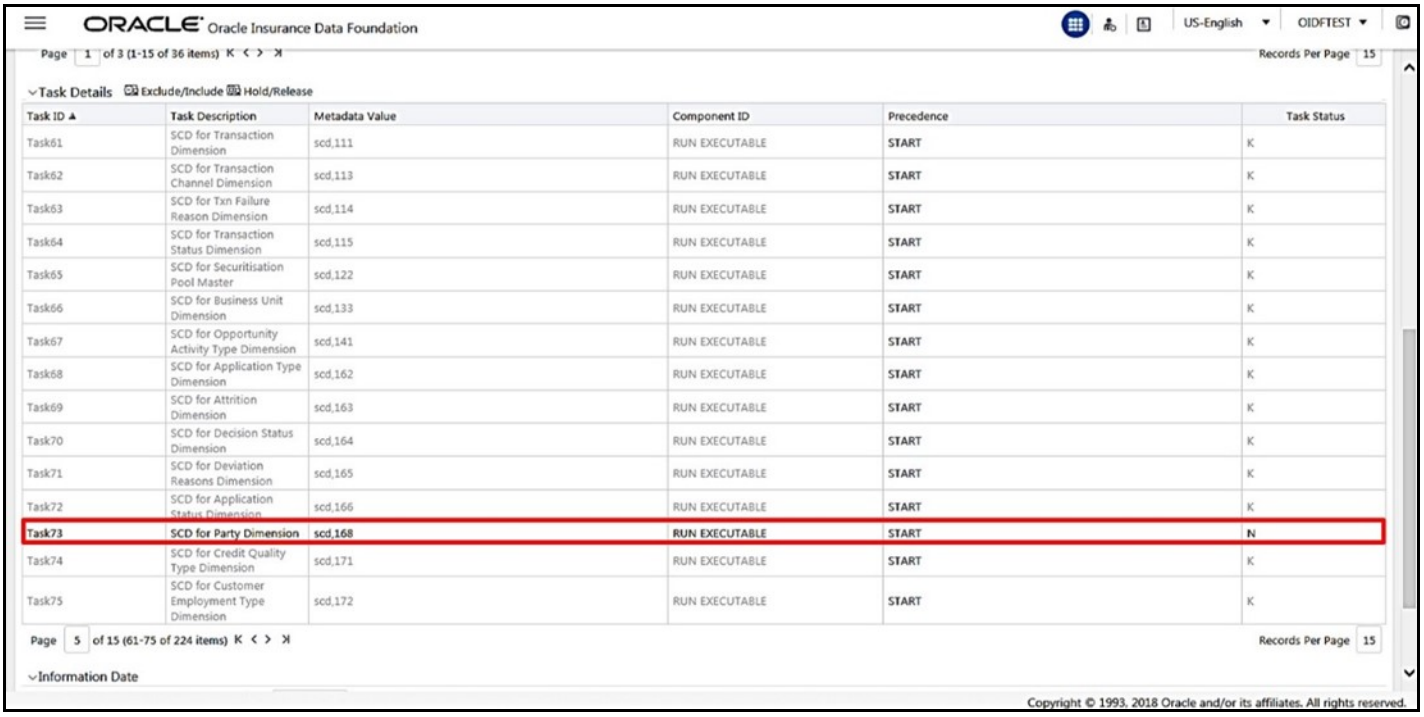
In the **Task Details** section, select the tasks that must be excluded from the batch execution procedure, and click > to move those tasks from the **Available Tasks** section to the **Set Tasks** section. The **Available Tasks** list consists of tasks that are available for the execution of Dimension table.

Figure 11: Exclude the Tasks



e. Confirm the task inclusion or exclusion.

Figure 12: The selected task is listed in the Task Details section



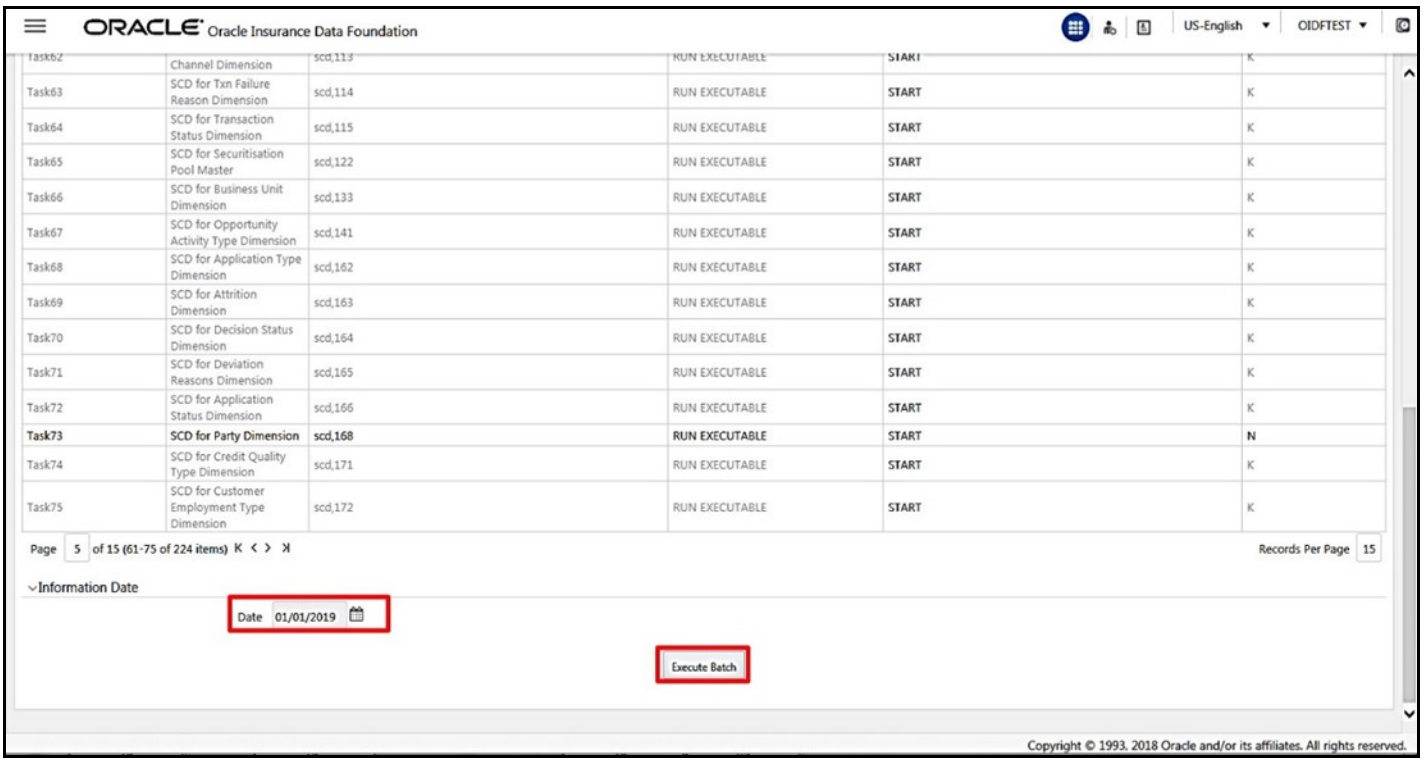
i. To save the changes, click **OK**. The following warning message is displayed: *If you exclude a task, it will be skipped when executing the batch but, the precedence will not be altered. Do you want to exclude the selected task(s)?*

ii. To proceed, click **OK**. The **Task Details** list consists only of the tasks present in the **Available Tasks** list. For example, in the following screenshot, the included (highlighted) task, of which the Task Status is N, and this task is executed during the batch execution. The excluded task(s) are displayed in the Grey color with Task Status as K.

4. Select the **FIC_MIS** date.

   In the **Information Date** section, click the calendar icon and select the required date. This date is the FIC_MIS date populated in the DIM_DATES table.

Figure 13: Select the FIC_MIS date and click Execute Batch



5. Execute the SCD batch.

   Click **Execute Batch** to run the selected SCD batch with one of the following selected task:

   - For the DIM_ACCOUNT table, execute the Batch ID `<INFODOM>_DIM_ACCOUNT_SCD`.
   - For the DIM_POLICY table, execute the Batch ID `<INFODOM>_INSURANCE_SCD` and include only DIM_POLICY related Stage table tasks.
   - For all other Dimension tables, execute the Batch ID `<INFODOM>_DATA_FOUNDATION_SCD`.

6. Confirm the SCD batch execution.

   a. The following pop-up message is displayed: *Do you want to execute the batch for MIS Date <MIS_Date>*. To execute the batch, click **OK**.

   b. The following acknowledgment message is displayed: *Batch triggered successfully. The Batch Run Identification is: <Batch Run ID>*. Click **OK** to continue. The SCD batch is executed. As a result, data is populated in the target Dimension table.

6.2.4.2 Execute the SCD Batch of the DIM_POLICY Table

> **NOTE** To load data into the DIM_POLICY table, execute the SCD batch for all the corresponding stage tables.

To execute the SCD batch of the DIM_POLICY table, see the Execute the SCD Batch section.

6.2.4.3 Execute the DIM_INSURANCE_SCD batch

To execute the DIM_INSURANCE_SCD batch, see the Execute the SCD Batch section.

6.2.4.4 Execute the <INFODOM>_DATA_FOUNDATION_SCD Batch for the Dimension Table

> **NOTE** If you are executing this Dimension SCD Process after a fresh installation or after installing an upgrade, or when there are changes made in the Stage table data, first perform steps given in the following procedures:
>
> 1. Prerequisites for the Execution Processes.
> 2. Execute the SCD Batch of the DIM_ACCOUNT table.
> 3. Execute the SCD Batch of the DIM_POLICY table.

To execute the `<INFODOM>_DATA_FOUNDATION_SCD` batch, follow these steps:

1. Verify the presence of data in the Stage Master table.
2. Execute the SCD batch.
3. Check the execution status of the SCD batch.

6.2.4.4.1 Verify Presence of Data in the Stage Master Table

Verify if data is present in the Stage master table corresponding to the required Dimension table with all the required records.

> **NOTE** Data is already loaded into the OIDF Stage Master tables (see the section Technical Flow to Populate the Result Tables).

6.2.4.4.2 Execute the SCD Batch

To execute the SCD batch, see the Execute the SCD Batch section.

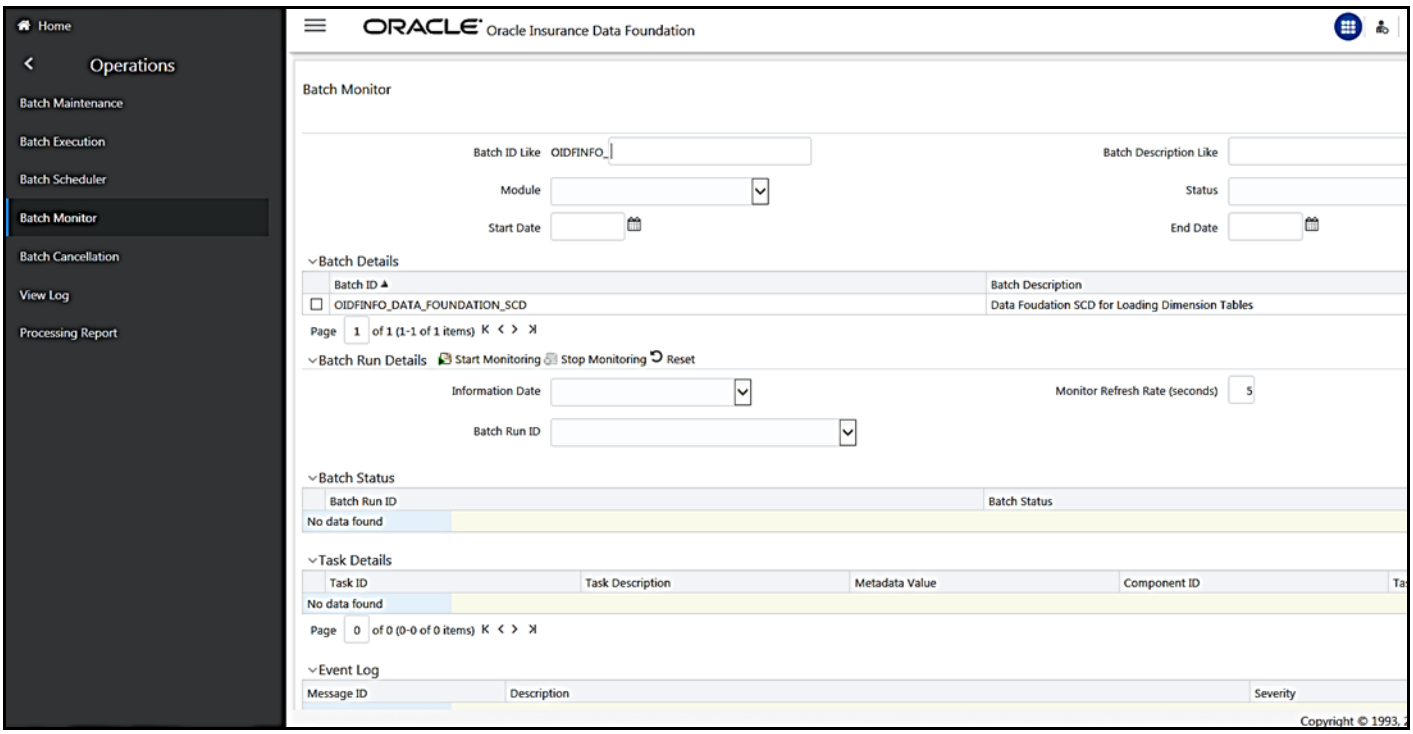
6.2.4.4.3 Check the Execution Status of the SCD Batch

To check the execution status of the SCD batch, follow these steps:

1. Navigate to the **Batch Monitor** page.

   From **OFSAA Home**, select **Oracle Insurance Data Foundation**, select **Operations**, and then select **Batch Monitor**.

Figure 14: Navigate to the Batch Monitor page

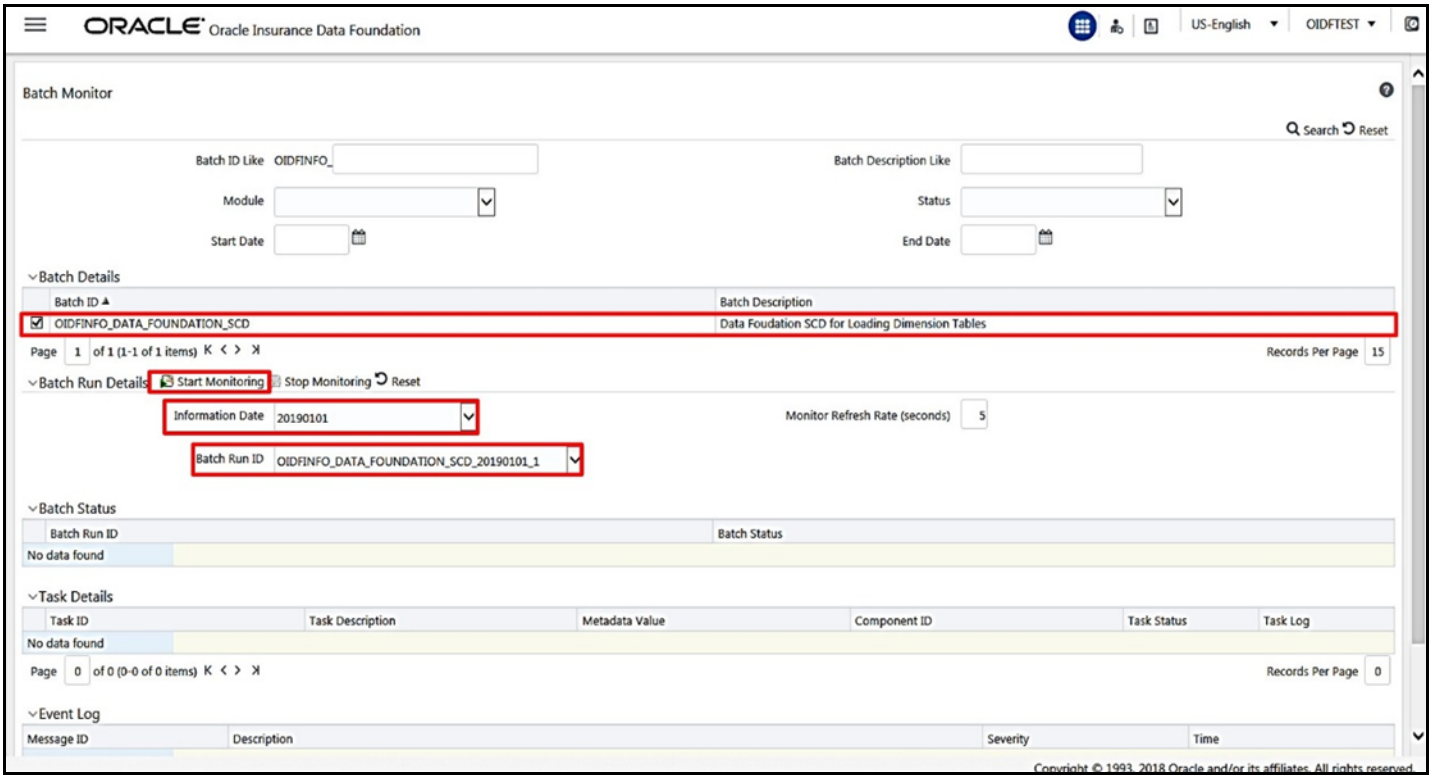


2. Select the Batch ID **<INFODOM>_DATA_FOUNDATION_SCD**.

   The **Batch Monitor** page appears on the right-hand side. In the Batch Details section, select the **Batch ID** that was executed during the Execute the SCD Batch step, that is, select the `<INFODOM>_DATA_FOUNDATION_SCD` Batch ID.

3. Select the FIC_MIS Date and Batch Run ID.

Figure 15: Select the Batch ID, FIC_MIS Date, and Batch Run ID



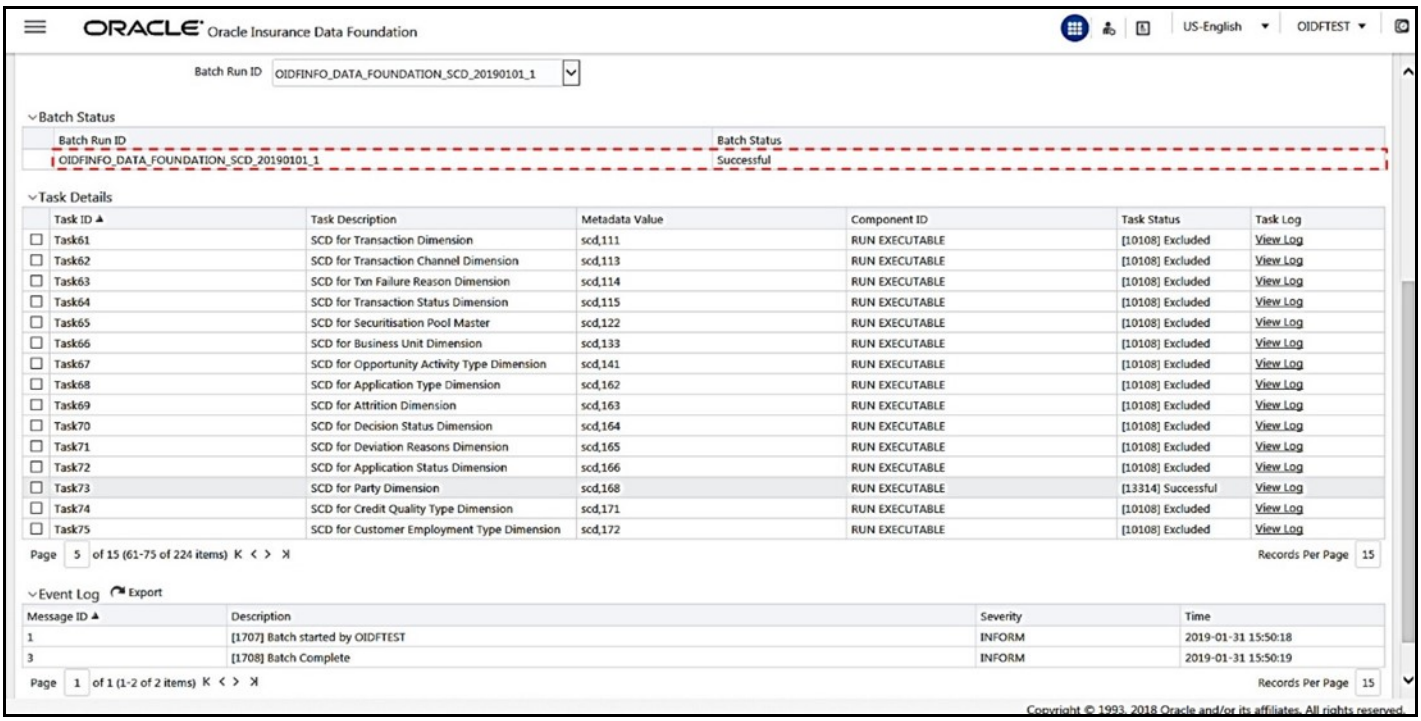
   a. In the **Batch, Run Details** section, click the **Information Date** box and select the required **FIC_MIS Date**. The SCD batch was executed on this date.

   b. Then click the **Batch Run ID** box and select the required value.

   c. Click the **Start Monitoring** icon.

4. Check the Batch status.

Figure 16: Batch execution status



In the **Batch Monitor** page, the Batch Status, Task Details, and Event Log sections are displayed in addition to the existing details.

The following are the types of Status messages:

- Not Started
- Ongoing
- Failure
- Successful

When a Batch Status is indicated as *Successful*, valid data is populated in the Dimension table.

6.2.4.5 Verify Log Files and Check Error Messages

Use one of these two methods to access the SCD batch execution log files to view the complete log generated during the SCD batch execution:

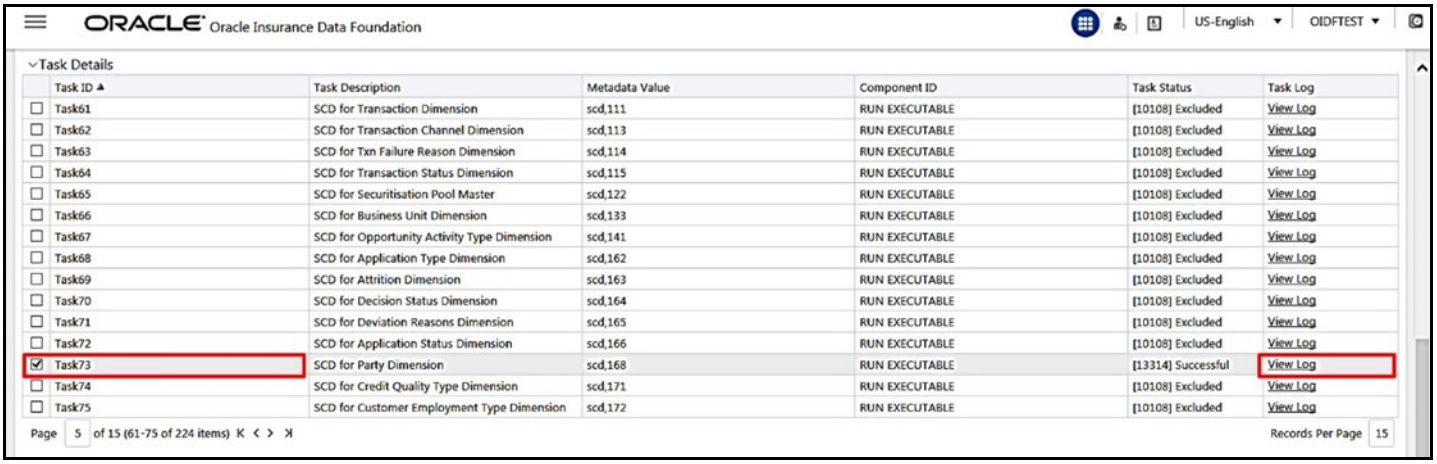
- View and download the log files from the application UI.
- View log files in the application server.
- Check error messages.

6.2.4.5.1 View and Download the Log Files from the Application UI

To view and download the log files from the application UI, follow these steps:

1. In the **Task Details** section, select the task that was executed and click **View Log**.

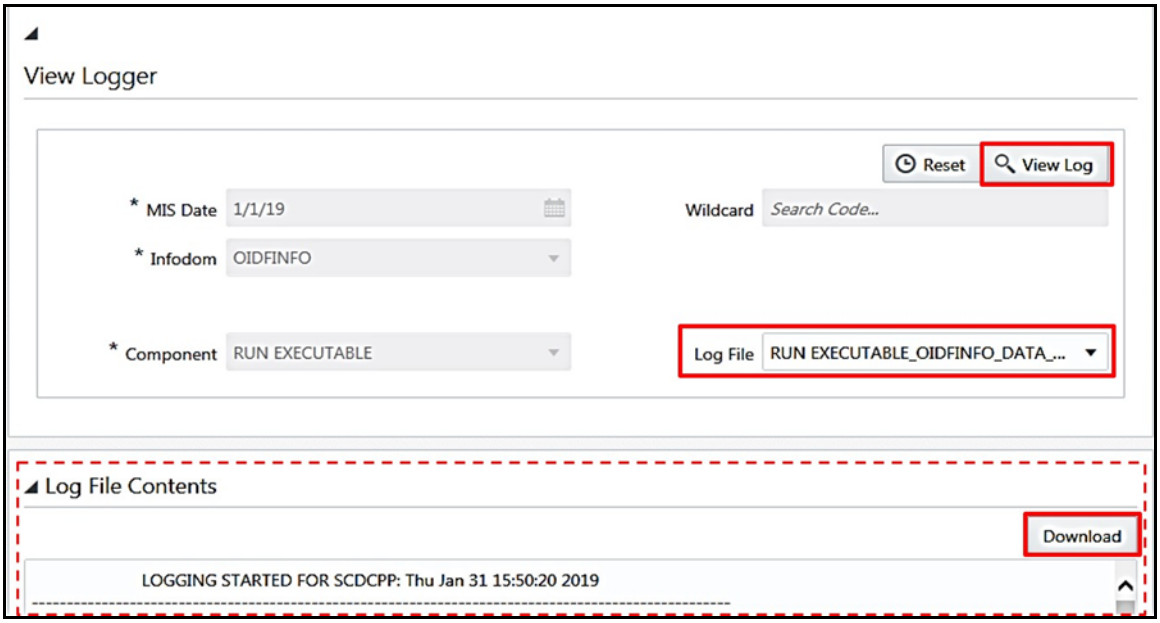
Figure 17: Select View Log for the Task that was included in the Batch execution



2. Select and view the log file.

   The **View Logger** page is displayed.

Figure 18: Verify the log file content in the View Logger page



   In the **Log File** box, select the required file, and then click **View Log**. The complete log information is displayed in the *Log File Contents* section.

   Two types of log files are generated:

   - `SCDCPP.log`
   - `FICGEN.log`

3. To download a copy of the log file content to the system, click **Download**.

6.2.4.5.2 View Log Files in the Application Server

To view the execution log file on the application server, verify the files in the following directory:

```
ftpshare/logs/<Run_Date>/<infodom>/RUN_EXECUTABLE
```

> **NOTE** For comprehensive information on the configuration and execution of a batch, see the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.

6.2.4.5.3 Check the Error Messages

To check the error messages, open the log file present in the `ftpshare/logs/<Run_Date>/<infodom>/RUN_EXECUTABLE` directory for the Dimension tables.

6.2.4.6 Post SCD Process

When the SCD process is complete, individual numeric Surrogate Keys are generated for each identifier associated with that Dimension table. This SKey is unique within each Dimension table. The SKeys of the Dimension table is used to structure the resulting tables (Fact tables) for populating the data.

The Dimension table is joined with the Reporting or Processing tables.

> **NOTE** To access the list of all Stage Master tables, see the SYS_TBL_MASTER table.
>
> To check the mapping from a Stage table column to the Dimension table column, see the SYS_STG_JOIN_MASTER table using Map Reference Number.
>
> For more information about the Dimension table loading process, see the chapter About Dimension Loading Process.

6.2.5Load Key Dimensions using AMHM

For more information about loading the Dimensions using AMHM, see the *Dimension Management* section in the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0 and *Dimension Load Procedure* section in the Oracle Financial Services Analytical Applications Data Model Utilities User Guide.

6.2.6 Table to Table (T2T) Loading Process

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

- Oracle Insurance Data Foundation Source Run: This Run pipeline (Process) loads all *non-Run* enabled tables in OIDF. For data movement from Stage tables to Result tables, this Run can be executed once per day, where the Result tables do not consist of Run SKey.
- Oracle Insurance Data Foundation Execution Run: This Run pipeline (Process) loads all *Run* enabled tables in OIDF. For data movement from Stage tables to Result tables, this Run can be executed any number of times per day with each unique Run SKey.

NOTE If you are following this T2T Process after the first-ever application installation or after the upgrade installation, then perform steps in these procedures:

1. Prerequisites for the Execution Processes.
2. Execute the SCD Batch of the DIM_ACCOUNT table.
3. Execute the SCD Batch of the DIM_POLICY table.
4. Execute the DIM_INSURANCE_SCD Batch.
5. Execute the <INFODOM>_DATA_FOUNDATION_SCD batch for the required Dimension table.

To access the Run Name and their order of execution, see the latest version of the Oracle Insurance Data Foundation Application Pack Run Chart.

For information about loading multiple load runs, see the chapter Loading Multiple Load Runs in OFSAA.

To create a custom Run process in PMF, see the Configure and Manage Custom Pipelines for OIDF Functions section.

To create a T2T, see the *Data Management Framework* section in the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.

For illustration, the Oracle Insurance Data Foundation *Source Run* process is primarily used.

To load all **non-Run** enabled tables in OIDF using the out-of-the-box Oracle Insurance Data Foundation Source Run process, or to load all **Run** enabled tables in OIDF using the out-of-the-box Oracle Insurance Data Foundation Execution Run process, perform the steps in the following section:

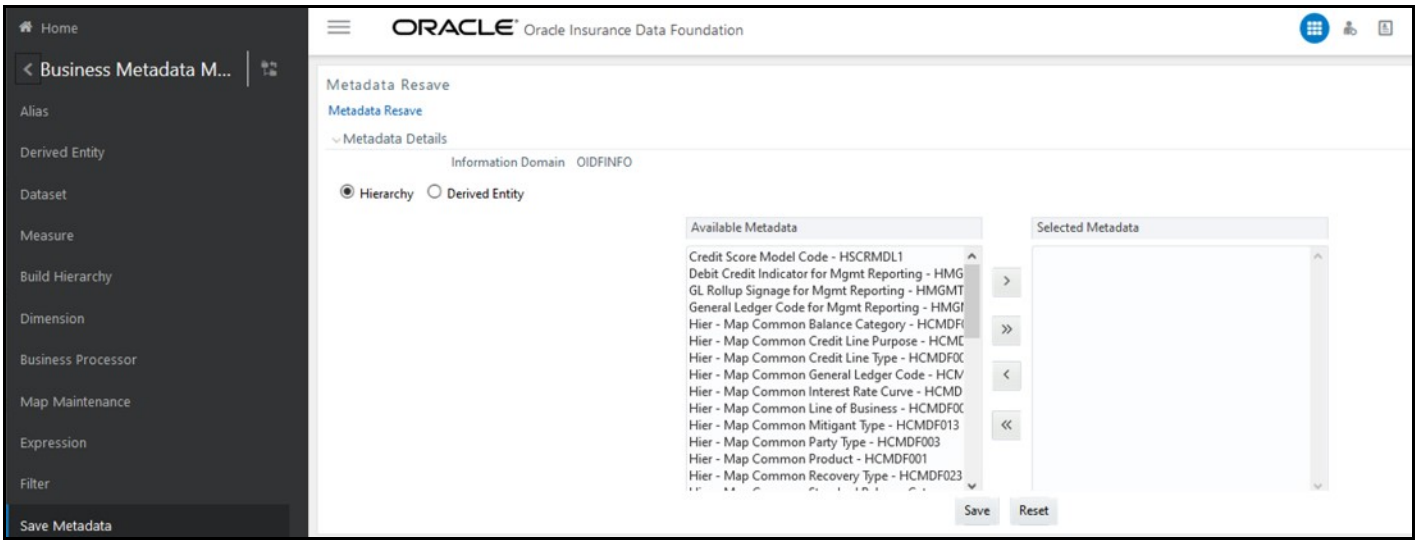
1. Prerequisites for loading T2T
2. Select the Run Parameters and Execute the Run
3. Verify the Run Execution
4. Check Error Messages
5. Post T2T Process

6.2.6.1 Prerequisites for Loading T2T

To resave the hierarchies, follow these steps:

1. From **OFSAA Home**, select **Oracle Insurance Data Foundation**, select **Unified Analytical Metadata**, select **Business Metadata Management**, and then select **Save Metadata**. The **Metadata Resave** page is displayed.

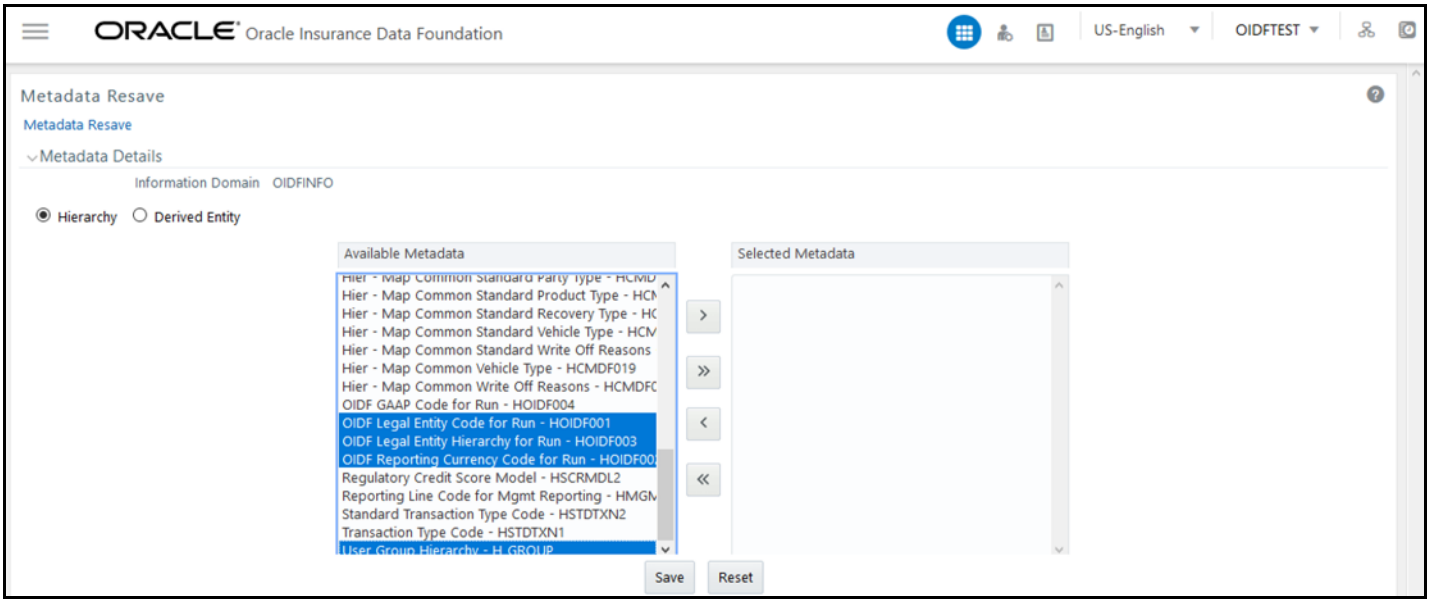
Figure 19: Navigate to the Metadata Resave page



2. In the **Metadata Resave** page, to load values for the Reporting Currency parameter and the Legal Entity parameter, select the following hierarchies:
   - Legal Entity Code for Run (HOIDF001)
   - Reporting Currency Code for Run (HOIDF002)
   - Legal Entity Hierarchy for Run (HOIDF003)
   - User Group Hierarchy (H_GROUP)

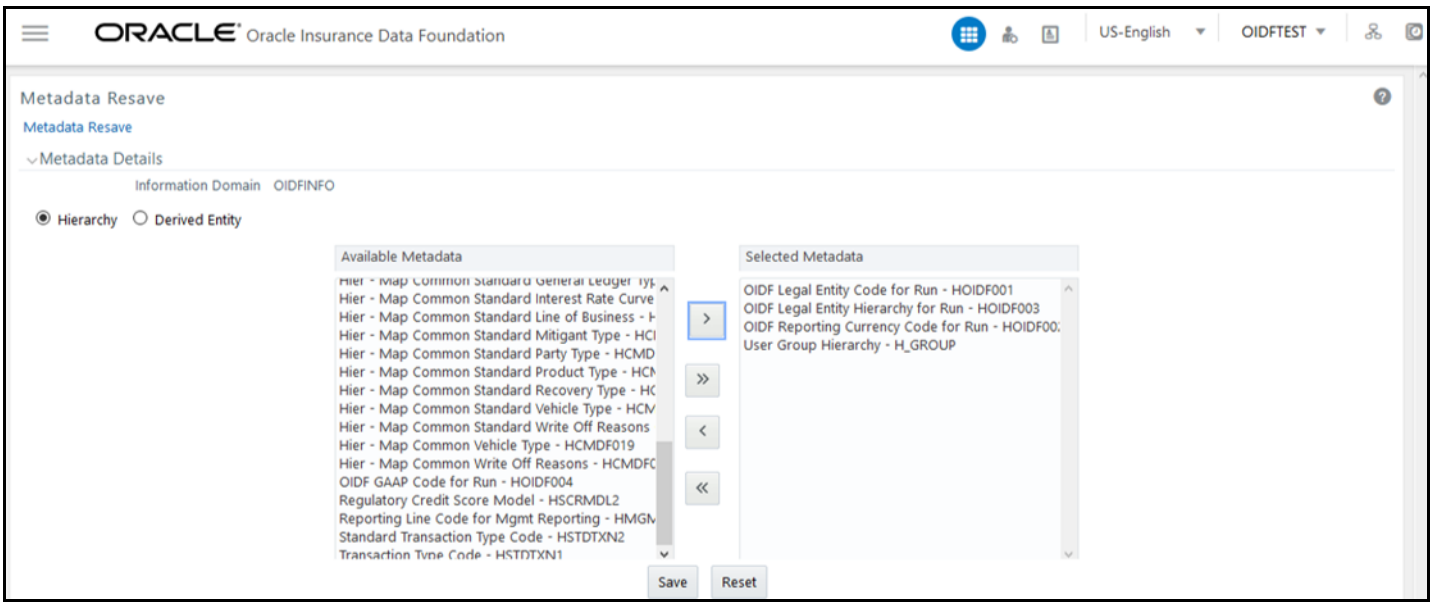
   Move the selected hierarchies from **Available Metadata** to **Selected Metadata**.

Figure 20: Select the Legal Entity and Reporting Currency hierarchies



3. To save the selection, click **Save**. Or to reset the details and start-over, click **Reset**.

Figure 21: Save the selected hierarchies



After saving the hierarchies, select the Run parameters and execute the Run.

6.2.6.2 Select the Run Parameters and Execute the Run

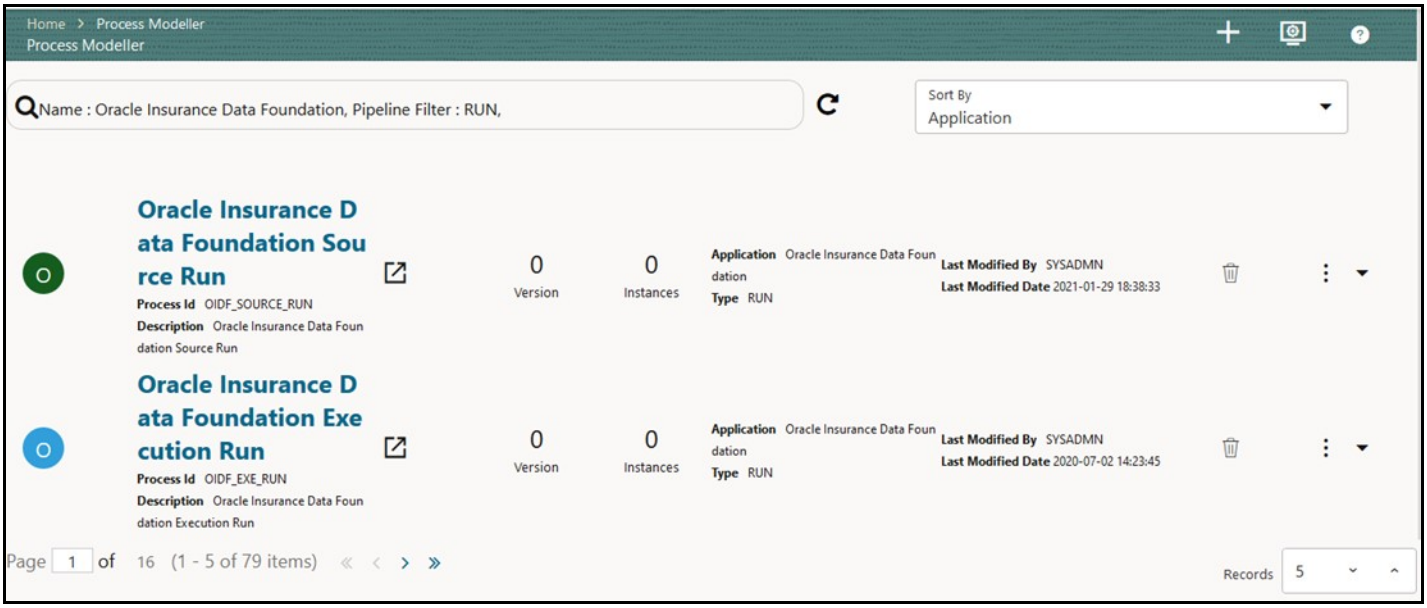
To select the Run parameters and execute the Run, follow these steps:

1. From **OFSAA Home**, select **Oracle Insurance Data Foundation**, click **Administration**  . In the **Information Domain** list, select the information domain required for OIDF, and then click the **Process Modelling Framework** tile.

   A submenu is displayed. Click **Process Modeller** to access the **Process Modeller** page.

   - In the following illustration, the Oracle Insurance Data Foundation Source Run process and Oracle Insurance Data Foundation Execution Run process are shown.

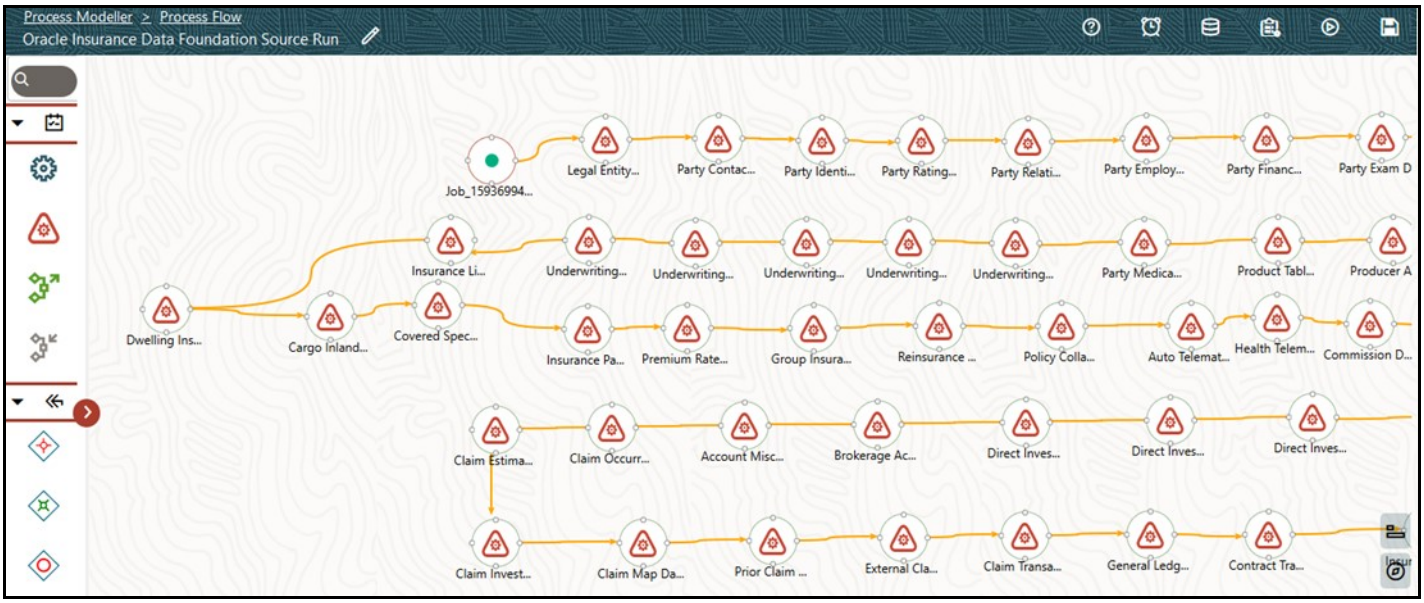
Figure 22: Process Modeller page with OIDF Source Run and OIDF Execution Run Process



2. Select the **Oracle Insurance Data Foundation Source Run** process. The corresponding process flow is displayed in a page. The **Process Flow** canvas contains a toolbar and a drawing canvas. Use the drawing canvas to design the process flow with the **Activity**, **Transition**, and **Widget** components available in the floating toolbar with Sub Pipeline (sub-process) activity as the base. Each Sub Pipeline activity represents each of the Data Load process in the OIDF Source Run, and Average Balances process. The Sub Pipeline activities are executed in a series.

   - The following illustration is for the Oracle Insurance Data Foundation Source Run process.

Figure 23: OIDF Source Run Process Flow



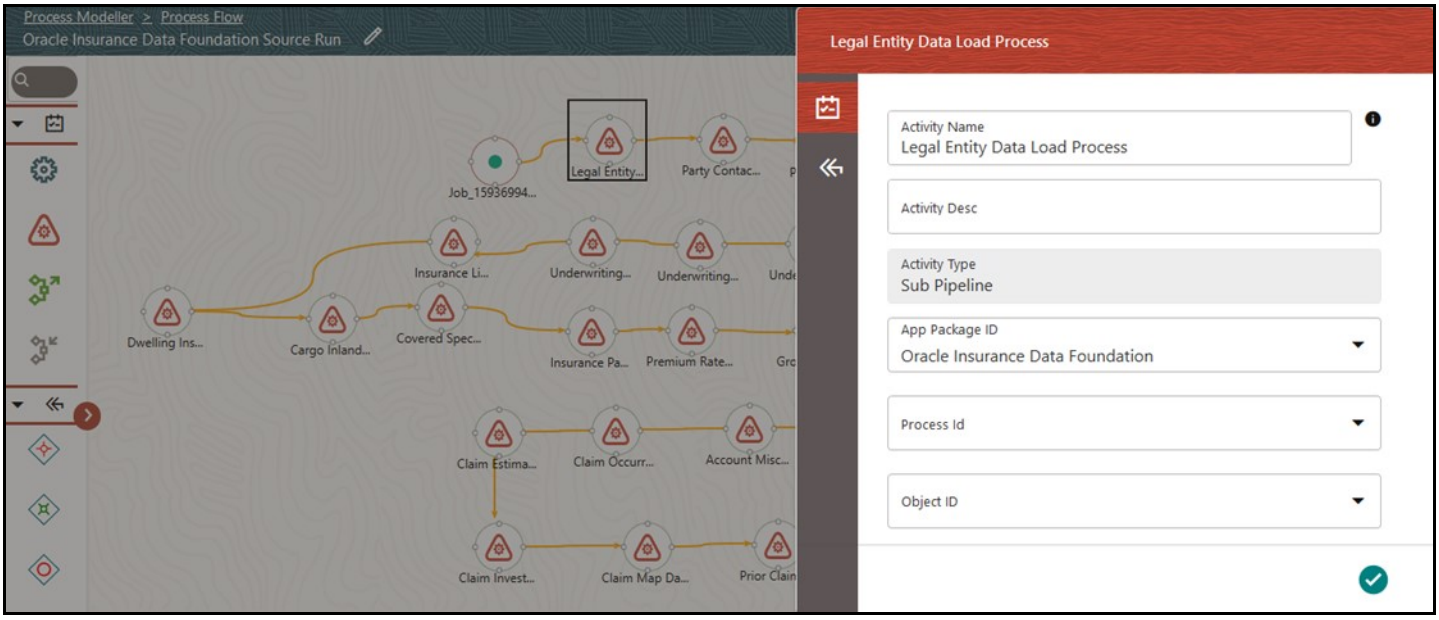
   - The following illustration is for the Oracle Insurance Data Foundation Execution Run process.

Figure 24: OIDF Execution Run Process Flow

3. Double-click a Sub Pipeline activity to see the details related to its **Sub Process Details**.

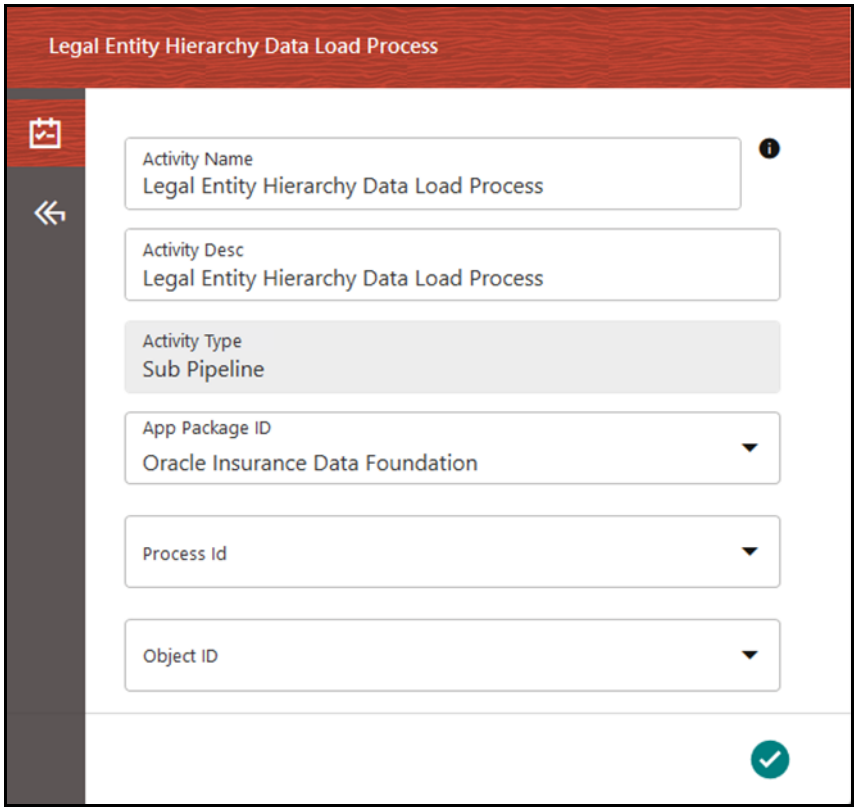
   - The following illustration is for the Oracle Insurance Data Foundation Source Run process.

Figure 25: OIDF Source Run Process with Sub Process Details



- The following illustration is for the Oracle Insurance Data Foundation Execution Run process.

Figure 26: OIDF Execution Run Process with Sub Process Details



4. On the drawing canvas, select the **Definition**, **Application Rule**, and **Data Fields** icons to see the respective details.
   - The following illustrations are for the Oracle Insurance Data Foundation Source Run process.

Figure 27: OIDF Source Run Process with Definition Details

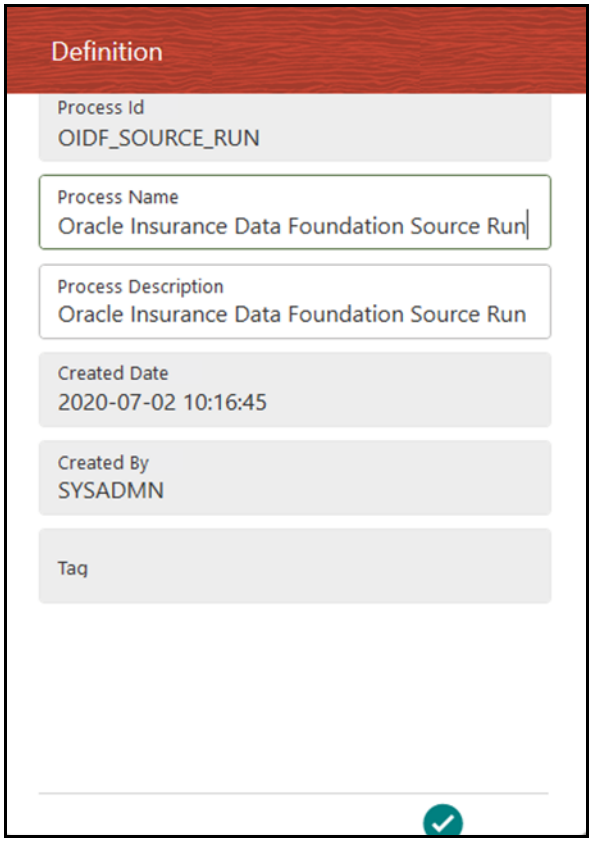


Figure 28: OIDF Source Run Process with Application Rule Details

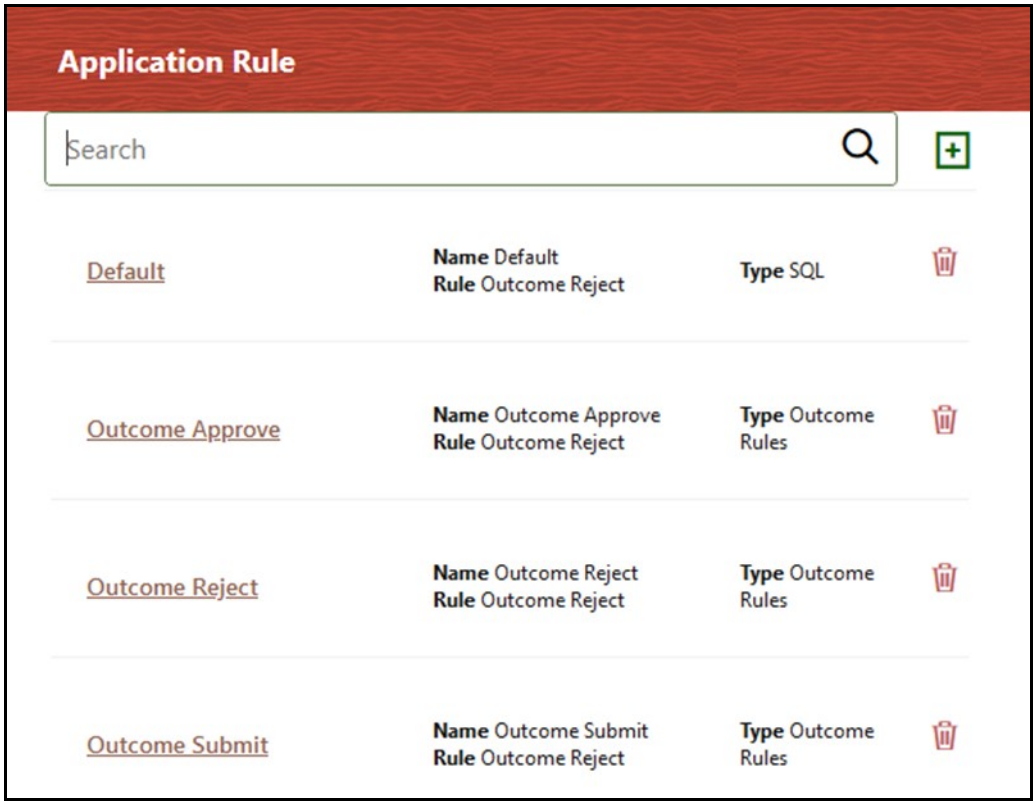
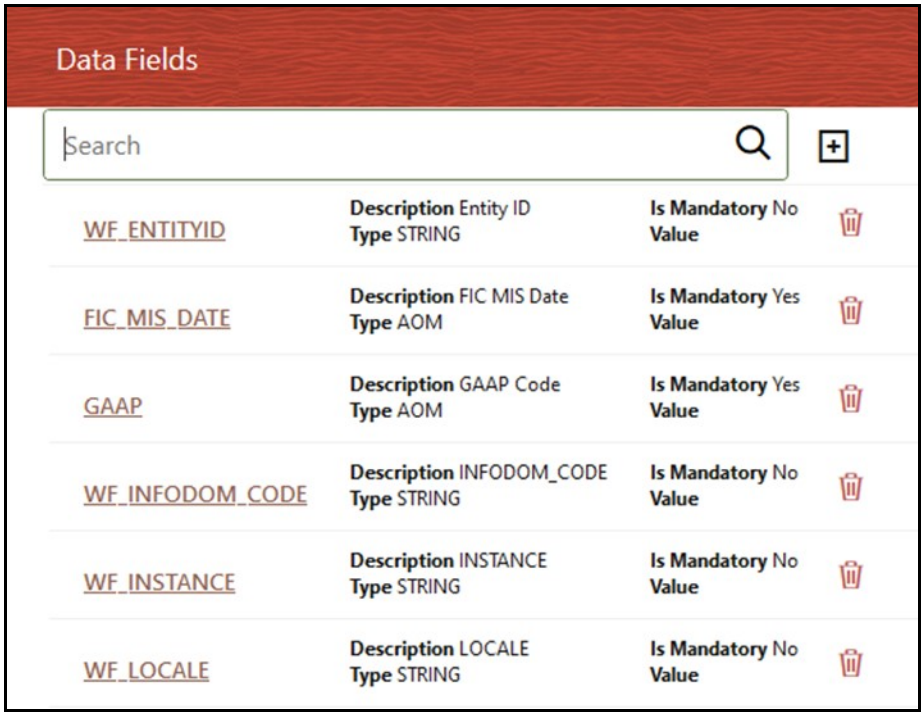


Figure 29: OIDF Source Run Process with Data Fields Details



- The following illustrations are for the Oracle Insurance Data Foundation Execution Run process.

Figure 30: OIDF Execution Run Process with Definition Tab Details

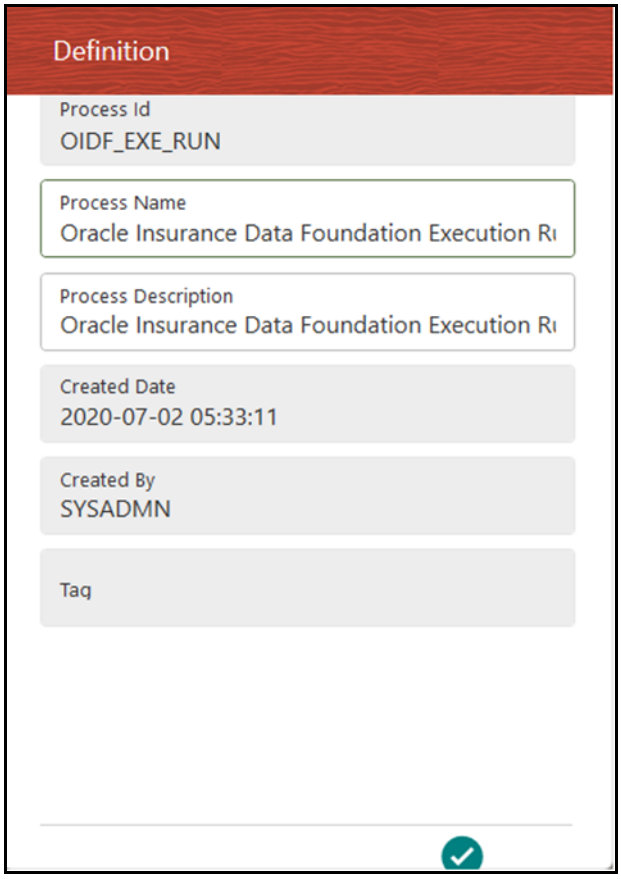


Figure 31: OIDF Execution Run Process with Application Rule Tab Details

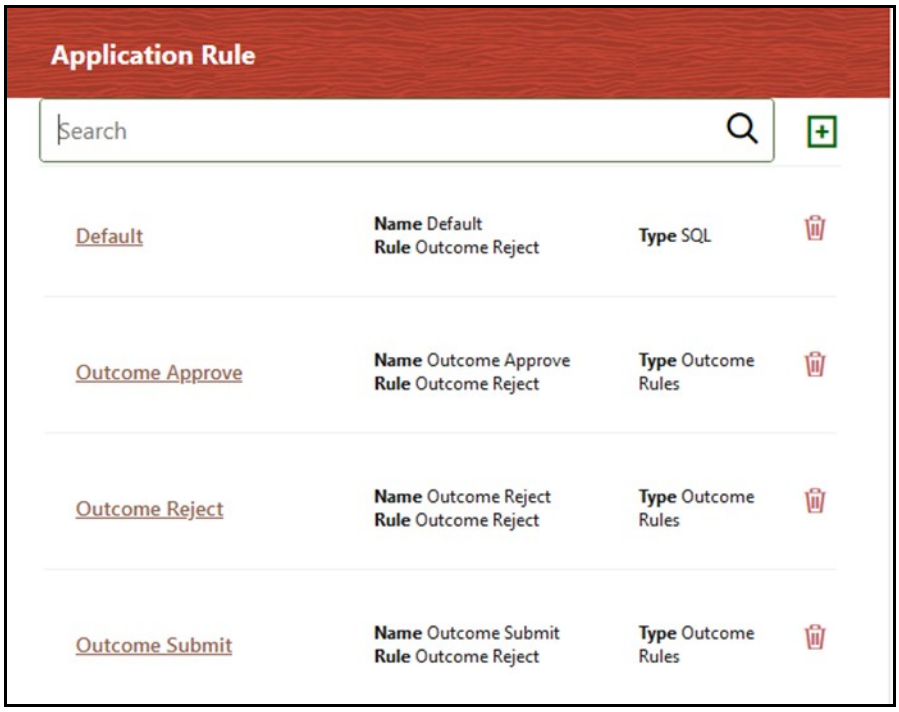
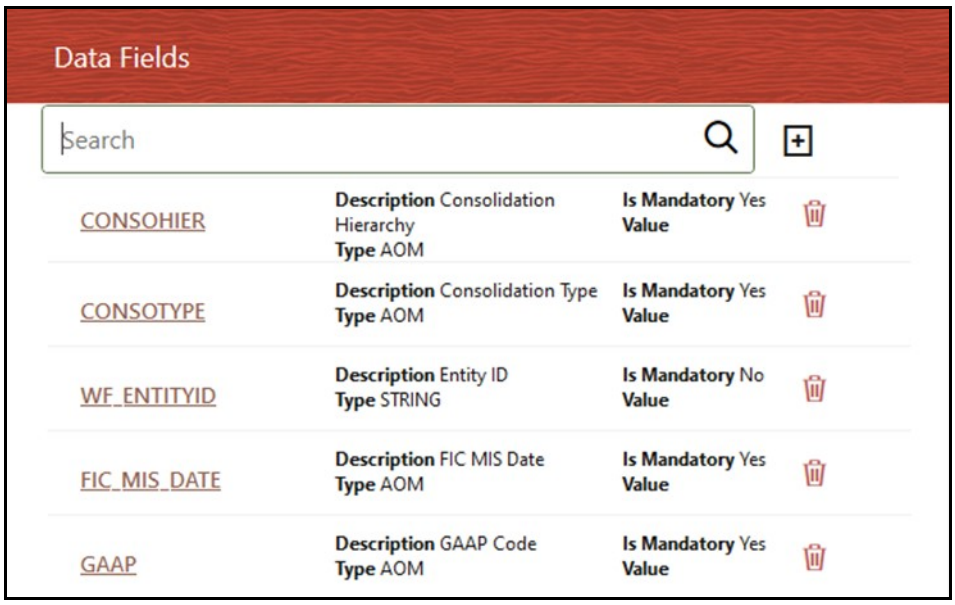


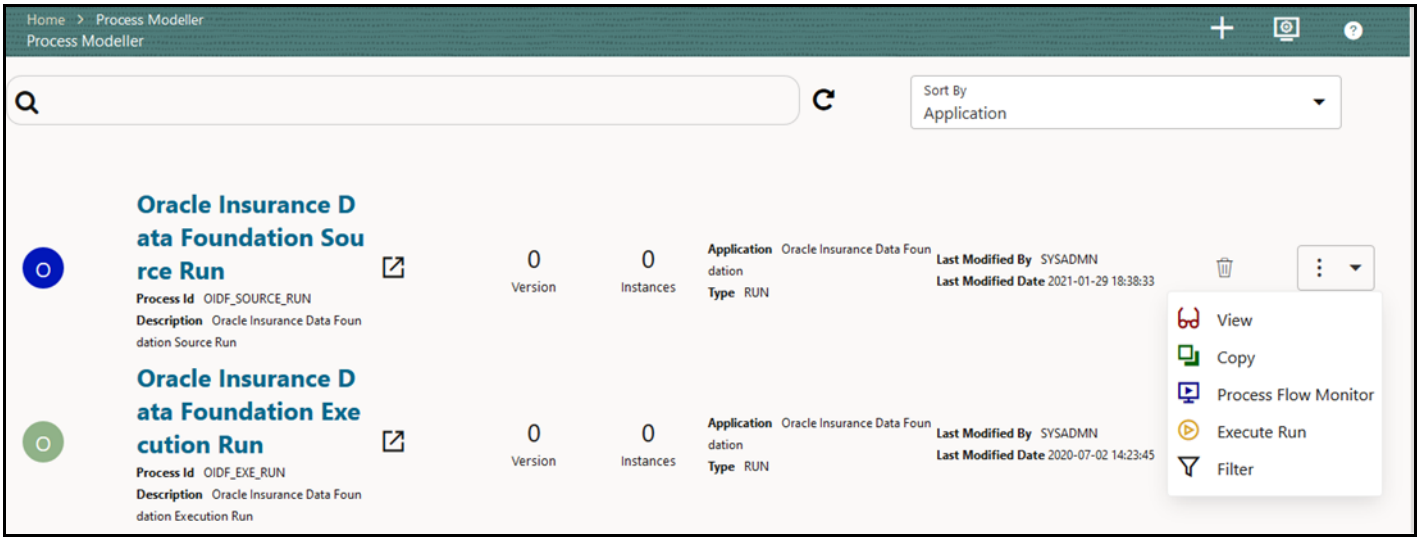
Figure 32: OIDF Execution Run Process with Data Fields Tab Details



5. In the **Process Modeller** page, click ⋮ corresponding to the Insurance Liability Contracts Data Load Process that must be executed. Click **Execute Run**. A Job ID is created in the Process Monitor page for each Sub Pipeline (sub-process) and every time a process is executed.

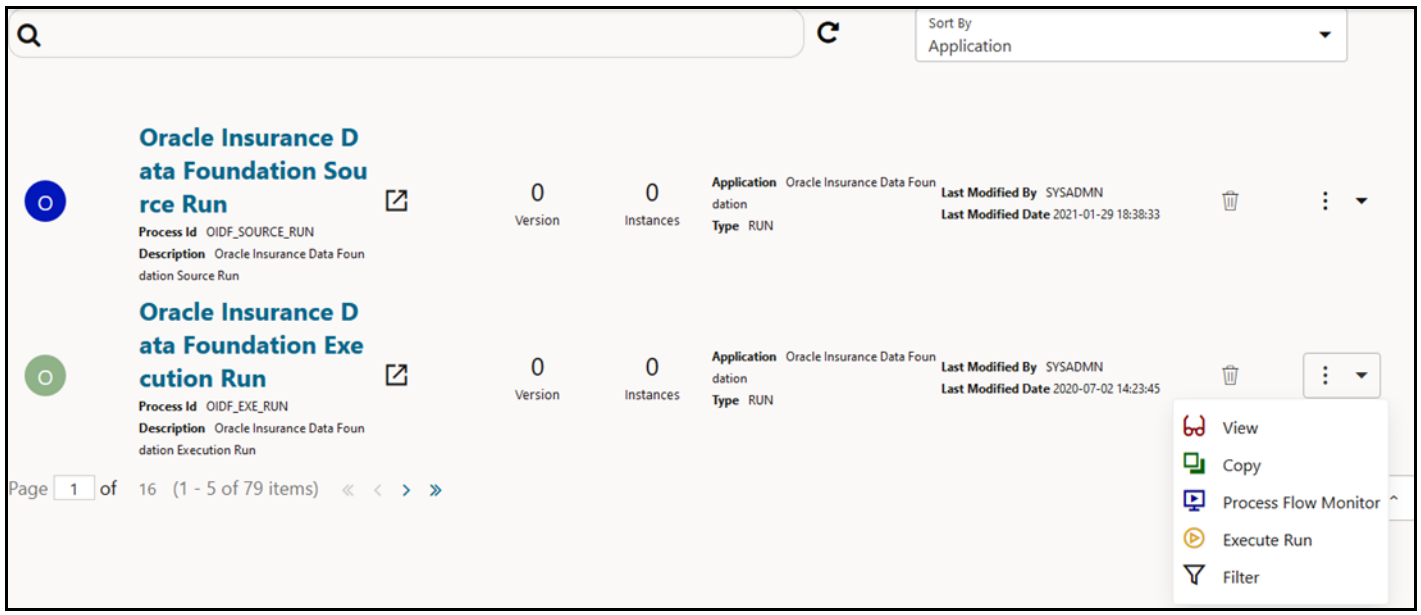
   - The following illustration is for the Oracle Insurance Data Foundation Source Run process.

Figure 33: OIDF Source Run Process - Select Execute Run



   - The following illustration is for the Oracle Insurance Data Foundation Execution Run process.

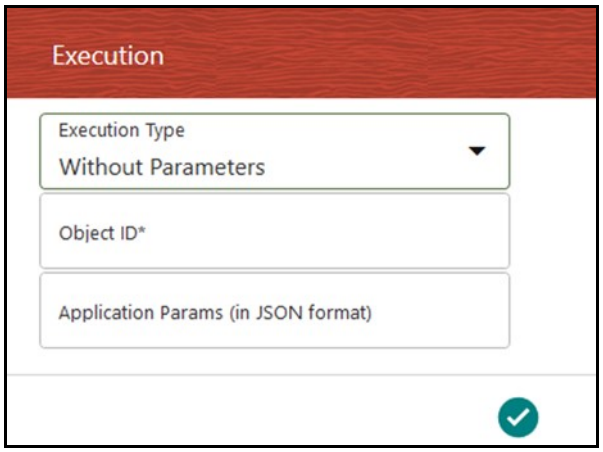
Figure 34: OIDF Execution Run Process - Select Execute Run



6. The **Execution** page is displayed with the Run parameters.

   - The following illustration is for the Oracle Insurance Data Foundation Source Run process (and for the Oracle Insurance Data Foundation Execution Run).

Figure 35: OIDF Source Run and OIDF Execution Run Processes - Execution Page Without Parameters



Enter and select the required details to execute the Run without parameters.

To execute the Run with parameters, select **With Parameters** in the **Execution Type** list.

Figure 36: OIDF Source Run Process - Execution Page With Parameters

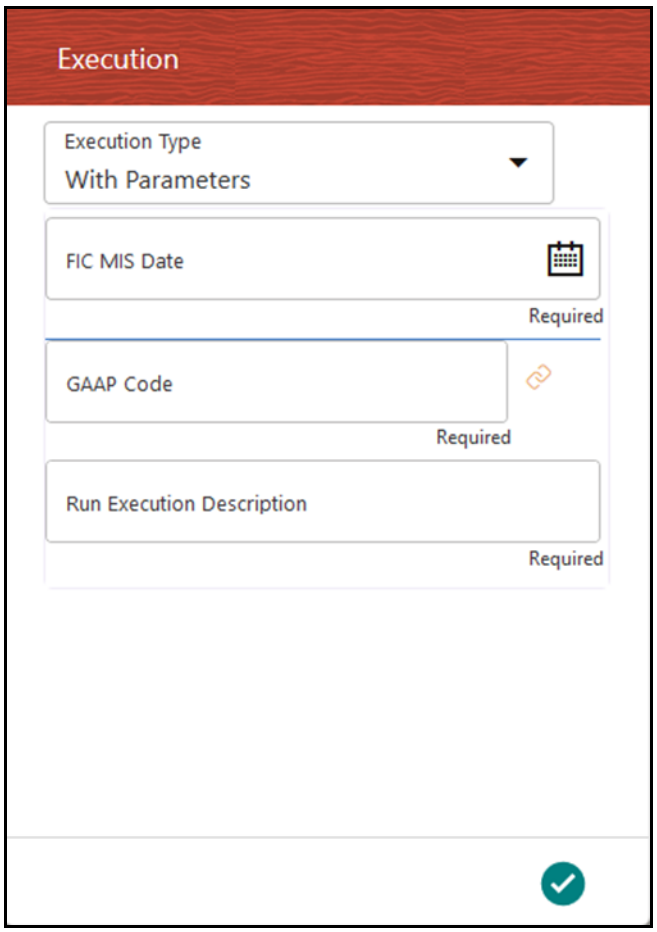
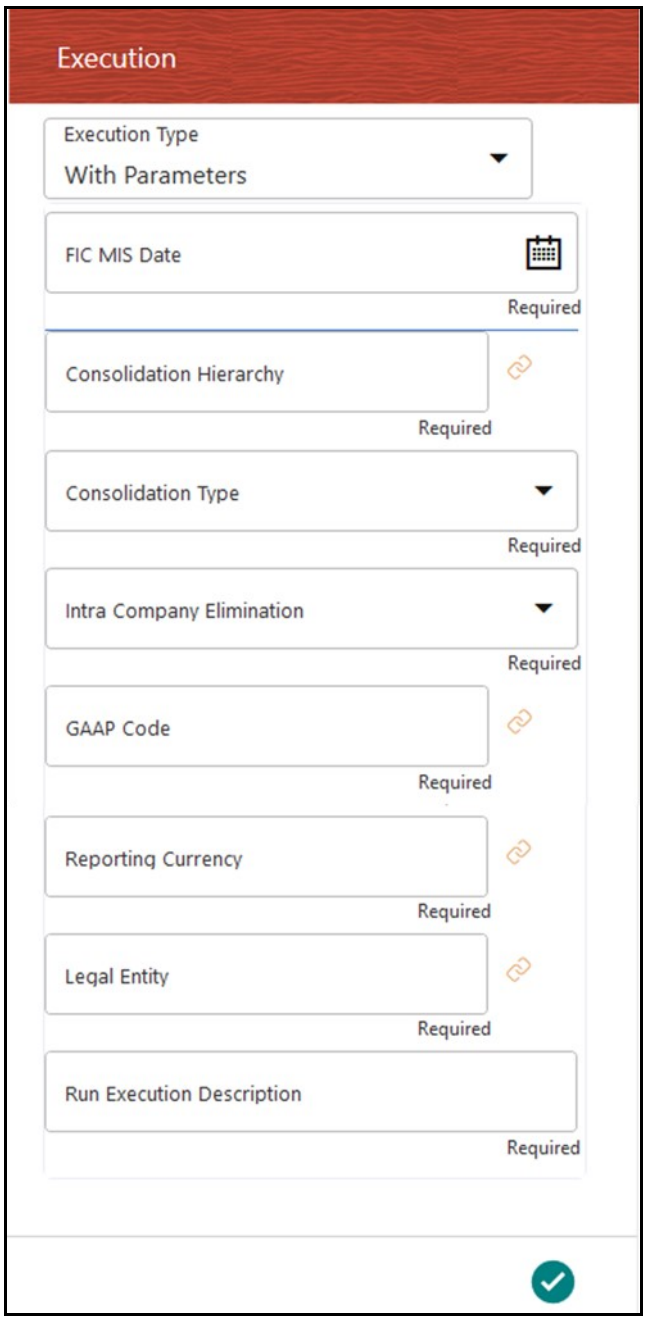


Figure 37: OIDF Execution Run Process - Execution Page With Parameters



Select or enter the required values for each field as follows.

Table 4: Insurance Liability Contracts Data Load Process - Execution Page With Parameters and description

Field Name	Description or Instruction
Reporting Currency	Use the icon to select the Reporting Currency Code used to calculate the amount during the data population in the target table.

Field Name	Description or Instruction
Legal Entity	Use the icon to select the Legal Entity Code to identify the legal entity used for the 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.
Consolidation Hierarchy	Use the icon to select the Legal Entity Hierarchy used for the consolidated run. This parameter is not required for the Solo Run.
GAAP Code	Use the icon to select the required accounting standard.
FIC MIS Date	Use the calendar icon to select the extraction date.
Run Execution Description	Enter a longer description of the Run.

7. To save the details, click.

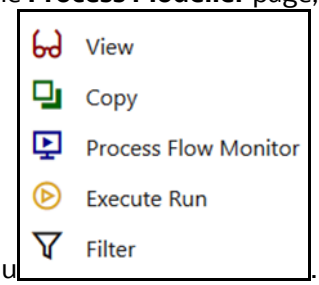
> **NOTE** The execution of the Oracle Insurance Data Foundation Source Run process is triggered using the selected FIC MIS DATE. The Run SKey is generated and inserted into the DIM_RUN table. For the Run SKey generated, the corresponding user-selected Run parameters are inserted into the RUN_EXE_PARAMETERS table.

8. To verify the Run execution status of the Oracle Insurance Data Foundation Source Run process (or the Oracle Insurance Data Foundation Execution Run process), see the Verify the Run Execution section.

> **NOTE** To abort, resume, or rerun the Run Pipeline process in the **Process Monitor** page, see the Abort, Resume, or Rerun the Process section.

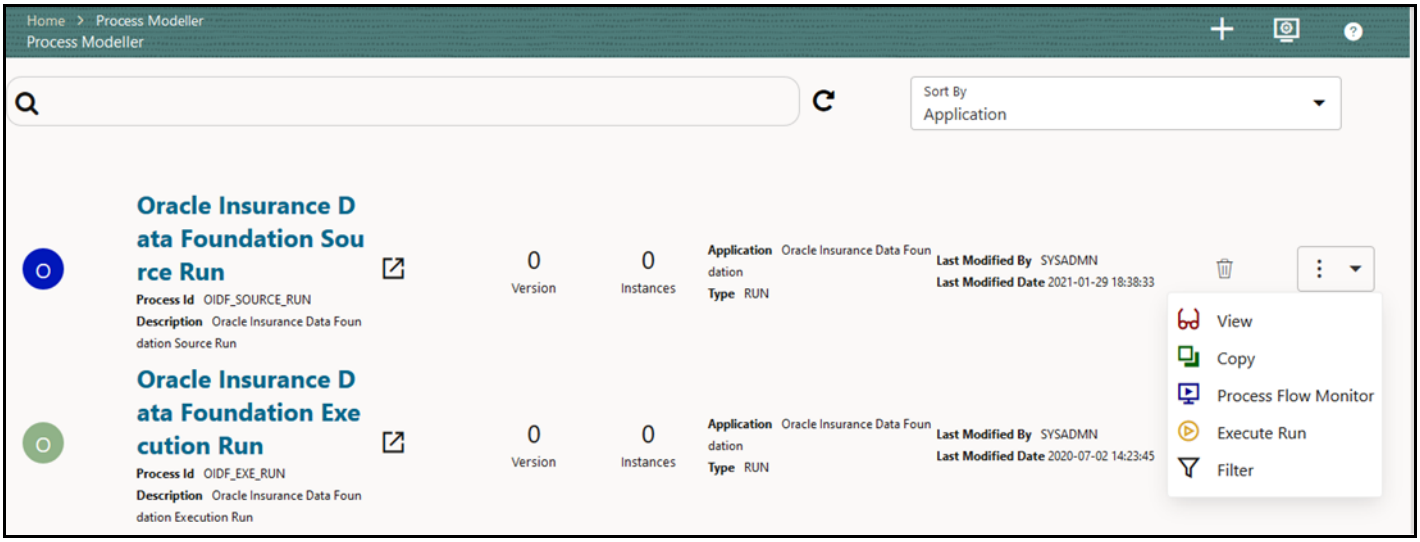
6.2.6.3 Verify the Run Execution

To verify the Run execution status of the Oracle Insurance Data Foundation Source Run process (or the Oracle Insurance Data Foundation Execution Run process), follow these steps:

1. To open the **Process Monitor** page, in the **Process Modeller** page, click  or select **Process Flow Monitor** from the Process Modeller menu  .

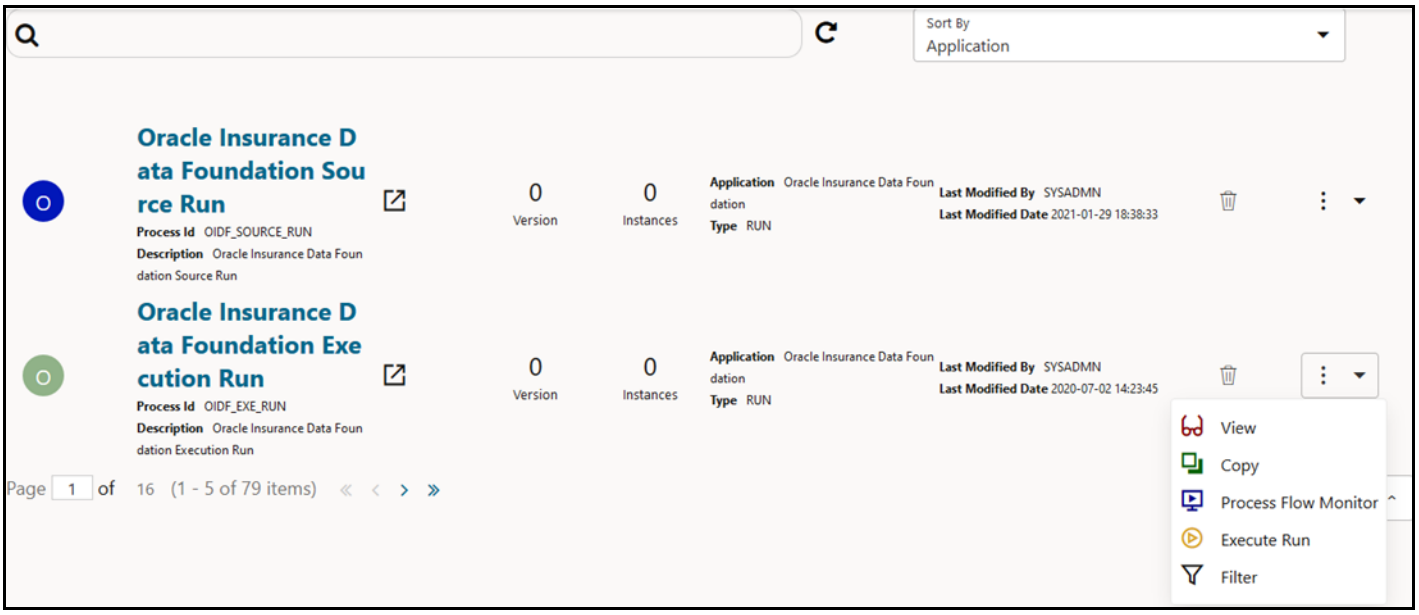
   - The following illustration is for the Oracle Insurance Data Foundation Source Run process.

Figure 38: OIDF Source Run Process – Select Process Flow Monitor



   - The following illustration is for the Oracle Insurance Data Foundation Execution Run process.

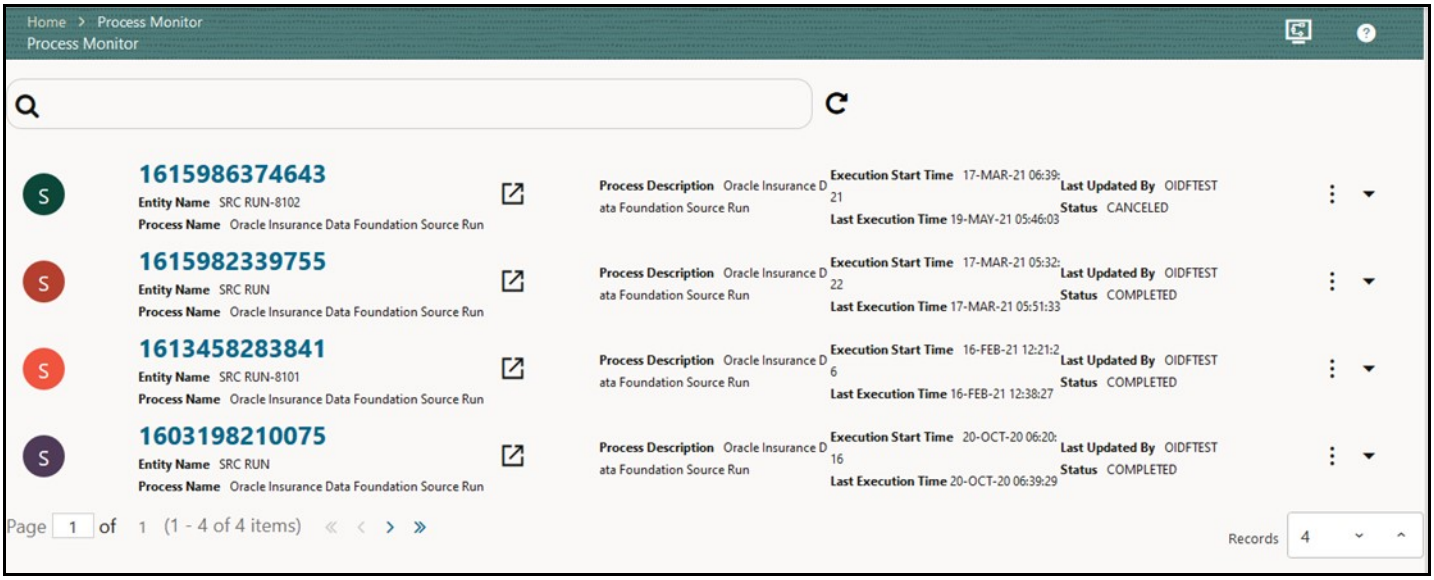
Figure 39: OIDF Execution Run Process – Select Process Flow Monitor



2. The **Process Monitor** page opens displaying all the Run instances corresponding to the Insurance Liability Contracts Data Load Process. In the **Process Monitor** page, for the required Run pipeline (process) that was executed, search the Run instance by Job ID, or by the Process Name **Oracle Insurance Data Foundation Source Run** (or **Oracle Insurance Data Foundation Execution Run**), and select the Run process instance.

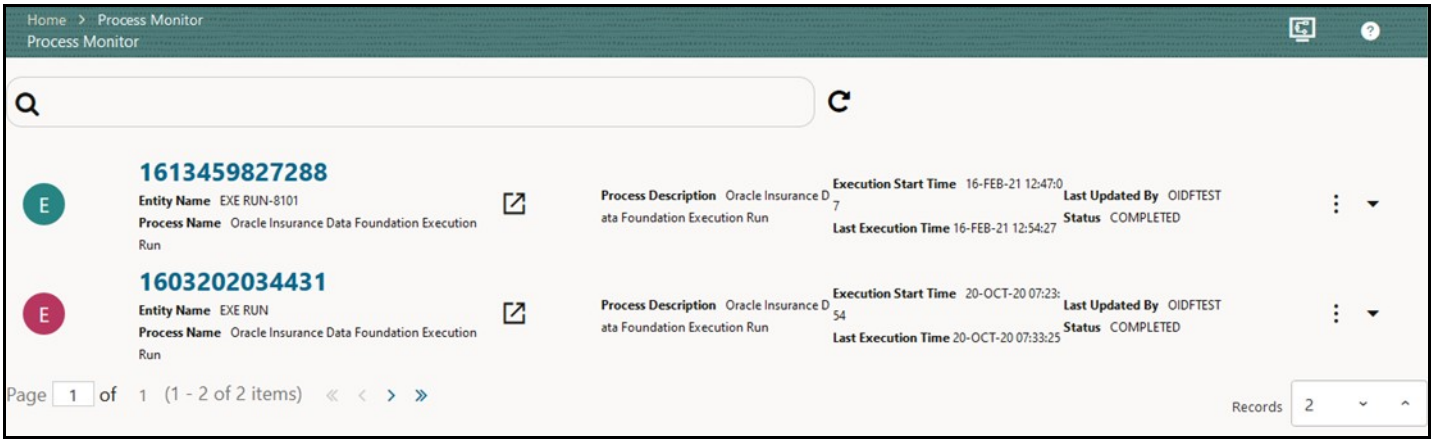
   - The following illustration is for the Oracle Insurance Data Foundation Source Run process.

Figure 40: OIDF Source Run Process – Process Monitor Page – Select Job ID



   - The following illustration is for the Oracle Insurance Data Foundation Execution Run process.

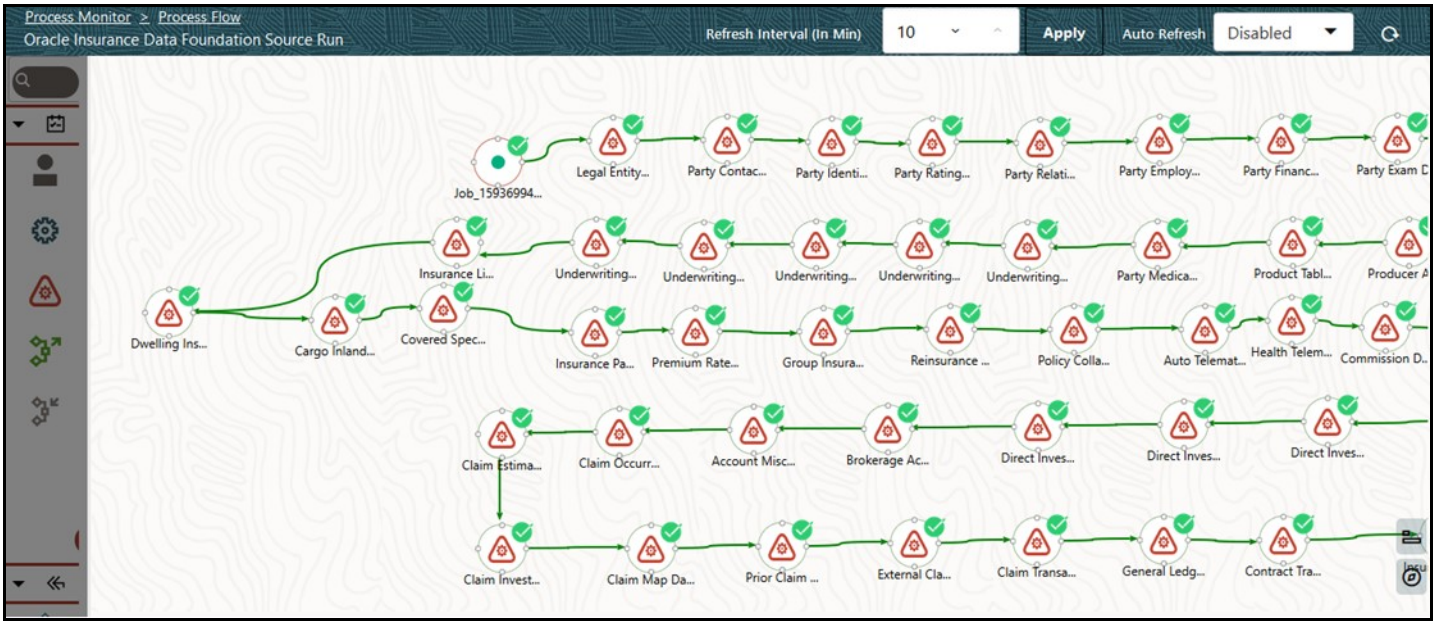
Figure 41: OIDF Execution Run Process – Process Monitor Page – Select Job ID



3. A status page opens, which displays the execution status of the executed Run instance.

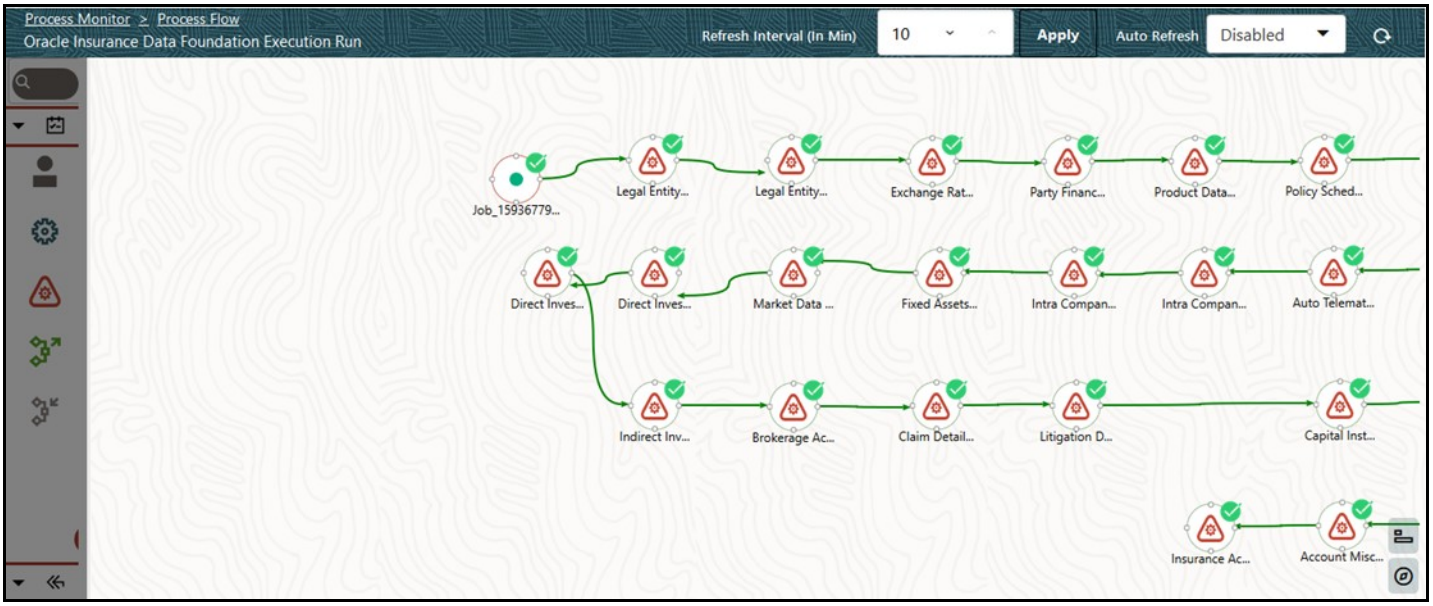
- The following illustration is for the Oracle Insurance Data Foundation Source Run process.

Figure 42: OIDF Source Run Process – Run Execution Status



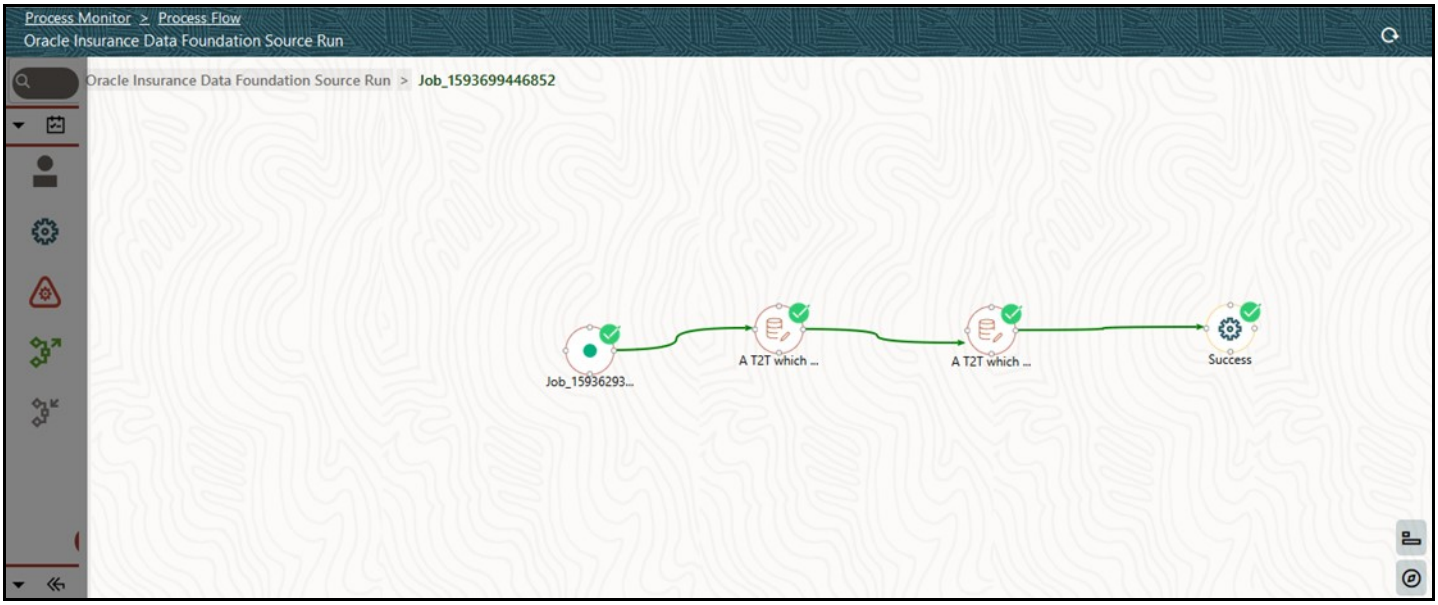
- The following illustration is for the Oracle Insurance Data Foundation Execution Run process.

Figure 43: OIDF Execution Run Process – Run Execution Status



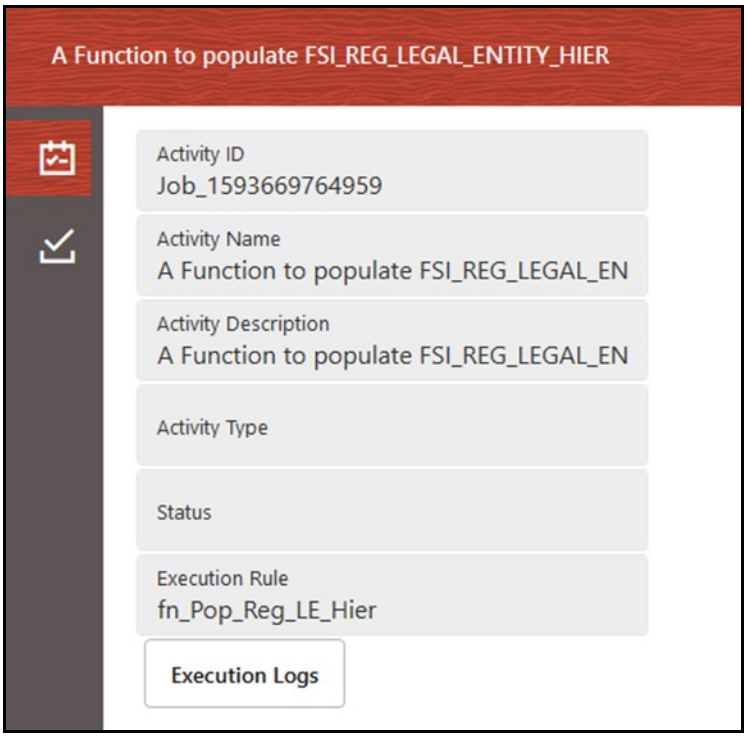
d. To verify the execution status for a Sub Pipeline, double-click the Sub Pipeline. The execution status related to the Sub Pipeline is displayed. In this illustration, the Legal Entity Data Load Process sub pipeline is used.

Figure 44: OIDF Sourced (Execution) Run Process – Run Execution Status of the Sub Pipeline



i. To verify the execution log of a T2T in the Sub Pipeline, double-click the required T2T widget. A page is displayed containing the Activity Definition and Activity Logs details. In the Activity Definition tab, click the **Execution Log**.

Figure 45: Activity Definition and Activity Logs details for T2T Run Execution in the OIDF Sourced (or Execution) Run Process



ii. The **Execution Logs** page is displayed. Select the required **Log File** in the list and click **View Log**.

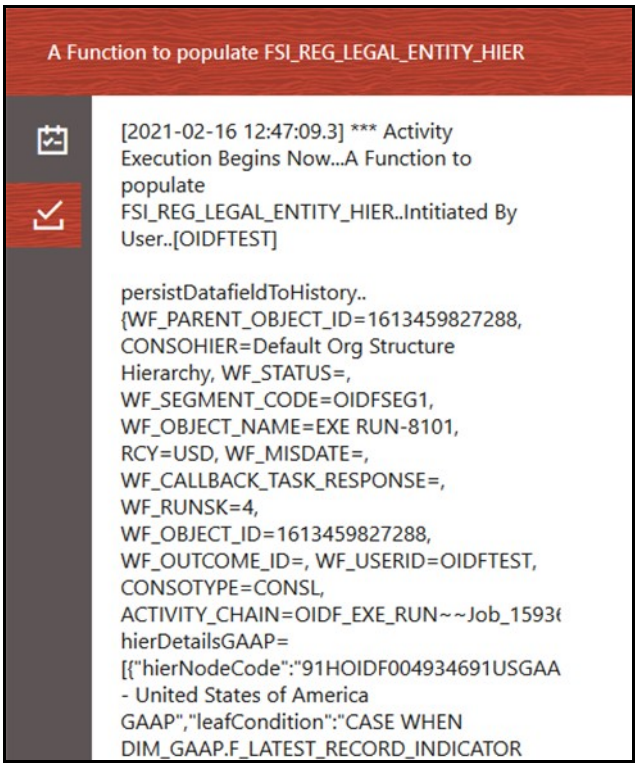
iii. The log details are displayed in the Log File Contents section. To download a copy of the log details, click **Download**.

The following two types of log files are generated:

- `T2T.log`
- `T2TCPP.log`

e. In the Sub Pipeline Run Execution Status page, to see the activity log corresponding to the executed T2T, click  .

Figure 46: Activity Logs of a T2T in OIDF Sourced (Execution) Run Process



For information about the complete functioning of the PMF, see the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

To design and execute a custom Oracle Insurance Data Foundation Source Run process, see the Configure and Manage Custom Pipelines for OIDF Functions section.

6.2.6.4 Check Error Messages

To check the error messages, see the log file present in the `ftpshare/logs/<Run_Date>/<infodom>/LOAD DATA` directory for any T2T.

6.2.6.5 Post T2T Process

When the T2T process is complete, data is populated in the Result tables.

NOTE

AAI_DMT_MAPPING_DETAILS table contains all the Stage, Dimension, and Fact tables. This table contains details about what source table-column level mapping must be done to the Result <TABLE>.<COLUMN>.

AAI_DMT_DEF_SOURCE_ENTITY table contains all the Expressions.

AAI_DMT_DEFINITION table contains all the Join conditions.

Each Join query must contain SKey. The join between a Dimension table and Fact table is based on the SKey.

7 Time Dimension Table

This section provides information about populating Time Dimension Transformation in the 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 are used to specify how to roll up the child member balances (for example, 3 months rolling average).

Topics:

7.1 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:

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

7.2 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 Insurance 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:
   - Iccserver
   - Router
   - AM Server
   - Message Server

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

7.3 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 OIDF Data Model.

7.4 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 OIDF has been seeded with the Batch ID <INFODOM>_DATA_FOUNDATION_SCD, which can be executed from Batch Execution section of OFSAAI. In the Parameter List, enter the Start Date and End Date. For example 19940101, 19941231.

NOTE You can load DIM_DATES for a fiscal year for ONE jurisdiction at a time. However, if the dates are populating incorrectly for the selected Jurisdiction, you should revisit the values entered in the DIM_FINANCIAL_YEARS TABLE and then repopulate the DIM_DATES.

You can also define a new Batch and an underlying Task definition from the Batch Maintenance page of OFSAAI. For more information on defining a new Batch, see the *How to Define a Batch* section.

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** page.
2. Click **Add** (+) button from the Task Details grid. The **Task Definition** page 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 OIDF 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.

7.5 Checking the Execution Status

To check the SCD batch execution status of Time Dimension Transformation, follow the procedure Check the Execution Status of the SCD Batch.

To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

8 Loading Multiple Load Runs in OFSAA

This section provides information about Loading Multiple Load Runs in OFSAA in the Oracle Insurance 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.

8.1 Objective

The following are the objectives of loading multiple Load Runs in OFSAA:

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

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

This release of the OIDF 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 this 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.

8.2 Design Details

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

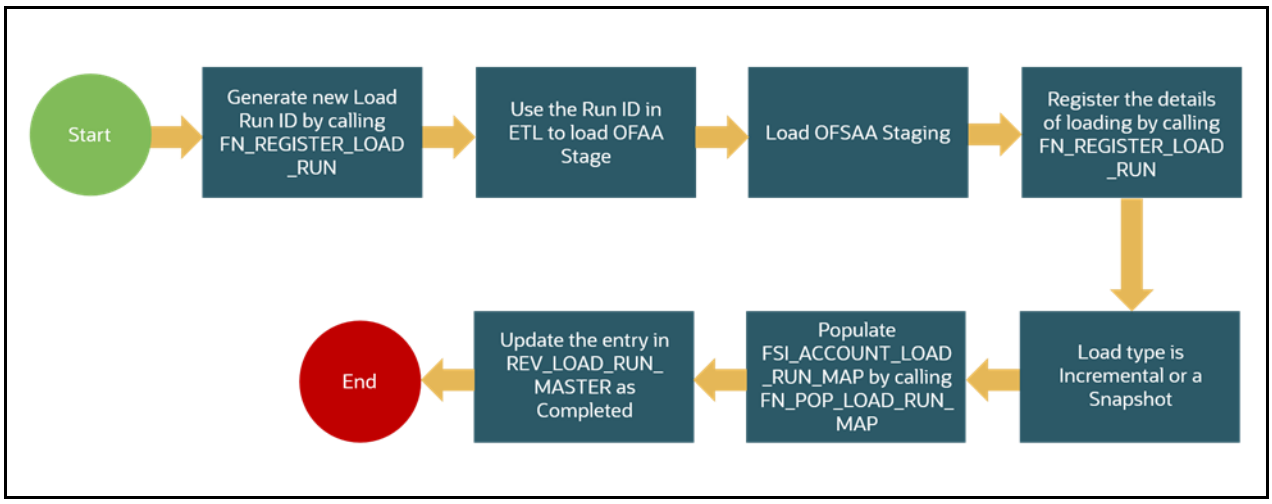
- ETL Tool
- OFSAA F2T
- OFSAA T2T
- OFSAA Excel upload
- OFSAA DIH

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.

8.3 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 47: Load Run process flow



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 5: Parameters of FN_REGISTER_LOAD_RUN

Parameters	Source Of Values	Example Values
Batch ID	Auto generated if you are using OFSAA Framework	OFSOIDFIN- 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
('OFSOIDFINFO_20150101_1','20150101','OIDF_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 6: 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	OIDF_Load	OFSOIDFINFO_20150101_1	In Progress

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

8.4 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

8.4.1 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 7: 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

FIC_MIS_DATE	V_ACCOUNT_NUMBER	V_GAAP_CODE	N_LOAD_RUN_ID	N_EOP_BAL
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 8: Populate Load Run Map Example

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

Example:

```
Declare

Result number;

Begin

Result: =
fn_pop_load_run_map('OFSOIDFINFO_20150101_1','20150101','STG_ANNUITY_CONTRACTS'
,1,'OIDF

_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 appears as follows.

Table 9: 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	Y
1-Jan-15	LOAN1001	USGAAP	1	N
1-Jan-15	LOAN1002	USGAAP	1	N
1-Jan-15	LOAN1001	USGAAP	2	Y
1-Jan-15	LOAN1002	USGAAP	2	Y
1-Jan-15	LOAN1000	USGAAP	2	Y

There may be a requirement to reload a complete snapshot of data in the OFSAA staging again. This could either be to satisfy an intraday reporting requirement or to load corrected source records in the OFSAA staging table. The earlier design forced users to truncate the staging table to accommodate the new set of dates. However, with the introduction of the Load Run identifier concept, you can retain both sets of data in the staging area and allow the downstream application to choose the correct set for processing. This involves 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 to be loaded again staging with a different load run identifier. Additionally, the 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 stamp with the new load run identifier.

STG_LOAN_CONTRACTS after load appears as follows.

Table 10: 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 appears as follows.

Table 11: The REV_LOAD_RUN_MASTER table after the second function call

LOAD_ RUN_ID	MIS_ DATE	LOAD_ TYPE	LOAD_ PURPOSE	START_DT_ TIME	LOAD_ RUN_ NAME	BATCH_ ID	LOAD_ RUN_ STATUS
1	01-JAN-15	B	BA	01-JAN-15 13:00 PM	OIDF_Load	OFSOIDF INFO_20 150101_1	Complete
2	01-JAN-15	B	BA	01-JAN-15 23:00 PM	Loan Corrections	OFSOIDF INFO_20 150101_2	In Progress

8.4.2 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 12: 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

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 13: Populate Load Run Map example

Parameters	Source of Values	Example Values
Batch ID	Auto-generated if you are using OFSAA Framework	OFSOIDFINFO_20150101_1
MIS-Date	Input from Customer	01/01/2015
Stage Table Name	Input from Customer	STG_ANNUITY_CONTRACTS
Load Run ID	Input from Customer	1
Load Run Name,	Input from Customer	OIDF_Load

Example

```
Declare

Result number;

Begin

Result: =
fn_pop_load_run_map('OFSOIDFINFO_20150101_1','20150101','STG_ANNUITY_CONTRACTS'
,1,'OIDF

_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 14: 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

8.5 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.

Topics:

8.5.1 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: Register Load Run Details Example

Parameters	Source of Values	Example Values
Batch ID	Auto-generated if you are using OFSAA Framework	OFSOIDFINFO_20150101_1
MIS-Date	Input from Customer	01/01/2015
Stage Table Name	Input from Customer	STG_ANNUITY_CONTRACTS
Load_Run_Id	Input from Customer	1
Load Run Name,	Input from Customer	OIDF_Load
Load Type	Input from Customer	S - Full SnapShot I - Incremental

Example:

```
Declare
Result number; Begin
Result: =
fn_register_load_details('OFSOIDFINFO_20150101_1','20150101',
'STG_ANNUITY_CONTRACTS',1,'OIDF_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 16: 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

8.5.2 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 = 'OIDF_LOAD' and LOAD_RUN_ID = 1;
```

8.6 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 OFSAAI.

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

Topics:

8.6.1 Complete Snapshot Load Scenario

To enable the Data Load into Results table from staging using the Load Run concept in case of Full SnapShot, 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: FCPS T2T for Stage Loan Contracts

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

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

- Modify the Corresponding Batch Task Each Time with Load Run Idxxx
  a. Select Batch, Task (T2T_FCPS_STG_ANNUITY_CONTRACTS)
  b. Click **Edit**.
  c. Add **Highlighted Condition** in **Default Value** and **Save** (Each Time we must provide the Load Run ID. Here in the following example, 1 is used.)

     [DRCY]=USD,[LOADRUN]=1

d. Execute the batch after this change. It loads the Result table with the given Load Run ID Records.

e. Modify the **Corresponding Process Task** of a Run each time with the Load Run ID.

f. Select the **Process** and click **Edit**.

g. Choose **Components**, select the **Object** (**T2T_FCPS_STG_ANNUITY_CONTRACTS**) and click **Components**.

h. Click the dropdown button associated with the T2T.

i. Add the **Highlighted Condition** with the following text, and **Save** DRCY, USD, LOADRUN, 1.

j. Execute the Run. It loads the Result table with the given Load Run ID records.

8.6.2 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, FCPS T2T for Stage Annuity Contracts:

Modify Join Condition inside T2T:

Join Condition to be added in all relevant T2T

For example, STG_ANNUITY_CONTRACTS T2T join

```
INNER JOIN FSI_ACCOUNT_LOAD_RUN_MAP

ON FSI_ACCOUNT_LOAD_RUN_MAP.V_ACCOUNT_NUMBER =

STG_ANNUITY_CONTRACTS.V_ACCOUNT_NUMBER

AND FSI_ACCOUNT_LOAD_RUN_MAP.N_LOAD_RUN_ID =
STG_ANNUITY_CONTRACTS.N_LOAD_RUN_ID AND FSI_ACCOUNT_LOAD_RUN_MAP.FIC_MIS_DATE =
STG_ANNUITY_CONTRACTS.FIC_MIS_DATE AND FSI_ACCOUNT_LOAD_RUN_MAP.V_GAAP_CODE =
STG_ANNUITY_CONTRACTS.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.

9 About Dimension Loading Process

This chapter provides information about Dimension Loading Process in the Oracle insurance data foundation application.

Topics:

9.1 About SCD Component

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.

Topics:

9.1.1 Type 1 SCDs - Overwriting

The Type 1 SCD overwrites old data with new data and therefore does not track historical data. This is useful for making changes to dimension data. This is the default type of dimension that is created. You do not need to specify any additional information to create a Type 1 SCD.

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 17: Type 1 SCD methodology example after the execution during a processing period

N_PRODUCT_ SK EY	V_PRODUCT_ NAME	D_START_DATE	D_END_DAT E	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 that 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, when SCD is executed for the new processing period, the record in the above example changes to the following.

Record Change Example:

Table 18: Type 1 SCD methodology example after the execution and record change during a new processing period

N_PRODUCT_SK EY	V_PRODUCT_NAME	D_START_DATE	D_END_DATE	F_LATEST_RECORD_ INDICATOR
1	Personal Loan	6/30/2010	12/31/9999	Y

9.1.2 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.

Therefore, a Type 2 SCD retains the full history of values. When the value of a chosen attribute changes, the current record is closed. A new record is created with the changed data values and this new record becomes the current record. Each record contains the effective time and expiration time to identify the time period between which the record was active.

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 19: Type 2 SCD methodology example after the execution during a processing period

N_PRODUCT_SKE Y	V_PRODUCT_NAME	D_START_DATE	D_END_DATE	F_LATEST_RECORD_INDI CATOR
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'.

9.1.3 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 OIDF SCD Technical Metadata file of this release.

9.2 Populating Data in Dimension Table

Topics:

Data Foundation solutions use the SCD component to handle dimensional data changes. For more details about the SCD component and SCD process, see the Slowly Changing Dimension (SCD) Process and the Execute the SCD Batch sections respectively.

9.2.1 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.

9.3 Tables Used by the SCD Component

For the metadata design, see the *Tables Used by the SCD Component* Section in the OIDF Metadata Design Document Release 8.1.x.

9.3.1 About DIM_<dimensionname>_V Table

The database view which SCD uses as the source.

Example:

DIM_BANDS_V

These views come as part of the 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 must be added for every user-defined dimension.

Example:

```
create sequence SEQ_DIM_<DIM> minvalue 1
maxvalue 999999999999999999999999999
increment by 1
```

9.4 Loading Dimension Tables

Load data into the Dimension tables using one of the following suitable processes.

Topics:

9.4.1 Seeded Data

This is an automatic process wherein the Seeded data for Dimensions, which are packaged within the installer, are populated during the installation.

> **NOTE** To access the Seeded data for Dimensions, see the latest version of the Oracle Insurance Data Foundation Application Pack Run Chart.

9.4.2 Loading Policy Tables through Insurance SCD

To execute Policy Dimension SCD batch <INFODOM>_INSURANCE_SCD, see Execute the SCD Batch of the DIM_POLICY table.

9.4.3 Loading Dimension Tables through Data Foundation SCD

To execute an SCD batch <INFODOM>_DATA_FOUNDATION_SCD for any other Dimension table, see Execute the <INFODOM>_DATA_FOUNDATION_SCD batch for the required Dimension table.

NOTE For 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.

9.4.4 Loading Key Dimensions using AMHM and Hierarchy Flattening

The Dimension Loader functionality in the Data Foundation application enables you to load Dimension tables such as DIM_ORG_UNIT, DIM_GL_ACCOUNT, DIM_COMMON_COA, DIM_PRODUCT, and DIM_ORG_STRUCTURE.

For more information about loading the Dimensions using AMHM, see the *Dimension Management* section in Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0 and *Dimension Load Procedure* section in the Oracle Financial Services Analytical Applications Data Model Utilities User Guide.

9.4.5Loading Data from STG_INTF Tables to DIM_INTF Tables

For the metadata design, see the *Loading Dimension Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

9.4.6 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 <infodom>_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 information, see the *Dimension Loaders* section in the *Data Loaders* section in the Oracle Financial Services Analytical Applications Data Model Utilities User Guide.

9.4.7 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 <infodom>_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 must be transformed.

For more details, see the *Hierarchy Transformation* section in the *Data Loaders* section the Oracle Financial Services Analytical Applications Data Model Utilities User Guide.

9.4.8 Checking the SCD Batch Execution Status

To check the SCD batch execution status, see Verify Log Files and Check Error Messages if any.

9.5 Supplementary Information for Dimension Loading Process

The following are the supplementary information required for the Dimension loading process.

Topics:

- Improving SCD Performance
- Handling Multiple GAAP Codes for the Same Account Number for the Same MIS Date in SCD
- Handling Multiple GAAP Codes for the Same Account Number for the Same MIS Date in the Function

9.5.1 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 above. If these have to be used in OFSAAI versions 7.3.2.2.0 or 7.3.2.3.0, 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 the 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 exist.

For improving performance, hints for the MERGE query, which is generated internally by the SCD, can be provided under MERGE_HINT. The following session alters can be mentioned in the SESSION_ENABLE_STATEMENT and SESSION_DISABLE_STATEMENT columns.

1. SESSION_ENABLE_STATEMENTs are executed before the MERGE in the SCD and SESSION_DISABLE_STATEMENTs are executed after the SCD MERGE.
2. 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 the SYS_TBL_MASTER table.
3. 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 20: Merge Hint and Session Enable Statement details

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

4. Execute all the tasks in parallel. This may cause N_RCV_LEG_ACCT_SKEY to have an incremental value as compared to N_ACCT_SKEY.
5. Execute the SQL file with all the SESSION_DISABLE_STATEMENTs, after the successful completion of the SCD batch.
6. After the DIM_ACCOUNT table is populated using this approach, you cannot use the initial approach (FN_POPDIMACCOUNT) as this will lead to SKey conflict.
7. Ensure that you have set the value of the sequence SEQ_DIM_ACCOUNT_SCD as max (value of SKey in DIM_ACCOUNT) +1, before moving from old to a new approach.
8. The F_LATEST_RECORD_INDICATOR for an existing DIM_ACCOUNT data already loaded by the function must be updated to 'Y' before running the SCD, failing which a new SKey may get generated for the same account number.

9. SCD execution occurs based on the GAAP code, which is configured in the SETUP_MASTER table. These are introduced to tackle the scenario of multiple GAAP codes. Whether or not there exist multiple GAAP codes, SETUP_MASTER must be manually configured as follows:

Table 21: Sample of the GAAP code to configure the SETUP_MASTER table

V_COMPONENT_CODE	V_COMPONENT_DESC	V_COMPONENT_VALUE
DEFAULT_GAAP	DEFAULT_GAAP	USGAAP

Where V_COMPONENT_VALUE must be manually populated with the required GAAP code. For all other GAAP codes, ensure to update SETUP_MASTER manually before running DIM_ACCOUNT SCD.

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

NOTE For illustration, Account Dimension is considered.

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 must 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'.

9.5.3 Handling Multiple GAAP Codes for the Same Account Number for the Same MIS Date in the Function

NOTE For illustration, Account Dimension is considered.

For the FN_POPDIMACCOUNT function, you have to create views and use these views instead of the tables in the FSI_DIM_ACCOUNT_SETUP_DETAILS table. For Product Processors having GAAP code as part of the Primary Key, create a view on the table with a filter on the GAAP code as:

- where V_GAAP_CODE = (SELECT V_COMPONENT_VALUE FROM SETUP_MASTER WHERE V_COMPONENT_DESC = 'DEFAULT_GAAP')
- Use this view under TABLE_NAME in the FSI_DIM_ACCOUNT_SETUP_DETAILS table. If there are different GAAP codes for two distinct account numbers for the same MIS date, then the function 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'.

NOTE If STG_OPTION_CONTRACTS is loaded for two MIS dates, and FN_POPDIMACCOUNT is executed, then all records irrespective of the MIS date will get loaded. To resolve this, remove the expression under SQL_TEXT in FSI_DIM_ACCOUNT_SETUP_DETAILS for STG_OPTION_CONTRACTS and use the same expression to create a view and use this view as the TABLE_NAME in FSI_DIM_ACCOUNT_SETUP_DETAILS.

10 Account Dimension Table

This section provides information about the Account Dimension loading process in the Data Foundation application.

Topics:

The account is where an insurer deposits premium from policies it underwrites and from which that Insurance Company funds day-to-day operations of the business.

For the metadata design, see the *Account Dimension Table* Section in the OIDF Metadata Design Document Release 8.1.x.

10.1 About SCD Process for Populating Account Dimension Table

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

- V_ACCOUNT_NUMBER
- N_ACCT_SKEY
- N_RCV_LEG_ACCT_SKEY
- FIC_MIS_DATE

This approach replaces the function load in which the table DIM_ACCOUNT is getting loaded through the function, FN_POPDIMACCOUNT. This loads the above-mentioned columns into the DIM_ACCOUNT table. Here, the sources are the different product processor tables present in the solution, which are configured in the FSI_DIM_ACCOUNT_SETUP_DETAILS table.

Topics:

10.1.1 About <INFODOM>_DIM_ACCOUNT_SCD Batch

Batch <INFODOM>_DIM_ACCOUNT_SCD is introduced with multiple tasks under it.

These tasks represent their corresponding SCD processes where different product processors are the source and DIM_ACCOUNT is the target. The MAP_REF_NUMs mentioned in the following table are introduced into the 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.

10.2 Deploying Account Dimension Table on Hive

All RDBMS related Account Dimension table can also be deployed on Hive (Stage and Results on Hive).

10.3 Populating Account Dimension Table

To populate data into the Account Dimension table through the SCD process, see Execute the SCD Batch of the DIM_ACCOUNT table.

NOTE You can also follow this SCD process to populate data into any Hive related Account Dimension table.

For supplementary information about populating data in the Dimension table, see the Supplementary Information for Dimension Loading Process section.

11 Policy Dimension Table

This section provides information about Policy Dimension and its loading process in the Oracle Insurance Data Foundation application.

Topics:

The policy is the legal document issued by an Insurance Company (Insurer) to a policyholder (Insured) which states the terms and conditions of the insurance coverage.

The policy dimension table stores list of all policies issued by the insurer.

For the metadata design, see the *Policy Dimension Table* Section in the OIDF Metadata Design Document Release 8.1.x.

11.1 About SCD Process for Populating Policy Dimension Table

The SCD process for the DIM_POLICY table generates individual numeric Surrogate Keys for every policy code. These columns are loaded during SCD process:

- V_POLICY_CODE
- N_POLICY_SKEY
- FIC_MIS_DATE
- V_PRODUCT_PROCESSOR_NAME
- V_PROD_CODE
- V_ENTITY_CODE

This approach replaces the function load in which the table DIM_POLICY is getting loaded through the function, FN_POPDIMPOLICY. This loads the above-mentioned columns into the DIM_POLICY table. Here, the sources are the different product processor tables present in the solution, which are configured in the FSI_DIM_POLICY_SETUP_DETAILS table.

Topic:

11.1.1 About <INFODOM>_INSURANCE_SCD Batch

Batch <INFODOM>_INSURANCE_SCD has been introduced with 30 tasks under it.

These tasks represent the corresponding SCD processes where different product processors are the source and DIM_POLICY is the target. MAP_REF_NUMs 224, 401, 402, and 403 are introduced into the 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.

11.2 Deploying Policy Dimension Table on Hive

All RDBMS related Policy Dimension table can also be deployed on Hive (Stage and Results on Hive).

11.3 Populating Policy Dimension Table

To populate data into the Policy Dimension table through the SCD process, see Execute the SCD Batch of the DIM_POLICY table.

> **NOTE** You can also follow this SCD process to populate data into any Hive related Policy Dimension table.

For supplementary information about populating data in the Dimension table, see the Supplementary Information for Dimension Loading Process section.

12 Executing OIDF Processes through Process Modelling Framework

Process Modelling Framework (PMF) is a process design and execution framework in OFSAA that facilitates the implementation of process sequences or pipelines. Use the framework to orchestrate Run Pipelines and Business Pipelines along with the objects that are a part of these pipelines.

This section provides information about the usage of the Process Modeling Framework (PMF) feature in executing the OIDF processes.

> **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.

Topics:

12.1 Overview

In OIDF Rule Run Framework (RRF), PMF is used to create a Run definition in a Run process. The visual representation of the Run is enabled through PMF by the construction of a Run Pipeline (process). The set of tasks required to carry out the OIDF functions is managed through PMF. Through the PMF, you can execute the following out-of-the-box Run processes for data loading:

- Oracle Insurance Data Foundation Source Run
- Oracle Insurance Data Foundation Execution Run

12.2 Process Modeling Framework Roles

Before you access PMF for the management of OIDF tasks, ensure to provide specific users with security or access rights. To grant access to the PMF functionality, assign the following PMF Roles to the user.

See the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0 for details.

Table 22: PMF User Role Codes and Role Names

Role Code	Role Name
WFMWRITE	Manage Workflow Monitor

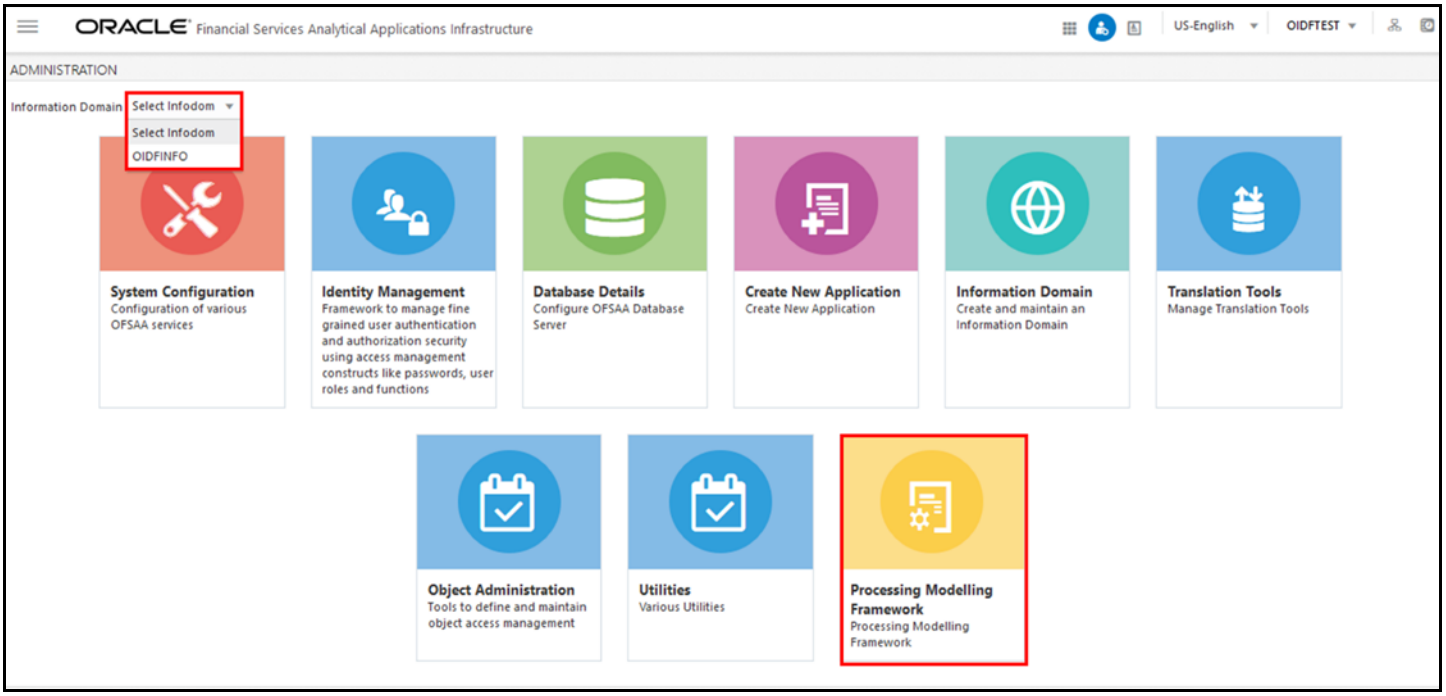
Role Code	Role Name
WFADMINACC	Process Admin User
WFDELACC	Process Delegation User
WFACC	Workflow Access
WFADV	Workflow Advanced
WFAUTH	Workflow Authorize
WFDELGADM	Workflow Delegation Admin
WFMACC	Workflow Monitor Access
WFREAD	Workflow Read
WFWRITE	Workflow Write

12.3 Access the Process Modelling Framework in OIDF

To access the PMF, follow these steps:

1. Log in to the OIDF application and click **Administration**.
2. In the **Information Domain** list, select the information domain required for OIDF. Then click the **Process Modelling Framework** tile.

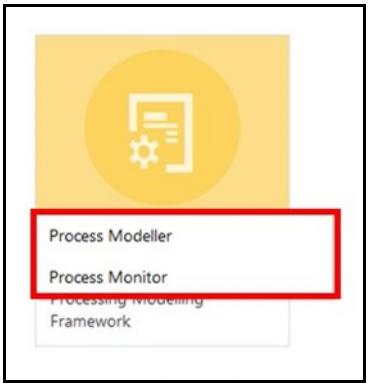
Figure 48: Select Process Modelling Framework Tile in the Administration Page



3. A submenu is displayed with the following menu items:
   - To access the Process Modeller, click **Process Modeller**.

- To monitor currently running processes, click **Process Monitor**.
- To delegate the Run process, click **Delegation**.

Figure 49: Select the Process Modeller or Process Monitor option in PMF submenu



12.4 Use the Process Modelling Framework for OIDF Functions

You can use PMF in one of the following ways to perform OIDF functions or tasks:

- Use Standard OIDF Run Pipelines
- Configure and Manage Custom Pipelines for OIDF Functions

ATTENTION Ensure you have executed all the Runs and Batch IDs mentioned in the latest RUN_CHART_SUMMARY sheet of the Oracle Insurance Data Foundation Application Pack Run Chart to access the out-of-the-box Run pipelines mentioned in the Use Standard OIDF Run Pipelines section.

Use the following Process Modelling Framework features to perform the OIDF functions either by using the standard pipelines (processes) or by creating custom pipelines:

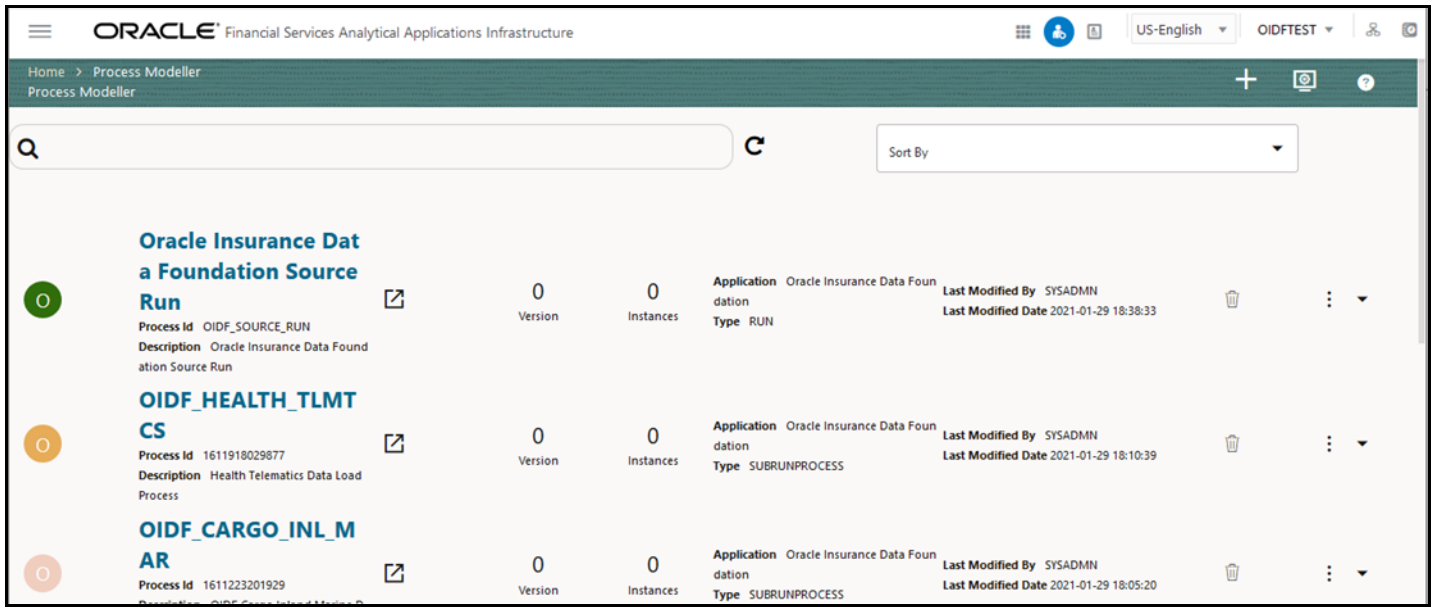
- Process Modeller
- Process Monitor

12.4.1 Process Modeller

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

This is a sample Process Modeller page in PMF.

Figure 50: Sample Process Modeller page

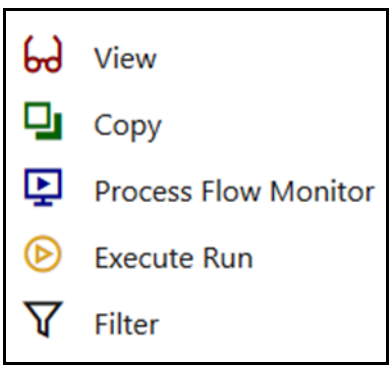


The Process Modeller page displays the existing Business Process Pipelines and Run Pipelines with the details such as Process ID, Process Name, Process Description, Version, Instance, Application, and Last Modified.

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 menu:

Figure 51: Process Modeller Menu



- 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 **Execute Run** to execute a Run Pipeline.
- Click **Filter** to apply a filter condition to a Run Pipeline.

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

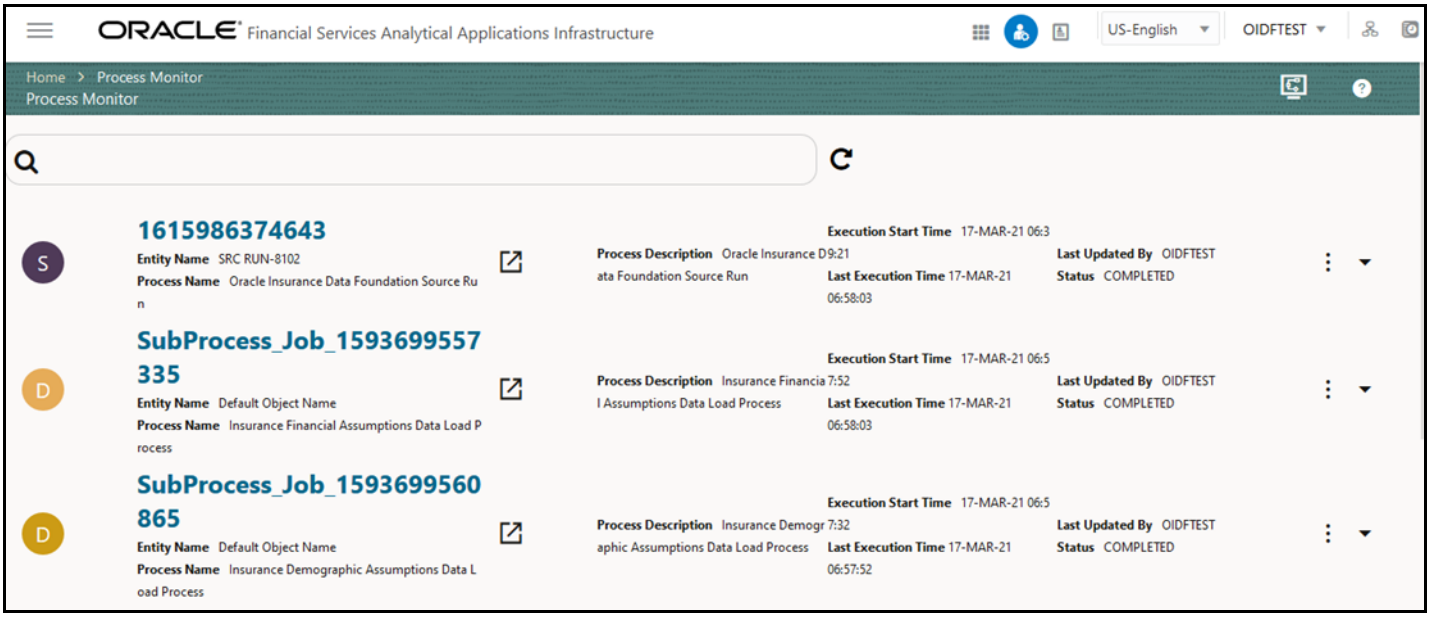
- Use the Search grid to search for a specific Pipeline by providing a keyword from the Process ID, Process Name, or Process Description and then click **Search**. Click **Reset** to reset the Search fields.
- You can sort the Pipelines based on Process ID, Process Name, or Application. Click the **Sort 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 to launch the Process Monitor page.

12.4.2 Process Monitor

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

This is a sample Process Monitor page in PMF.

Figure 52: Sample Process Monitor Page

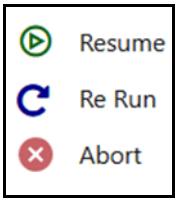


This page displays all the workflows, which are invoked from the application with details such as Entity Name, Process Name, Process Description, Execution Start Time, Last Execution Time, Last Updated By, and Status.

You can perform the following tasks on the **Process Monitor** page:

- Click ⋮ to view the following submenu:

  Figure 53: Process Monitor with Re Run, Abort, and Resume options

  - Abort: To abort an ongoing Run Pipeline process.
  - Resume: To resume a Run Pipeline process.
  - Re Run: To execute a Run Pipeline process again irrespective of the previous execution status.
- Use the **Search** grid to search for a specific Pipeline by providing a keyword from the Job ID, Process Name, or Process Description, and then click **Search**. Click **Reset** to reset the Search fields.
- You can sort the Pipelines based on Job ID, Process Name, or Application. Click the **Sort by** list 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  to open the Process Modeller page.

12.4.3 Use Standard OIDF Run Pipelines

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

- Oracle Insurance Data Foundation Source Run: This Run pipeline (Process) loads all *non-Run* enabled tables in OIDF.
- Oracle Insurance Data Foundation Execution Run: This Run pipeline (Process) loads all *Run* enabled tables in OIDF.

To load all **non-Run** enabled tables in OIDF using the out-of-the-box Oracle Insurance Data Foundation Source Run process, or to load all **Run** enabled tables in OIDF using the out-of-the-box Oracle Insurance Data Foundation Execution Run process, perform the steps in the following section:

- Table to Table (T2T) Loading Process

12.4.4 Configure and Manage Custom Pipelines for OIDF Functions

This section provides information about configuring and managing the custom pipelines (processes) for OIDF functions.

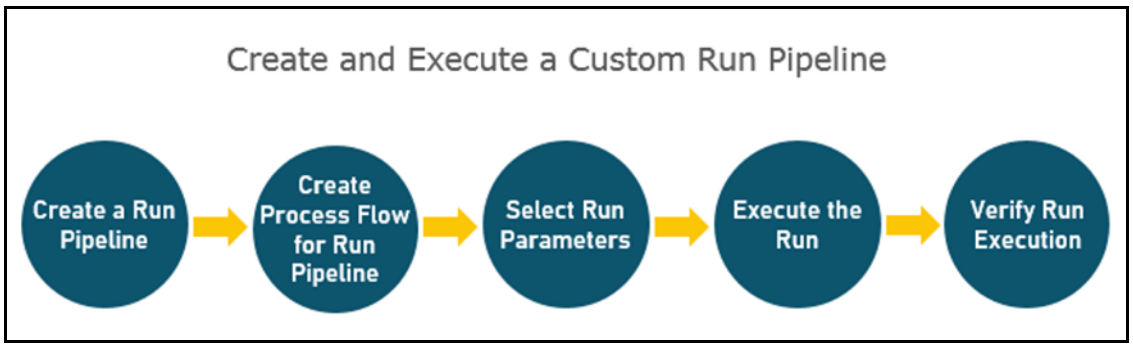
> **NOTE** Ensure you have completed the required settings mentioned in the following section:
>
> - Prerequisites for loading T2T

In this section, to create and execute a custom Run pipeline (process) for OIDF in PMF using Fact Common Policy Summary T2T for illustration, follow these steps:

- Create a Run Pipeline
- Create a Process Flow for a Run Pipeline
- Select the Run Parameters and Execute the Run
- Abort, Resume, or Rerun the Process
- Verify the Run Execution

The Insurance Liability Contracts Data Load Process is created and executed in this section for Fact Common Policy Summary T2T. The visual representation of the preceding steps is depicted in the following illustration.

Figure 54: Create and Execute a Custom Run Pipeline



NOTE

Use a Business Pipeline to design a Business Process, which consists of a sequence of internal or external tasks through well-defined interfaces. Using the designer, you can design the entire business flows consisting of several types of tasks or another business pipeline.

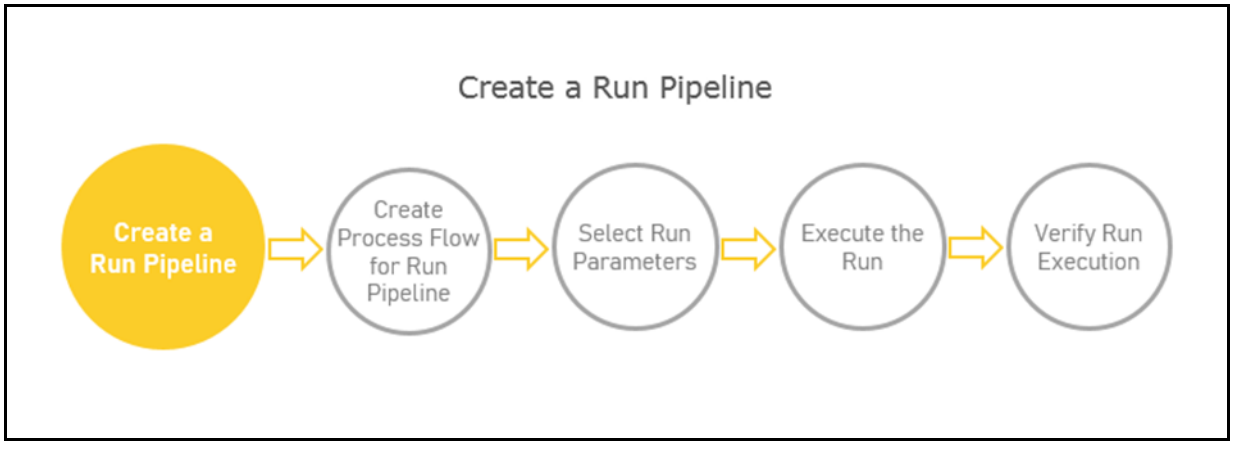
For information about designing a Business Pipeline, see the *Orchestration of a Business Pipeline* section in the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

To use a standard out-of-the-box Run pipeline for OIDF, see the Use Standard OIDF Run Pipelines section.

12.4.4.1 Create a Run Pipeline

Visual representation of the Run is enabled through PMF by the construction of a Run pipeline. Several 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.

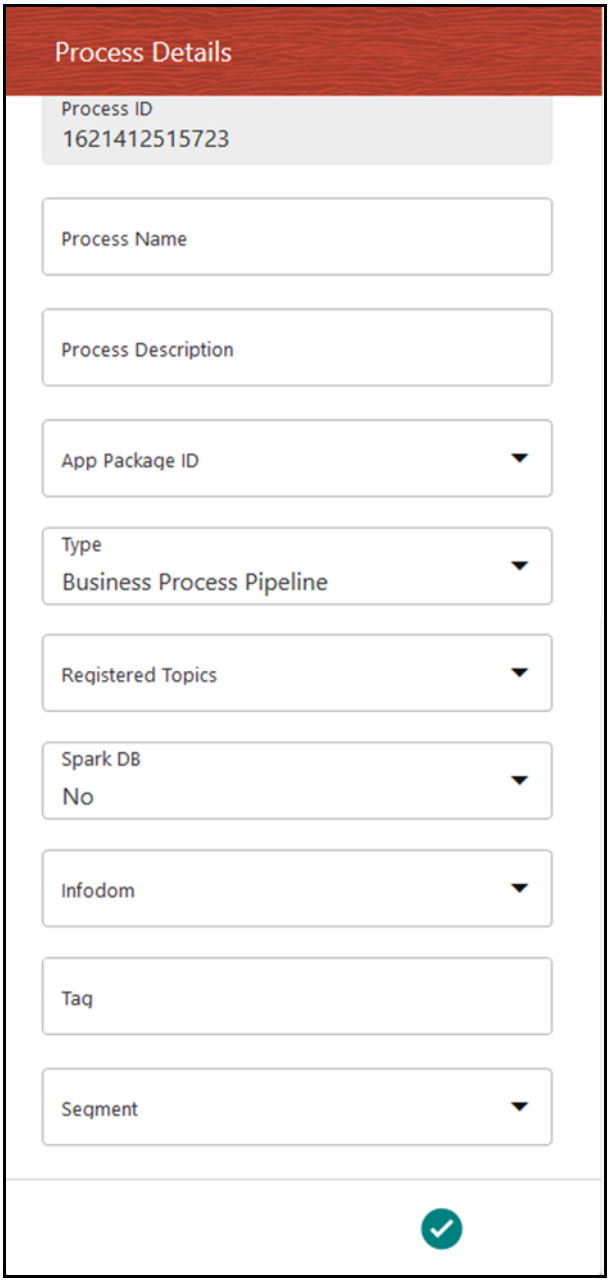
Figure 55: Create a Run Pipeline



To create a new Run pipeline for Insurance Liability Contracts Data Load Process in the Process Modeller, follow these steps:

1. From **OFSAA Home**, select **Oracle Insurance Data Foundation**, click **Administration**. In the **Information Domain** list, select the information domain required for OIDF. Then click the **Process Modelling Framework** tile. A submenu is displayed. Click **Process Modeller** to access the **Process Modeller** page.
2. In the **Process Modeller** page, click .
3. The **Process Details** page is displayed.

Figure 56: Process Details page



Enter or select the required values for each field.

Table 23: Process Details page field names and description

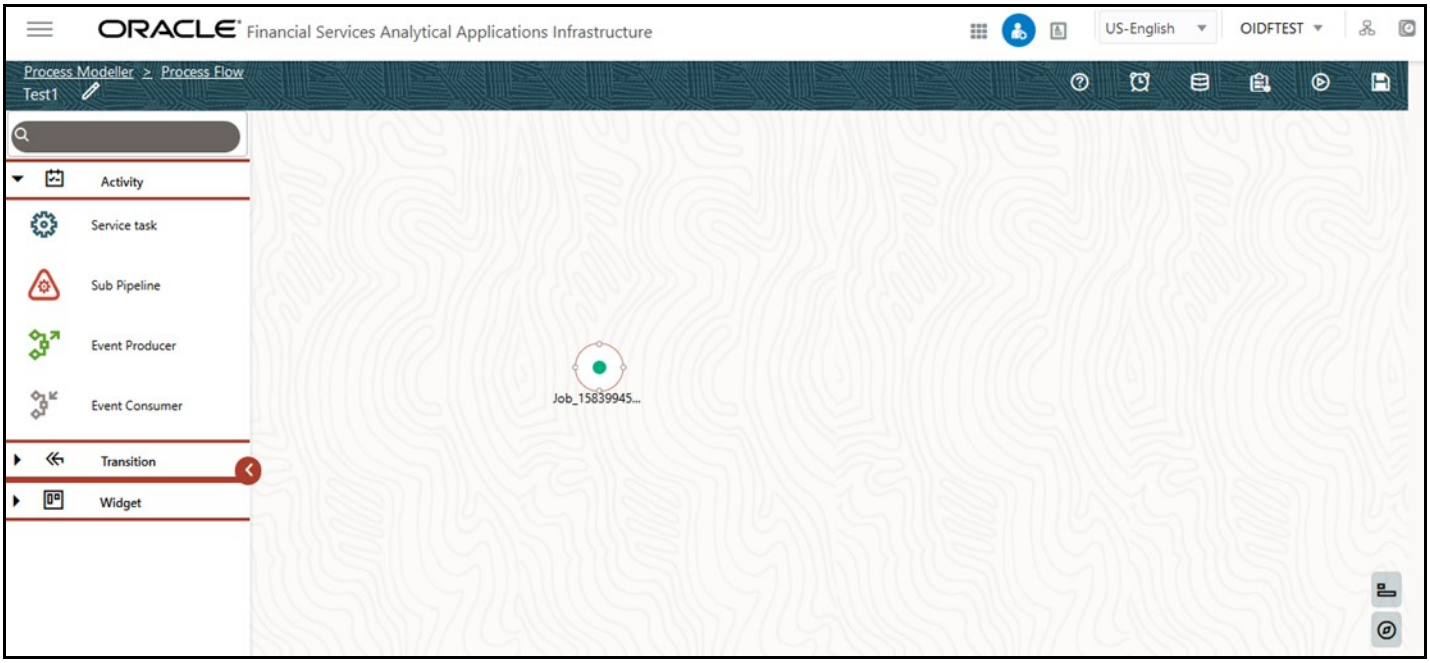
Field Name	Description and instruction
Process ID	This value is automatically generated.
Process Name	Enter a unique and easily recognizable value
Process Description	Enter a unique and easily recognizable value
App Package ID	Select the required application package in which the process must be created from the drop-down list.
Type	Select the required process type from the **Type** drop-down list. The options are as follows: • Business Process Pipeline • Sub Run Process • Run Pipeline • Workflow Pipeline • Attribution Run
Registered Topics	**NOTE**: This functionality is currently not applicable to OIDF.
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 keyword or helpful in defining a relationship between two or more Pipelines.
Segment	Select the application segment. The options are as follows: • **OIDFSEG**: OIDF segment

4. To save the details, click  . The **Process Flow** canvas is displayed.

12.4.4.2 Create a Process Flow for a Run Pipeline

After you click  in the **Process Details** page, a Process Flow canvas page is displayed.

Figure 57: Process Flow Canvas Page

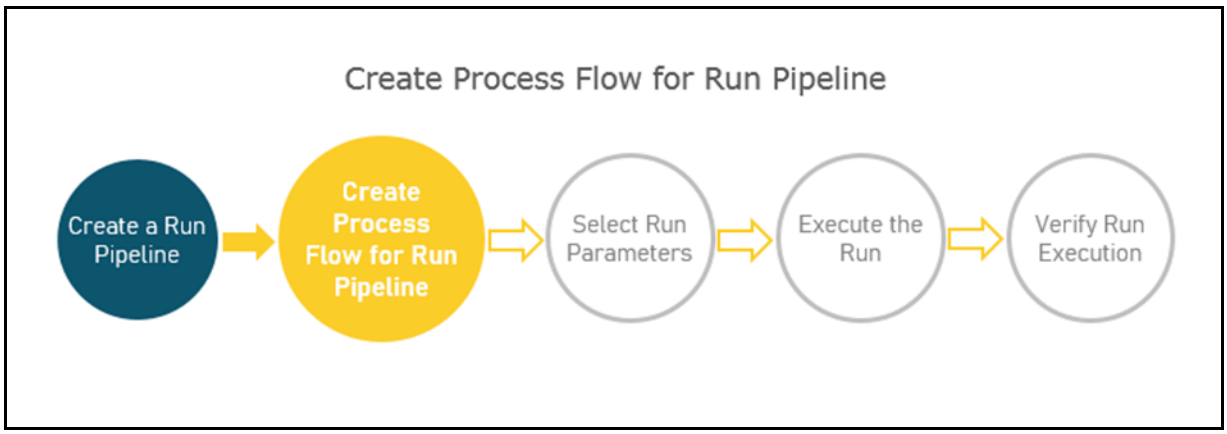


By default, START  from the toolbar appears. This Start activity indicates the beginning of the process.

The **Process Flow** tab contains a floating toolbar and a drawing canvas. Use the drawing canvas to design the process flow with the **Activity**, **Transition**, and **Widget** components available in the floating toolbar.

To design a process flow diagram for any Run Pipeline, see the *Run Pipeline* section, and to use several 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.

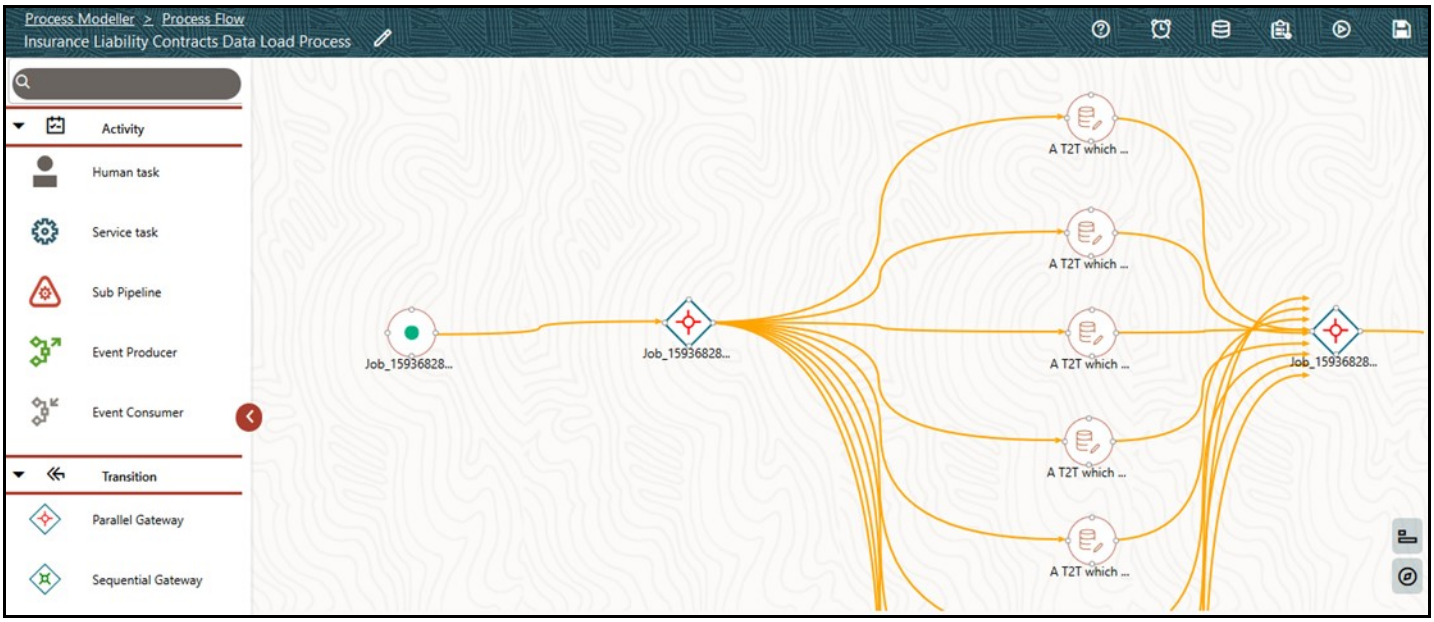
Figure 58: Create Process Flow for Run Pipeline



To create a process flow diagram for an Insurance Liability Contracts Data Load Process (a Run Pipeline for the Fact Common Policy Summary T2T execution) follow these steps:

1. Use the drawing canvas with the LoadT2T widget as the base. In the illustration, each LoadT2T widget represents each of the Insurance Liability Contracts Data Load Process for the Fact Common Policy Summary T2T. The LoadT2T widgets are executed in parallel.

Figure 59: Insurance Liability Contracts Data Load Process Flow Design



2. Double-click a Load T2T widget to configure the details related to its **Activity**, **Transition**, and **Notification**.

Figure **60**: Insurance Liability Contracts Data Load Process with Load T2T widget – Add Activity Details

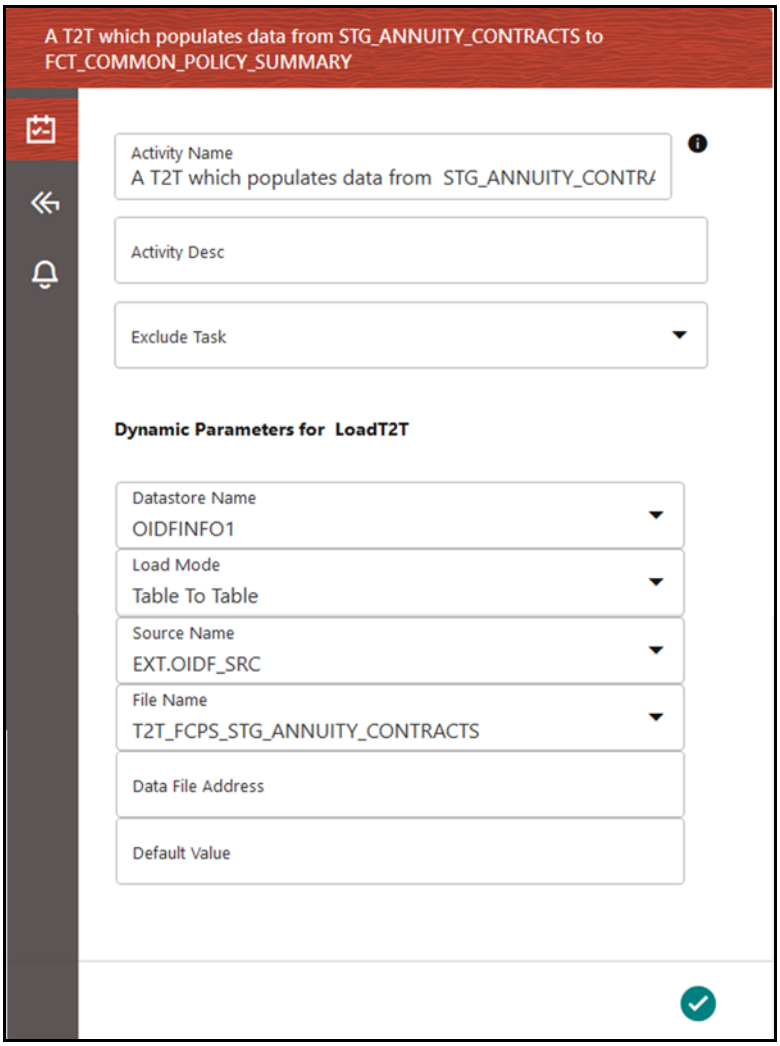
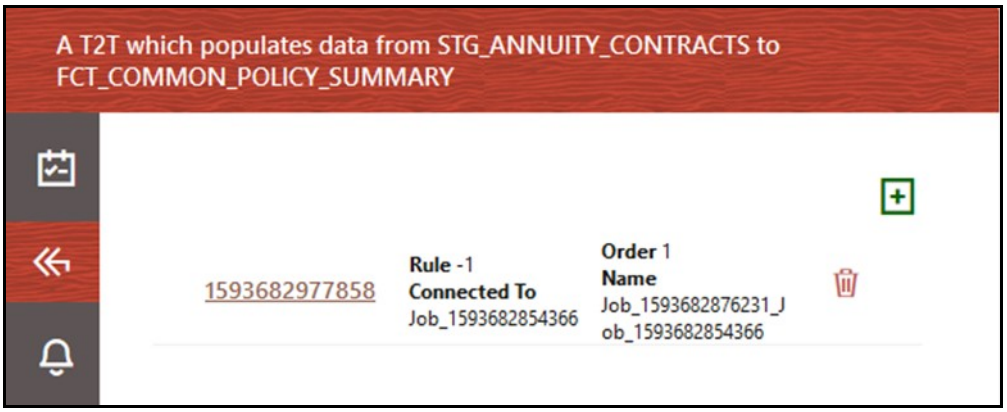


Figure 61: Insurance Liability Contracts Data Load Process with Load T2T widget – Add Transitions Details



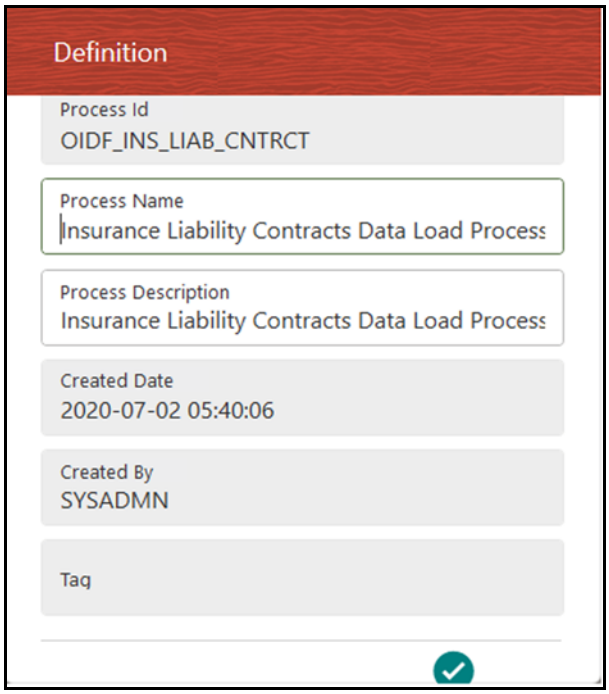
To save the details, click.

3. On the drawing canvas, select the **Definition**, **Data Fields**, **Application Rule** icons to see the respective details.

NOTE To create an Application Rule, see the *Application Rules* section, and to create a Data Field, see the *Data Fields* section in the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

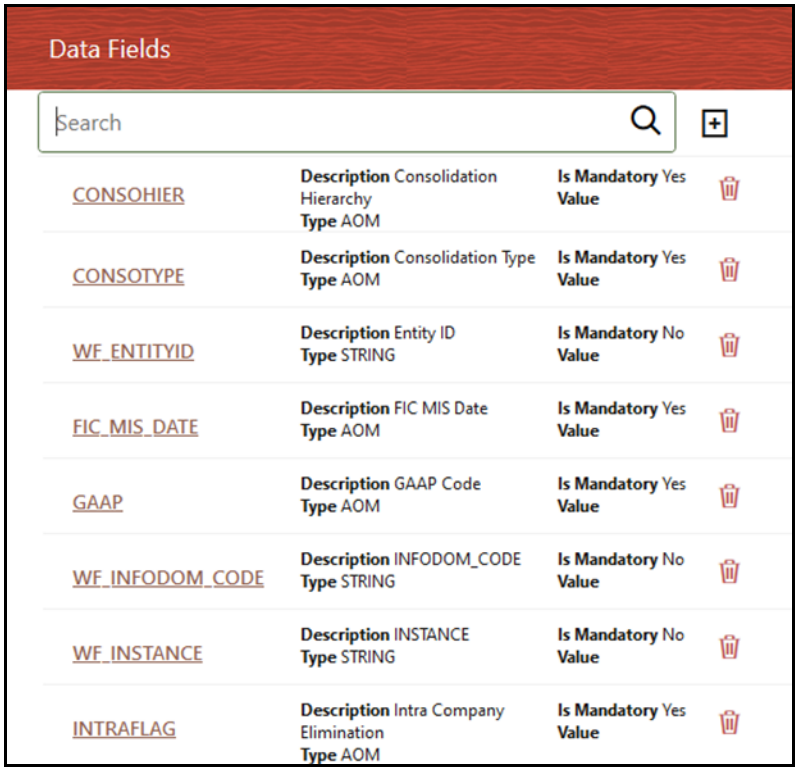
- Click to open the Definition details. To save the details, click.

Figure 62: Insurance Liability Contracts Data Load Process with Definition Details



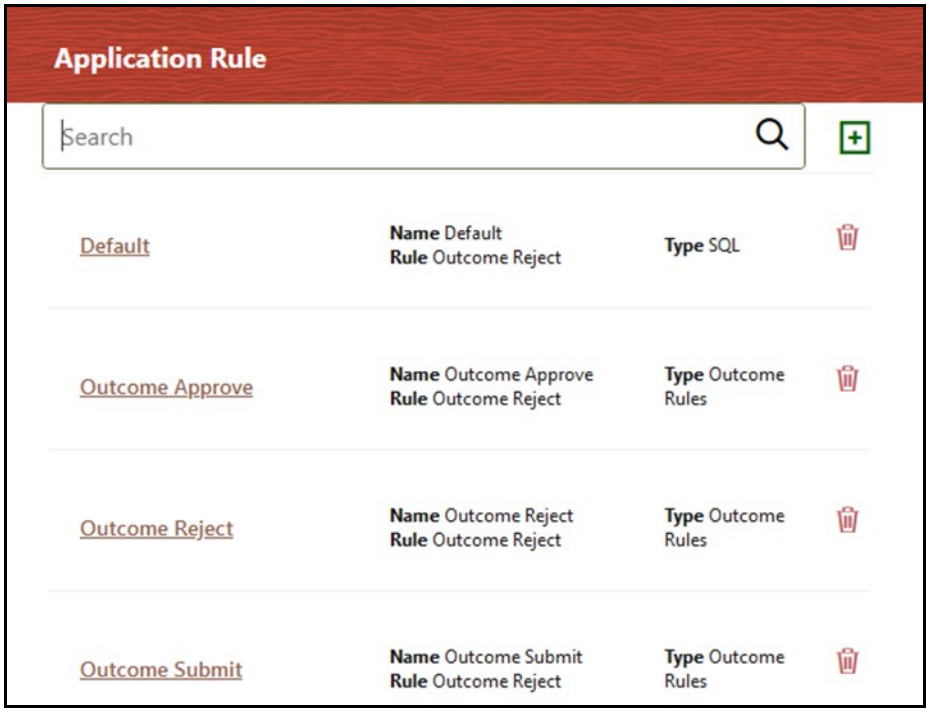
- Click  to open the Data Fields details. Add, modify, delete, or view the Data Fields details.

Figure 63: Insurance Liability Contracts Data Load Process with Data Fields Details



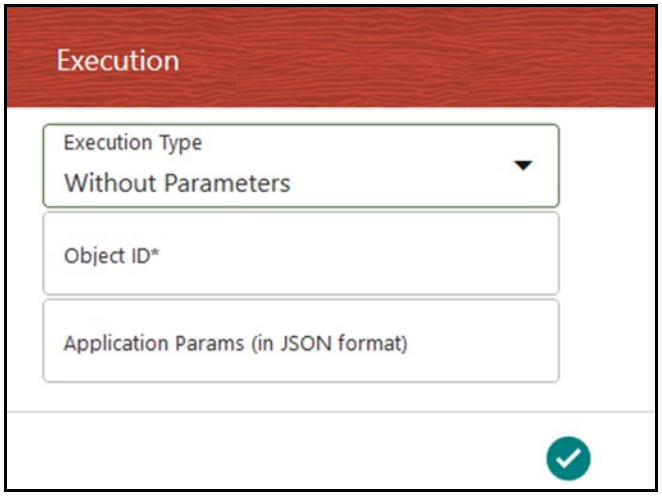
- Click  to open the Application Rule details. You can add or modify or delete the Rule details.

Figure 64: Insurance Liability Contracts Data Load Process with Application Rule Details



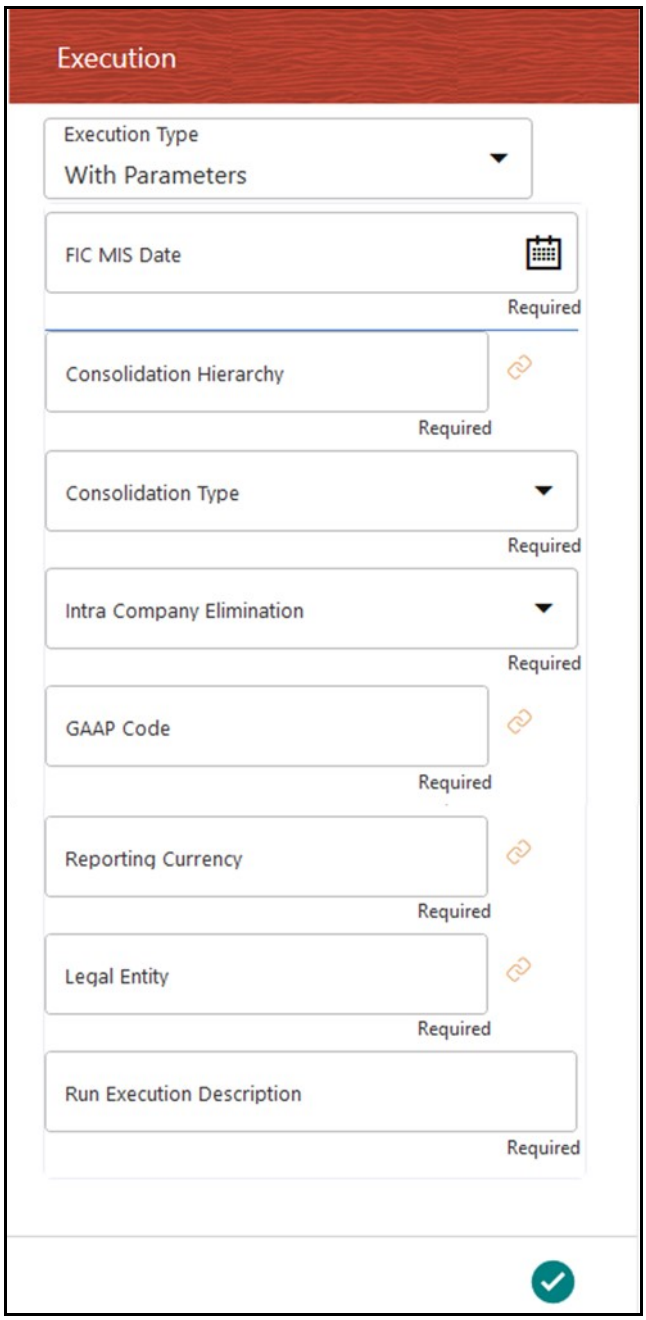
- Click  to open the Execution details. To save the details, click  .

Figure 65: Insurance Liability Contracts Data Load Process without Execution Parameters



Enter and select the required details to execute the Run without parameters. To execute with parameters, select **With Parameters** in the **Execution Type** list. To save the details, click  .

Figure 66: Insurance Liability Contracts Data Load Process with Execution Parameters



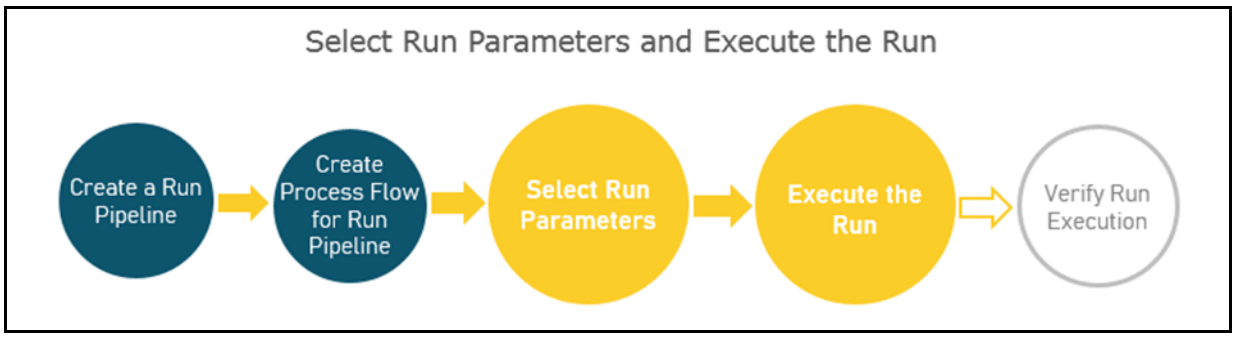
4. In the **Process Flow** canvas, click to save the process flow.

To execute the created Run pipeline (process), see the Select the Run Parameters and Execute the Run section.

12.4.4.3 Select the Run Parameters and Execute the Run

After a Run pipeline (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 on the UI or using a command-line utility called `wfExecExternal.sh`.

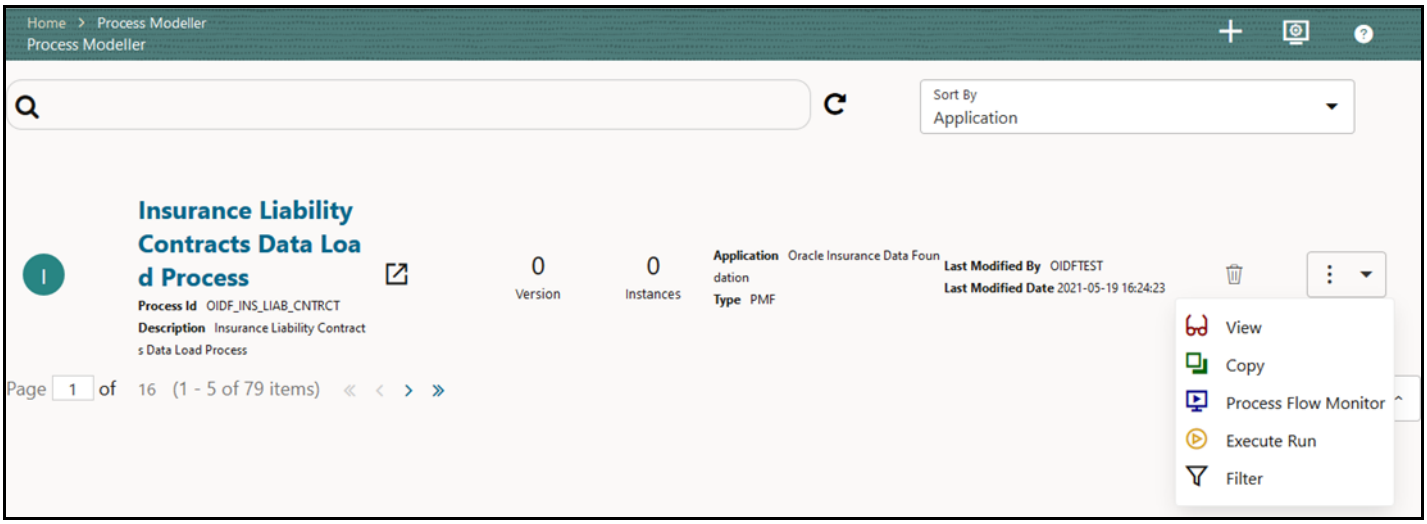
Figure 67: Select Run Parameters and Execute the Run



To select the Run parameters and execute the Run, follow these steps:

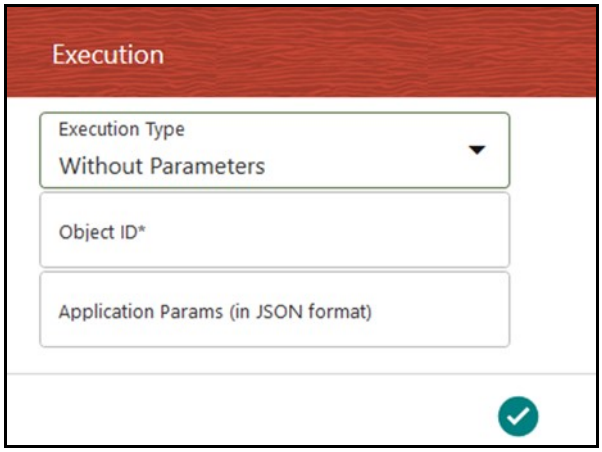
1. In the **Process Modeller** page, click ⋮ corresponding to the Insurance Liability Contracts Data Load Process that must be executed. Click **Execute Run**.

Figure 68: Insurance Liability Contracts Data Load Process - Select Execute Run



2. The **Execution** page is displayed. This page consists of the Run parameters specific to the Insurance Liability Contracts Data Load Process.

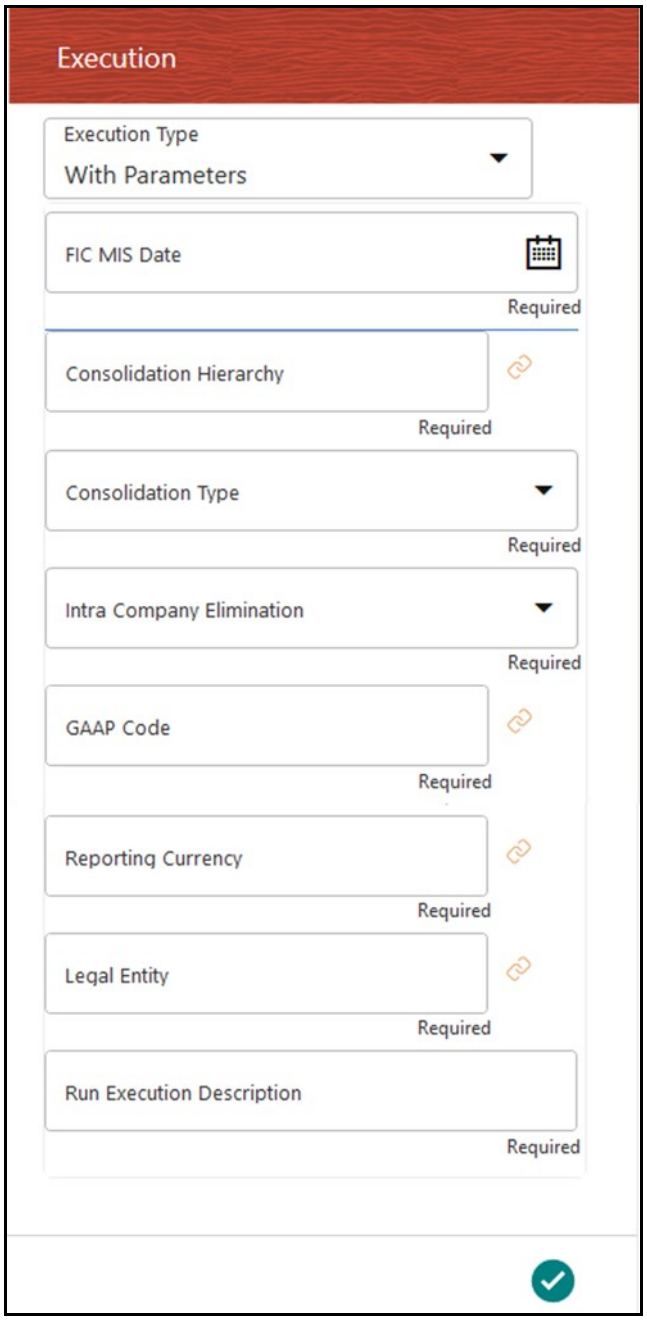
Figure 69: Insurance Liability Contracts Data Load Process - Execution Page Without Parameters



Enter and select the required details to execute the Run without parameters.

To execute with parameters, select **With Parameters** in the **Execution Type** list.

Figure 70: Insurance Liability Contracts Data Load Process - Execution Page With Parameters



Select or enter the required values for each field as follows.

Table 24: Insurance Liability Contracts Data Load Process - Execution Page With Parameters and description

Field Name	Description or Instruction

Field Name	Description or Instruction
Reporting Currency	Use the icon to select the Reporting Currency Code used to calculate the amount during the data population in the target table.
Legal Entity	Use the icon to select the Legal Entity Code to identify the legal entity used for the 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.
Consolidation Hierarchy	Use the icon to select the Legal Entity Hierarchy used for the consolidated run. This parameter is not required for the Solo Run.
GAAP Code	Use the icon to select the required accounting standard.
FIC MIS Date	Use the calendar icon to select the extraction date.
Run Execution Description	Enter a longer description of the Run.

3. To save the details and execute the Run, click .

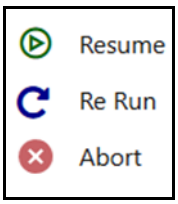
> **NOTE** The execution of the Insurance Liability Contracts Data Load Process is triggered using the selected FIC MIS DATE. The Run SKey is generated and inserted into the DIM_RUN table. For the Run SKey generated, the corresponding user-selected Run parameters are inserted into the RUN_EXE_PARAMETERS table.

4. To verify the Run execution status of the Insurance Liability Contracts Data Load Process, see the Verify the Run Execution section.
5. To abort, resume, or rerun the process, see the section Abort, Resume, or Rerun the Process.

12.4.4.4 Abort, Resume, or Rerun the 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 71: Process Monitor with Re Run, Abort, and Resume options



12.4.4.4.1 Abort the Run Process Execution

The Abort feature helps you to abort a Run Pipeline (process), which is in the process of execution. To abort a Run process execution, follow these steps:

1. In the **Process Monitor** page, click  corresponding to the Run process that must be aborted, and then select **Abort**.
2. When the abort is complete, the *Abort Successful* confirmation message is displayed.

12.4.4.4.2 Resume the Run Process Execution

You can resume a Run Pipeline (process), which is not executed successfully, explicitly interrupted, canceled, or put on hold during the execution process. By resuming a Run process execution, you can continue its execution directly from the point of interruption or failure and complete executing the remaining tasks.

To resume a Run process execution, follow these steps:

1. In the **Process Monitor** page, click  corresponding to the Run process that must be resumed, and then select **Resume**.
2. When the resume is complete, the *Resume Execution is Successful* confirmation message is displayed.

12.4.4.4.3 Rerun the Run Process Execution

You can rerun a Run Pipeline (process), which was previously executed irrespective of its previous execution state.

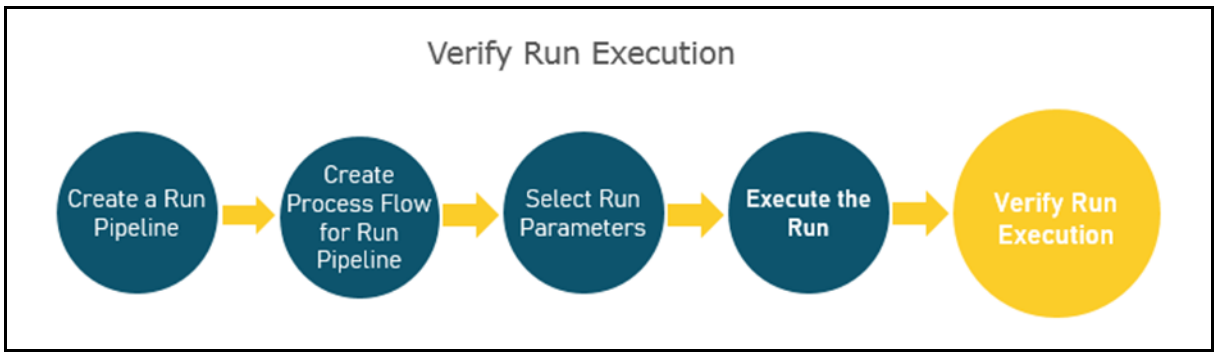
To rerun a Run process execution, follow these steps:

1. In the **Process Monitor** page, click  corresponding to the Run process that must be rerun, and then click **Re Run**.
2. When the re-run is complete, the *Re Run Execution is Successful* confirmation message is displayed.

12.4.4.5 Verify the Run Execution

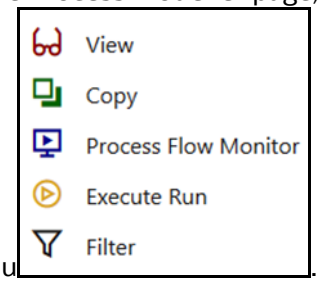
After the Run pipeline (process) execution, you can check the status of the execution.

Figure 72: Verify Run Execution



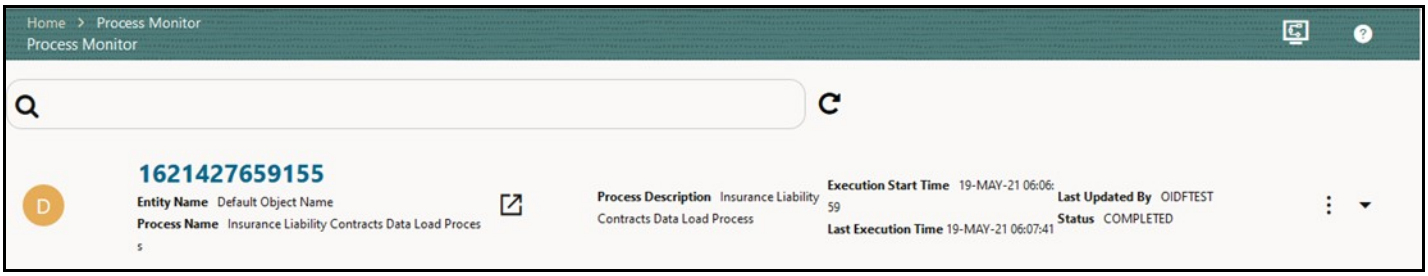
To verify the Run execution, follow these steps:

1. To open the **Process Monitor** page, in the **Process Modeller** page, click or select **Process Flow Monitor** from the Process Modeller menu.



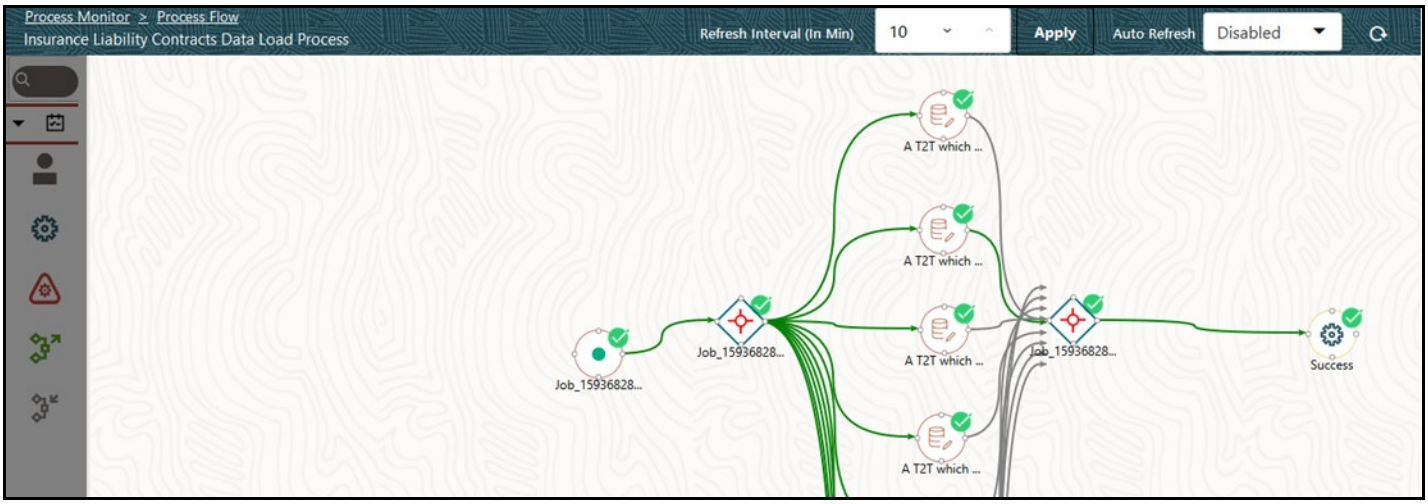
2. In the **Process Monitor** page, for the required Run pipeline that was executed, search the Run instance by Job ID or Process Name, and select the process instance.

Figure 73: Insurance Liability Contracts Data Load Process – Process Monitor Page – Select Job ID



3. A status page opens, which displays the execution status of the executed Run. In this illustration, the Insurance Liability Contracts Data Load Process is successfully executed.

Figure 74: Insurance Liability Contracts Data Load Process – Run Execution Status



To use a standard out-of-the-box Run pipeline for OIDF functions, see the Use Standard OIDF Run Pipelines section.

For information about the complete functioning of the PMF, see the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide Release 8.1.2.0.0.

13 Restatement Support Feature in OIDF

This section provides information about the Restatement Support feature (prior dated Run execution) in the Oracle Insurance Data Foundation application explicitly for the Start Date or End Date attributes.

Topics:

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

When a regulatory body asks a 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 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 or 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 or End Date attributes must be used when rerunning the report for a prior date. A backdated Run execution for Restatement Support is similar to a regular Run execution in OIDF.

13.1 Existing Features for Restatement Support

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

13.1.1 Data Versioning

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

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

13.1.1.1 Data Versioning in Seeded Data

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

13.1.1.2 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 table are introduced to store the Mapper data versioning for each MIS date.
- For a 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.

13.1.1.3 Incremental Data Load

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

13.1.2 Data Flow

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

- The right version of dimensional attributes is considered in the data flow based on the Start Date or End Date attributes.
- In the Joins with Dimensions, the T2Ts are enhanced to utilize Start Date or 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.

13.2 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.
- For the Dimensions with no available data flow, you must use the custom methods 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 Latest Record Indicator (LRI) and 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.

13.3 Use Case for Restatement in OIDF

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

Consider that an insurance company has missed reporting some accounts or transactions when reporting to the regulator. Then the regulator has asked the insurance company to resubmit the report with complete details. To address this requirement, the insurance company 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 the right version of the Surrogate keys that is applicable for the prior MIS date. After the data is corrected, the insurance company can re-run the report and resubmit it.

13.4 Assumption for the Implementation of Restatement

The assumption for the implementation of the Restatement Support is as follows:

- The feature is applicable only on the MIS dates versioned after the date on which the Restatement Support feature is enabled.

14 Legal Entity Consolidation

The reporting entity may be a part of an Insurance Company that consists of multiple legal entities such as parent or child entities (subsidiaries) under its name. Users can select the entity for which processing must be done. In addition, whether a 'Solo' or 'Consolidation' execution must be done using the Run Execution screen. However, if it is executed using RRF execution then these options must be set up 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 the staging data population. Run Management screen selects this process by default.

ATTENTION All the following sections are applicable also to Stage and Results on Hive.

Topics:

14.1 Run Parameters Assignment

Run Management Framework in the product allows the reporting Insurance Company to define and execute a Run by selecting a combination of parameters capital computation.

The rule 'Run Definition User Defined Run Param Assignment' is used to assign the run parameters in case of a run executed through the Run Rule Framework. However, if the execution is through Run Management, the parameters are populated based on the run defined in the run definition screen.

14.2 Consolidation Procedures

The Consolidation procedures are as follows:

- Combine 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 investment of the parent in each subsidiary and the equity portion of the parent of each subsidiary (IFRS 3 Business Combinations explain how to account for any related goodwill).

- Eliminate in full intragroup assets and liabilities, equity, income, expenses, and cash flows that relate 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 in full).

In the preceding list, the first and third points are partly handled in the consolidation process, second is accounting idea output, which is provided in General Ledger as a part of Stage General Ledger data inputs

14.3 Types of Consolidation

The types of Consolidation are as follows:

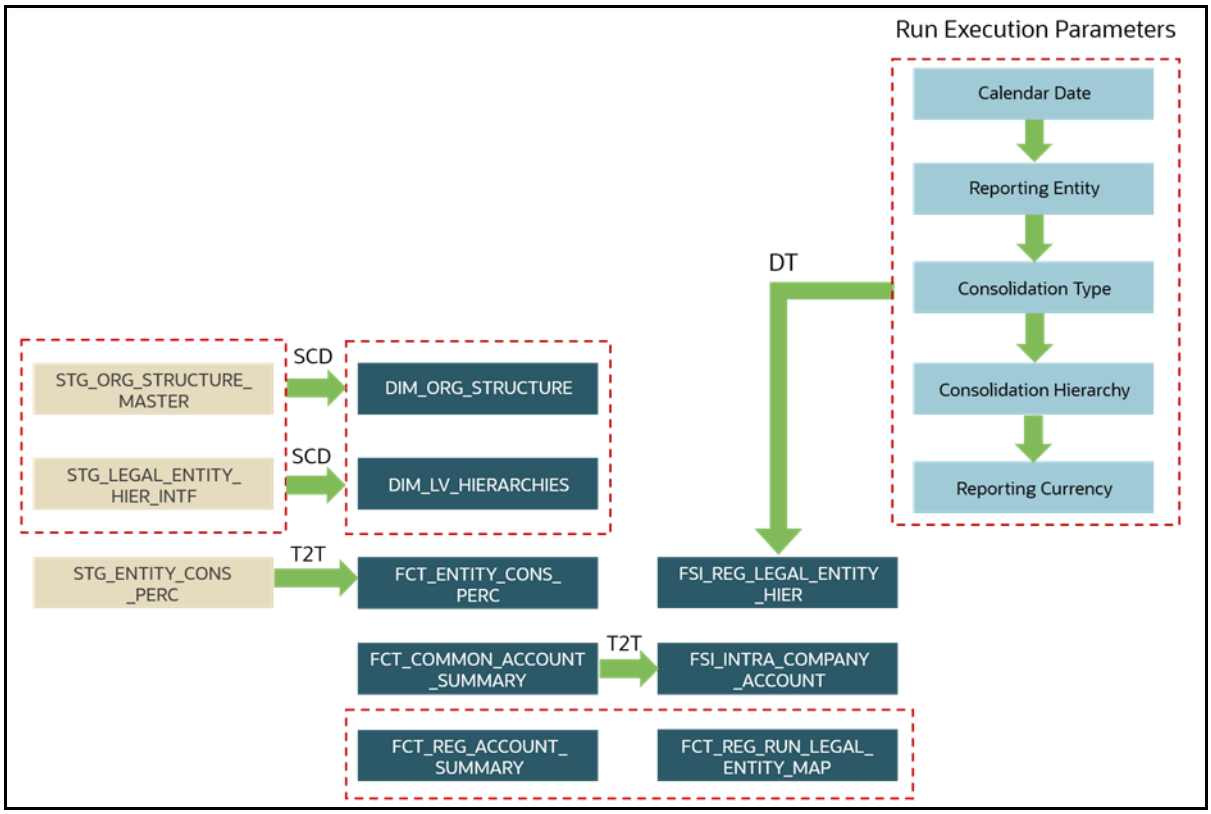
- Simple Aggregation: Aggregate across entities without any elimination.
- Full Consolidation: Aggregate and eliminate intra group transactions.
- Proportionate Consolidation: Aggregate and eliminate intra group transactions and balances reflecting consolidation percentage owned by parents in a subsidiary.

14.4 Consolidation Activities

Scope of Consolidation is the list of Entities that participate in consolidation. Legal Entity Structure is looked through the Organization Structure Dimension. This stores the parent-child relationship and is stored only once. When moving the data, Legal Entity can move related entities to the processing or reporting area. The legal structure is finalized once, and this structure only stores one parent-child relationship.

This is the Data Flow diagram of Legal Entity consolidation activities.

Figure 75: Legal Entity Consolidation Data Flow



14.5 Legal Entity Tables

For the metadata design, see the *Legal Entity Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

14.6 FSI Intracompany Policy Table

This chapter details the FSI Intracompany Policy table in the Oracle Insurance Data Foundation application.

Topics:

14.6.1 About Intracompany Policy Table

The Intracompany Policy table is the Policy table that records the transactions between the legal entities of a company. In the Intracompany Policy table, to record the transactions between the companies belonging to the same group structure, the intercompany policy transactions (the policy transactions between the companies belonging to the different group structures) must be excluded.

14.6.2 Criteria to Qualify as an Intracompany Policy Transaction

To exclude an intercompany policy transaction from the Intracompany Policy table, the F_INTRAGROUP_EXP_IND flag is used in the Fact Common Policy Summary table. If the F_INTRAGROUP_EXP_IND flag is set to Y, it indicates that the two companies belong to the same organizational structure and the corresponding policy transactions are then included in the Intracompany Policy Entity. The remaining transactions are not considered as intracompany transactions.

Now, the policy transaction must qualify these two criteria to be included as an Intracompany Policy transaction in the Intracompany Policy table:

- Both companies (the Policy issuing Entity and Party) must belong to the same organization group structure.
- The Party must be a Beneficiary of the Policy.

When the two criteria are met by a policy transaction, only then in the Fact Common Policy Summary table, the F_INTRAGROUP_EXP_IND flag is set to Y; and the policy transaction is added as a record in the FSI_INTRA_COMPANY_POLICY table.

14.6.3 Analyze Different Policy Transaction Scenarios to Qualify as an Intracompany Policy Transaction

There are different scenarios to analyze a policy transaction between two companies. This analysis decides whether the policy transaction qualifies as an Intracompany Policy transaction.

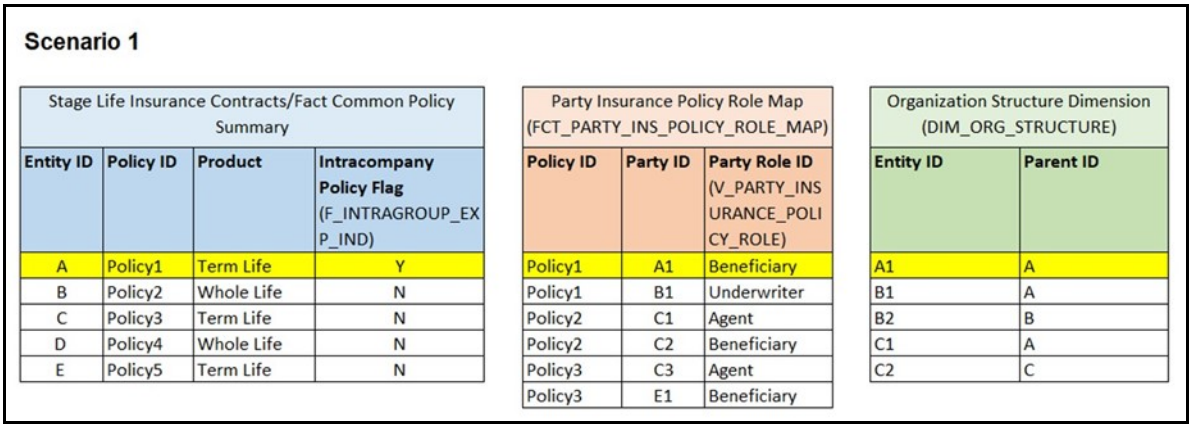
The policy transaction scenarios are explained using these tables:

- Life Insurance Contracts table:
  - Stage Life Insurance Contracts (STG_LIFE_INS_CONTRACTS): This table consists of the Policy issuing Entity and the issued Policy ID.
  - Fact Common Policy Summary (FCT_COMMON_POLICY_SUMMARY): This table consists of the Policy issuing Entity, issued Policy ID, and the Intracompany Policy Flag.
- Party Insurance Policy Role Map (FCT_PARTY_INS_POLICY_ROLE_MAP): This table consists of the list of Parties mapped to the issued Policy as a Party Role (Beneficiary, Agent, Underwriter, and so on).
- Organization Structure Dimension (DIM_ORG_STRUCTURE): This table is used to determine whether the Policy issuing Entity and its Party have the same Parent or the same organization group structure. When the Policy issuing Entity and its Party are a part of the same organization group structure, it is called as Intracompany.

The policy transaction scenarios are as follows:

- Scenario 1: When the Policy issuing Entity and its Party belongs to the same organization group structure, and the Party Role is a Beneficiary, then the Intracompany Policy flag is set to Y.

Figure 76: Policy Transaction scenario 1



In illustration Scenario 1, observe the following:

a. In the Stage Life Insurance Contracts table, refer to Policy1. A is the Policy issuing Entity, which owns Policy1.

b. Verify whether the Party associated with the Policy1 is a Beneficiary or not. Refer to the Party Insurance Policy Role Map table. The V_PARTY_INSURANCE_POLICY_ROLE column lists the Party Roles associated with Policy1. Two Party Roles are associated with Policy1, namely, A1 and B1. A1 is a Beneficiary, however, B1 is an Underwriter as depicted in the V_PARTY_INSURANCE_POLICY_ROLE column.

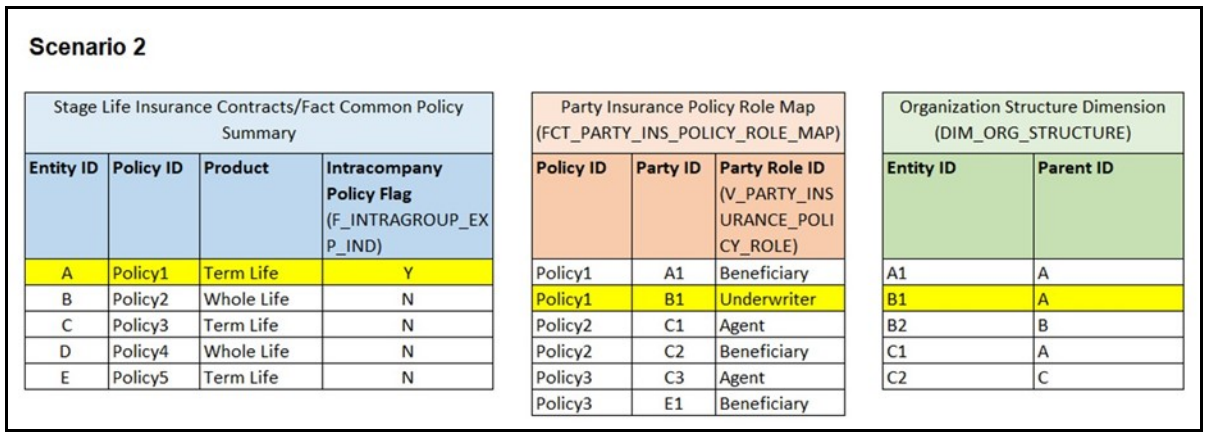
c. Refer to the DIM_ORG_STRUCTURE table to see if the Beneficiary and the Policy issuing Entity are a part of the same organization group structure. As highlighted in the illustration, A1 associated with the Beneficiary and the Entity A is a part of the same organization group structure. Therefore, the Intracompany group structure flag is set to Y.

d. Therefore, in the Fact Common Policy Summary table, the Intracompany Policy Flag (F_INTRAGROUP_EXP_IND) column value is set to Y.

Therefore, this Policy contract between these two companies is included in the Intracompany Policy table.

- Scenario 2: When the Policy issuing Entity and its Party belongs to the same organization group structure, and the Party Role is not a Beneficiary, then the Intracompany Policy flag is set to N.

Figure 77: Policy Transaction scenario 2



Stage Life Insurance Contracts/Fact Common Policy Summary

Entity ID	Policy ID	Product	Intracompany Policy Flag (F_INTRAGROUP_EXP_IND)
A	Policy1	Term Life	Y
B	Policy2	Whole Life	N
C	Policy3	Term Life	N
D	Policy4	Whole Life	N
E	Policy5	Term Life	N

Party Insurance Policy Role Map (FCT_PARTY_INS_POLICY_ROLE_MAP)

Policy ID	Party ID	Party Role ID (V_PARTY_INSURANCE_POLICY_ROLE)
Policy1	A1	Beneficiary
Policy1	B1	Underwriter
Policy2	C1	Agent
Policy2	C2	Beneficiary
Policy3	C3	Agent
Policy3	E1	Beneficiary

Organization Structure Dimension (DIM_ORG_STRUCTURE)

Entity ID	Parent ID
A1	A
B1	A
B2	B
C1	A
C2	C

In illustration Scenario 2, observe the following:

a. In the Stage Life Insurance Contracts table, refer to Policy1. A is the Policy issuing Entity, which owns Policy1.

b. Verify whether the Party associated with the Policy1 is a Beneficiary or not. Refer to the Party Insurance Policy Role Map table. The V_PARTY_INSURANCE_POLICY_ROLE column lists the Party Roles associated with Policy1. Two Party Roles are associated with Policy1, namely, A1 and B1. A1 is a Beneficiary, however, B1 is an Underwriter as depicted in the V_PARTY_INSURANCE_POLICY_ROLE column.

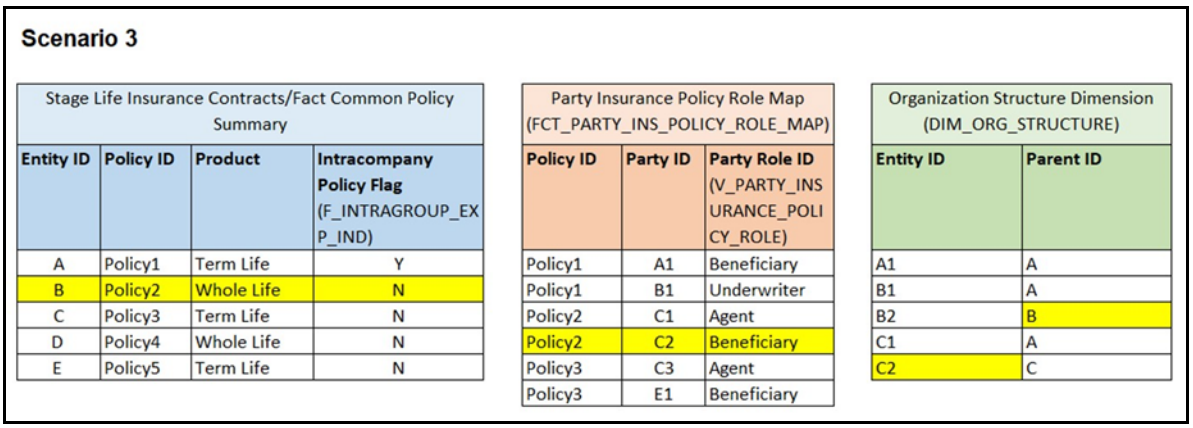
c. Refer to the DIM_ORG_STRUCTURE table. The Party B1 and Entity A are a part of the same organization group structure. Therefore, the Intracompany group structure flag is set to Y.

d. However, the Party Role is an Underwriter. As a result, in the Fact Common Policy Summary table, the Intracompany Policy Flag (F_INTRAGROUP_EXP_IND) column value is set to N.

Therefore, the Policy transactions between these two companies are excluded from the Intracompany Policy table.

- Scenario 3: When the Party Role is a Beneficiary, and the Policy issuing Entity and its Party belongs to different organization group structures, then the Intracompany Policy flag is set to N.

Figure 78: Policy Transaction scenario 3



Scenario 3

Stage Life Insurance Contracts/Fact Common Policy Summary			
Entity ID	**Policy ID**	**Product**	**Intracompany Policy Flag** (F_INTRAGROUP_EXP_IND)
A	Policy1	Term Life	Y
B	Policy2	Whole Life	N
C	Policy3	Term Life	N
D	Policy4	Whole Life	N
E	Policy5	Term Life	N

Party Insurance Policy Role Map (FCT_PARTY_INS_POLICY_ROLE_MAP)		
Policy ID	**Party ID**	**Party Role ID** (V_PARTY_INSURANCE_POLICY_ROLE)
Policy1	A1	Beneficiary
Policy1	B1	Underwriter
Policy2	C1	Agent
Policy2	C2	Beneficiary
Policy3	C3	Agent
Policy3	E1	Beneficiary

Organization Structure Dimension (DIM_ORG_STRUCTURE)	
Entity ID	**Parent ID**
A1	A
B1	A
B2	B
C1	A
C2	C

In illustration Scenario 3, observe the following:

a. In the Stage Life Insurance Contracts table, refer to Policy2. B is the Policy issuing Entity, which owns Policy2.

b. Verify whether the Party associated with the Policy2 is a Beneficiary or not. Refer to the Party Insurance Policy Role Map table. The V_PARTY_INSURANCE_POLICY_ROLE column lists the Party Roles associated with Policy2. Two Party Roles are associated with Policy2, namely, C1 and C2. C1 is an Agent and C2 is a Beneficiary as depicted in the V_PARTY_INSURANCE_POLICY_ROLE column.

c. Refer to the DIM_ORG_STRUCTURE table to see if the Beneficiary and the Policy issuing Entity are a part of the same organization group structure. As highlighted in the illustration, C2 associated with the Beneficiary and Entity B are a part of the different organization group structures. Therefore, the Intracompany group structure flag is set to N.

d. Therefore, in the Fact Common Policy Summary table, the Intracompany Policy Flag (F_INTRAGROUP_EXP_IND) column value is set to N.

The Policy transactions between these two companies are not intracompany and therefore, not added to the Intracompany Policy table.

14.6.4 Run-enabled and Non-Run-enabled Tables

These are the descriptions for the Run-enabled and Non-Run-enabled tables:

- Run-enabled tables: The Oracle Insurance Data Foundation Execution Run can be executed any number of times per day with each unique Run SKey for data movement in the Run-enabled tables.
- Non-Run-enabled tables: The Oracle Insurance Data Foundation Source Run can be executed once per day for Data Movement from Staging Area to Results Area for Non-Run SKey tables.

NOTE The records that enter the FSI_INTRA_COMPANY_POLICY table must not be entered into any other Run-enabled tables. The process mentioned for the Intracompany identification and consolidation will not be as expected unless processing or reporting application refers and reflect the same as required in the application-specific metadata. The same principle also applies to customized metadata.

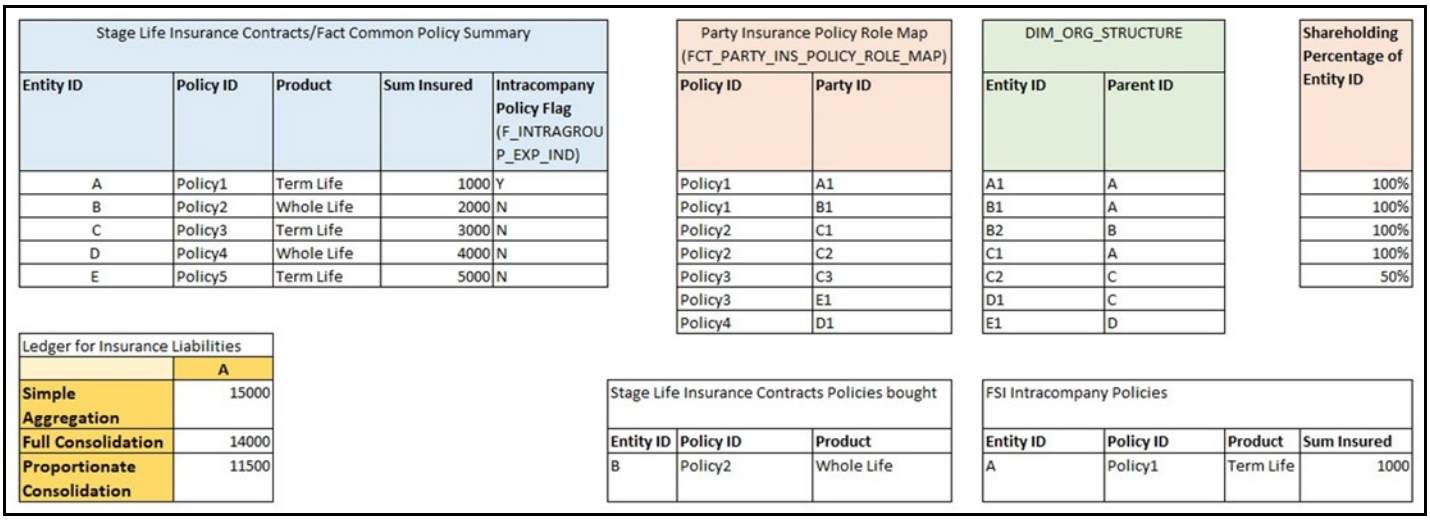
14.6.5 Consolidation Procedures in the Intracompany Policy Table

The supported Consolidation procedures applicable to the Intracompany Policy tables are as follows:

- Simple Aggregation
- Full Consolidation
- Proportionate Consolidation

This is a use case representation for the Consolidation procedure types:

Figure 79: Use case for the Consolidation procedure types



Stage Life Insurance Contracts/Fact Common Policy Summary

Entity ID	Policy ID	Product	Sum Insured	Intracompany Policy Flag (F_INTRAGROUP_EXP_IND)
A	Policy1	Term Life	1000	Y
B	Policy2	Whole Life	2000	N
C	Policy3	Term Life	3000	N
D	Policy4	Whole Life	4000	N
E	Policy5	Term Life	5000	N

Party Insurance Policy Role Map (FCT_PARTY_INS_POLICY_ROLE_MAP)

Policy ID	Party ID
Policy1	A1
Policy1	B1
Policy2	C1
Policy2	C2
Policy3	C3
Policy3	E1
Policy4	D1

DIM_ORG_STRUCTURE

Entity ID	Parent ID
A1	A
B1	A
B2	B
C1	A
C2	C
D1	C
E1	D

Shareholding Percentage of Entity ID
100%
100%
100%
100%
50%

Ledger for Insurance Liabilities

	A
Simple Aggregation	15000
Full Consolidation	14000
Proportionate Consolidation	11500

Stage Life Insurance Contracts Policies bought

Entity ID	Policy ID	Product
B	Policy2	Whole Life

FSI Intracompany Policies

Entity ID	Policy ID	Product	Sum Insured
A	Policy1	Term Life	1000

The pictorial representation is explained in the following sections.

14.6.5.1 Simple Aggregation

In a Simple Aggregation consolidation procedure, all the account transactions in an entity are aggregated. The entity can be a Parent or Child.

The following logic is used for a Simple Aggregation consolidation procedure:

1. Identify the list of entities to be added based on the organization structure.

   In the illustration, see the DIM_ORG_STRUCTURE table. For the consolidation of Entity A, the child entities are A, B, C, D, and E.

2. Aggregate both the Accounts and the General Ledger transactions. The aggregation of the General Ledger transactions is the addition of General Ledgers of the entities in the list.

   In the illustration, see the Stage Life Insurance Contracts or Fact Common Policy Summary table and the Sum Insured column. Add the Sum Insured amount of the entities A, B, C, D, and E. The total amount is 15000 for the Simple Aggregation in Entity A.

3. This process does not identify intragroup transactions. As a result, intragroup transactions are also included.

14.6.5.2 Full Consolidation

In a Full Consolidation procedure, all the account transactions in an entity are aggregated and the Intergroup transactions are eliminated. This is repeated for each entity involved and the results are added.

The following logic is used for a Full Consolidation procedure:

1. Identify the list of entities to be added based on the organization structure.

   In the illustration, see the DIM_ORG_STRUCTURE table. For the consolidation of Entity A, the child entities are A, B, C, D, and E.

2. Aggregate both the Accounts and the General Ledger transactions.

   In the illustration, see the Stage Life Insurance Contracts or Fact Common Policy Summary table and the Sum Insured column. Add the Sum Insured amount of the entities A, B, C, D, and E. The total amount is 15000.

3. Exclude all the Intragroup transactions.

   In the illustration, see the FSI Intracompany Policies table. The transaction between the entities A and A1 is considered as an intragroup transaction. Therefore, exclude the Sum Insured of Entity A (Sum Insured=1000) from the total amount. The new total amount is 14000 for Full Consolidation in Entity A.

4. Repeat steps 2 and 3 for each entity.

14.6.5.3 Proportionate Consolidation

In a Proportionate Consolidation procedure, the account transactions for each Entity (performing Simple Aggregation) are added. To this, the share of profits and expenses in the Entity, where the entity holds the stake are added. Then the Intergroup transactions are eliminated.

The following logic is used for a Proportionate Consolidation procedure:

1. Identify the list of entities to be added based on the organization structure.

   In the illustration, see the DIM_ORG_STRUCTURE table. For the consolidation of Entity A, the child entities are A, B, C, D, and E.

2. When one entity holds a stake in another entity, follow these steps:
   - When the Balance Sheet is prepared for the Parent entity, follow these steps:
     i. Multiply each of the Account and the General Ledger transaction with the percentage that the Parent entity holds as a stake in the Child entity.

        In the illustration, see the Shareholding Percentage of Entity ID table. Entity A holds a 100% stake in A, B, C, and D, and a 50% stake in E. Therefore, the resultant amounts are 1000, 2000, 3000, 4000, and 2500 respectively.

     ii. Aggregate the Accounts and the General Ledger transaction results from the previous step.

        The summation of the amount from the previous step is 12500.
   - When the Balance Sheet is prepared for the Child entity, follow these steps:
     i. Multiply each of the Account and the General Ledger transaction with the percentage that the Child entity holds as a stake in its own company.

     ii. Aggregate the Accounts and the General Ledger transaction results from the previous step.

3. Exclude all the intragroup transactions.

In the illustration, see the FSI Intracompany Policies table. The transaction between the entities A and A1 is considered as an intragroup transaction. Therefore, exclude the Sum Insured of Entity A (Sum Insured=1000) from the amount in the previous step. The new total amount is 11500 for the Proportionate Consolidation in Entity A.

For the metadata design, see the *About FSI Intracompany Policy T2T (Result Table)* Section in the OIDF Metadata Design Document Release 8.1.x.

14.7 Populating Legal Entity Tables

This section provides information about populating the Legal Entity tables.

14.7.1 Deploying Legal Entity Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

14.7.2 Populating Legal Entity Dimension Tables

Follow this SCD process to populate data into a Dimension table:

NOTE You can also follow this SCD process to populate data into any Hive-related Dimension table.

1. To populate data into a Dimension table, execute the SCD batch. For a detailed procedure, see the Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Dimension table, follow the procedure Check the Execution Status of the SCD Batch.
3. To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

14.7.3 Populating Legal Entity T2T Result Table

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:

a. Prerequisites for loading T2T.

b. Select the Run Parameters and Execute the Run.

2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.

3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

14.8 Populating FSI Intracompany Tables

This section provides information about populating the FSI Intracompany tables.

14.8.1 Deploying FSI Intracompany Policy Table on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

14.8.2 Populating FSI Intracompany Policy T2T Result Table

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:

   a. Prerequisites for loading T2T.

   b. Select the Run Parameters and Execute the Run.

2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.

3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

15 Exchange Rates Tables

This section provides information about the Exchange Rates tables in the Oracle Insurance Data Foundation application.

Topics:

The Exchange Rates table stores the list of all exchange rates for all types of currency. These are standalone tables.

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

15.1 Handle Alternate Currency

In a downloadable format, the customers supply the Exchange Rates information. 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 is 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 the download, with a common currency in each of the pair, the exchange rate for currencies not common as part of the download pair is derived and populated.

15.2 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 an insurance company has branches in multiple countries, and therefore, multiple exchange rates scenarios exist. Then the insurance company must decide to choose the closing price of the exchange rate. Therefore, the insurance company computes the balance sheet based on the entity in each country. Assume that the insurance company 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, the insurance company captures multiple exchange rates because of different countries. For Currency A to Currency B, the insurance company must capture three pairs of exchange rates. OIDF handles this type of scenario using the Rate Data Source Code, which is based on the data source. The insurance company chooses the time zone.

The insurance company creates different data sources such as country AB, country AC, and country AD. A mapping exists from Legal Entity to each data source. If the Legal Entity is of country B, then the insurance company uses AB as the data source and all the prices of data source AB are used to compute all the transactions. This is the business use case that resulted in the requirement of two Exchange Rates tables in OIDF.

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.

15.3 Overview of the Exchange Rates Table

OIDF has two Exchange Rates Staging tables. They are:

- STG_EXCHANGE_RATE_HIST
- STG_FORWARD_EXCHG_RATES

Spot 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 OIDF 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 is also provided. There are two different data loading categories in OIDF for the Exchange Rates tables. They are:

- The data loading method supported for the Exchange Rates table from the OIDF 8.1.0.0.0 release.
- The data loading method supported the Exchange Rates table for the OIDF 8.0.9.0.0 and earlier versions.

15.3.1 Data Loading Method Supported by the OIDF 8.1.0.0.0 Release

The existing customers can 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.

15.3.2 Data Loading Method Supported for the OIDF 8.0.9.0.0 and Earlier Releases

Only the existing customers can follow the earlier data loading methods for the Exchange Rates tables.

ATTENTION The method of data loading into the STG_EXCHANGE_RATE_HIST table is no more supported in the OIDF 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 can result in 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 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.

15.4 About Exchange Rates T2Ts (Result Tables)

For the metadata design, see the *About Exchange Rates T2Ts (Result Tables)* Section in the OIDF Metadata Design Document Release 8.1.x.

15.5 Deploying Exchange Rates Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

15.6 Populating Exchange Rates T2T Result Table

Execute the T2T process through the Oracle Insurance Data Foundation Execution Run in the Process Modelling Framework.

NOTE When executing the Run, the Run SKey is auto-generated and stamped against each record.

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

FSI_EXCHANGE_RATES table must be loaded prior to loading any of the Account Summary tables.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

16 Party Subject Area

This section provides information about the Party Dimension loading process and populating several Party Attributes Results tables in the Oracle Insurance Data Foundation application.

Topics:

16.1 Overview

Party refers to any person or organization that interacts with the Insurance Company while Party Attributes refers to information elements associated with any person or organization that interacts with the Insurance Company.

Correspondingly, Party tables can be classified into Party Dimension tables and Party T2T tables. The Party Dimension tables store the history of a Party and different types of Party T2Ts store the history of a Party specific to the attributes.

16.2 Party Dimension Table

Party Dimension table stores the history of a Party. A Party here can be a customer, issuer, guarantor, and so on.

NOTE Party Dimension table must be populated before populating any other Party related Dimension table.

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

- DIM_PARTY

- DIM_CUSTOMER

In the current release, data is sourced from STG_PARTY_MASTER to the above Dimension tables.

To address the Bug 20486362 - SCD TO POPULATE DIM_CUSTOMER FROM STG_PARTY_MASTER TO BE ADDED, a new SCD (MAP_REF_NUM = 335) was introduced in OIDF 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 dependency 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.

For the metadata design, see the *Party Dimension Table* Section in the OIDF Metadata Design Document Release 8.1.x.

16.3 Party Definition Tables

Party Definition consists of the details related to Party such as Annual Income, Primary Source of Wealth, and Probability of Default Percentage, and so on.

The purpose of Party Definition tables is to store personal information of a Party to check whether the premium is paid on a regular basis with no default on the payments, and the coverage is as per the income calculation of the Party.

For the metadata design, see the *Party Definition Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

16.4 Party Contact Details Tables

Party Contact Details consist of details about the phone.

The purpose of these tables is that the Party contact data can be used in KYC document related tables.

For the metadata design, see the *Party Contact Details Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

16.5 Party Identification Document Tables

The Party Identification Document contains the identification details of an individual Party. Identification means the action or process of identifying someone or something, or the fact of being identified.

The purpose of Party Identification Document tables is that the data from these tables can be used in the Know Before You Owe related tables.

For the metadata design, see the *Party Identification Document Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

16.6 Party Medical Attributes Tables

Party Medical Subject Area covers information elements pertaining to medical treatment or tests or conditions.

The purpose of the Party Medical attributes tables is to cover detail medical history of applicant or policyholder and use the data in either underwriting or continuing the existing insurance coverage.

For the metadata design, see the *Party Medical Attributes Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

16.7 Party Financial Data or Employment Attributes Tables

Party Financial Data or Employment Attributes contain details related to the profession of a Party.

For the metadata design, see the *Party Financial Data or Employment Attributes Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

16.8 Party Producer Exam, Certification, and License Tables

An Insurance Producer (also called an agent or insurance broker) means any person (individual, corporation, association, partnership, and so on) who or which is licensed to solicit, negotiate, or sell insurance. There are different categories of insurance and a producer must be licensed in each category he or she wishes to transact business.

Purpose of the Party Producer Exam, Certification, and License tables is to assess the eligibility of the Producer by evaluating the type of exam and its results, certifications completed, and license obtained by the Producer.

For the metadata design, see the *Party Producer Exam, Certification, and License Tablse* Section in the OIDF Metadata Design Document Release 8.1.x.

16.9 Party Results Tables

Party Results consist of the details related to Party such as Annual Income, and Primary Source of Wealth, and so on.

For the metadata design, see the *Party Results Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

16.10 Other Party Attributes Tables

This section consists of details about other Party attribute tables.

For the metadata design, see the *Other Party Attributes Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

16.11 Deploying Party Related Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

16.12 Populating Party Related Dimension Tables

Follow this SCD process to populate data into a Dimension table:

NOTE You can also follow this SCD process to populate data into any Hive-related Dimension table.

1. To populate data into a Dimension table, execute the SCD batch. For a detailed procedure, see the Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Dimension table, follow the procedure Check the Execution Status of the SCD Batch.
3. To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

16.13 Populating Party Related T2Ts (Result Tables)

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

16.14 Related Topics

You can see the following topics related to other function-specific tables:

- Insurance Underwriting Entities
- Insurance Contracts Tables
- Insurance Claims Tables

17 Product Tables

This section provides information about Product tables in the Oracle Insurance Data Foundation application.

Topics:

Product tables are used to load the product details that the customer has opted for.

For the metadata design, see the *About Product Dimension Tables* and *About Product T2T (Result Table)* Sections in the OIDF Metadata Design Document Release 8.1.x.

17.1 Deploying Product Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

> **NOTE** In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

17.2 Populating Product Dimension Tables

Follow this SCD process to populate data into a Dimension table:

> **NOTE** You can also follow this SCD process to populate data into any Hive-related Dimension table.

1. To populate data into a Dimension table, execute the SCD batch. For a detailed procedure, see the Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Dimension table, follow the procedure Check the Execution Status of the SCD Batch.
3. To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

17.3 Populating Product Related T2T (Result Table)

Follow this T2T process to populate data into any T2T Result table:

> **NOTE** Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

17.4 Related Topics

You can see the following topics related to other function-specific tables:

- Insurance Underwriting Entities
- Insurance Contracts Tables

18 Insurance Underwriting Entities

This section provides information about Insurance Underwriting entities in the Oracle Insurance Data Foundation application and step-by-step instructions to use this section.

Topics:

Insurance Underwriting describes the process of assessing risk, ensuring that the cost of the cover is proportionate to the risks faced by the Party or the organization concerned. It evaluates risk and exposures of potential Party or organizations. Insurance Company decides the coverage the Party or organization must receive, and the premium associated with the coverage or whether to accept the risk and insure them.

18.1 Insurance Underwriting Application Tables

When an applicant applies for insurance, the Insurance Company must look at different criteria to decide if they are going to accept the application for coverage. As a result, the Insurance Company creates and documents detailed information about the applicant and this process is called Underwriting Application.

For the metadata design, see the *Insurance Underwriting Application Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

18.2 Insurance Underwriting Quotes Tables

An insurance quote is an estimate of what an applicant's rate can be with a potential Insurance Company. Quotes are subject to change depending on how much information an applicant gives at the time of the quote. The more forthcoming an applicant is with information, the more accurate is their insurance quote.

For the metadata design, see the *Insurance Underwriting Quotes Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

18.3 Insurance Underwriting Tables for Risk Analysis

Insurance Company determines and assesses the risk for itself when offering insurance to an insurer through the Underwriting process. Lesser the risk, lesser the cost to an insurer.

Topics:

- About Insurance Underwriting T2T (Result Table) for Risk Analysis
- Party Criminal Conviction Tables
- Party Driving Violation Tables
- Party Lifestyle Activity Tables

18.3.1 About Insurance Underwriting Dimension Tables for Risk Analysis

For the metadata design, see the *Insurance Underwriting Dimension Tables for Risk Analysis* Section in the OIDF Metadata Design Document Release 8.1.x.

18.3.2 About Insurance Underwriting T2T (Result Table) for Risk Analysis

For the metadata design, see the *Insurance Underwriting T2T (Result Table) for Risk Analysis* Section in the OIDF Metadata Design Document Release 8.1.x.

18.3.3 Party Criminal Conviction Tables

Party Criminal Conviction is the outcome of a criminal prosecution, which concludes in a judgment that the defendant is guilty of the crime(s) charged.

For the metadata design, see the *Party Criminal Conviction Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

18.3.4 Party Driving Violation Tables

Party Driving Violation is the violation of specific Road and Motor vehicle laws, by the driver (Party) of a vehicle. This information is collected for the purpose of either underwriting or continuing suitable insurance coverage.

For the metadata design, see the *Party Driving Violation Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

18.3.5 Party Lifestyle Activity Tables

Lifestyle Activity covers physical activities of the Party such as auto racing, aviation, boat racing, mountain climbing, para scuba diving, and many more, which may be considered as risky by the insurance company.

The Lifestyle Activity tables store the details of the particular Lifestyle Activities of the Party.

Lifestyle Activity tables can be mainly classified into Lifestyle Activity Details and its specific sub-categories.

For the metadata design, see the *Party Lifestyle Activity Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

18.4 Producer and Producer Agreement Tables

This section provides information about Insurance Producer and Producer Agreement tables in the Oracle Insurance Data Foundation application.

Insurance Producer refers to a person or a firm that must be licensed under the respective laws of the land to sell, solicit, or negotiate Insurance Contracts.

The agreement means a formal agreement between an Insurance firm and an Insurance Producer on the lines of authority specified in the agreement to sell, negotiate, or solicit the Insurance products with prospective customers. In some companies, an agent must also be appointed for selling particular products.

An insurance license is an authority to sell, solicit, or negotiate particular Insurance products and awarded by jurisdiction-specific insurance regulatory authority to Insurance Producers whereas producer agreements are between producers and insurance firms. The availability of a valid insurance license is a pre-requisite of the producer agreements. There are two types of agreements:

- Producer Agreements: This can be between producers, or a producer and a carrier. The basic components of Producer agreements are Producer and Carrier. The Producer Agreement specifies distribution details along with product types. Essentially, one Producer Agency handles multiple individual Producers on behalf of the carrier, where individual Producers are not appointed by a carrier is what is excluded from this.
- Carrier Appointments: This is about a direct appointment by an insurance carrier or appointment by producer agency. Basic components of Carrier Appointments are Carrier, Producer Agreement identifier, and Appointment Identifier. If the said appointment is through producer agency or agreement, then the end-user must populate the Producer Identifier. For direct appointments, producer agreement identifiers can be chosen not to be populated. Carrier appointments specify distribution details along with product types and other details. However, it is expected that when carrier appointments are done as part of producer agreement, then distribution and product types are within the allowed values as per producer agreement.

Producer and Producer Appointment tables store the details related to Producer, Producer Agreements, Carrier Appointments, and license, which covers both the use cases mentioned above. Producer Appointments and Agreements are used to track down the sales.

To determine the eligibility of a Producer, the Examinations and Certifications are used. For more information, see Party Producer Exam, Certification, and License Tables.

These are the Producer and Producer Agreement entities:

- Stage Producer Appointment Details loads the Producer Appointment details. The appointment collection represents several appointments between the agent and several companies. The appointment may be to an insurance company carrier or can be used for appointments to distributors. Appointments are assigned to each Producer by each company by jurisdiction. In some companies, an agent also must be appointed for selling particular products.
- Stage Producer Insurance Distribution Details loads the Insurance distribution details. It may be used with Carrier Code as entity recognition for referencing the associated Insurance Distribution Agreement.
- Stage Producer Line of Authority Details loads the individual lines of authority, which are required to offer this product for sale.

For the metadata design, see the *Producer and Producer Agreement Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

18.5 Deploying Insurance Underwriting Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

18.6 Populating Insurance Underwriting Dimension Tables

Follow this SCD process to populate data into a Dimension table:

NOTE You can also follow this SCD process to populate data into any Hive-related Dimension table.

1. To populate data into a Dimension table, execute the SCD batch. For a detailed procedure, see the Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Dimension table, follow the procedure Check the Execution Status of the SCD Batch.
3. To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

18.7 Populating Insurance Underwriting T2T Result Tables

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

18.8 Related Topics

You can see the following topics related to other function-specific tables:

- Product Tables
- Insurance Contracts Tables
- Insurance Claims Tables

19 Insurance Contracts Tables

This section provides information about Insurance Contracts tables such as Common Policy Summary tables, Reinsurance Contract Held tables, Reinsurance Contract Issued tables, other Reinsurance tables, Insurance Coverage tables, Insurance Policy, and Fund tables, and Policy Schedules tables in the Oracle Insurance Data Foundation application.

Topics:

Accounting standards define insurance contract as a contract under which one party (the insurer) accepts significant insurance risk from another party (the policyholder) by agreeing to compensate the policyholder if a specified uncertain future event (the insured event) adversely affects the policyholder.

19.1 How to Load Insurance Contracts in OIDF?

This section provides information about how to load the Insurance Contracts in OIDF.

Topics:

19.1.1 Direct Insurance Contract Structure

In OIDF, Direct Insurance Contracts comprise of the following important components:

- Contract
- Coverage
- Supplemental Data
- Insurance Participation
- Policy and Funds
- Policy Loans
- Policy Conversion

In OIDF, Reinsurance is also a part of Contracts.

Party, which means a beneficiary, producer and agent, underwriter, and so on, are expected to be created and available for reference in the Party set of tables based on the related role. For details, see the Party Subject Area section.

19.1.2 Reinsurance Contracts

Reinsurance contract refers to an insurance contract issued by one entity (the reinsurer) to compensate another entity for claims arising from one or more insurance contracts issued by that other entity (underlying insurance contracts). Reinsurance Contracts consists of two types of contracts:

- Reinsurance Contracts Issued: This is a contract entered into by a reinsurance company when they sell the reinsurance contract to an insurance company. Often, this is called as a reinsurance contract written by a specific reinsurer to differentiate between purchased and sold.
- Reinsurance Contracts Held: This is a contract entered into by an insurance company when they purchase a reinsurance contract to mitigate the claims or losses.

For the list of tables and mapping details, see Reinsurance Contracts Tables.

19.2 Contract Tables

This section provides information about the Contract tables.

Topics:

- Common Policy Summary Tables
- Long Duration Contracts Table
- Life Insurance Contract Tables
- Health Insurance Contract Tables
- Annuity Contract Tables
- Property and Casualty Contract Tables
- Retirement Contract Tables

19.2.1 Common Policy Summary Tables

Common Policy Summary stores the detailed information elements of insurance policies directly written or issued by the insurance companies.

For the metadata design, see the *Common Policy Summary Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

19.2.2 Long Duration Contracts Table

Long Duration Contracts (principally life and annuity contracts) table stores the calculation of the liability of future policy benefits, a simplified amortization method for deferred acquisition costs, recording market risk benefits at fair value, and enhanced disclosures.

For the metadata design, see the *Long Duration Contracts Table* Section in the OIDF Metadata Design Document Release 8.1.x.

19.2.3 Life Insurance Contract Tables

Life insurance is a contract between an insurer and a policyholder in which the insurer guarantees payment of a death benefit to the named beneficiaries when the insured dies. The insurance company promises a death benefit in exchange for premiums paid by the policyholder.

For the metadata design, see the *Life Insurance Contract Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

19.2.4 Health Insurance Contract Tables

Health insurance is a contract between the insurer and the policyholder or an individual's sponsor in which the insurer provides the payments of the complete or a part of the risk that the policyholder (or registered beneficiaries) incurs in the form of medical expenses in exchange for premiums paid by the policyholder or by the individual's sponsor.

An insurer develops a routine finance structure, such as a monthly premium or payroll tax by estimating the overall health risk and health system expenses over the risk pool and provides the money to pay for the health care benefits specified in the insurance agreement.

For the metadata design, see the *Health Insurance Contract Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

19.2.5 Annuity Contract Tables

Annuity tables store the details of annuity contracts, which are defined as a written agreement between a financial institution and a customer outlining obligations of each Party in an annuity coverage agreement.

For the metadata design, see the *Annuity Contract Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

19.2.6 Property and Casualty Contract Tables

This section details the Property and Casualty Contract tables in the Oracle Insurance Data Foundation application.

In the Property and Casualty Contract category, the Auto Insurance tables store details of the policy related to the auto insurance contracts. The OIDF application supports auto insurance contracts for individual vehicles and a fleet of vehicles.

For the metadata design, see the *Property and Casualty Contract Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

19.2.7 Retirement Contract Tables

Retirement contracts are insurance and savings plans that help an individual to create a corpus for their future during the policy term. On maturity (retirement), a percentage amount of the accumulated corpus can be withdrawn as a lump sum and the rest in parts in the form of a pension.

For the metadata design, see the *Retirement Contract Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

19.3 Coverage Tables

The Policy Coverage table stores the list of all coverages and amounts set by the Insurer.

For the metadata design, see the *Coverage Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

Topics:

- Embedded Options Tables

19.3.1 Embedded Options Tables

This section provides information about the Embedded Options table.

For the metadata design, see the *Embedded Options Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

19.4 Insurance Participation Tables

This section provides information about the Insurance Participation tables.

For the metadata design, see the *Insurance Participation Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

19.5 Supplemental Data

Supplemental Data refers to the additional and contextual information applicable to only certain insurance contracts. This consists of primarily two information sets namely group insurance details and policy schedules.

Topics:

- Group Beneficiary Tables
- Policy Schedules Table
- Policy and Funds Tables
- Policy Loans Tables
- Policy Conversion Table
- Special Events Tables

19.5.1 Group Beneficiary Tables

Group Insurance refers to insurance that covers a defined group of people, such as employees of a company or members of a particular profession, typically against illness or death.

For the metadata design, see the *Group Beneficiary Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

19.5.2 Policy Schedules Table

Policy Schedules refers to the agreed non-linear premium and benefit schedules associated with the insurance contracts. For example, step up Annuity Contracts, and Term Life Insurance, where benefits are increasing or decreasing with the remaining term of the contract. Agreed schedules can be stored in this entity.

For the metadata design, see the *Policy Schedules Table* Section in the OIDF Metadata Design Document Release 8.1.x.

19.5.3 Policy and Funds Tables

Policy Funds refers to the investment made by the policyholder as part of the insurance contract and significant market risk is borne by the policyholder.

For the metadata design, see the *Policy and Funds Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

19.5.4 Policy Loans Tables

Some insurance policies allow policyholders to take the loans based on the amount accumulated to date as per the terms and conditions laid in the contract. Such loans are considered as a contract by itself in OIDF and must be loaded in stage loan contracts. This entity holds policy identifiers to link multiple loans taken under a given insurance policy.

For the metadata design, see the *Policy Loans Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

19.5.5 Policy Conversion Table

This table is used to store the policy details when converting one policy to another. The policy can be converted from one to multiple policies or multiple to one policy. An insurance policy with this type of provision allows the insured to switch to a different type of policy.

For the metadata design, see the *Policy Conversion Table* Section in the OIDF Metadata Design Document Release 8.1.x.

19.5.6 Special Events Tables

Any social event such as sports, film shooting, concerts, and so on are considered as a special event. An Insurer can insure their special event under the Policy Coverages against the liabilities occurring because of the event cancellation, terrorism, liquor, the artist not showing up, and so on.

Any Party that needs insurance coverage against the liquor liabilities can obtain insurance using the Policy Coverages. The liquor liabilities coverage may be applicable as a part of the Special Event Insurance if the liquor is served at the policy covered special event venue.

For the metadata design, see the *Special Events Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

19.5.7 Multilanguage Support Tables

For the metadata design, see the *Multilanguage Support Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

19.6 Reinsurance Contracts Tables

A reinsurance contract is defined as an insurance contract issued by one entity (the reinsurer) to compensate another entity for claims arising from one or more insurance contracts issued by that other entity. The requirements for the assessment of significant insurance risk in a reinsurance contract are the same as for an insurance contract. However, a reinsurance contract transfers significant insurance risk if it transfers substantially all of the insurance risk resulting from the insured portion of the underlying insurance contract, even if it does not expose the reinsurer to the possibility of a significant loss.

For the metadata design, see the *Reinsurance Contract Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

19.7 Deploying Insurance Contracts Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

> **NOTE** In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

19.8 Populating Insurance Contracts Dimension Tables

Follow this SCD process to populate data into a Dimension table:

> **NOTE** You can also follow this SCD process to populate data into any Hive-related Dimension table.

1. To populate data into a Dimension table, execute the SCD batch. For a detailed procedure, see the Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Dimension table, follow the procedure Check the Execution Status of the SCD Batch.
3. To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

19.9 Populating Insurance Contracts T2T Result Tables

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

19.10 Related Topics

You can see the following topics related to other function-specific tables:

- Insurance Claims Tables
- General Ledger Data and Management Reporting Tables
- Transaction Summary Tables

20 Policy Assets Tables

This section provides information about the Policy Assets tables in the Oracle Insurance Data Foundation application, and step-by-step instructions to use this section.

Topics:

Policy Assets is an asset with an insurance policy. An insured asset is one for which an insurance company compensates the policyholder if the asset is damaged or destroyed. Companies and individuals have insurance policies on their assets, or at least their tangible assets, to transfer the risk associated with owning them. There are two types of Policy Asset types. The type under the movable category includes automobiles, and so on. The type under the immovable category includes real estate, equipment, and so on.

For the metadata design, see the *About Policy Assets Dimension Tables* and *About Policy Assets T2Ts (Result Tables)* Sections in the OIDF Metadata Design Document Release 8.1.x.

20.1 Dwelling Insurance and Dwelling Inspection Tables

Dwelling Insurance consists of the tables related to Dwell and Dwell Inspection. Dwelling Insurance is a part of the insurance policy of the homeowners. Dwelling Insurance indemnifies the homeowner against any physical damage to their building due to any of the insured perils such as earthquake, fire, flood, and crime.

The Dwelling Inspection valuation is carried out at the time of purchase of a new house, renovations to the house, purchase of a new policy, and claims settlement.

Topics:

20.1.1 Benefits of the Dwelling Inspection Valuation

The Dwelling Inspection valuation benefits are as follows:

- Identifies the potential risks that can cause safety issues or losses. So that they can be addressed to avoid the claims.
- Helps the Dwelling Inspector (also called as Home Inspector) to identify different areas where the insured will be eligible for the discounts on the home insurance policy.
- Ensures that the insured dwelling value is not too high or too low. So that the insured saves money.

20.1.2 Role of the Dwelling Inspector in the Dwelling Inspection Valuation

The Dwelling Inspector considers the following factors during an inspection:

- The make and age of the house: The materials used to build the house, whether the house is primarily made of cement, bricks, or wood, and when was the house built. Some materials are more combustible.
- Electrical and plumbing system: If there is overloading, frayed wiring, or any potential for burst pipes. The inspector looks for exposed pipes and wiring.
- Roofing: The roof condition to see if there are several layers or if there are missing shingles, wear and tear, or holes.
- Chimneys: If the chimneys are well-maintained because the ill-maintained chimneys are a fire hazard.
- Walls and ceilings: The signs of leaks such as water spots and peeling paint.
- Water connections: The condition of rubber connecting hoses of the washing machine and dishwasher.
- Safety and security systems: The presence of fire alarms, smoke alarms, burglary alarms, dead-bolt locks, and motion sensors.
- Possible areas of liability: The presence of dogs and safety measures such as anti-slip floors.

20.1.3 Dwelling Insurance Tables

This section provides information about the Dwelling Insurance tables.

For the metadata design, see the *Dwelling Insurance Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

20.1.4 Dwelling Inspection Tables

This section provides information about the Dwelling Inspection tables.

For the metadata design, see the *Dwelling Inspection Tables* and *About Policy Assets T2Ts (Result Tables)* Section in the OIDF Metadata Design Document Release 8.1.x.

20.1.5 Appliances and Articles, and Cellphone Insurance Tables

This section provides information about the Appliances and Articles, and Cellphone Insurance tables.

The Appliances and Articles Insurance protects the contents of the building from the damages caused as agreed upon in the Policy. The Appliances and Articles Insurance is a part of the Dwelling Insurance.

The Cellphone Insurance covers the cost of cellphone repairs and parts replacement as agreed upon in the Policy.

For the metadata design, see the *Appliances and Articles, and Cellphone Insurance Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

20.1.6 Vehicle and Cargo Entities

For the metadata design, see the following sections (categories of vehicle and cargo entities) in the OIDF Metadata Design Document Release 8.1.x:

- Vehicle Tables
- Transportation Tables
- Policy Covered Asset Storage Tables

20.1.7 Aviation Insurance Tables

The Aviation Insurance covers the physical and operational damages to different types of aircrafts and their parts. It also covers the resultant liabilities.

For the metadata design, see the *Aviation Insurance Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

20.1.8 Cyber Liability Insurance Tables

Cyber Liability Insurance is an insurance policy that helps protect organizations from the consequence of cyberattacks and hacking threats such as data breaches, viruses, or other forms of malicious cybersecurity activity. It covers the financial losses due to cyberattacks or other tech-related risks, as well as privacy investigations or lawsuits following an attack.

For the metadata design, see the *Cyber Liability Insurance Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

20.2 Deploying Policy Assets Table on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

20.3 Populating Policy Assets Dimension Tables

Follow this SCD process to populate data into a Dimension table:

NOTE You can also follow this SCD process to populate data into any Hive-related Dimension table.

1. To populate data into a Dimension table, execute the SCD batch. For a detailed procedure, see the Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Dimension table, follow the procedure Check the Execution Status of the SCD Batch.
3. To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

20.4 Populating Policy Assets T2T Result Table

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

21 Stream Data Tables for the Auto Telematics

This section provides information about the vehicle IoT tables and their results derived from other result tables. Vehicle IoT tables consist of the Telematics tables, to which the data is sourced from the Telematics device attached to the vehicles, other vehicle monitoring systems, and vehicle service records.

Topics:

21.1 Telematics Tables

Telematics is a generic term used to refer to an interdisciplinary field in Information Technology that combines telecommunications and informatics. An electronic device designed using Telematics is installed in the vehicles and it can telecommunicate computerized information. OIDF supports sourcing of the Telematics data along with other data from the devices such as Tachograph and driver behavior monitoring systems in the Auto Insurance.

Vehicle IoT (Internet of Things) Result data flow is used to design the OIDF tables that capture processed outputs of the trip level details, vehicle performance, and vehicle driver behavior details.

For the metadata design, see the *Telematics Tables* and the following Sections in the OIDF Metadata Design Document Release 8.1.x.

- Vehicle IoT Data Flow
- About Telematics Dimension Tables
- About Telematics T2Ts (Result Tables)
- Vehicle IoT (Internet of Things) Result Data Flow
- Populating the FCT_TRIP_DETAILS Table
- Populating the FCT_TRIP_SUMMARY Table
- Population of the FCT_VEH_PERFORMANCE_SUMMARY Table
- Results derived in the FCT_DRIVER_BEHAVIOUR_SUMMARY Tables

21.2 Deploying Stream Data Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

> **NOTE** In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts.

21.3 Populating Stream Data Dimension Tables

Follow this SCD process to populate data into a Dimension table:

> **NOTE** You can also follow this SCD process to populate data into any Hive-related Dimension table.

1. To populate data into a Dimension table, execute the SCD batch. For a detailed procedure, see the Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Dimension table, follow the procedure Check the Execution Status of the SCD Batch.
3. To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

21.4 Populating Stream Data T2T Result Tables

Follow this T2T process to populate data into any T2T Result table:

> **NOTE** Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

21.5 Related Topics

You can see the following topics related to other function-specific tables:

- Insurance Contracts Tables
- Insurance Claims Tables

22 Stream Data Tables for the Health Wearables

This section provides information about the Health Wearable related IoT (Internet of Things) tables and their results derived from the other Result tables. The Health Wearable related IoT tables consist of the Telematics tables to which the data is sourced from the Health Wearable activity device wore by the Party.

Topics:

Telematics is a generic term used to refer to an interdisciplinary field in Information Technology that combines telecommunications and informatics. A wearable electronic device designed using Telematics along with Biosensors is called a Health Wearable device and it can telecommunicate computerized information. OIDF supports sourcing of the Telematics data along with other data from the devices such as wearable fitness trackers, smart health watches, wearable ECG monitors, wearable blood pressure monitors, and Biosensors.

Health IoT Result Tables data flow is used to design the OIDF tables that capture processed outputs of the Party physical activity details and health profile guidelines provided to the Party. The results are derived into the Physical Activity Summary and Physical Activity Tracking Result tables.

For the metadata design, see the following Sections in the OIDF Metadata Design Document Release 8.1.x.

- About Health Wearable Dimension Tables
- About Health Wearable T2Ts (Result Tables)
- Health IoT (Internet of Things) Result Data Flow
- Populating the FCT_PHYSICAL_ACTIVITY_SUMMARY Table
- Populating the FCT_PHYSICAL_ACTIVITY_TRACKING Table

22.1 Deploying Stream Data Tables for the Health Wearables on Hive

All RDBMS-related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts.

22.2 Populating Stream Data Dimension Tables for the Health Wearables

Follow this SCD process to populate data into any Stream Data Dimension table for the Health Wearables:

NOTE You can also follow this SCD process to populate data into any Hive-related Stream Data Dimension table.

1. To populate data into any Stream Data Dimension table for the Health Wearables, execute the SCD batch. For a detailed procedure, see the Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Stream Data Dimension table for the Health Wearables, follow the procedure Check the Execution Status of the SCD Batch.
3. To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

22.3 Populating Stream Data T2T Result Tables for the Health Wearables

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

22.4 Related Topics

You can see the following topics related to other function-specific tables:

- Insurance Contracts Tables
- Insurance Claims Tables

23 Pandemic Data and Hospital Capacity Data Tables

This section provides information about the Pandemic Data and Hospital Capacity Data tables.

Topics:

23.1 Pandemic Data Table

Pandemic is the outbreak of a disease across a wide geographical area affecting a large number of people. A few examples of the pandemic are SARS, MERS, Zika, Ebola, COVID-19, and so on.

During a pandemic, the public authority of each affected region generally provides data about the number of people affected by that pandemic and the areas where the outbreak has occurred.

OIDF supports the storage of the country level pandemic data details in the categories of positive cases, count of hospitalized people, count of recovered people, death count, ventilator count, and so on.

For the metadata design, see the *Pandemic Data Table* and *Hospital Capacity Data Table* Sections the following Sections in the OIDF Metadata Design Document Release 8.1.x.

23.2 Hospital Capacity Data Table

The public authority responsible for handling the Pandemic Data may also provide data regarding the capacity of the healthcare sector of the area.

OIDF supports the storage of the region level hospital capacity data in the categories of bed count for different severity levels and ICU beds per unit of the population.

23.3 Deploying Pandemic Data and Hospital Capacity Data Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

23.4 Populating Pandemic Data and Hospital Capacity Data T2T Result Tables

Follow this T2T process to populate data into any T2T Result table:

> **NOTE** Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

23.5 Related Topics

You can see the following topics related to other function-specific tables:

- Party Subject Area
- Insurance Contracts Tables
- Insurance Claims Tables

24 Common Account Summary Tables

This section provides information about Common Account Summary tables in the Oracle Insurance Data Foundation application such as Asset and other Account Summary, Trading and Investment, Instruments Contracts, and Capital and Borrowings, and step-by-step instructions to use this section.

Topics:

24.1 Abstract

For an Insurance company, this entity stores account covering their exposures to Trading, Investment as well as Loans.

OIDF Subject Area of Trading and Investment refers to the significant portion of assets of the Insurance Company. Trading and Investment Subject Area comprise of following types of entities:

- **Contracts:** Investment Contracts refers to a contract or transaction or scheme where entity invest money either on its own behalf or on behalf of their clients with the expectation of profit from sale or holding of such investment.
- **Instrument:** Tradable investments use instruments with ISIN or CUSIP or Other known identifiers. Instrument definition most commonly is published by the issuer of instrument, which can be another public or private or government entity or stock exchange.

For more information, see the *Abstract* Section of the *Common Account Summary Tables* Chapter in the OIDF Metadata Design Document Release 8.1.x.

24.2 Trading and Investments Contract Tables

This section provides Instrument, Trading and Investments Contract table details.

For the metadata design, see the following Sections in the OIDF Metadata Design Document Release 8.1.x:

- Instruments Tables
- About Trading and Investments Contract T2Ts (Result Tables)

24.3 Macroeconomic Variable Tables

Macroeconomic variables are the indicators that point to the current trends in the economy and the factors affecting the economy. These variables are used to get behavioral data of the market systems operating on a large scale and to understand how different sectors of the economy are related to each other.

OIDF supports the data storage of different macroeconomic variable values across different regions. This stored data contains both historical values and future projections. This data is used in LLFP (OFS IFRS Application Pack) to compute the probability of default using the Cox proportional-hazards model.

For the metadata design, see the *Macroeconomic Variable Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

24.4 Capital Instruments Position and Transactions Tables

Capital Instruments Position and Transactions tables pertain to the capital instrument and transactional part of the balance sheet of a financial institution. It covers capital instruments such as Equity and Debt along with non-retail borrowings done by financial institutions.

For the metadata design, see the *Capital Instruments Position and Transactions Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

24.5 Other Common Account Summary Tables

For the metadata design, see the *Other Common Account Summary Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

24.6 Abstract of Common Account Summary Result Tables

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 Common Account Summary tables 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 Common Account Summary tables that are organized by the application subject area.

The preceding Common Account Summary tables are part of the 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.

24.7 Deploying Common Account Summary Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List

of Unsupported T2Ts

24.8 Populating Common Account Summary Dimension Tables

Follow this SCD process to populate data into a Dimension table:

NOTE You can also follow this SCD process to populate data into any Hive-related Dimension table.

1. To populate data into a Dimension table, execute the SCD batch. For a detailed procedure, see the Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Dimension table, follow the procedure Check the Execution Status of the SCD Batch.
3. To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

24.9 Populating Common Account Summary T2T Result Tables

NOTE Ensure to load FCT_COMMON_ACCOUNT_SUMMARY and FSI_EXCHANGE_RATES tables prior to loading any of the other Account Summary tables.

Ensure to manually configure the SETUP_MASTER table with required GAAP_CODEs before executing Account Summary Population T2Ts. For an account, load only one GAAP_CODE to Fact Common Account Summary table. By default, OIDF installer seeds the following entry into SETUP_MASTER. When executing through a batch, the RUNSkey defaults to -1.

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

25 Loan Account Summary Tables

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

Topics:

Loan Account Summary provides a detailed report of insurance transactions done.

Loan Account in Insurance is used, when an insured takes a loan against his or her policy. The policy code is used to track the loan and its outstanding amount for the reporting if the policy is closed and then decided how to treat the loan.

For the metadata design, see the *Loan Account Summary Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

25.1 Deploying Loan Account Summary Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

> **NOTE** In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

25.2 Populating Loan Account Summary T2T Result Tables

Follow this T2T process to populate data into any T2T Result table:

> **NOTE** Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

25.3 Related Topics

You can see the following topics related to other function-specific tables:

- Insurance Contracts Tables
- Insurance Claims Tables
- Placed Collateral Tables
- Transaction Summary Tables

26 Fiduciary Services Investment Summary Population

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

Topics:

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 may have different investments that the person may do. The customer may have different assets under the trust such as Investments, Mutual Funds, and so on. Therefore, 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.

For the metadata design, see the *Fiduciary Services Investment Summary Population* Section in the OIDF Metadata Design Document Release 8.1.x.

26.1 Deploying Fiduciary Services Investment Summary Result Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

26.2 Populating Fiduciary Services Investment Summary T2T Result Tables

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.

2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

26.3Related Topics

You can see the following topics related to other function-specific tables:

- Common Account Summary Tables

27 Mitigants Results Tables

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

Topics:

Mitigant means to reduce the loss of life and property by reducing the impact of undesirable events.

The purpose of the Mitigants Results table is to store the list of actions to be taken to reduce the impact of undesirable events.

For the metadata design, see the *Mitigants Results Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

27.1 Deploying Mitigants Results Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

27.2 Populating Mitigants Results T2T Result Tables

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

27.3 Related Topics

You can see the following topics related to other function-specific tables:

- Common Account Summary Tables

28 Placed Collateral Tables

This section provides information about Placed Collateral Result tables in the Oracle Insurance Data Foundation application and step-by-step instructions to use this section.

Topics:

Financial Institutions place collateral in derivative transactions, reinsurance issued, or similar use cases where the risk of obligations not being served by reporting entity are perceived high and are secured by seeking agreed collateral in lieu to reduce the risk. Therefore, Placed Collateral refers to the collateral placed by reporting entity to other entities. The Mitigant reflect collateral received and collateral placed functions exactly opposite as of Mitigant.

In Data Foundation, assets hold on books and collateral placed are treated differently. For example, assume a use case where Insurance Company has purchased a lot of Government securities with a market value of 10 Million USD. When they write a reinsurance contract, Insurance Company has sought collateral comprising of government securities worth 4 million USD. In this case, stage investments hold data reflecting 6 million USD, whereas stage placed collateral holds data of 4 million USD.

For the metadata design, see the *Placed Collateral Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

28.1 Deploying Placed Collateral Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

> **NOTE** In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

28.2 Populating Placed Collateral Dimension Table

Follow this SCD process to populate data into a Dimension table:

> **NOTE** You can also follow this SCD process to populate data into any Hive-related Dimension table.

1. To populate data into a Dimension table, execute the SCD batch. For a detailed procedure, see the Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Dimension table, follow the procedure Check the Execution Status of the SCD Batch.
3. To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

28.3 Populating Placed Collateral T2T Result Tables

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

28.4 Related Topics

You can see the following topics related to other function-specific tables:

- Loan Account Summary Tables

29 Transactions and Accounting Tables

This section provides information about Insurance Policy Transactions tables in the Oracle Insurance Data Foundation application.

This consists of primarily two sections. Policy Transactions generated out policy life cycle accounting events and policy commission which are specific to producer or agents business and are generated out of underwriting and some period in the policy life cycle. However, they are not expected to overlap with policy transactions which exclude any producer or agent-specific transactions.

Topics:

Insurance Policy Transactions are about accounting transactions done for a given policy. They are sourced from respective accounting systems in the Insurance Company.

The purpose of Insurance Policy Transactions tables is to store attributes pertaining to policy transactions related data on an 'as-is' basis received from the source system for further operation reporting. OIDF holds for every product processor or contract table, one transaction table.

For the metadata design, see the following Sections in the OIDF Metadata Design Document Release 8.1.x:

- About Insurance Policy Transactions Dimension Tables
- About Insurance Policy Transactions T2Ts (Result Tables)
- Policy Commission Tables

29.1 Deploying Insurance Policy Transactions Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

> **NOTE** In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

29.2 Populating Insurance Policy Transactions Dimension Tables

Follow this SCD process to populate data into a Dimension table:

> **NOTE** You can also follow this SCD process to populate data into any Hive-related

Dimension table.

1. To populate data into a Dimension table, execute the SCD batch. For a detailed procedure, see the Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Dimension table, follow the procedure Check the Execution Status of the SCD Batch.
3. To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

29.3 Populating Insurance Policy Transactions T2T Result Tables

NOTE As a prerequisite, ensure to load the DIM_STD_TRANSACTION_TYPE seeded table.

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

29.4 Related Topics

You can see the following topics related to other function-specific tables:

- Insurance Contracts Tables
- Loan Account Summary Tables
- Insurance Claims Tables

30 General Ledger Data and Management Reporting Tables

This section provides information about GL Data and GL to Management Reporting tables in the Data Foundation application.

Topics:

A General Ledger is the foundation of a financial system, which is used to store and organize financial data. General Ledgers are used to create the financial statements of a company. A General Ledger account is an account or record used to sort, store, and summarize the transactions of a company. Transactions are posted to individual sub-ledger accounts as defined by the chart of accounts of that company. A Legal Entity in the Insurance institution can maintain the same General Ledger data (either solo or consolidated) in one or more source systems.

In OIDF, Stage GL Data stores summarized data at the primary granularity of General Ledger whereas several transaction tables store the individual transactions. OFSAA suite of applications expect the consolidation of General Ledger data occurring in the financial systems, and OIDF entities are loaded after consolidation.

30.1 About GL Data T2T (Result Table)

The purpose of GL Data tables is to keep track of its financial transactions and to prepare financial reports. Each account is a unique record that summarizes each type of asset, liability, equity, revenue, and expense.

For the metadata design, see the *About GL Data T2T (Result Table)* Section in the OIDF Metadata Design Document Release 8.1.x.

30.2 About GL to Management Reporting T2T (Result Table)

The purpose of GL to Management Reporting tables is to provide a detailed description of every general ledger account and the transactions that make up the balance in that account. The general ledger holds all the financial information used to create the income statement and balance sheet reports and serves several main purposes in the financial operation of the business.

For the metadata design, see the *About GL to Management Reporting T2T (Result Table)* Section in the OIDF Metadata Design Document Release 8.1.x.

30.3 About Mapper for GL to Management Reporting Result Table

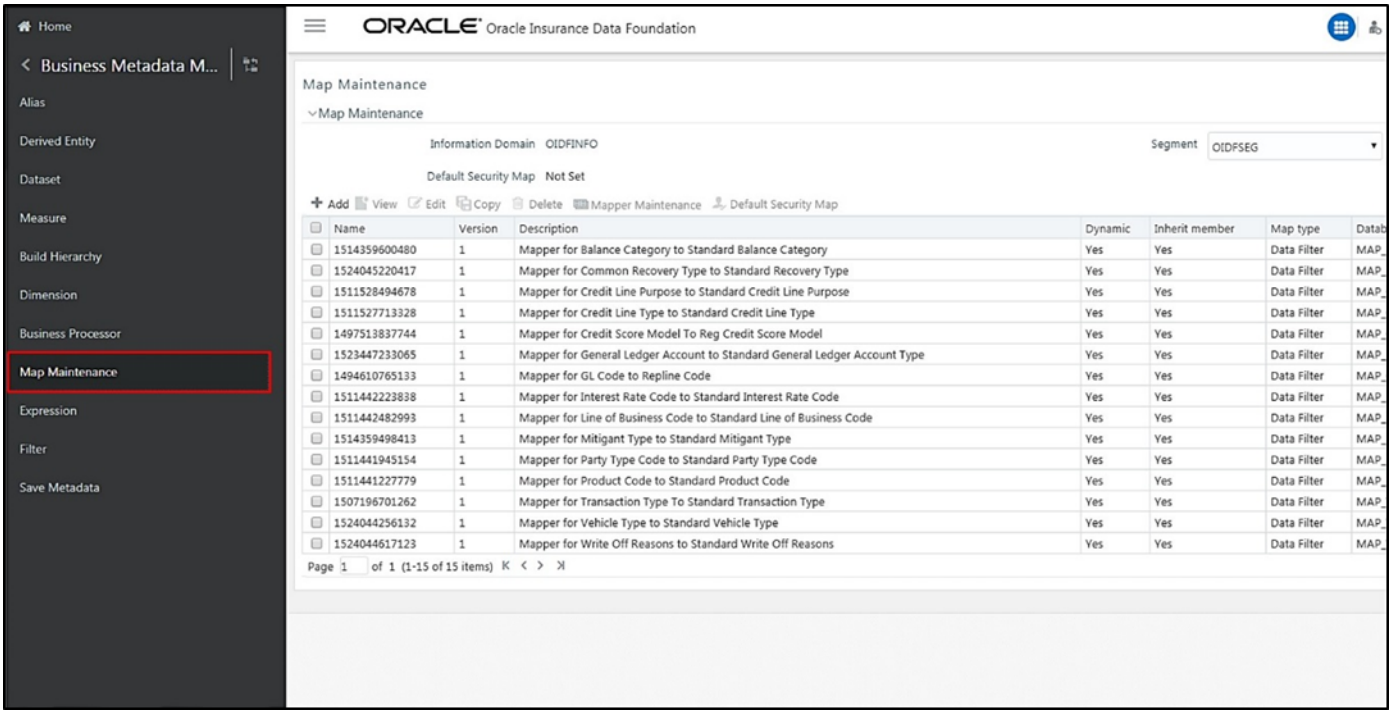
The Fact Management Reporting table uses reporting line codes, which are OFSAA specific values that represent each line item of reporting requirements. On the other hand, the Stage GL Data table uses the General Ledger Codes. These General Ledger Codes are user-specific values that represent each General 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.

30.4 Maintaining Mappers for GL to Management Reporting Result Table

To maintain Mappers through the Map Maintenance component of OFSAAI:

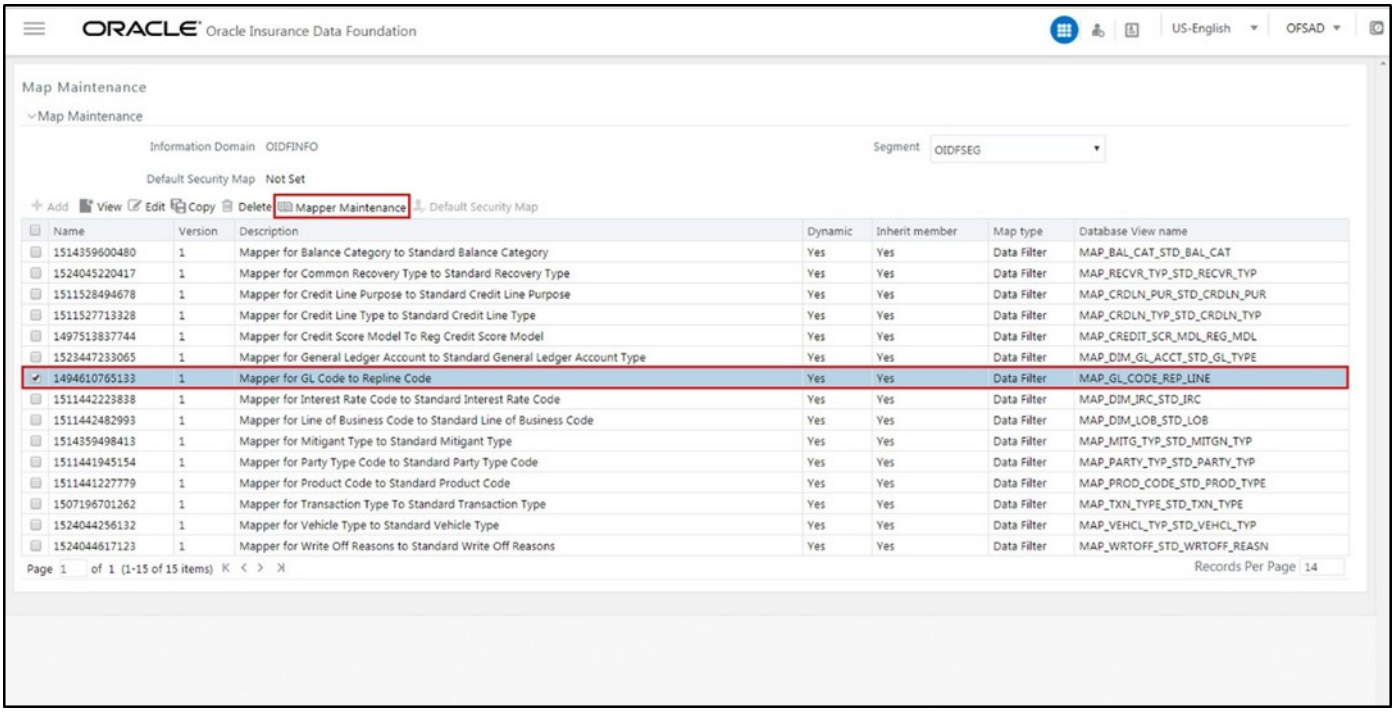
1. From **OFSAA Home**, select **Oracle Insurance Data Foundation**, select **Unified Analytical Metadata**, select **Business Metadata Management**, and then select **Map Maintenance**.
2. The **Map Maintenance** page is displayed.

Figure 80: Map Maintenance page



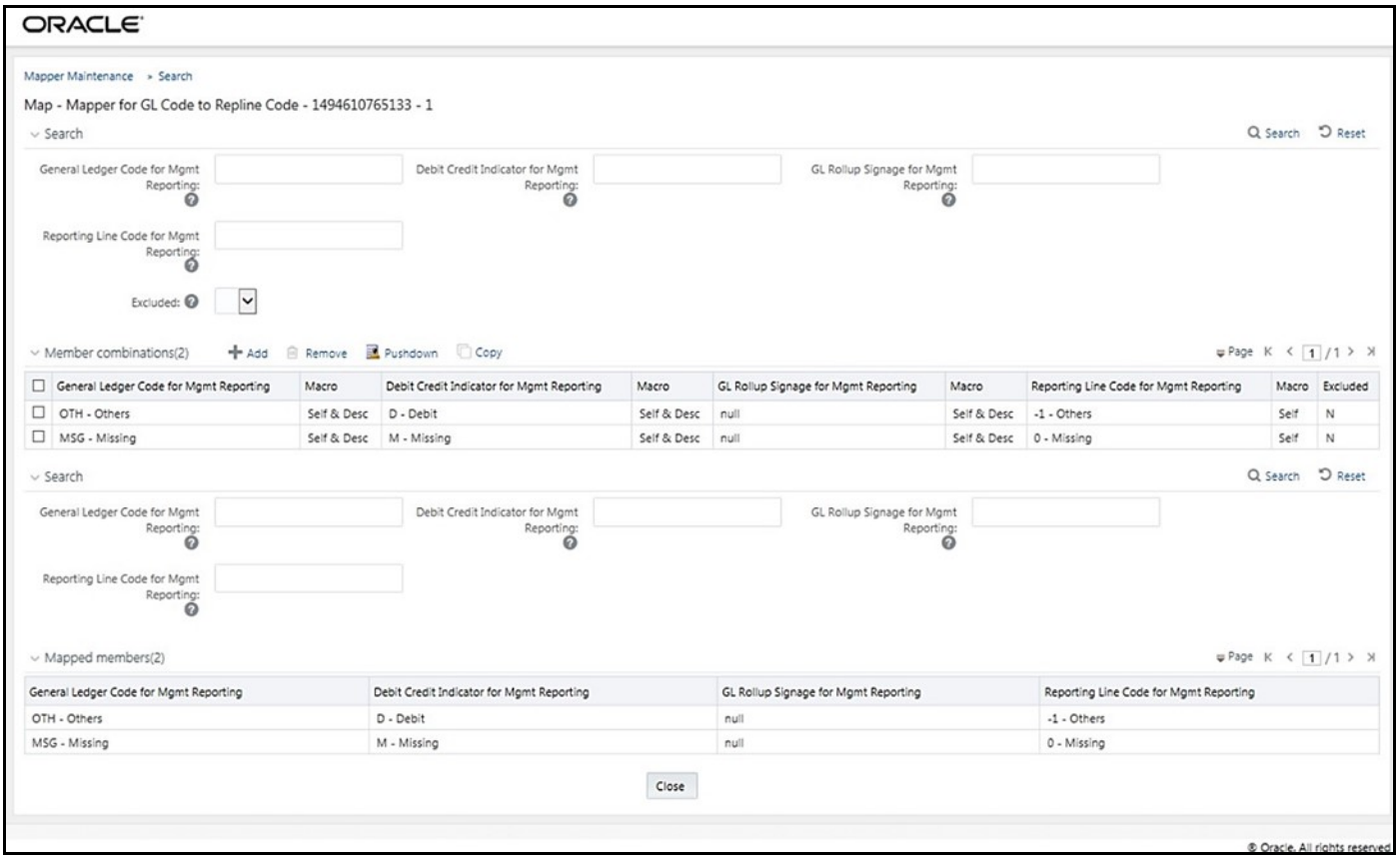
3. Select the mapper record **Mapper for GL Code to Repline Code** and then click the Mapper Maintenance icon.

Figure 81: Select the required Mapper



4. 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 82: Mapper Maintenance page



Topics:

- Prerequisites for Mapper Maintenance
- Possible Mapping Combinations
- Performing Multiple Sets of Mapping Combinations

30.4.1 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

30.4.2 Possible Mapping Combinations

These are the four types of mapping combinations:

- 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

30.4.2.1 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:

- 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.

30.4.2.1.1 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 the 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.

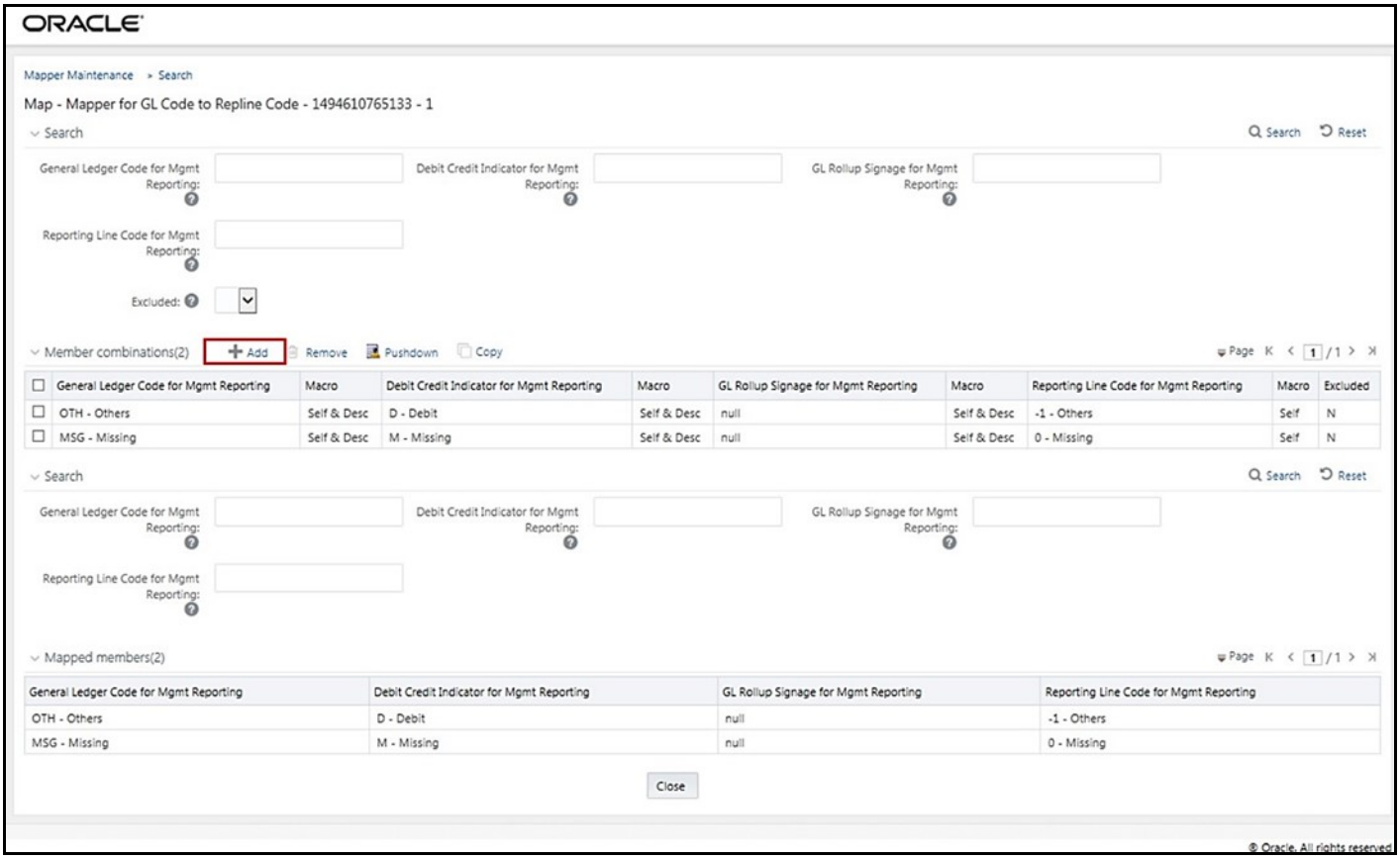
NOTE The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:

- The value must be M when you map the GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets the value to M.
- The value must be D when you map the GL Code to the Debit member.
- The value must be C when you map the GL Code to the Credit member.

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

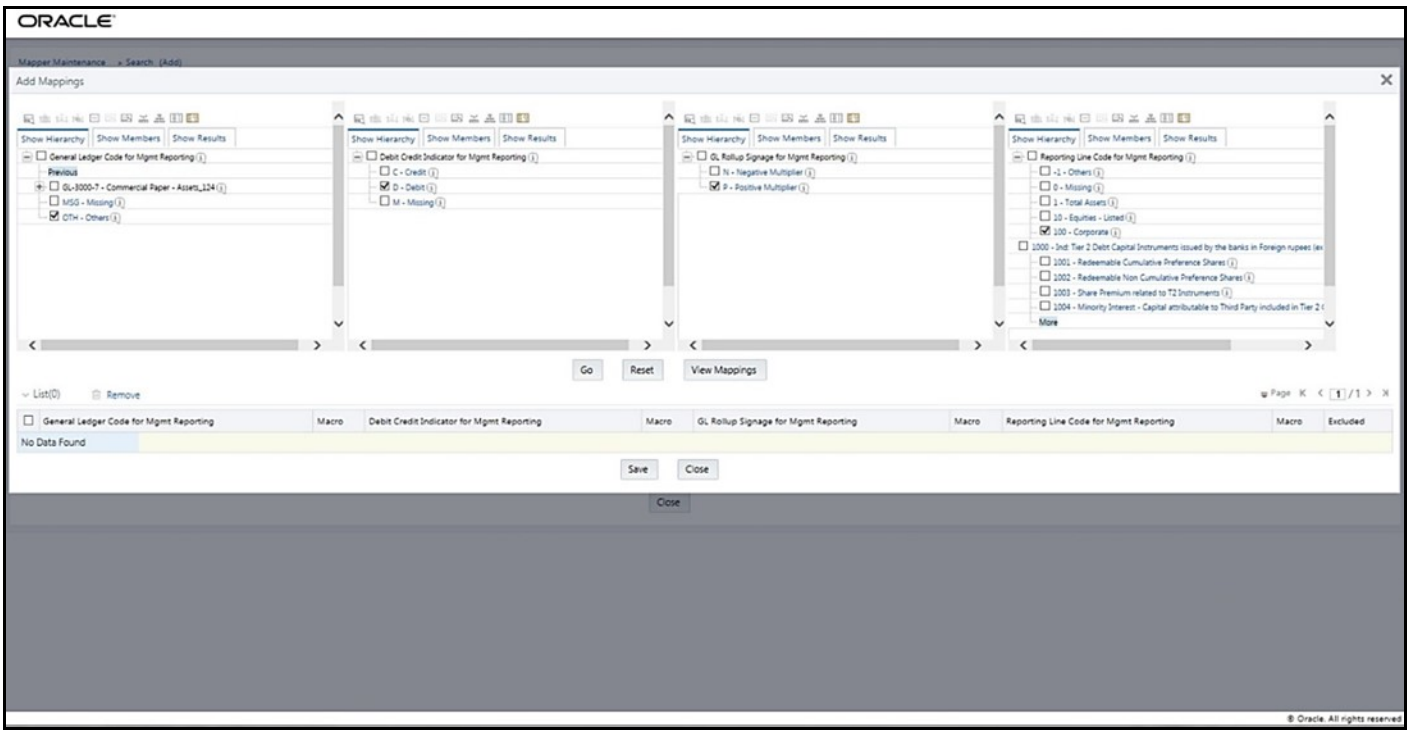
1. In the **Mapper Maintenance** page, in the Member combinations section, click **Add**.

Figure 83: Select Add in the Mapper Maintenance page for the One-to-One mapping at child hierarchy level



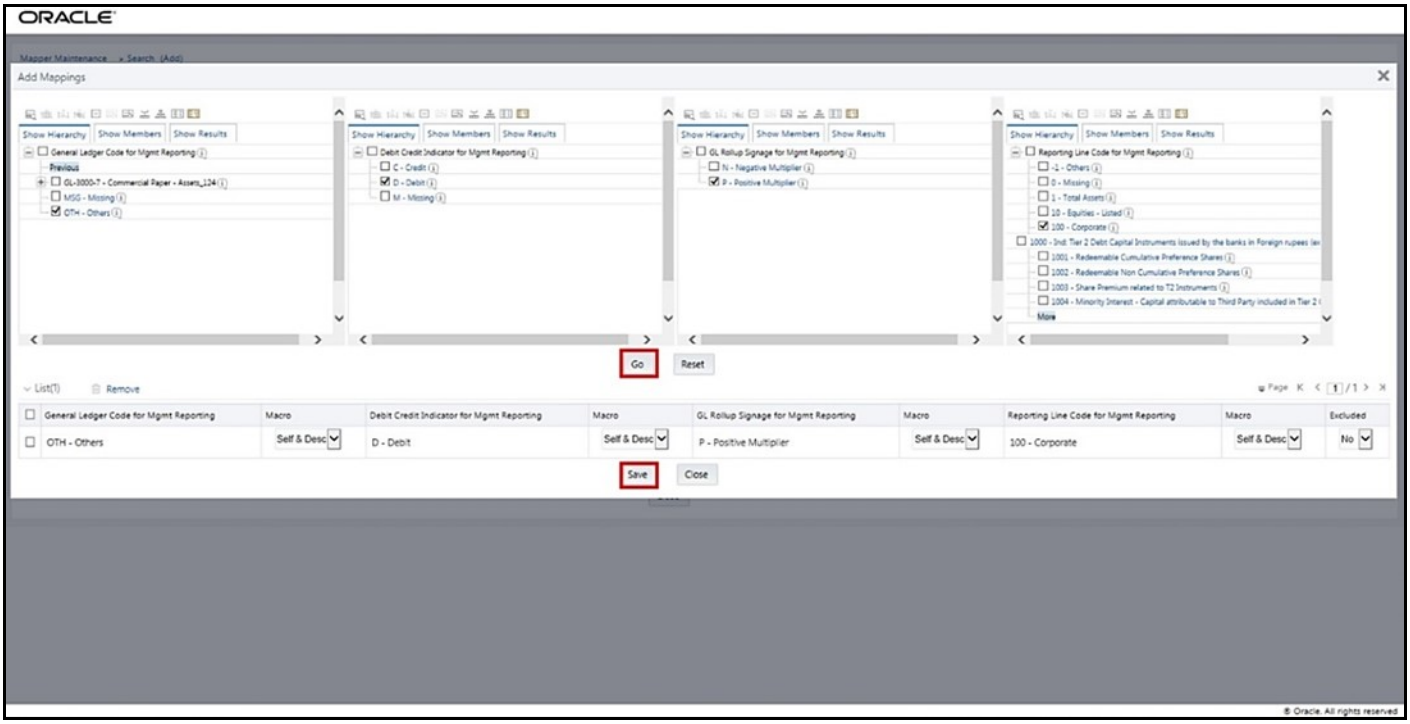
2. The **Add Mappings** page is displayed. For illustration, select the member of the General Ledger Code for Mgmt Reporting hierarchy **OTH - Others**, and the member of the Debit Credit Indicator for Mgmt Reporting hierarchy **D - Debit** to map to the member of the Reporting Line Code for Mgmt Reporting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

Figure 84: Add Mappings page for the One-to-One mapping at child hierarchy level



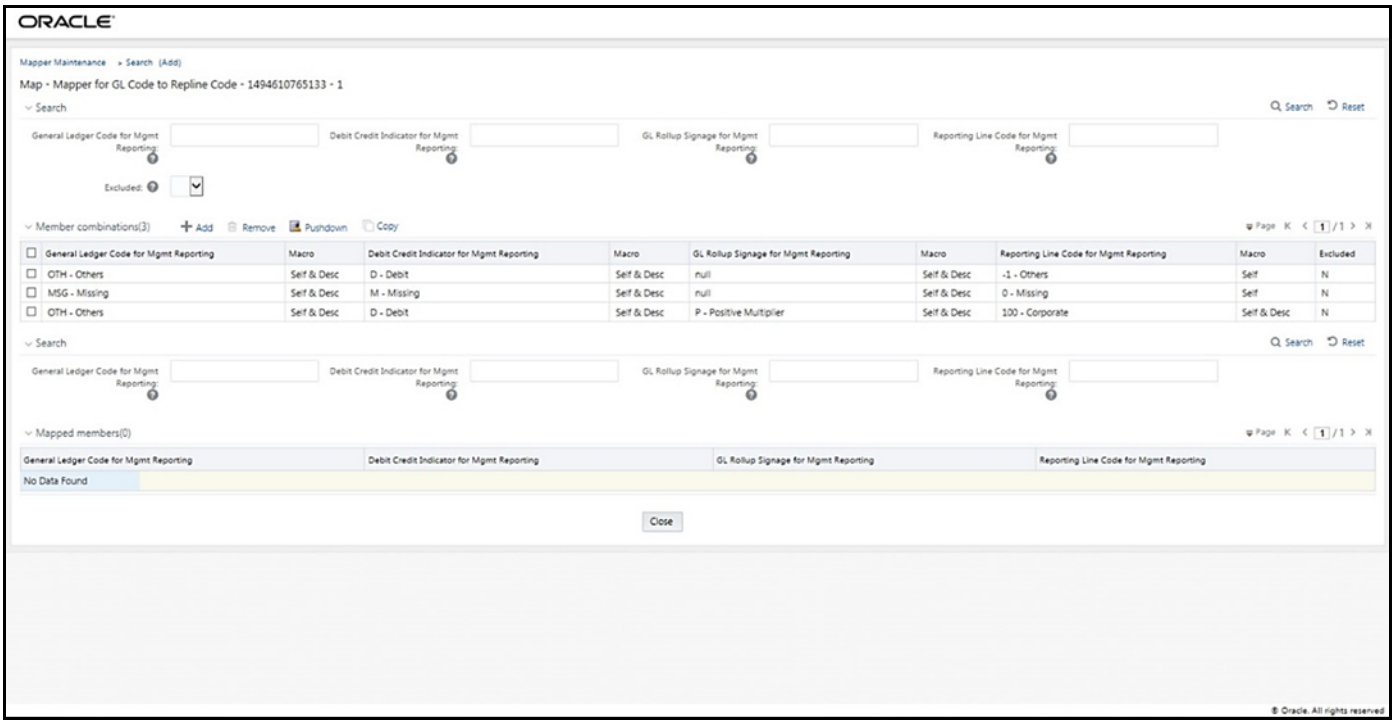
3. To map the members, click **Go**. The list of mapped members is displayed at the bottom. To save the mappings, click **Save**.

Figure 85: Map the members and save the mappings



4. An acknowledgment pop-up message appears. To confirm saving the mapping, click **Yes**.
5. The mapped member combinations are listed in the **Mapper Maintenance** page.

Figure 86: Added mappings listed in the Mapper Maintenance page



30.4.2.1.2 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 the 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.

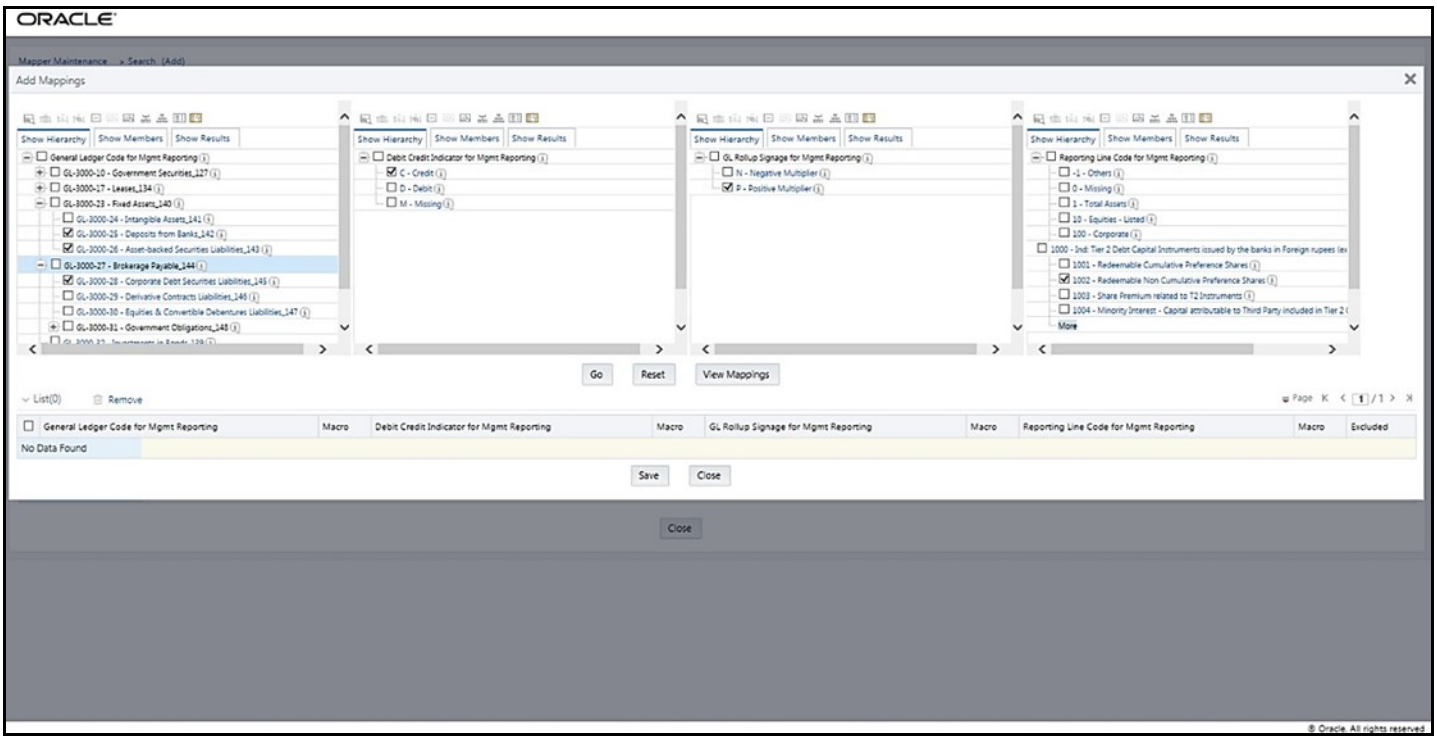
NOTE In the T2T process, in the Stage GL Data table, the GL Code aggregation takes place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:

- The value must be M when you map the GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets the value to M.
- The value must be D when you map the GL Code to the Debit member.
- The value must be C when you map the GL Code to the Credit member.

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

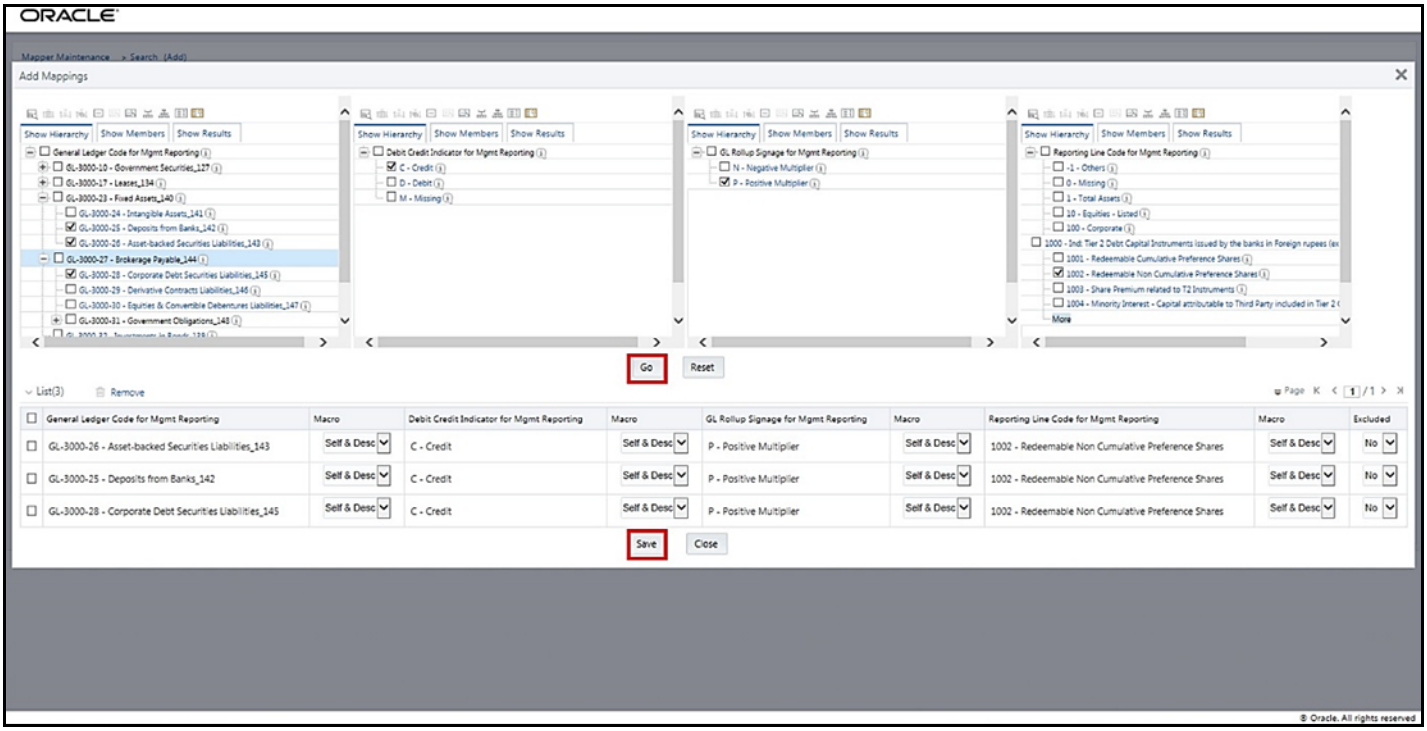
1. In the **Mapper Maintenance** page, in the Member combinations section, click **Add**.
2. The **Add Mappings** page is displayed. For illustration, select the child members of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-26 – Asset-backed Securities Liabilities_143 and GL-3000-28 - 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 87: 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 appears at the bottom. To save the mappings, click **Save**.

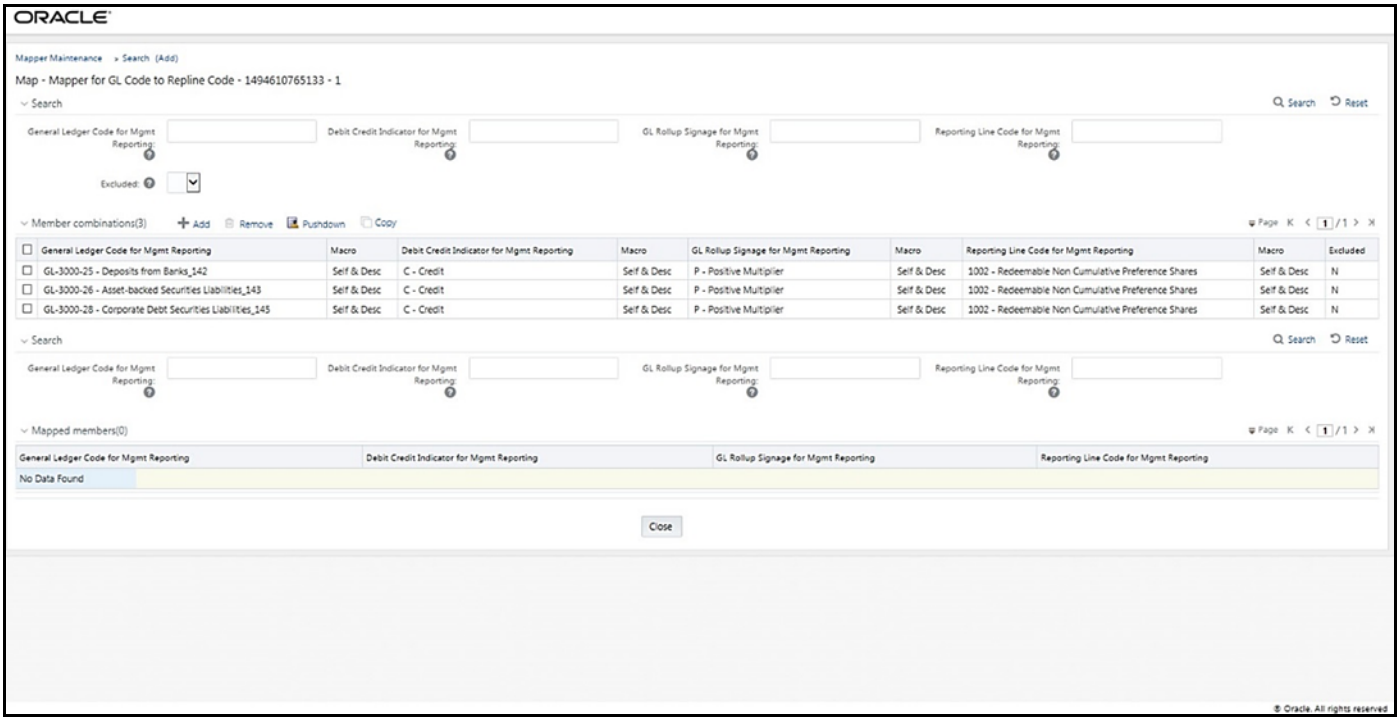
Figure 88: Map the members and save the mappings



4. An acknowledgment pop-up message appears. To confirm saving the mappings, click **Yes**.

5. The mapped member combinations are listed in the **Mapper Maintenance** page.

Figure 89: Added mappings listed in the Mapper Maintenance page



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

To map multiple GL Codes to multiple Reporting Line Codes, use the Mapper Maintenance utility. In this utility, select two or more members in the General Ledger Code for the Mgmt Reporting hierarchy, 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 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:

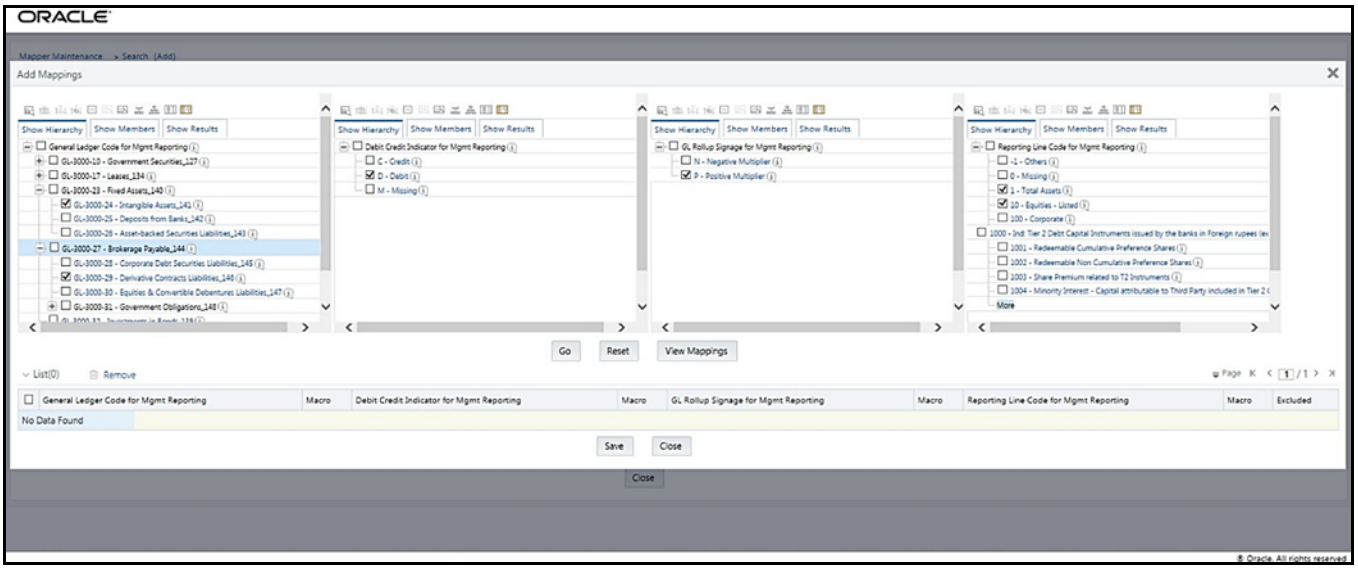
- The value must be M when you map the GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets the value to M.
- The value must be D when you map the GL Code to the Debit member.
- The value must be C when you map the GL Code to the Credit member.

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

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

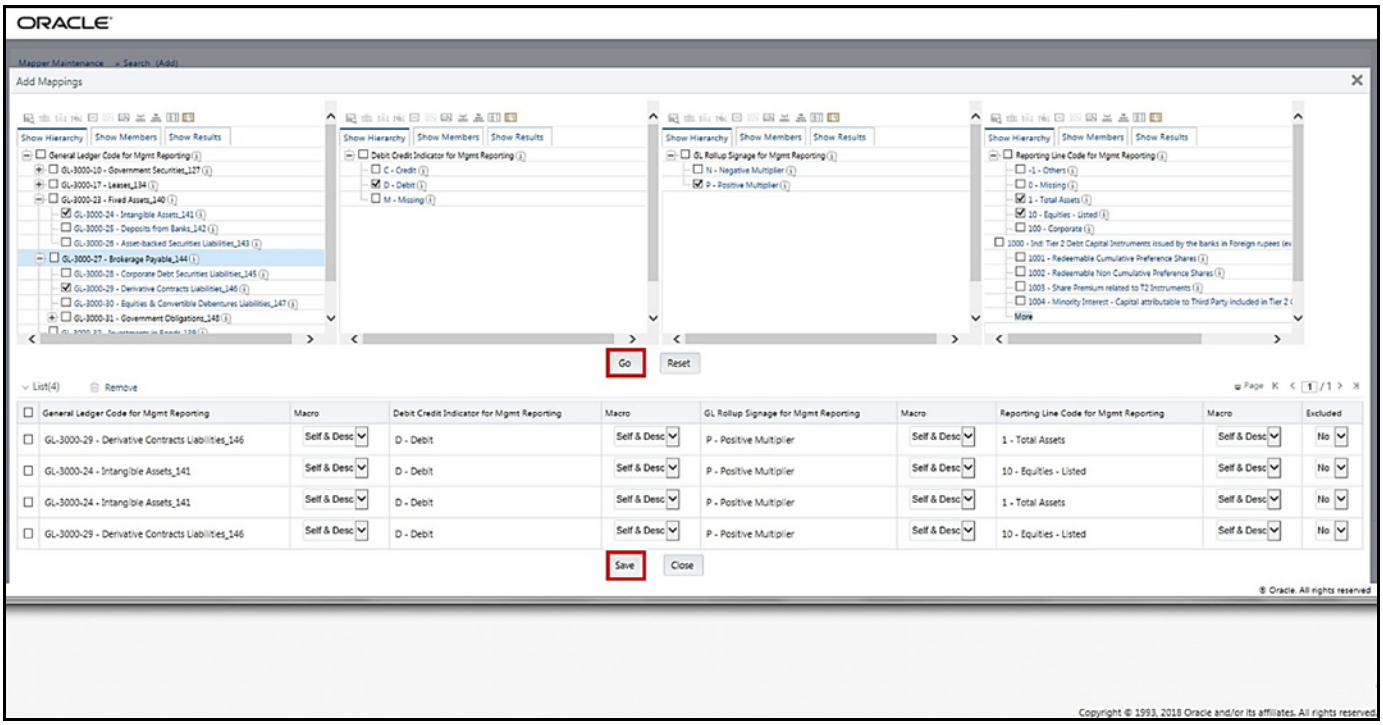
2. The **Add Mappings** page is displayed. For illustration, select the child members of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-24 – Intangible Assets_141 and GL-3000-29 – Derivative Contracts Liabilities_145, and the member of the Debit Credit Indicator for Mgmt Reporting hierarchy, D - Debit to map to the child members of the Reporting Line Code for Mgmt Reporting hierarchy, 1 - Total Assets and 10 - Equities Listed with the member of the GL Rollup Signage for Mgmt Reporting hierarchy, P - Positive Multiplier.

Figure 90: Add Mappings page for the Many-to-Many mapping at child hierarchy level



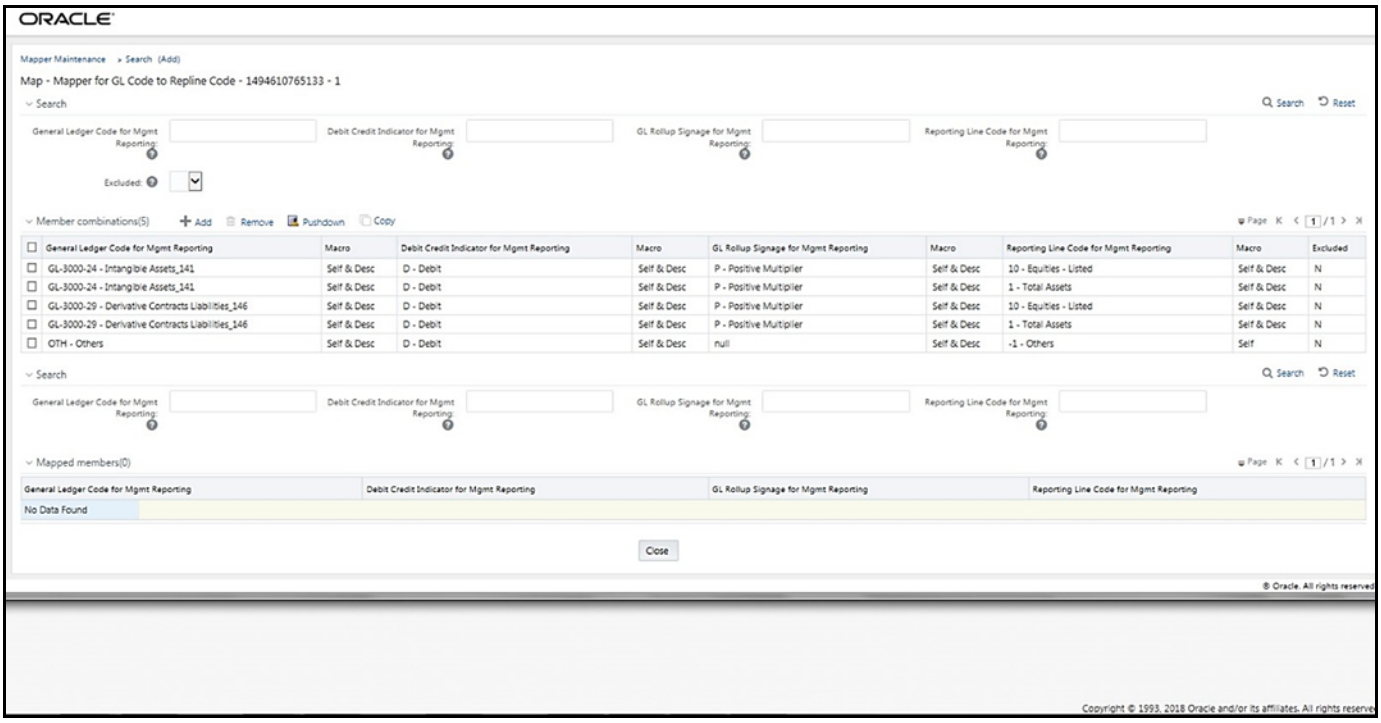
3. To map the members, click **Go**. The list of mapped members appears at the bottom. To save the mappings, click **Save**.

Figure 91: Map the members and save the mappings



4. An acknowledgment pop-up message appears. To confirm saving the mappings, click **Yes**.
5. The mapped member combinations are listed in the **Mapper Maintenance** page.

Figure 92: Added mappings listed in the Mapper Maintenance page



30.4.2.2 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:

- 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 as Parent-Child hierarchy are as follows.

30.4.2.2.1 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.

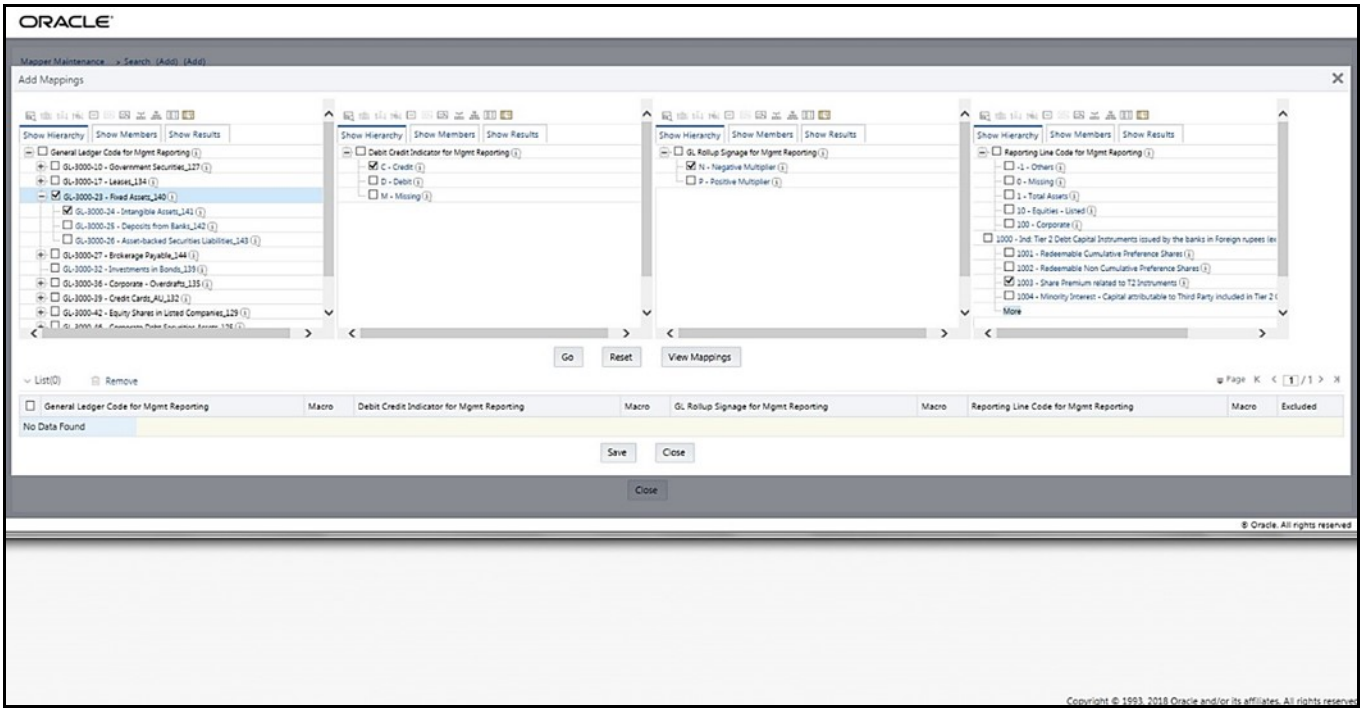
NOTE In the T2T process, in the Stage GL Data table, the aggregation of parent GL Codes and their descendants take place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:

- The value must be M when you map the GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets the value to M.
- The value must be D when you map the GL Code to the Debit member.
- The value must be C when you map the 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:

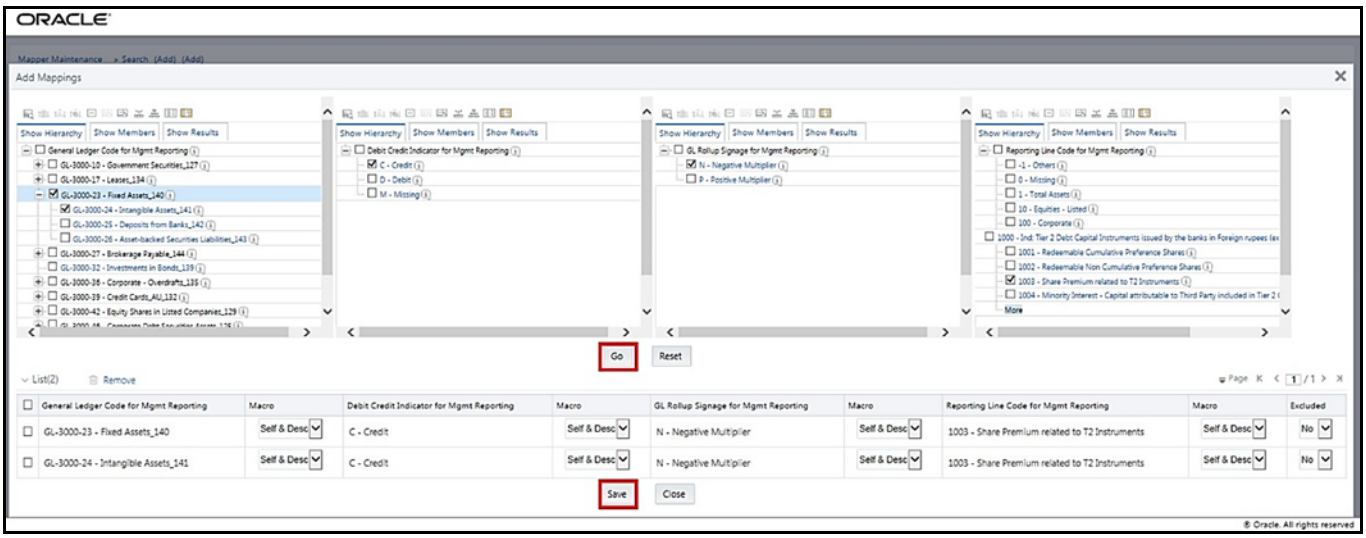
1. In the **Mapper Maintenance** page, in the Member combinations section, click **Add**.
2. The **Add Mappings** page is displayed. For illustration, select one parent member of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-23 – Fixed Assets_140 and its child member GL-3000-24 – Intangible 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 93: Add Mappings page for the one parent to one Reporting Line Code mapping at parent and child hierarchy level



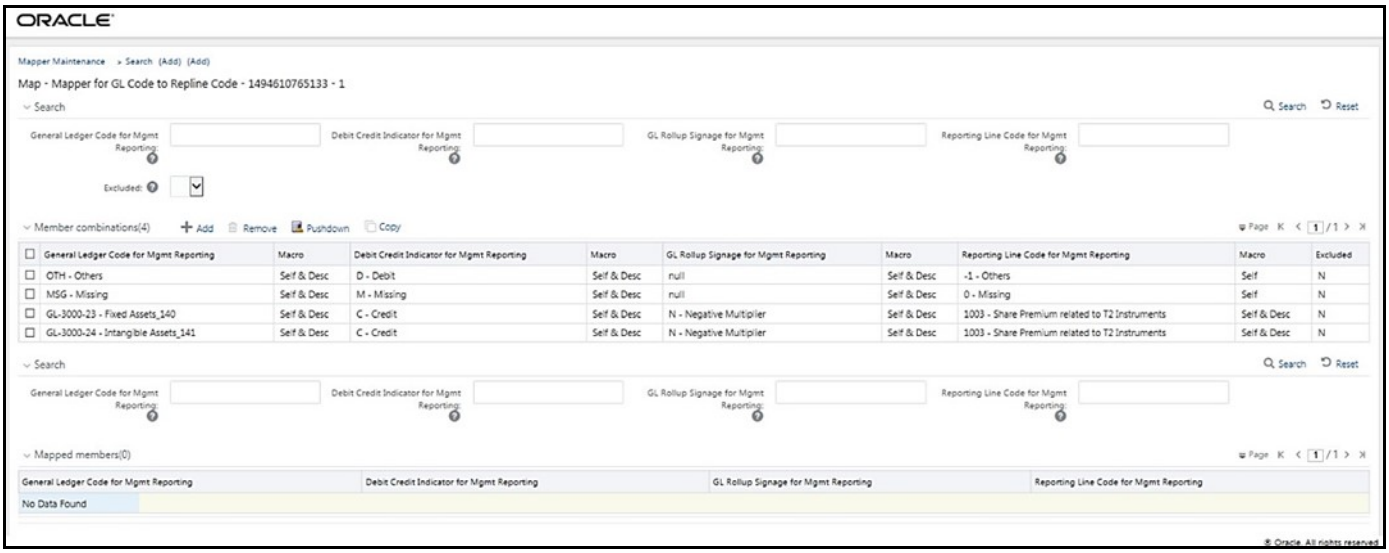
3. To map the members, click **Go**. The list of mapped members appears at the bottom. To save the mappings, click **Save**.

Figure 94: Map the members and save the mappings



4. An acknowledgment pop-up message appears. To confirm saving the mappings, click **Yes**.
5. The mapped member combinations are listed in the **Mapper Maintenance** page.

Figure 95: Added mappings listed in the Mapper Maintenance page



30.4.2.2.2 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

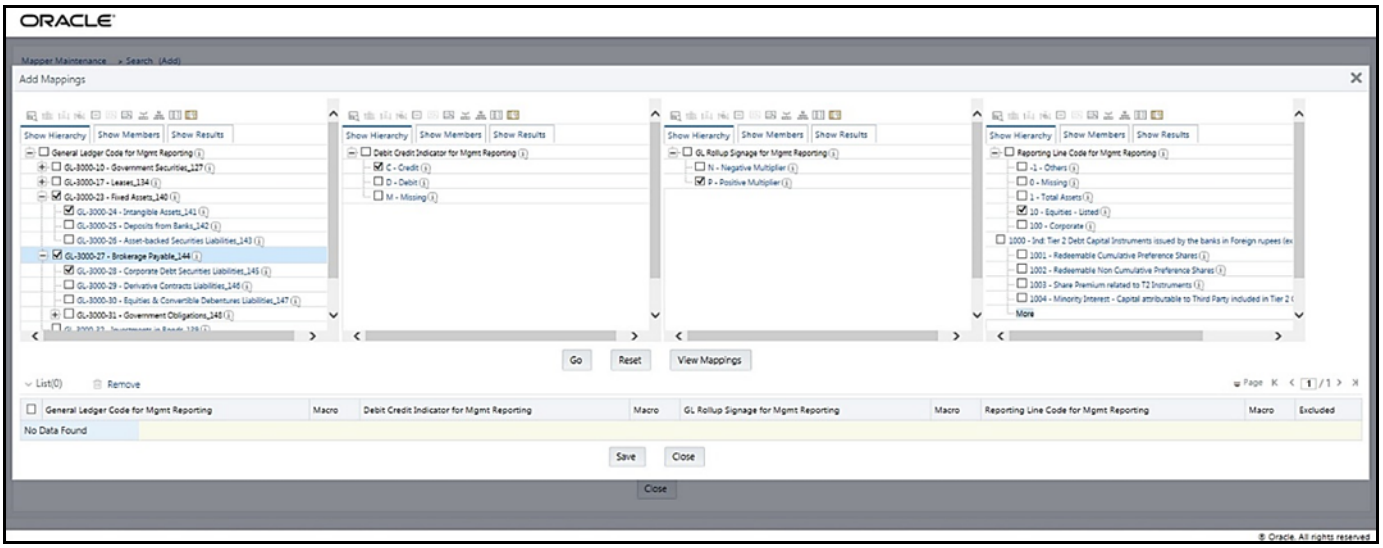
that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:

- The value must be M when you map the GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets the value to M.
- The value must be D when you map the GL Code to the Debit member.
- The value must be C when you map the 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:

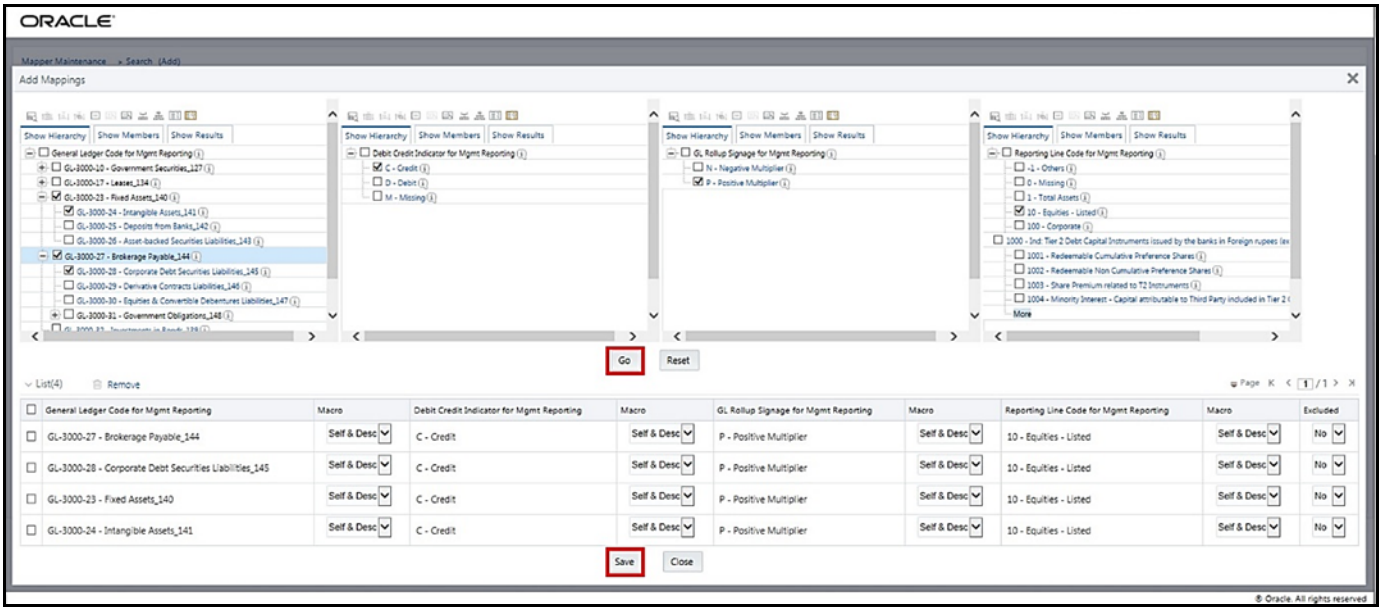
1. In the **Mapper Maintenance** page, in the Member combinations section, click **Add**.
2. The **Add Mappings** page is displayed. For illustration, select parent members of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-23 – Fixed Assets_140 and GL-3000-27 – Brokerage Payable_144, and 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 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 96: Add Mappings page for the many parents to one Reporting Line Code mapping at parent and child hierarchy level



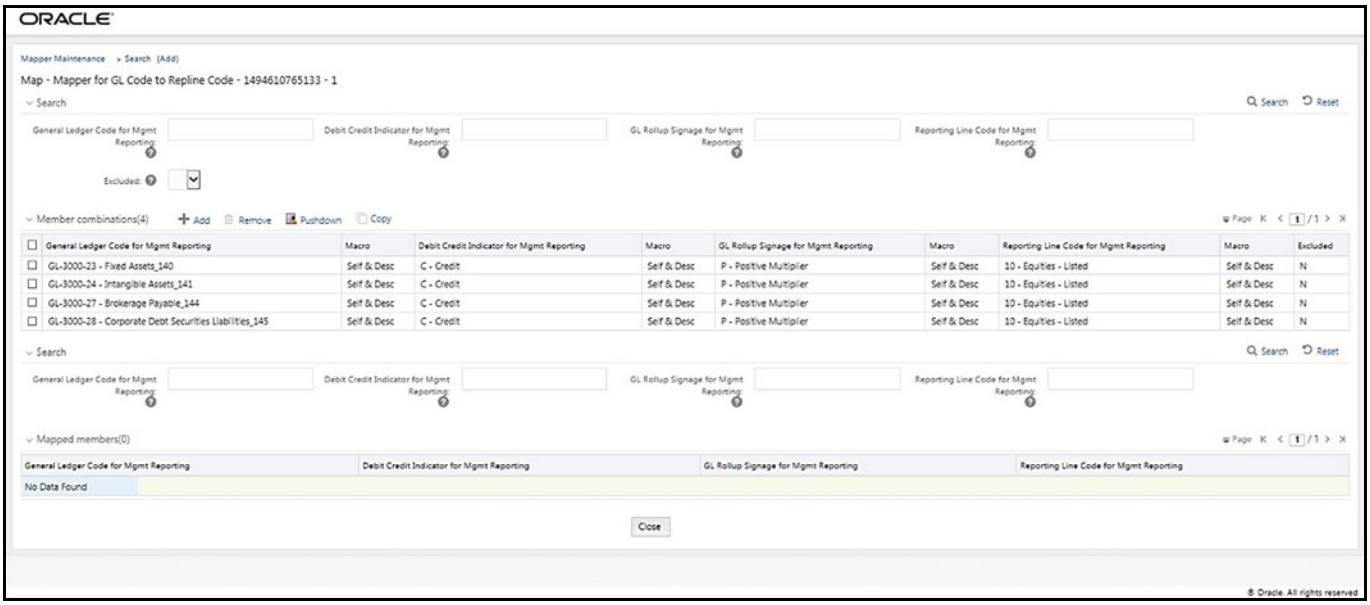
3. To map the members, click **Go**. The list of mapped members appears at the bottom. To save the mappings, click **Save**.

Figure 97: Map the members and save the mappings



4. An acknowledgment pop-up message appears. To confirm saving the mappings, click **Yes**.
5. The mapped member combinations are listed in the **Mapper Maintenance** page.

Figure 98: Added mappings listed in the Mapper Maintenance page



30.4.2.2.3 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.

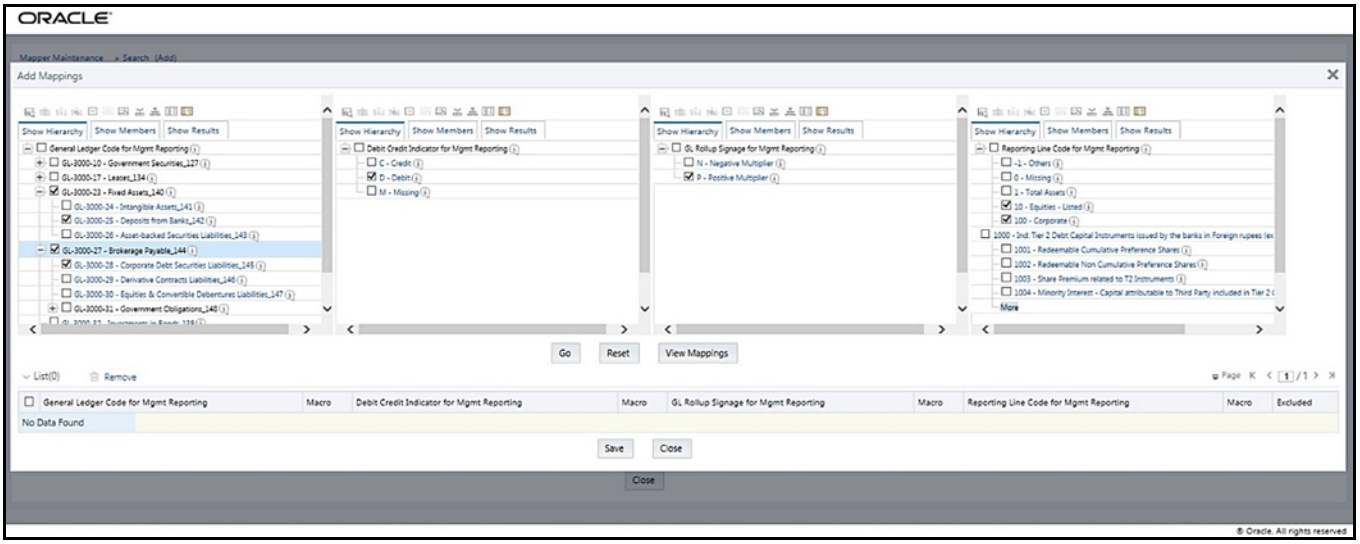
NOTE In the T2T process, in the Stage GL Data table, the aggregation of parent GL Codes and their descendants take place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:

- The value must be M when you map the GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets the value to M.
- The value must be D when you map the GL Code to the Debit member.
- The value must be C when you map the 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:

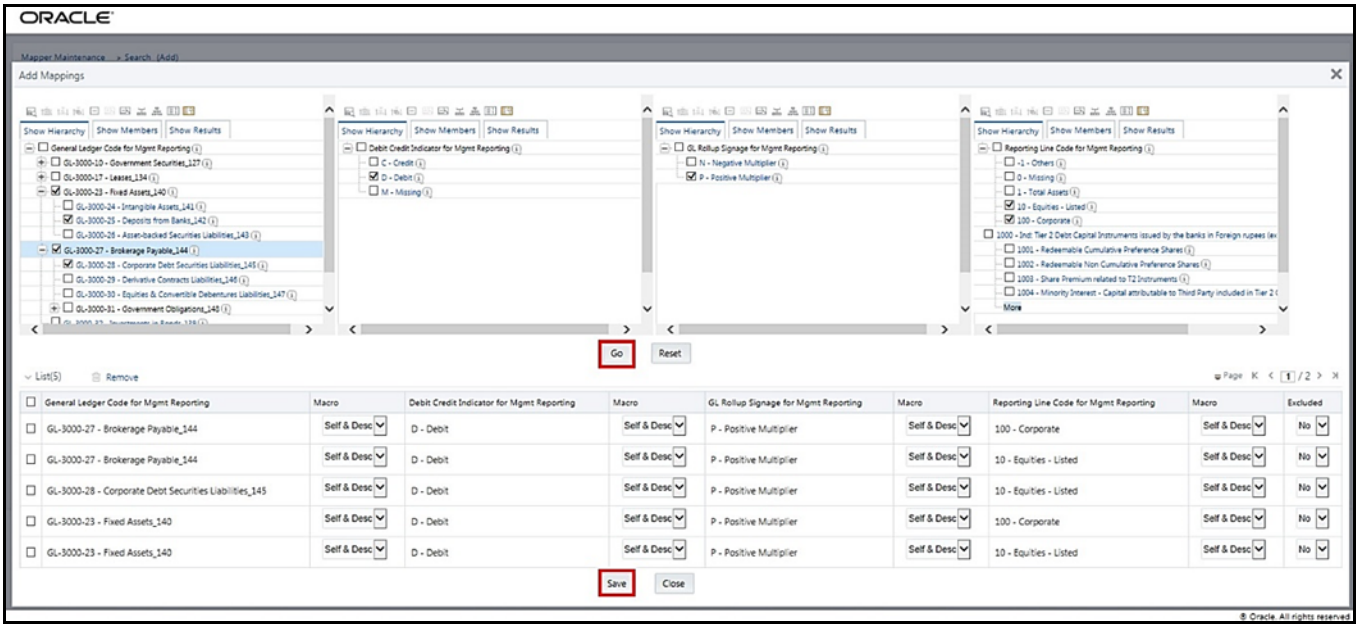
1. In the **Mapper Maintenance** page, in the Member combinations section, click **Add**.
2. The **Add Mappings** page is displayed. For illustration, select parent members of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-23 – Fixed Assets_140 and GL-3000-27 – Brokerage Payable_144, and 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 99: Add Mappings page for the many parents to many Reporting Line Codes mapping at parent and child hierarchy level



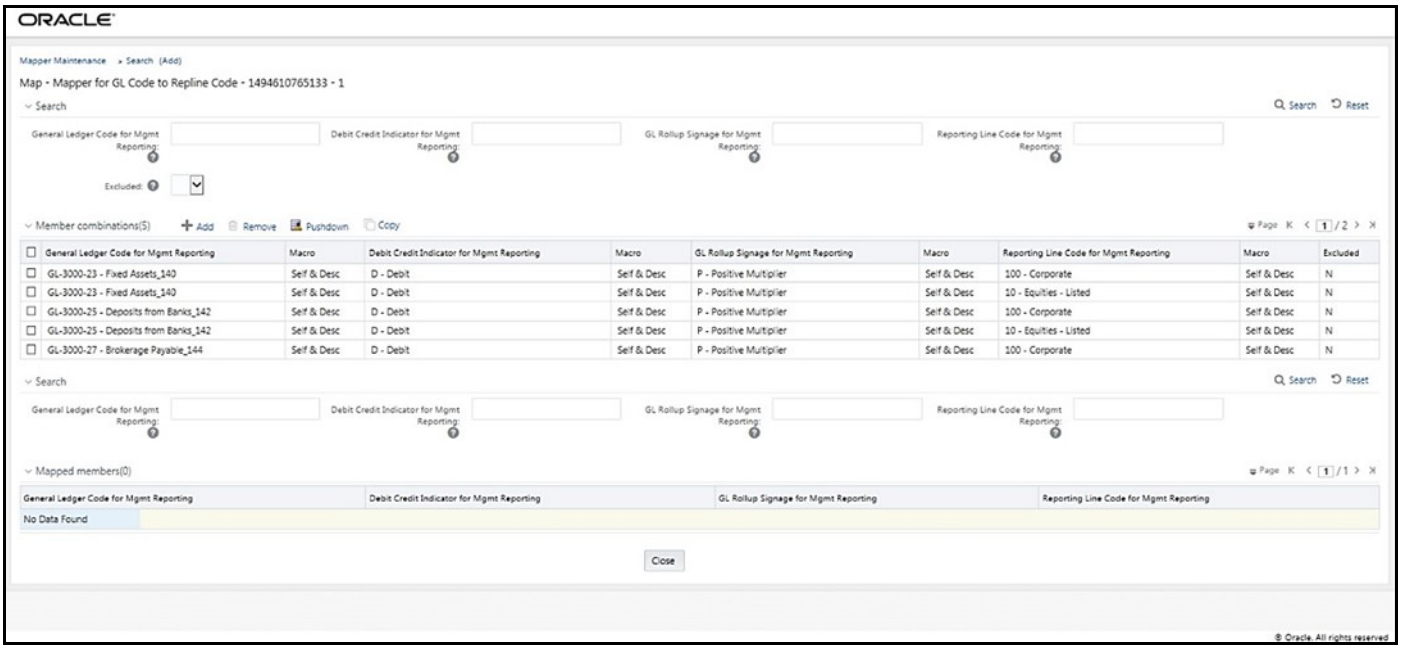
3. To map the members, click **Go**. The list of mapped members appears at the bottom. To save the mappings, click **Save**.

Figure 100: Map the members and save the mappings



4. An acknowledgment pop-up message appears. To confirm saving the mappings, click **Yes**.
5. The mapped member combinations are listed in the **Mapper Maintenance** page.

Figure 101: Added mappings listed in the Mapper Maintenance page



30.4.2.3 Mapping Combinations at Parent Hierarchy Level without Descendants

This section explains the mapping combinations for the General Ledger Code for the 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

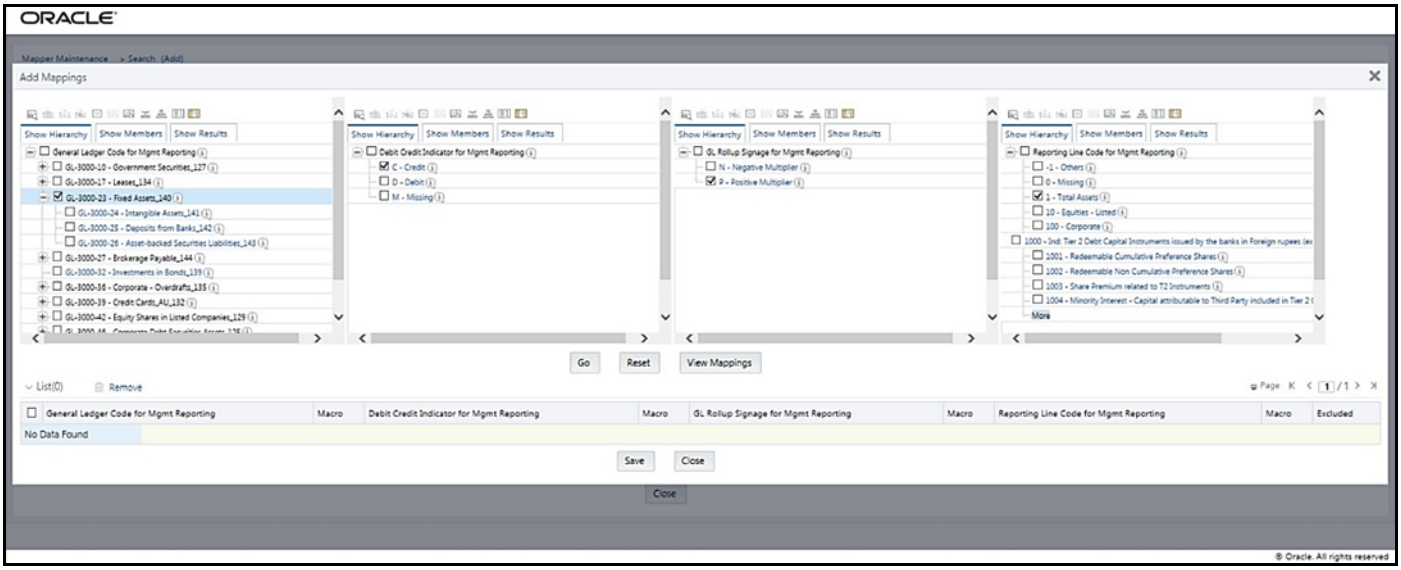
In the T2T process, in the Stage GL Data table, the aggregation of parent GL Code takes place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:

- The value must be M when you map the GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets the value to M.
- The value must be D when you map the GL Code to the Debit member.
- The value must be C when you map the GL Code to the Credit member.

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

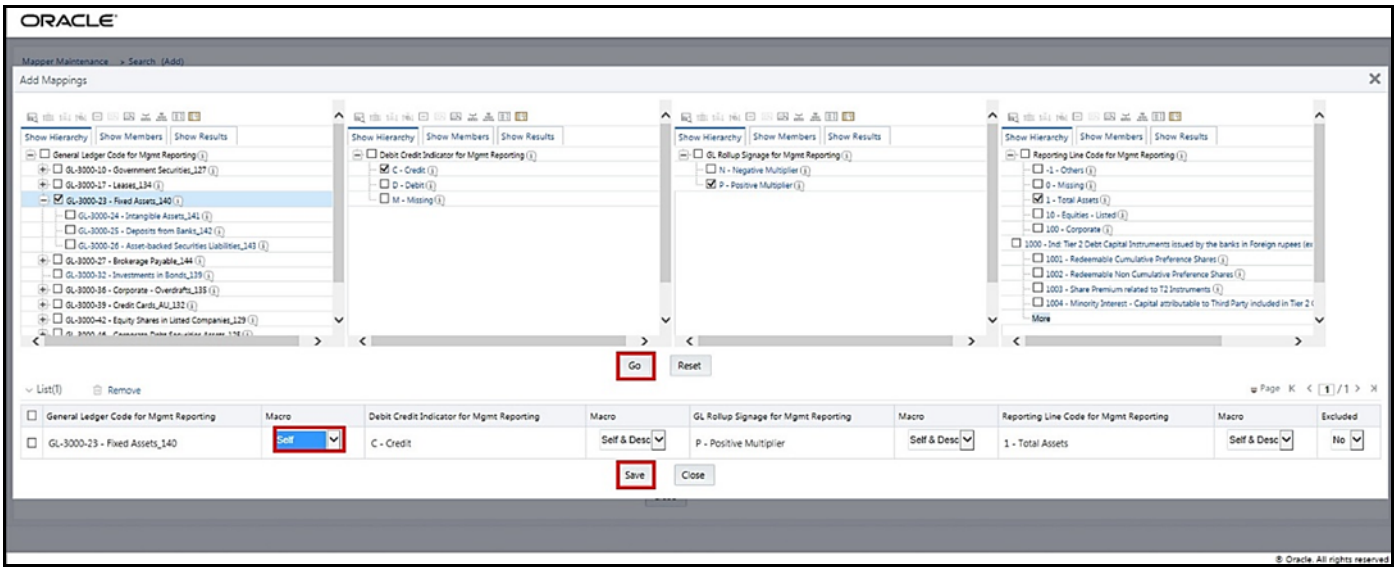
1. In the **Mapper Maintenance** page, in the Member combinations section, click **Add**.
2. The **Add Mappings** page is displayed. For illustration, select one parent member of the General Ledger Code for Mgmt Reporting hierarchy GL-3000-23 – Fixed Assets_140, and the member of the Debit Credit Indicator 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 102: Add Mappings page for the one parent to one or many Reporting Line Codes mapping without Descendants



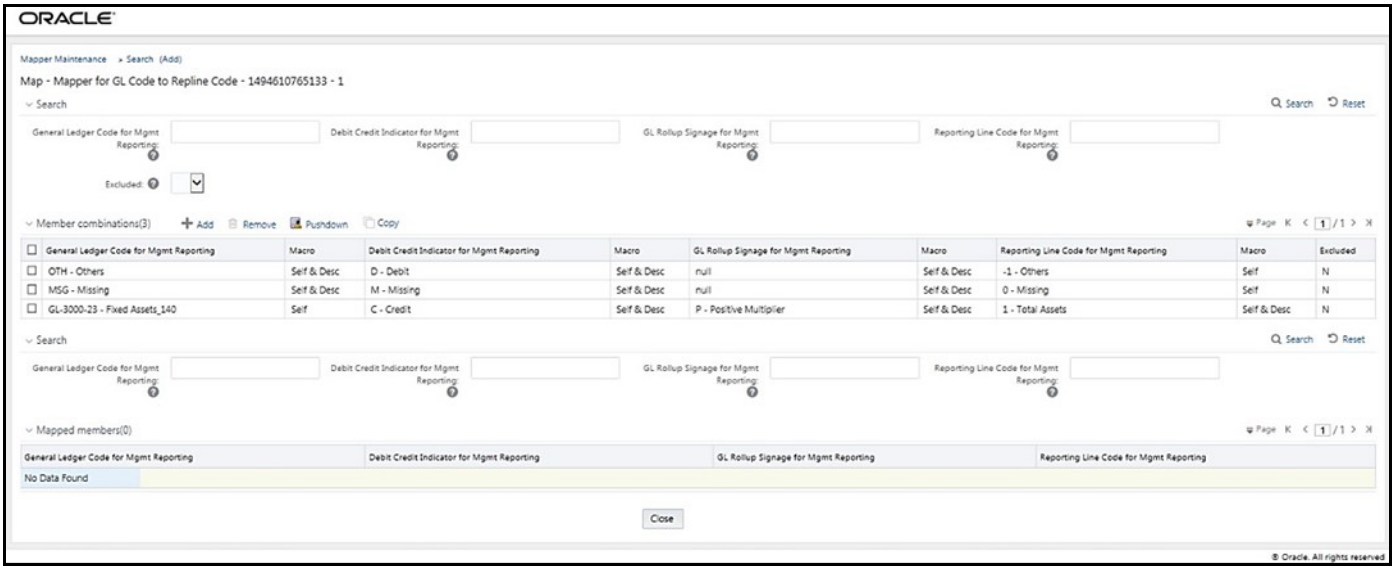
3. To map the members, click **Go**. The list of mapped members appears at the bottom. To exclude the Descendants of the GL Code from the mapping, in the Macro column, select **Self**. To save the mappings, click **Save**.

Figure 103: Map the members and save the mappings by excluding Descendants



4. An acknowledgment pop-up message appears. 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 104: Added mappings listed in the Mapper Maintenance page



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

This section explains the mapping combinations for the General Ledger Code for the 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.

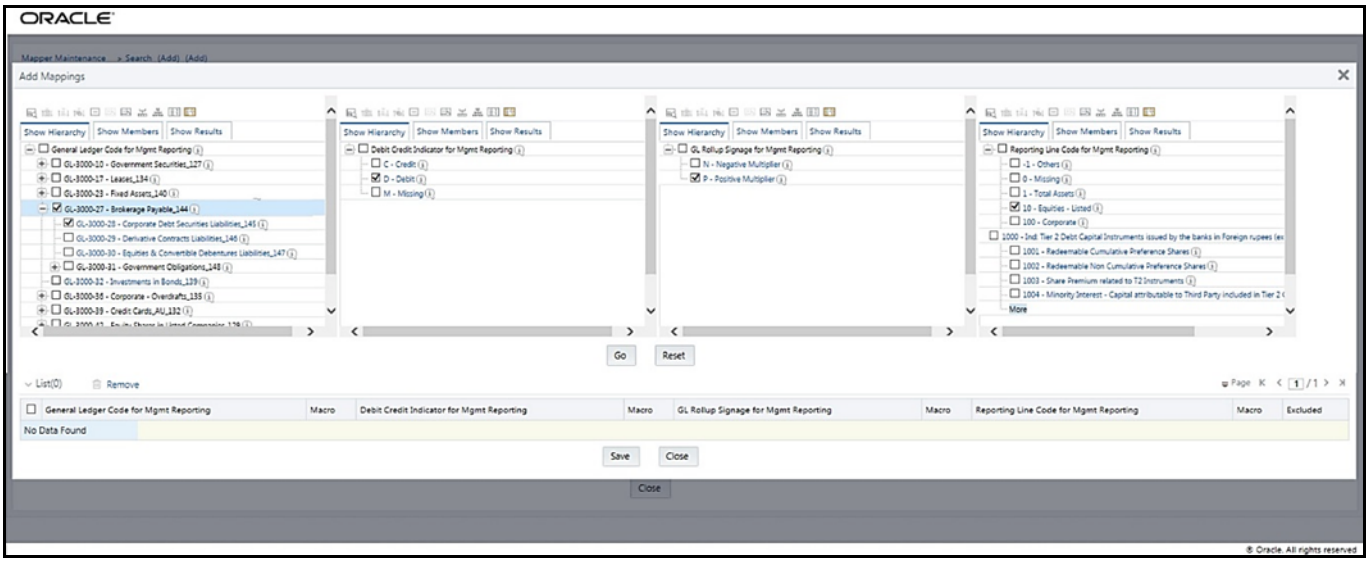
NOTE In the T2T process, in the Stage GL Data table, the aggregation of parent GL Code and its descendants take place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:

- The value must be M when you map the GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets the value to M.
- The value must be D when you map the GL Code to the Debit member.
- The value must be C when you map the 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:

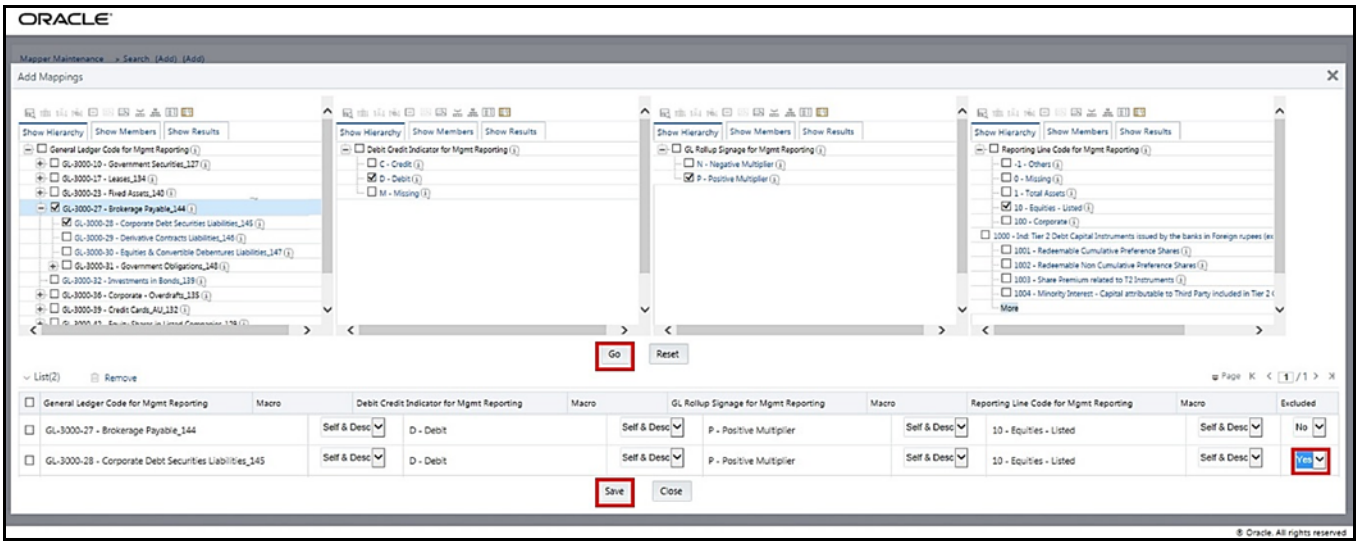
1. In the **Mapper Maintenance** page, in the Member combinations section, click **Add**.
2. The **Add Mappings** page is displayed. For illustration, select one parent member of the General Ledger Code for Mgmt Reporting hierarchy GL-3000-27 – Brokerage Payable_144 and its child member (Descendant) 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 105: Add Mappings page for the mapping combinations at parent hierarchy level by removing one or more Descendants



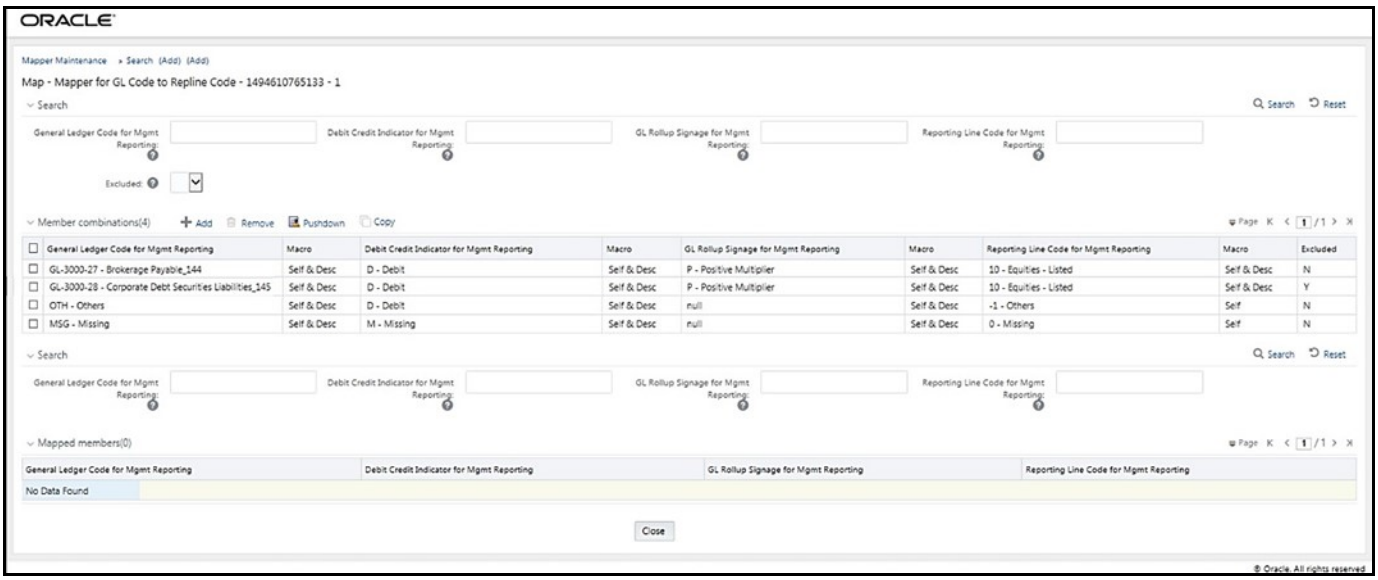
3. To map the members, click **Go**. The list of mapped members appears at the bottom. To exclude the Descendant of the GL Code from the mapping, in the Macro column, select Self & Descendants, and in the Excluded column, select **Yes**. To save the mappings, click **Save**.

Figure 106: Map the members and save the mappings by excluding Descendants



4. An acknowledgment pop-up 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 107: Mapped member combinations are listed in the Mapper Maintenance page

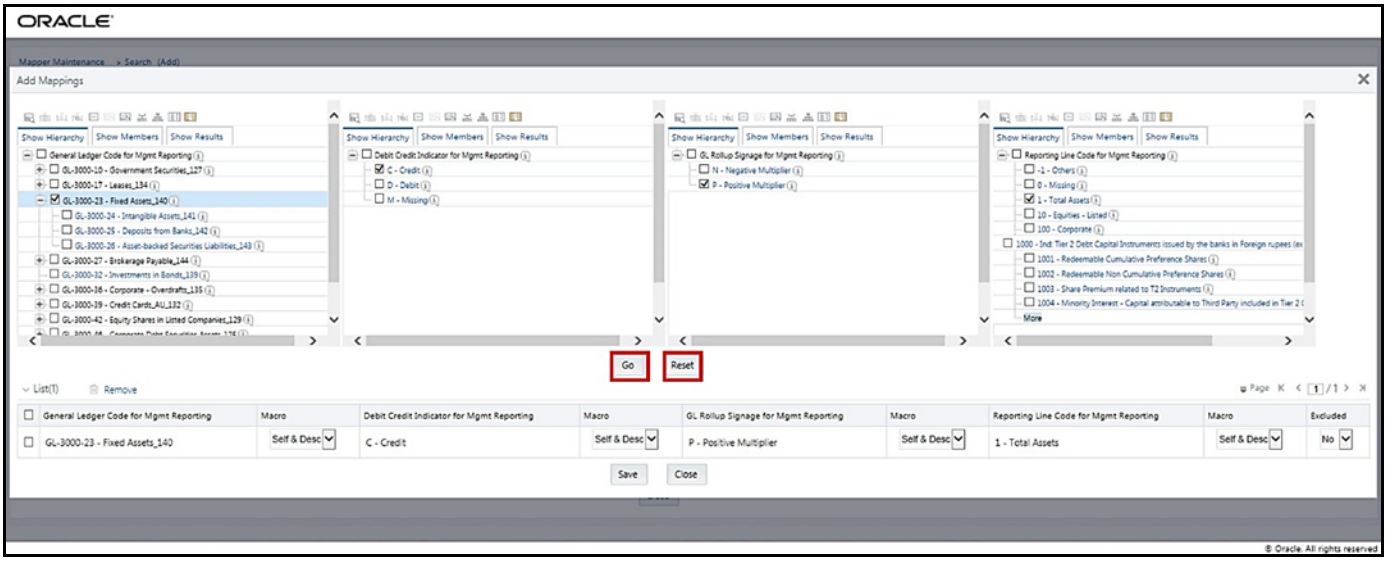


30.4.3 Performing Multiple Sets of Mapping Combinations

To perform mapping more than one time in the **Add Mappings** page, use the **Reset** functionality. To perform this procedure:

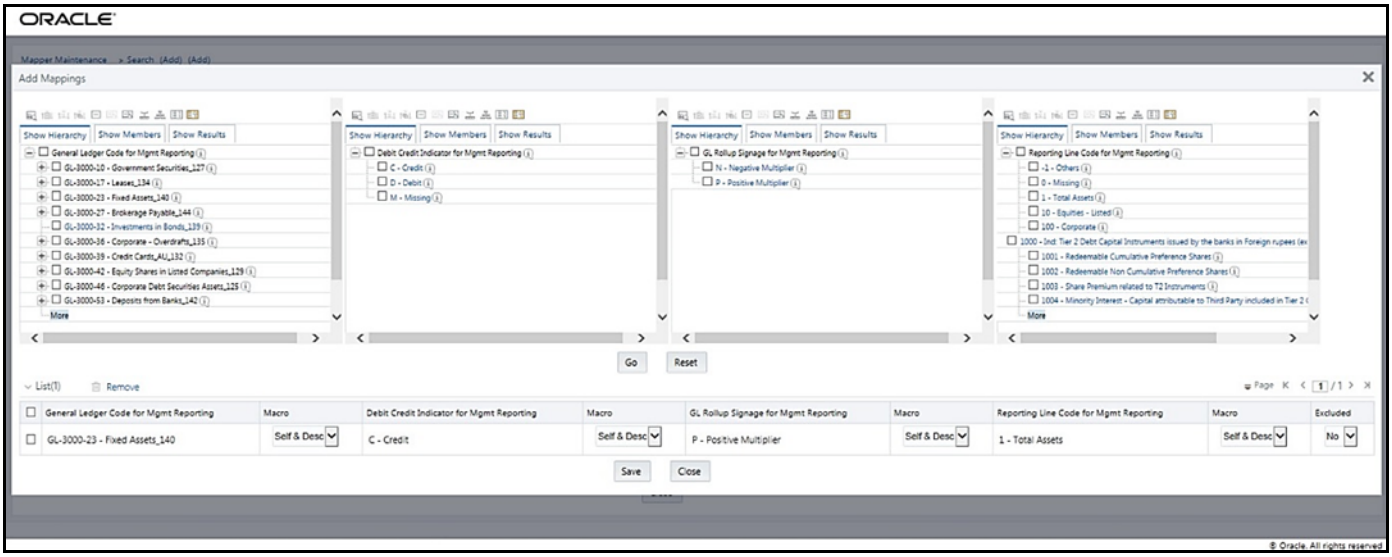
1. In the **Mapper Maintenance** page, in the Member combinations section, click **Add**.
2. The **Add Mappings** page is displayed. For illustration, select one parent member of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-23 – Fixed Assets_140, and the member of the Debit Credit Indicator 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 appears at the bottom. To initiate mapping of another set of member combinations, click **Reset**.

Figure 108: Add Mappings page for adding mappings for the first set of members and select Reset



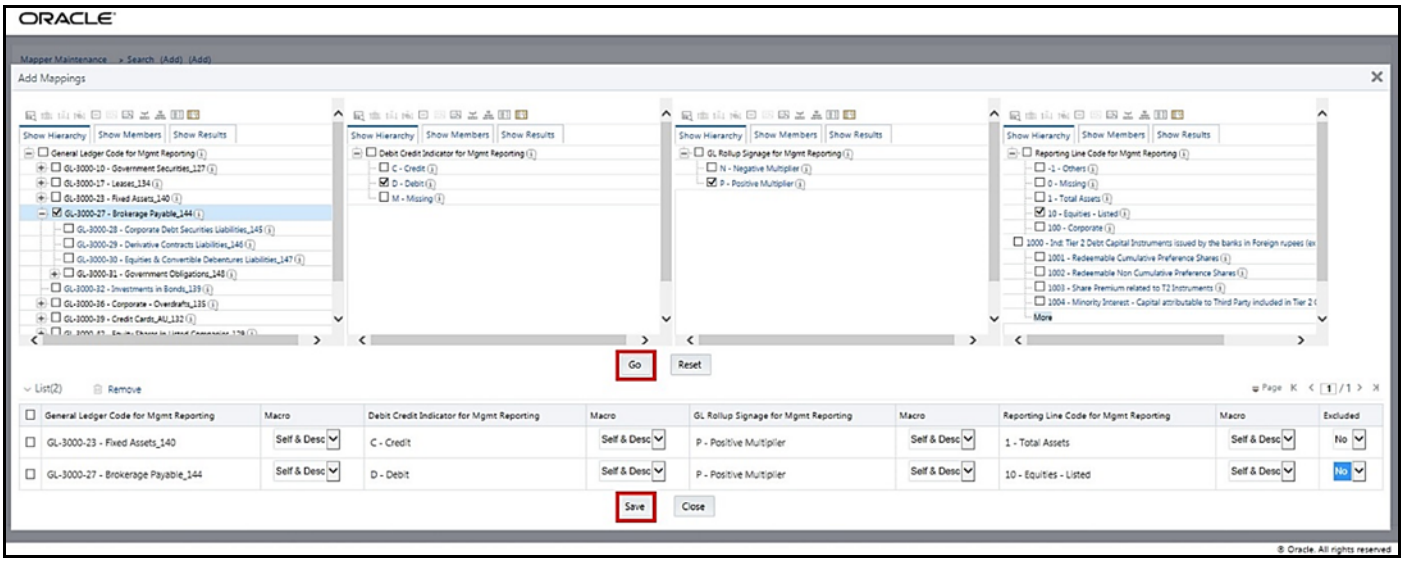
3. The mapping selections clear.

Figure 109: Mapping selections clear up



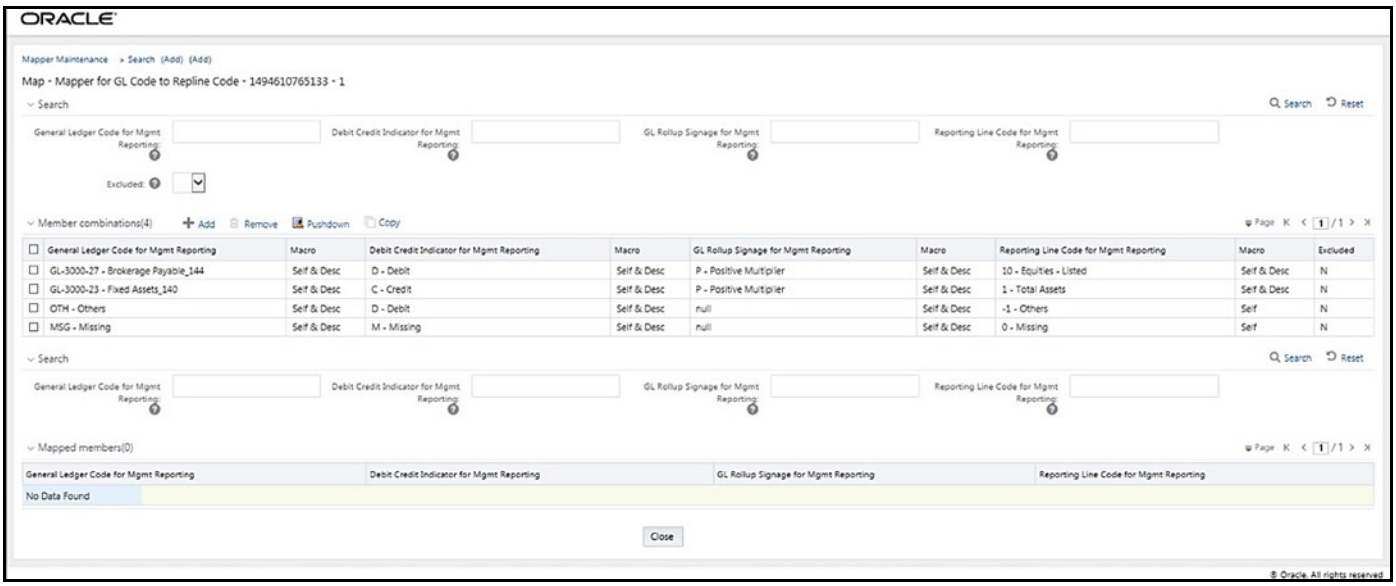
4. For illustration, selected a parent member of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-27 – Brokerage Payable_144, and the member of the Debit Credit Indicator for Mgmt Reporting hierarchy D - Debit to map to the member of the Reporting Line Code for Mgmt Reporting hierarchy 10 – 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 110: Add Mappings page for adding mappings for the next set of members



5. An acknowledgment pop-up message appears. To confirm saving the mappings, click **Yes**.
6. Both set of mapped member combinations are listed on the **Mapper Maintenance** page.

Figure 111: Mapped member combinations are listed in the Mapper Maintenance page



30.5 Loading 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 194610765133, 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 must be C or D or M).
- V_MEMBER_3 = Reporting Line Code (values from DIM_REP_LINE.N_REP_LINE_CD).

Figure 112: Loading the Mapper Maintenance from backend

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

NOTE These values are real business key columns and not display codes.

30.6 Deploying GL Data and GL to Management Reporting Result Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

30.7 Populating GL Data and GL to Management Reporting T2T Result Tables

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

30.8 Related Topics

You can see the following topics related to other function-specific tables:

- Insurance Contracts Tables
- Insurance Claims Tables
- Transaction Summary Tables

31 Insurance Claims Tables

This chapter details the Insurance Claims tables in the Oracle Insurance Data Foundation application.

Topics:

For the metadata design, see the *Insurance Claims Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

31.1 Deploying Insurance Claims Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

> **NOTE** In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

31.2 Populating Insurance Claims Dimension Tables

Follow this SCD process to populate data into a Dimension table:

> **NOTE** You can also follow this SCD process to populate data into any Hive-related Dimension table.

1. To populate data into a Dimension table, execute the SCD batch. For a detailed procedure, see the Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Dimension table, follow the procedure Check the Execution Status of the SCD Batch.
3. To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

31.3 Populating Insurance Claims T2Ts (Result Tables)

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

31.4 Related Topics

You can see the following topics related to other function-specific tables:

- Insurance Underwriting Entities
- Insurance Contracts Tables
- General Ledger Data and Management Reporting Tables
- Insurance Claims Tables
- Transaction Summary Tables

32 Actuarial Assumptions Tables

This chapter provides information about Insurance Actuarial Policy Summary, Mortality and Morbidity tables, and several Insurance Lapse Rate Assumptions Result tables in the Oracle Insurance Data Foundation Application.

Topics:

An actuarial assumption is an estimate of an uncertain variable input into a financial model, normally for the purposes of calculating premiums or benefits.

For the metadata design, see the following Sections in the OIDF Metadata Design Document Release 8.1.x:

- Actuarial Assumptions Dimension Tables
- Financial Assumptions Tables
- Demographic Assumptions Tables

32.1 Deploying Actuarial Assumptions Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

> **NOTE** In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

32.2 Populating Actuarial Assumptions Dimension Tables

Follow this SCD process to populate data into a Dimension table:

> **NOTE** You can also follow this SCD process to populate data into any Hive-related Dimension table.

1. To populate data into a Dimension table, execute the SCD batch. For a detailed procedure, see the Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Dimension table, follow the procedure Check the Execution Status of the SCD Batch.

3. To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

32.3 Populating Actuarial Assumptions T2T Result Tables

Follow this T2T process to populate data into any T2T Result table:

> **NOTE** Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

32.4 Related Topics

You can see the following topics related to other function-specific tables:

- Insurance Contracts Tables

33 Actuarial Output Tables

This section provides information about Actuarial Outputs such as Actuarial Cash Flows, Calculations, and Result Areas in the Oracle Insurance Data Foundation application.

Topics:

For the metadata design, see the following Sections in the OIDF Metadata Design Document Release 8.1.x.

- About Actuarial Assumed Policy Summary
- About Actuarial Cash Flows, Calculations, and Result Areas Summary Tables

33.1 Deploying Actuarial Cash Flows, Calculations, and Result Areas Summary Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

33.2 Populating Actuarial Cash Flows, Calculations, and Result Areas Summary Dimension Tables

Follow this SCD process to populate data into a Dimension table:

NOTE You can also follow this SCD process to populate data into any Hive-related Dimension table.

1. To populate data into a Dimension table, execute the SCD batch. For a detailed procedure, see the Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Dimension table, follow the procedure Check the Execution Status of the SCD Batch.
3. To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

33.3 Populating Actuarial Cash Flows, Calculations, and Result Areas Summary T2T Result Tables

Follow this T2T process to populate data into any T2T Result table:

> **NOTE** Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

33.4 Related Topics

You can see the following topics related to other function-specific tables:

- Insurance Contracts Tables
- Transaction Summary Tables

34 Common Customer Summary Tables

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

Topics:

Common Customer Summary table stores attribute pertaining to customer-related data on an 'as-is' basis received from the source system. Customer balances are derived from the account summary. The customer relationship table derives the relationship between accounts and customers. Common customer summary data is populated for all the active customers in the customer dimension.

For the metadata design, see the following Sections in the OIDF Metadata Design Document Release 8.1.x.

- About Dimension Tables That Load Common Customer Summary T2Ts
- About T2Ts (Result Tables) that Load Common Customer Summary T2Ts
- About Common Customer Summary T2T (Result Table)

34.1 Deploying Common Customer Summary Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

> **NOTE** In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

34.2 Populating Data in the Common Customer Summary T2T Result Tables

Follow these steps to populate data in the Common Customer Summary T2T Result tables:

1. Prerequisites.
2. Populating Common Customer Summary T2T Result Tables.

34.2.1 Prerequisites

To load the resultant tables required for Common Customer Summary T2T, follow these steps:

1. Populating Dimension Tables to Load Common Customer Summary T2T.
2. Populating T2Ts to Load Common Customer Summary T2T.

3. Populating Other Tables to Load Common Customer Summary T2T.

34.2.1.1 Populating Dimension Tables to Load Common Customer Summary T2T

For Dimension mapping details for this section, see About Dimension Tables That Load Common Customer Summary T2Ts. Follow this SCD process to populate data into any Dimension tables that are used to load Common Customer Summary T2Ts:

> **NOTE** You can also follow this SCD process to populate data into any Hive-related Dimension table.

1. To populate data into a Dimension table, execute the SCD batch. For a detailed procedure, see the Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Dimension table, follow the procedure Check the Execution Status of the SCD Batch.
3. To verify log files, and check the error messages (if any), follow the procedure Verify Log Files and Check Error Messages.

34.2.1.2 Populating T2Ts to Load Common Customer Summary T2T

For T2T mapping details for this section, see About T2Ts (Result Tables) that Load Common Customer Summary T2Ts.

Follow this T2T process to populate data into any T2T Result table:

> **NOTE** Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

34.2.1.3 Populating Other Tables to Load Common Customer Summary T2T

Load these tables with data:

- DIM_BANDS

  To follow the Dimension loading process, see Populating Dimension Tables to Load Common Customer Summary T2T.

- DIM_GENDER

To follow the Dimension loading process, see Populating Dimension Tables to Load Common Customer Summary T2T.

- FCT_COMMON_ACCOUNT_SUMMARY

  For mapping details, see Other Common Account Summary Tables and About Common Policy Summary T2Ts (Result Tables).

  To follow the T2T process, see Populating T2Ts to Load Common Customer Summary T2T.

34.2.2 Populating Common Customer Summary T2T Result Tables

Follow T2T process (Populating T2Ts to Load Common Customer Summary T2T) to populate data into any Common Customer Summary T2T Result table:

> **NOTE** You can also follow this T2T process to populate data into any Hive related Common Customer Summary T2T Result table.

For T2T mapping details for this section, see About Common Customer Summary T2T (Result Table).

34.3 Related Topics

You can see the following topics related to other function-specific tables:

- Party Subject Area

35 Credit Score Model and Probability of Default Model Tables

This section provides information about the Credit Score Model and Probability of Default Model tables in the Data Foundation application and step-by-step instructions to use this section.

Topics:

For the metadata design, see the following Sections in the OIDF Metadata Design Document Release 8.1.x.

- About Credit Score Model and Probability of Default Model T2Ts (Result Tables)
- Staging Data Expectation for Credit Score Model

35.1 About Mapper for Credit Score Model to Regulatory Credit Score Model

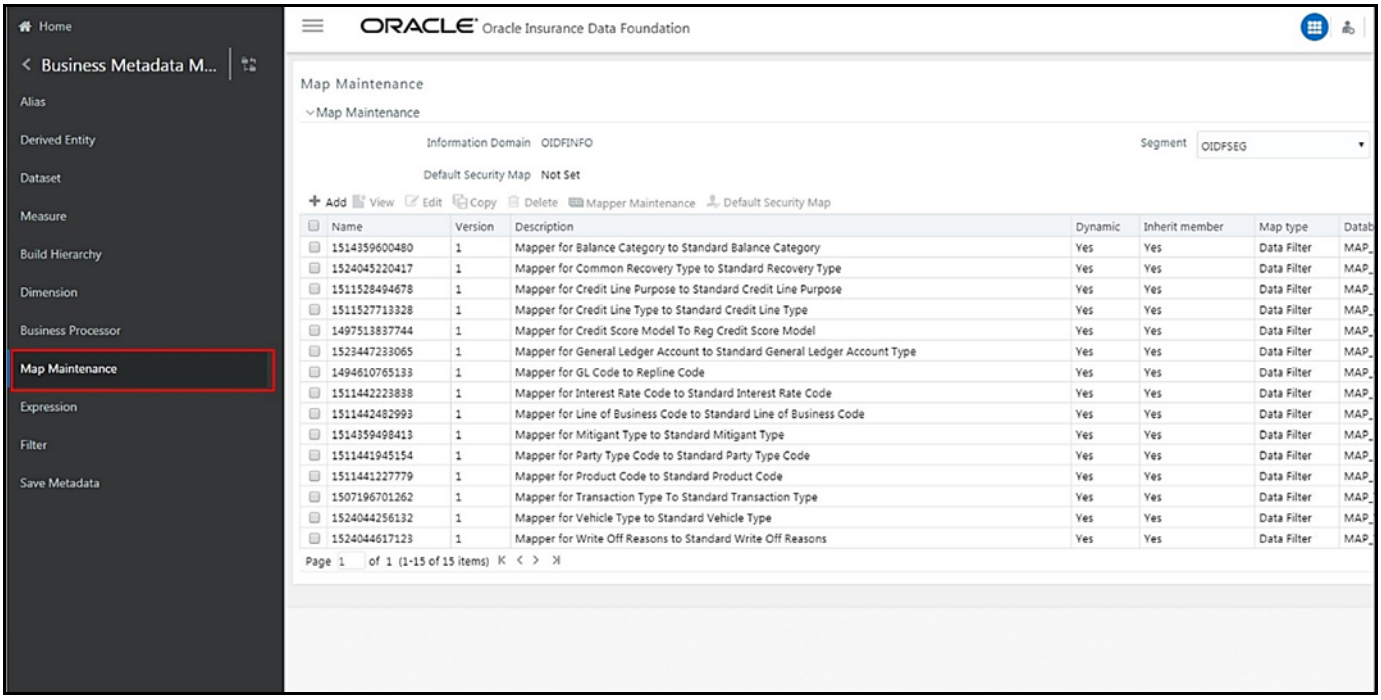
Credit Score Model dimension table is a slowly changing dimension (SCD), which has user-specific values for available credit score models. Regulatory Credit Score Model dimension table is a seeded dimension, which has the reporting specific values for regulatory reporting requirements. There are one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

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

To maintain the Mapper for Credit Score Model to Regulatory Credit Score Model, perform these steps.

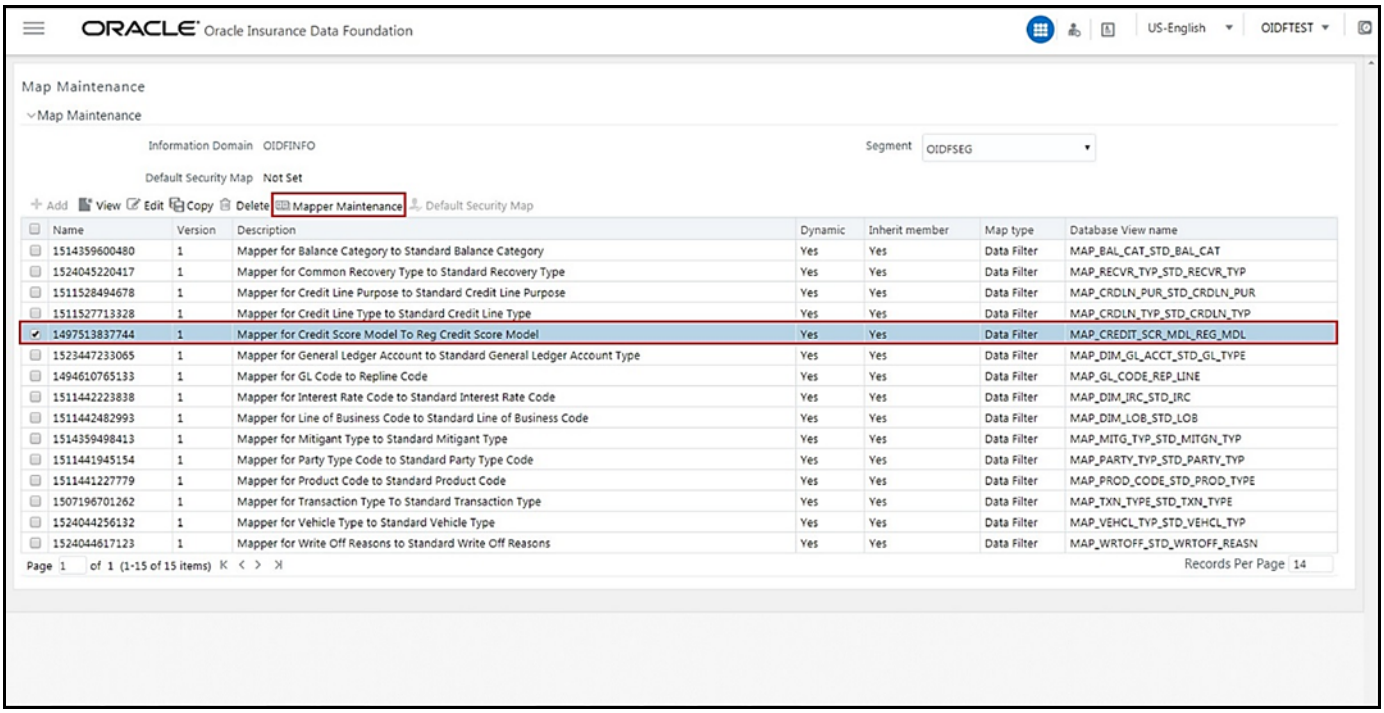
1. From **OFSAA Home**, select **Oracle Insurance Data Foundation**, select **Unified Analytical Metadata**, select **Business Metadata Management**, and then select **Map Maintenance**.

Figure 113: Navigate to the Map Maintenance page



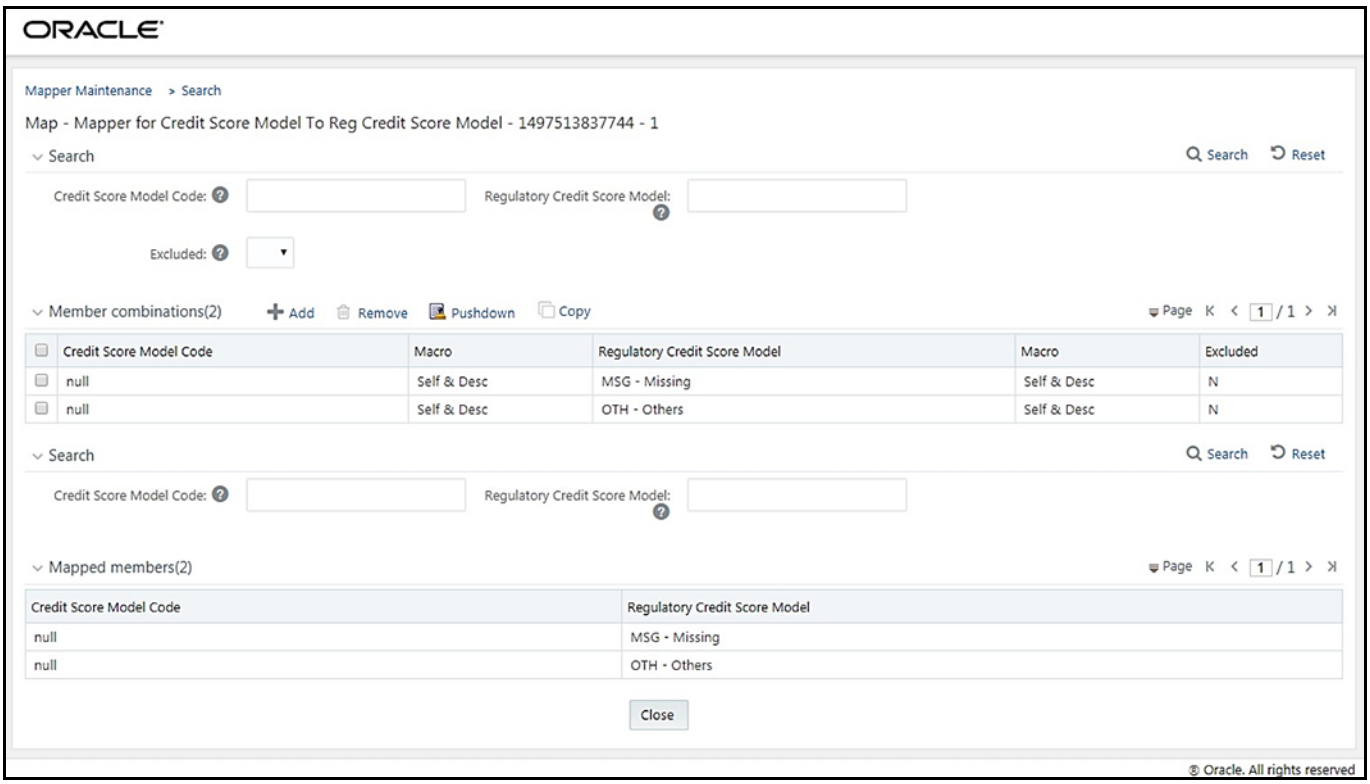
2. The **Map Maintenance** page appears. Select Mapper for Credit Score Model to Regulatory Credit Score Model. Click the Mapper Maintenance icon.

Figure 114: Select the required Mapper and click Mapper Maintenance



3. The **Mapper Maintenance** page is displayed. OIDF 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 115: The Mapper Maintenance page



Topics:

- Prerequisites for Mapper Maintenance
- Possible Mapping Combinations

35.2.1 Prerequisites for Mapper Maintenance

1. Load Credit Score Model Dimension using SCD.
2. Resave the following hierarchies:
   - HSCRMDL1 - Credit Score Model Code
   - HSCRMDL2 - Regulatory Credit Score Model

35.2.2 Possible Mapping Combinations

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

Topics:

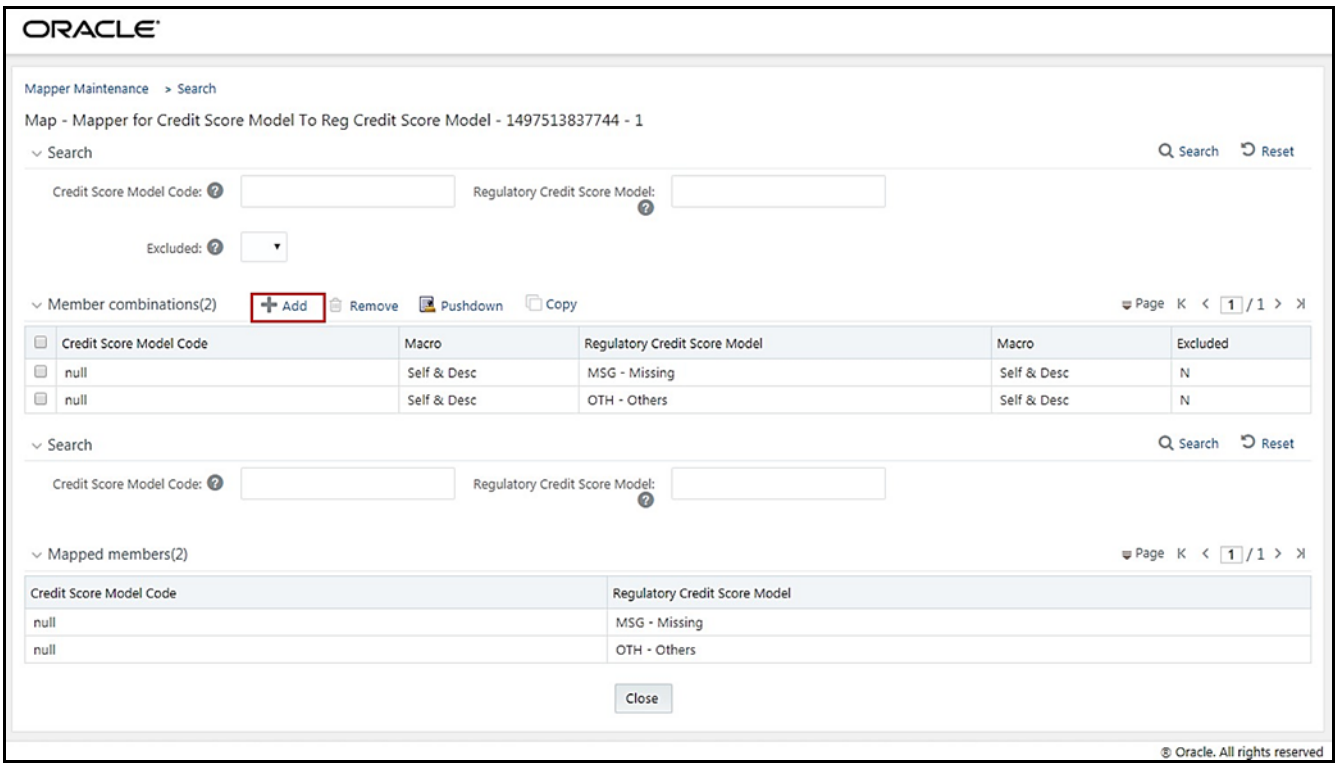
- One-to-One Mapping
- Many-to-One Mapping

35.2.2.1 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 the Fact Account Credit Score Details Regulatory Credit Score Model column while loading the T2T.

1. In the **Mapper Maintenance** page, click **Add**.

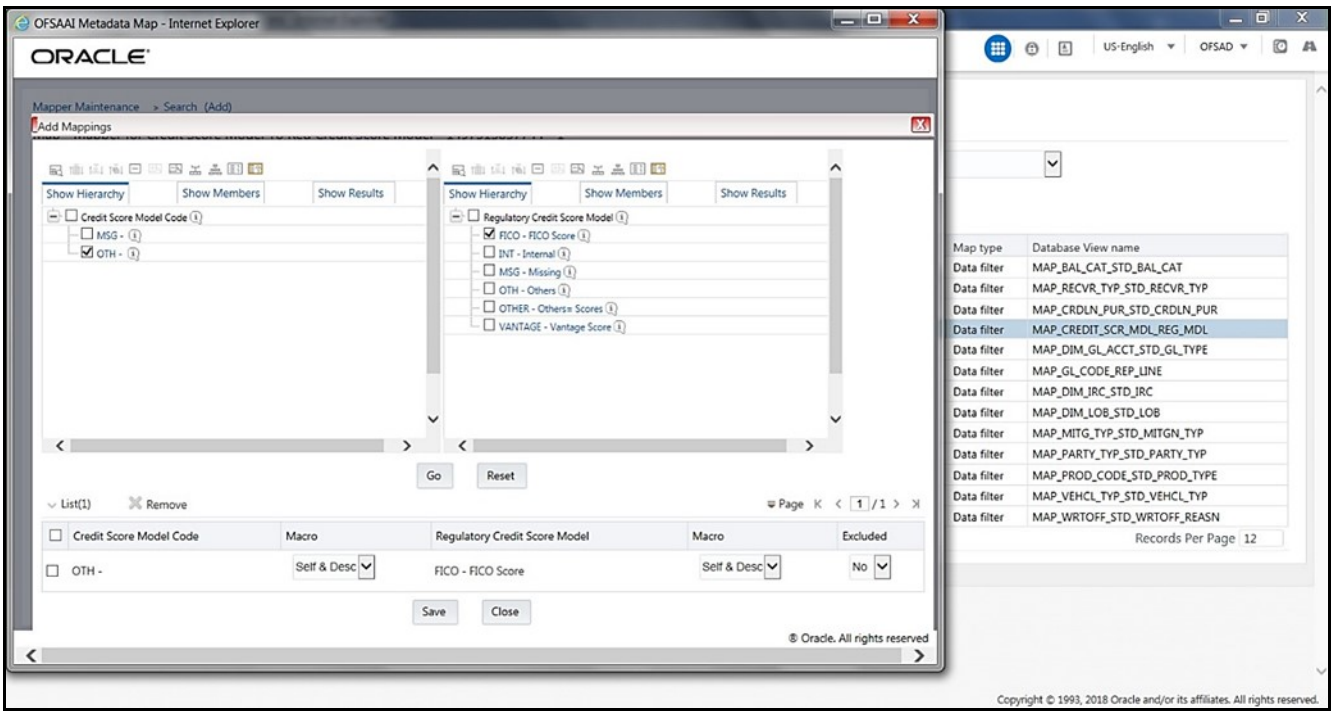
Figure 116: Select Add in the Mapper Maintenance page



2. The **Add Mappings** page opens.

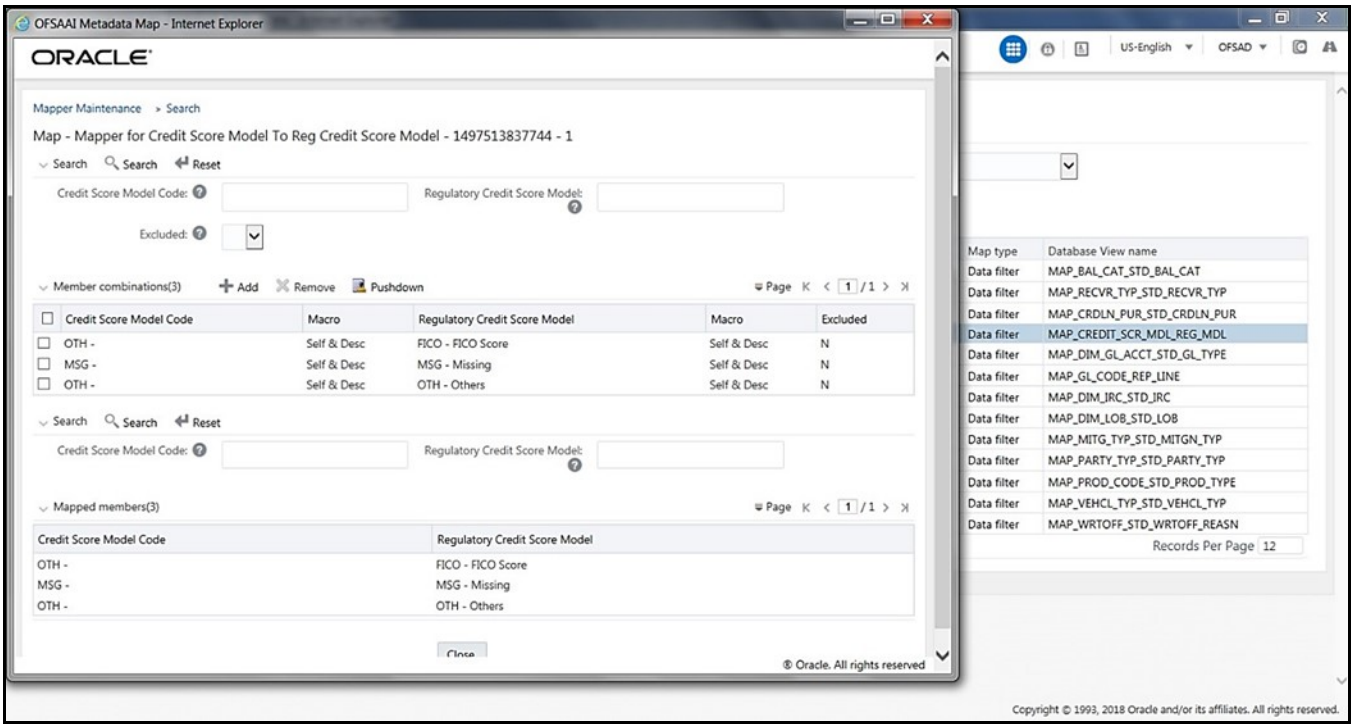
   In this example, OTH - Others are mapped to FICO - FICO Score. To map, click **Go** and then to save the mapping, click **Save**.

Figure 117: Add Mappings page for the One-to-One Mapping



3. An acknowledgment is displayed. Click **Yes** to confirm.
4. The Mapped Members are displayed in the **Mapper Maintenance** page.

Figure 118: Added mappings listed in the Mapper Maintenance page

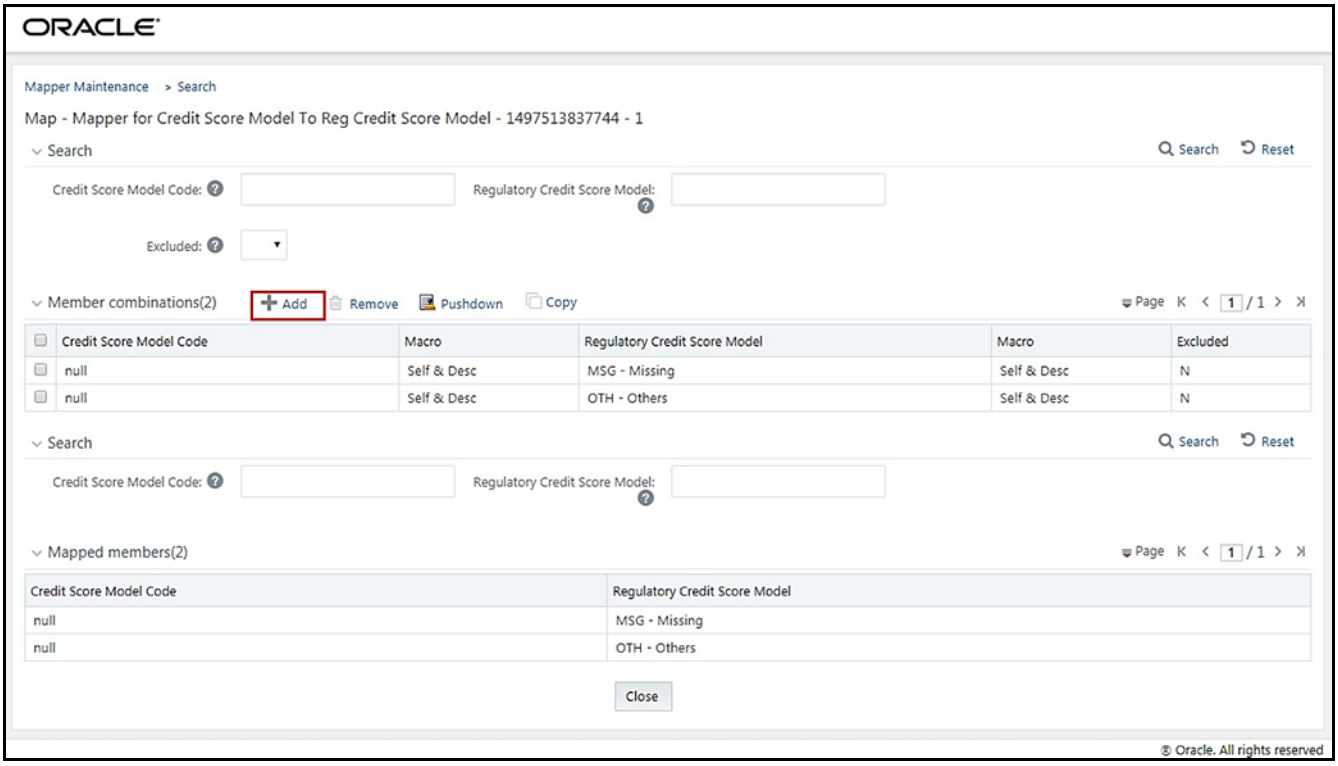


35.2.2.2 Many-to-One Mapping

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 the Fact Account Credit Score Details Regulatory Credit Score Model column while loading the T2T.

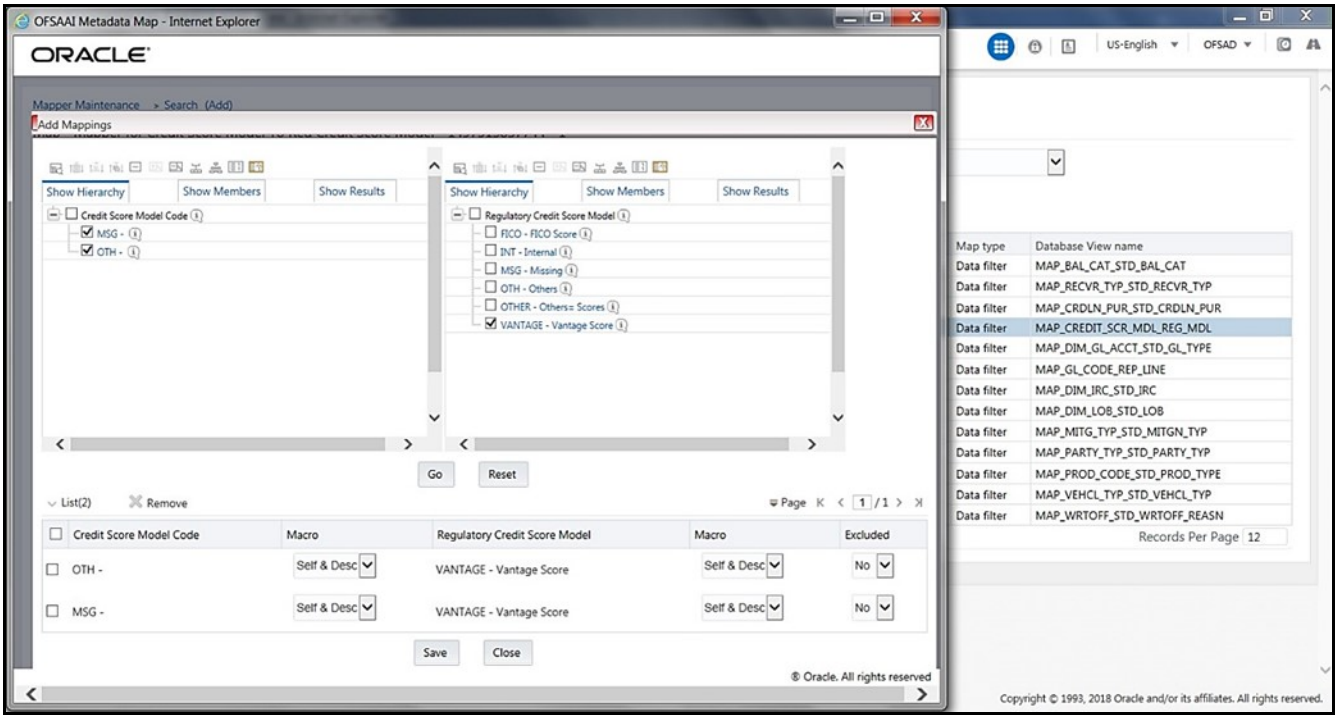
1. In the **Mapper Maintenance** page, click **Add**.

Figure 119: Select Add in the Mapper Maintenance page



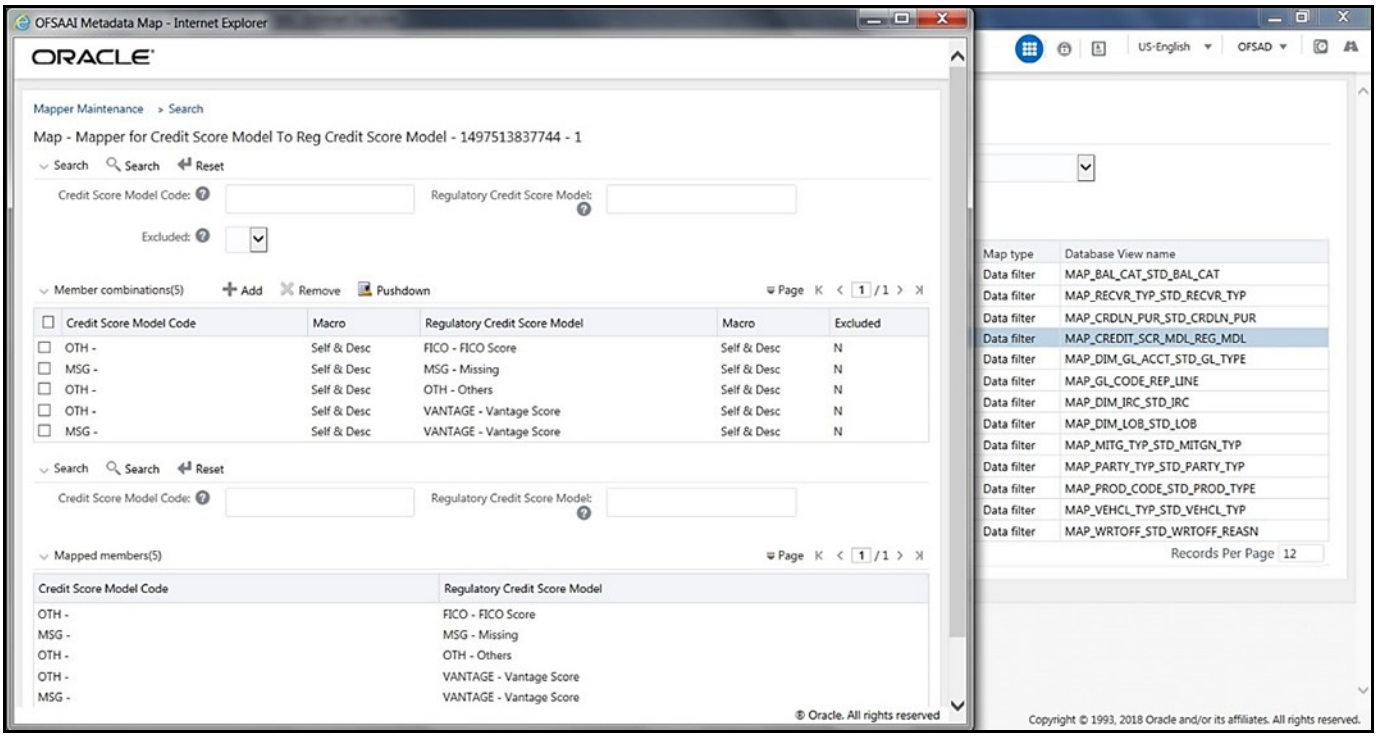
2. The **Add Mappings** page opens. In this example, MSG – Missing, and OTH - Others are mapped to VANTAGE - Vantage Score. To map, click **Go** and then to save the mapping, click **Save**.

Figure 120: Add Mappings page for the Many-to-One Mapping



3. An acknowledgment is displayed. Click **Yes** to confirm.
4. The Mapped Members are displayed in the **Mapper Maintenance** page.

Figure 121: Added mappings listed in the Mapper Maintenance page



35.3 Loading Mapper Maintenance through Backend

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

Credit Score Model Code (values from DIM_REG_CREDIT_SCORE_MODEL.V_REG_CREDIT_SCORE_MODEL_CODE).

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

Figure 122: Loading the Mapper Maintenance from backend

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

35.4 Deploying Credit Score Model and Probability of Default Model Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

35.5 Populating Credit Score Model and Probability of Default Model T2T Result Tables

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.

b. Select the Run Parameters and Execute the Run.

2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

35.6 Related Topics

You can see the following topics related to other function-specific tables:

- Common Account Summary Tables

36 Other Miscellaneous Tables

This section provides information about populating several other Results tables in the Oracle Insurance Data Foundation application and step-by-step instructions to use this section.

Topics:

Other Results tables store Party details related to fixed assets, spend obligations, and assets sold. These tables are used for the purpose of the financial report of the party or organization.

For the metadata design, see the *Other Miscellaneous Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

36.1 Deploying Other Result Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts.

36.2 Populating Other T2T Result Tables

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

37 Transaction Summary Tables

This section provides information about Transaction Summary tables in the Oracle Insurance Data Foundation application and step-by-step instructions to use this section.

Topics:

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.

For the metadata design, see the *Transaction Summary Tables* Section in the OIDF Metadata Design Document Release 8.1.x.

37.1 Deploying Transaction Summary Result Tables on Hive

All RDBMS related Result tables can also be deployed on Hive (Stage and Results). Deploy the Hive T2Ts using 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.

NOTE In general, Stage and Result tables are also supported in Hive. However, there are some exceptions. For a list of tables that are not supported in Hive, see List of Unsupported T2Ts

37.2 Populating Transaction Summary T2T Result Tables

Follow this T2T process to populate data into any T2T Result table:

NOTE Only RDBMS T2Ts can be executed using the PMF.

1. To populate data into any T2T Result table, execute the PMF process for that T2T. For a detailed procedure, see the following sections:
   a. Prerequisites for loading T2T.
   b. Select the Run Parameters and Execute the Run.
2. To check the T2T execution status and verify the log files of any Result table, follow the procedure in the Verify the Run Execution section.
3. To check the error messages, if any, follow the procedure in the Check Error Messages section.

37.3 Related Topics

You can see the following topics related to other function-specific tables:

- Common Account Summary Tables

38 Big Data Implementation in OIDF

This section provides information about Big Data processing supported in the Oracle Insurance Data Foundation application.

Topics:

38.1 About Big Data in OIDF

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

38.2 OIDF Big Data Architecture

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

- Staging and Results on Hive
- Staging on Hive and Results on RDBMS

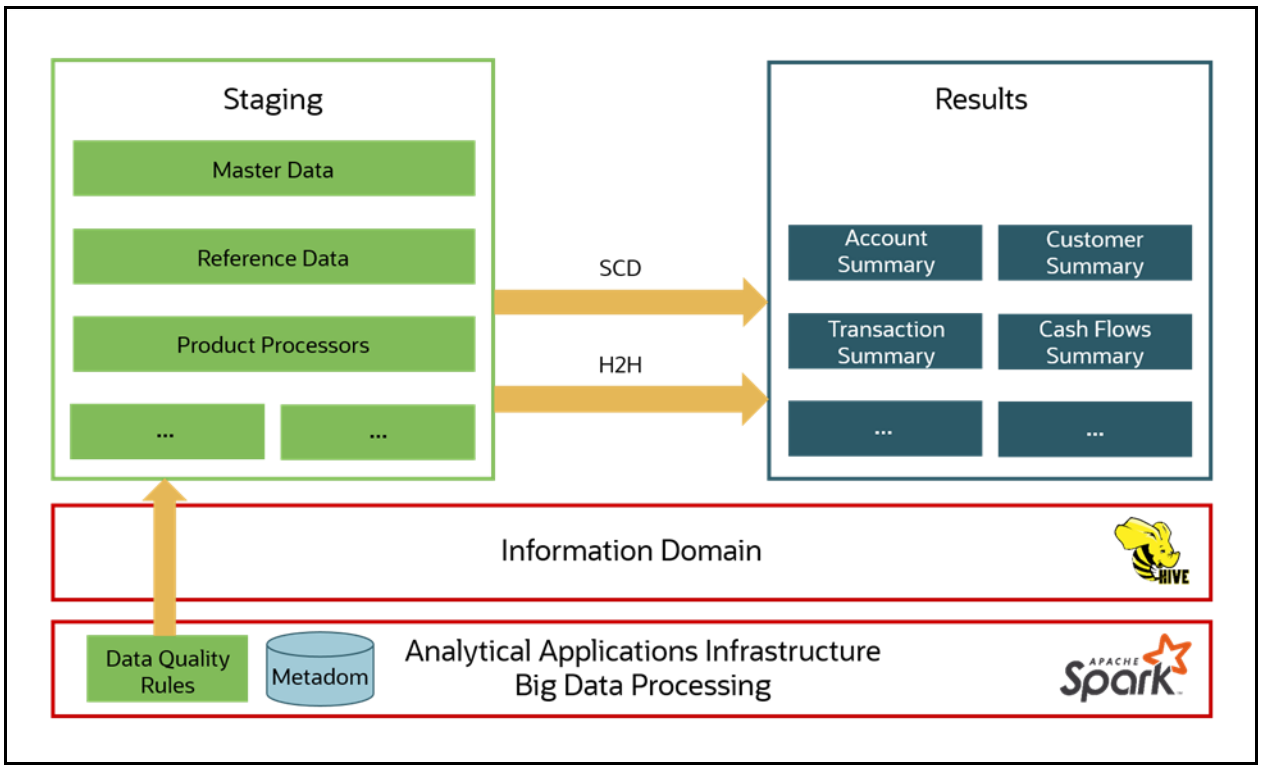
This section consists of the following topics:

- Modifications
- Workarounds
- List of Supported SCDs and T2Ts
- List of Unsupported SCDs
- List of Unsupported T2Ts
- Executing Run through Rule Run Framework for Hive

38.2.1 About 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 reside in an RDBMS schema known as Metadom. See the following architecture diagram for the representation of this deployment process.

Figure 123: Data Foundation Big Data Architecture with Staging and Results on Hive



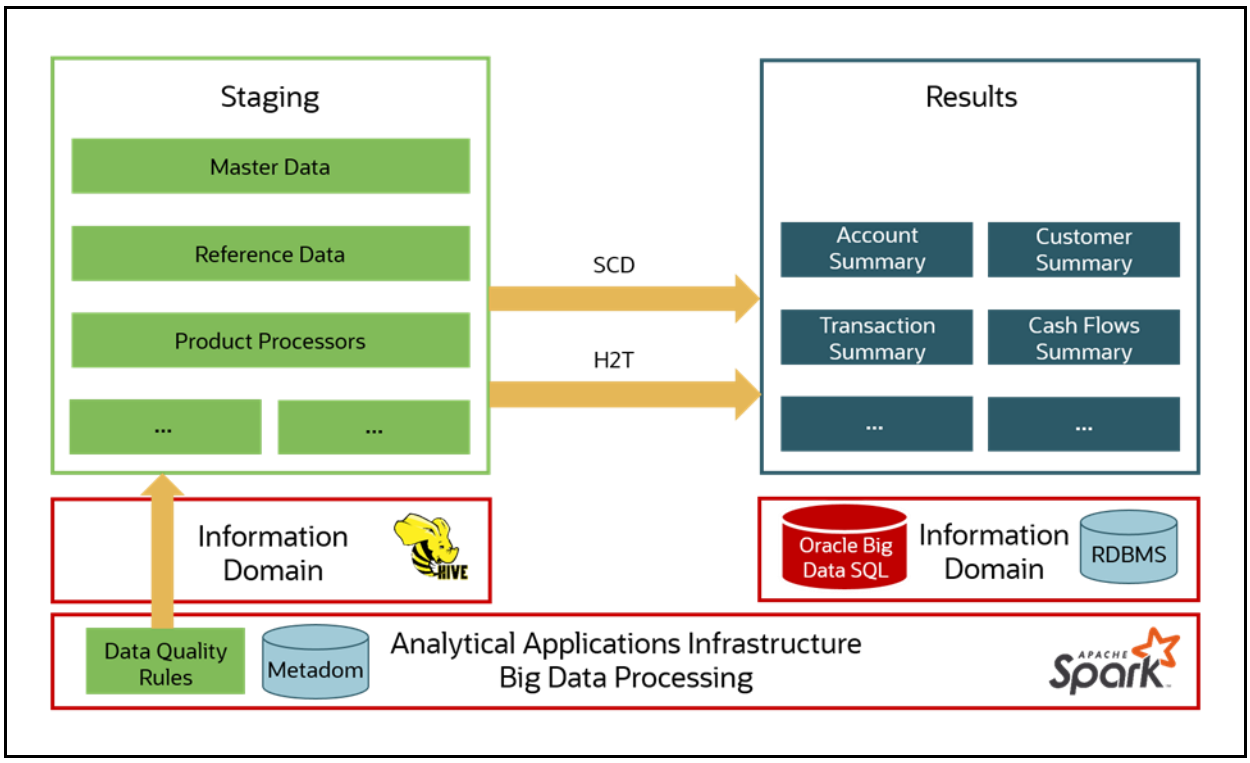
For the list of supported OIDF Hive Metadata definitions, see the Run Chart, SCD Metadata, and Technical Metadata (Staging Source) documents at My Oracle Support (MOS).

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

38.2.2 About Staging on Hive and Results on RDBMS

In the Staging on Hive and Results on the 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, additional software is required, which is Oracle Big Data SQL. 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. See the following architecture diagram for the representation of this deployment process.

Figure 124: Data Foundation Big Data Architecture with Staging on Hive and Results on RDBMS



38.2.3 Modifications

Modifications in OIDF Hive (Staging and Results on Hive deployment process) compared to OIDF 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 OIDF 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
```

38.2.4 Workarounds

The list of workarounds in OIDF 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 AMHM screens in OIDF (RDBMS). However, the development of the underlying functionality for the AMHM feature in the OIDF 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
  - MAP_DIM_GL_ACCT_STD_GL_TYPE
  - 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_RECVR_TYP_STD_RECVR_TYP
  - MAP_VEHCL_TYP_STD_VEHCL_TYP
  - MAP_WRTOFF_STD_WRTOFF_REASN

NOTE Hierarchies are supported using AMHM screens in OIDF (RDBMS). However, the development of the underlying functionality for the AMHM feature in the OIDF 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.

- 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 OIDF 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 base currency.

  Workaround:

  Source the corresponding Exchange Rate value.

- When performing Big Data installation for OIDF Hive, the following error is logged in the file `OIDF_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.

38.2.5 List of Supported SCDs and T2Ts

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

38.2.5.1 List of Supported SCDs

The SCDs for Hive used in Data Foundation solutions are listed in the Oracle Insurance Data Foundation for Hive - SCD Metadata for Hive spreadsheet under Technical Metadata for OIDF HIVE 8.1.2.0.0.

38.2.5.2 Run Enabled T2Ts

Deploy the OIDF Hive T2Ts using 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.

38.2.5.3 List of Supported T2Ts

The T2Ts for Hive used in Data Foundation solutions are listed in the Oracle Insurance Data Foundation for Hive - Technical Metadata (Staging Source) spreadsheet under Technical Metadata for OIDF HIVE 8.1.2.0.0.

38.2.6 List of Unsupported SCDs

- SCD-195
- SCD-196
- SCD-205
- SCD-208
- SCD-465
- SCD-411
- SCD-236

- SCD-237
- SCD-242
- SCD-248
- SCD-244
- SCD-336
- SCD-337
- SCD-378
- SCD-381
- SCD-335
- SCD-409
- SCD-419
- SCD-460
- SCD-332
- SCD-132

38.2.7 List of Unsupported T2Ts

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

- T2T_FSI_GL_CODE_REP_LINE_HIST
- T2T_FCT_PARTY_EMPLOYMENT_DETAILS
- T2T_FCT_PARTY_EXAM_DETAILS
- T2T_FCT_PARTY_PD_DETAILS
- T2T_FCT_PRDCR_INS_DISTRBTION_DTLS
- T2T_FCT_PARTY_MDCAL_CONDITN_DTL
- T2T_FCT_PARTY_FAMILY_MEDICAL_DA
- T2T_FCT_COMMON_CUSTOMER
- T2T_FCT_APPLICATION_DOCUMENT
- T2T_FCT_APPLICNS_DOC_PRINT_LOG
- T2T_FCT_PARTY_INS_POLICY_ROLE_MAP
- T2T_FCT_VEHICLE_INSPECTION_DETAILS
- T2T_FCT_VEHICLE_DEVICE_EVENT_DTLS
- T2T_FCT_VEHICLE_FLEET_MAP
- T2T_FCT_ACCT_CREDIT_SCORE_DETAILS
- T2T_FCT_CLAIM_BREAK_UP_DETAILS
- T2T_FPT_STG_ANNUITY_TXNS
- T2T_FPT_STG_HEALTH_INS_TXNS

- T2T_FPT_STG_LIFE_INS_POLICY
- T2T_FPT_STG_PROP_CASU_POLICY_TXNS
- T2T_FPT_STG_RETIREMENT_ACCOUNTS_TXNS
- T2T_FCT_POLICY_TRANSACTIONS_HLD
- T2T_FCT_POLICY_TRANSACTIONS_ISS
- T2T_FCT_INSURANCE_MORTALITY
- FN_POP_REG_LE_HEIR
- T2T_FSI_EXCHANGE_RATES
- T2T_FSI_EXCHANGE_RATES_FRWD
- T2T_FCT_PRODUCT_RATE_MATRIX
- T2T_FCT_TRIP_DETAILS
- T2T_FCT_TRIP_FLEET_SUMMARY
- T2T_FCT_TRIP_SUMMARY
- T2T_FCT_DRIVER_BEHAVIOUR_SUMMARY
- T2T_FCT_VEH_PERFORMANCE_SUMMARY
- T2T_FCT_INTRA_COMPANY_ACCT_SU
- T2T_FSI_INTRA_COMPANY_POLICY
- T2T_FCT_FIXED_ASSETS
- T2T_FCT_HEDGE_PORTFL_SET_ACCT_MAP
- T2T_FLAS_STG_LOAN_CONTRACTS
- T2T_FCT_FUND_CIS_COMPOSITION
- T2T_FFSIS_STG_INVESTMENTS
- T2T_FFSIS_STG_MANAGED_INV_ADV
- T2T_FFSIS_STG_MUTUAL_FUNDS
- T2T_FFSIS_STG_TRUSTS
- T2T_FFSIS_STG_CUSTODIAL_ACCOUNTS
- T2T_FCT_LITIGATION_DETAILS
- T2T_FCT_PAYMENTS_SUMMARY
- T2T_FCT_ACCT_RECOVERY_DETAILS
- T2T_FCT_ACCT_WRITE_OFF_DETAILS
- T2T_FCT_SPEND_OBLIGATIONS
- T2T_FCT_CREDIT_PARTCPN_TRNCH_DETL

38.2.8 Executing Run through Rule Run Framework for Hive

Deploy the OIDF Hive T2Ts using 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.

39 Metadata Browser

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

Topics:

39.1 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.

39.2 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

39.3 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.

39.4 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.

For complete information about the Metadata Browser, see the OFSAA Metadata Browser User Guide Release 8.1.0.0.0.

40 Recommendation for Backdated Run

This section provides information about performing a backdated Run in the 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.

Topics:

40.1 Overview of Backdated Run Execution

Backdated Run Execution is similar to any regular Run Execution in OIDF. You must reload or correct the data, which must be loaded for the given prior date. See the OIDF Run Chart Release 8.1.2.0.0 and execute the relevant Batches/Runs for the required prior date.

40.2 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.

40.3 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 OIDF 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.

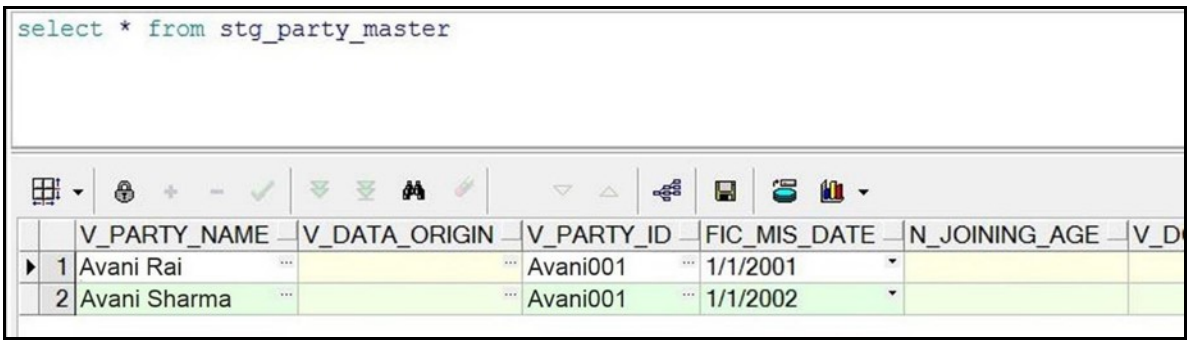
40.4 Backdated Run Execution using Latest Record Indicator (LRI) batch

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.

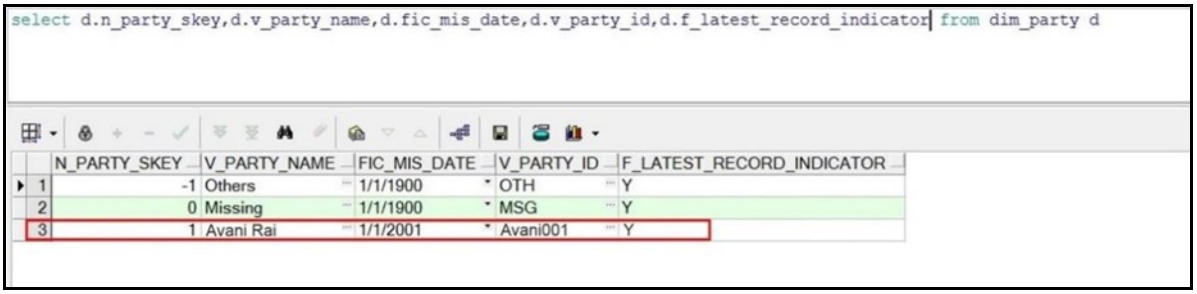
Figure 125: Sample of adding records for the backdated Run execution using LRI



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:

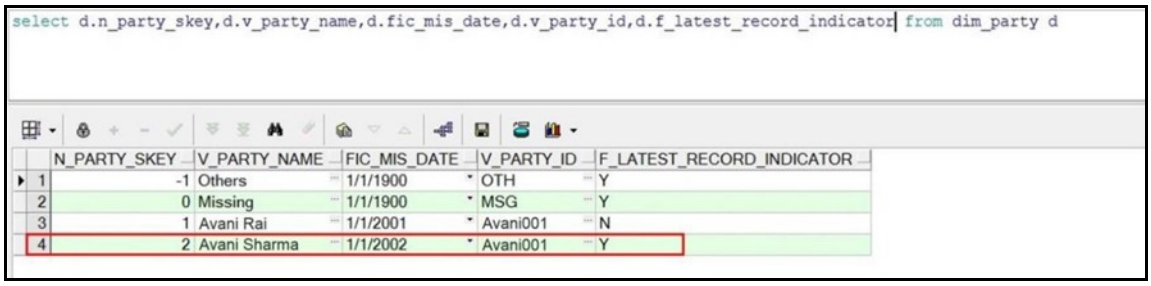
Figure 126: Sample after executing the SCD batch for first record for the backdated Run execution using LRI



	N_PARTY_SKEY	V_PARTY_NAME	FIC_MIS_DATE	V_PARTY_ID	F_LATEST_RECORD_INDICATOR
1	-1	Others	1/1/1900	OTH	Y
2	0	Missing	1/1/1900	MSG	Y
3	1	Avani Rai	1/1/2001	Avani001	Y

- After adding the 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 127: Sample after executing the SCD batch for the second record for the backdated Run execution using LRI



	N_PARTY_SKEY	V_PARTY_NAME	FIC_MIS_DATE	V_PARTY_ID	F_LATEST_RECORD_INDICATOR
1	-1	Others	1/1/1900	OTH	Y
2	0	Missing	1/1/1900	MSG	Y
3	1	Avani Rai	1/1/2001	Avani001	N
4	2	Avani Sharma	1/1/2002	Avani001	Y

- F_LATEST_RECORD_INDICATOR associated with the previous data record (with FIC_MIS_DATE 1/1/2001) is set to N automatically.

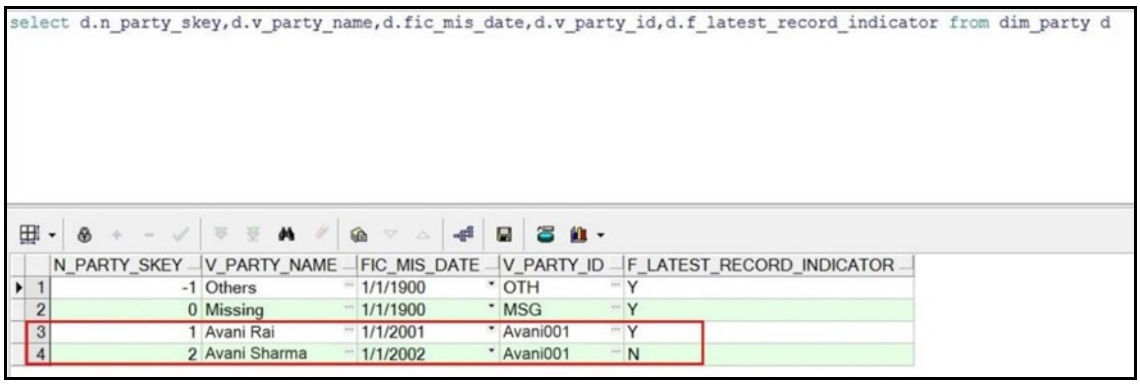
3. In **Batch Maintenance**, add a **Task** for the LRI Dimension table. In the **Parameter List** field, mention the Dimension table name ('DIM_TABLE_NAME') for which Backdated Run using LRI must be executed. In this example, 'DIM_TABLE_NAME' = 'DIM_PARTY'.

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

   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.

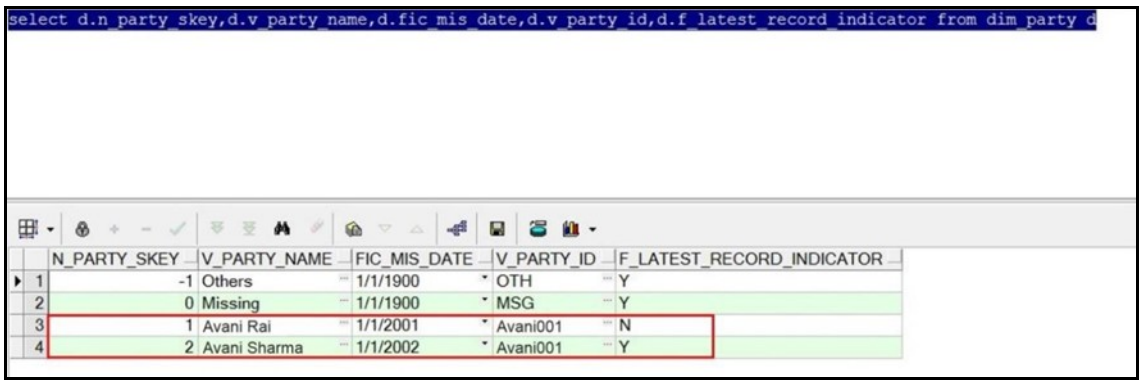
Figure 128: Sample data record after running the LRI batch with a different FIC_MIS_DATE



	N_PARTY_SKEY	V_PARTY_NAME	FIC_MIS_DATE	V_PARTY_ID	F_LATEST_RECORD_INDICATOR
1	-1	Others	1/1/1900	OTH	Y
2	0	Missing	1/1/1900	MSG	Y
3	1	Avani Rai	1/1/2001	Avani001	Y
4	2	Avani Sharma	1/1/2002	Avani001	N

- To change the default data selection, execute the UPDATE_BACK_DATED_DIM_LRI batch for the required data record for its FIC_MIS_DATE. In the following image, the batch is executed for the data record with FIC_MIS_DATE 1/1/2002. Therefore, the flag F_LATEST_RECORD_INDICATOR associated with this FIC_MIS_DATE 1/1/2002 is set to Y and this data record is now flagged as the Latest Record. This record will be used as default by the system for all future transactions and processes. The flag F_LATEST_RECORD_INDICATOR associated with the earlier default data record (with FIC_MIS_DATE 1/1/2001) is set automatically to N, indicating that the record with FIC_MIS_DATE 1/1/2001 is not the Latest Record anymore.

Figure 129: Sample data record after running again LRI batch with a different FIC_MIS_DATE



	N_PARTY_SKEY	V_PARTY_NAME	FIC_MIS_DATE	V_PARTY_ID	F_LATEST_RECORD_INDICATOR
1	-1	Others	1/1/1900	OTH	Y
2	0	Missing	1/1/1900	MSG	Y
3	1	Avani Rai	1/1/2001	Avani001	N
4	2	Avani Sharma	1/1/2002	Avani001	Y

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

41 Compare Data Model Reports

This chapter provides information about comparing Data Model Reports of two release versions in the Data Foundation application and step-by-step instructions to use this feature.

The first section gives you an understanding of the Data Model Report extracted from the erwin Data Modeler application. The Comparing Data Model Reports section details the steps to use the OFSAA application and download the Difference Report between two Data Model release versions.

Topics:

41.1 Creating the Data Model Report from erwin

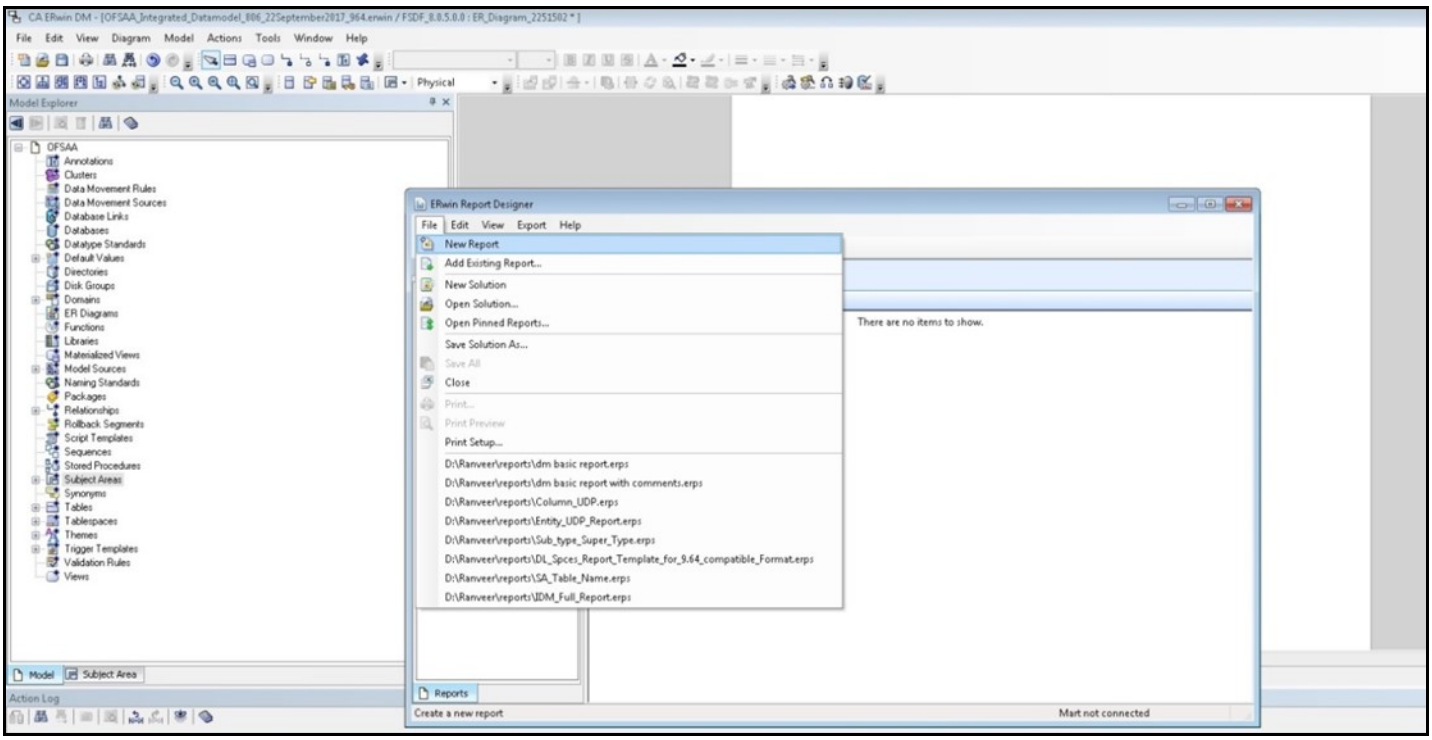
You can create new Data Model Reports from the erwin Data Modeler application if there are no .eprs files to extract the information.

> **NOTE** If the Data Model Reports are existing, see Extracting Data Model Report from erwin for more information.

To create the Data Model Report from the erwin Data Modeler application, follow these steps:

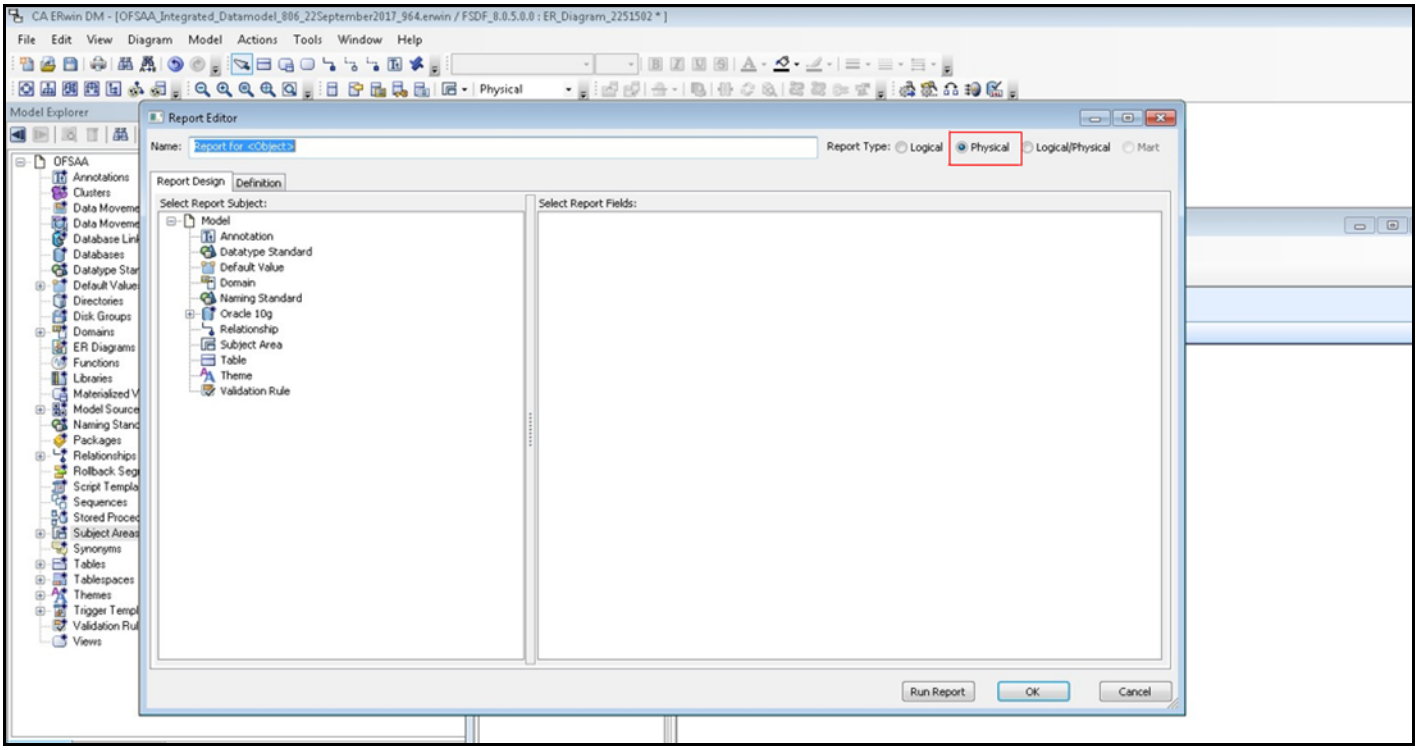
1. In the erwin **Report Designer** page, select **File**, select **New Report** (to create the .erps report file).

Figure 130: Select New Report in the erwin Report Designer page



2. In the **Report Editor** page, select the Report Type as **Physical** and select the Report Subject as **Table**.

Figure 131: Select the report type and report subject in the Report Editor page



3. Select the **Report Fields** to match the sample Data Model Report shown as follows:

Figure 132: Select the Report Fields to match the sample Data Model Report

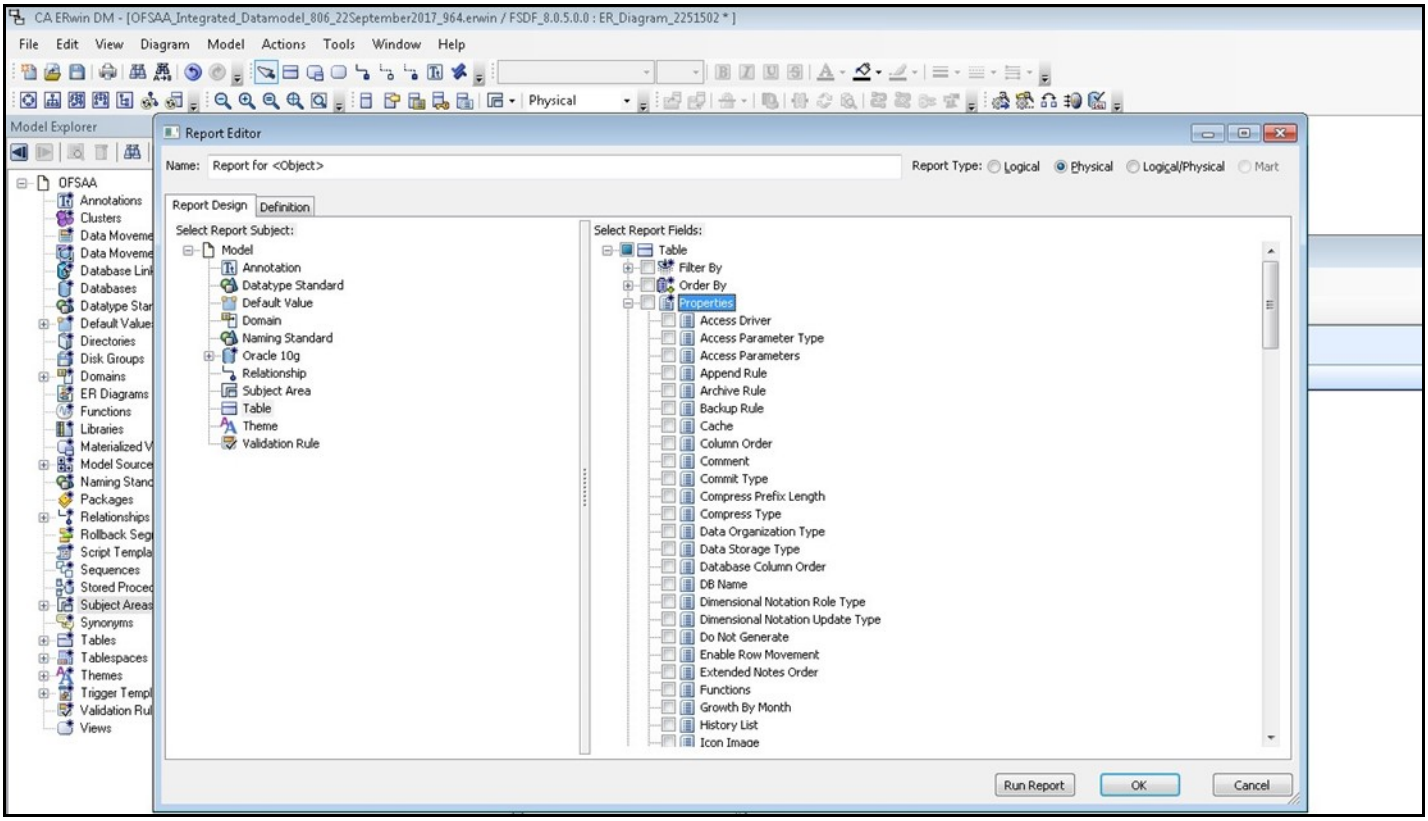
FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW Acrobat

K17

	A	B	C	D	E	F	G	H
1	**Entity/Table Physical_Name**	**Attribute/Column Physical_Name**	**Physical Data Type**	**Null Option**	**Is PK**	**Is FK**	**Domain Parent**	
2	Dim_Financial_Elements_Attr	financial_elem_id	NUMBER(14)	Not Null	Yes	No	ID_NUMBER	
3	Dim_Financial_Elements_Attr	attribute_id	NUMBER(22)	Not Null	Yes	No	Number	
4	Dim_Financial_Elements_Attr	dim_attribute_numeric_member	NUMBER(22)	Null	No	No	Number	
5	Dim_Financial_Elements_Attr	dim_attribute_varchar_member	VARCHAR2(30)	Null	No	No	Text_Short_Description	
6	Dim_Financial_Elements_Attr	number_assign_value	NUMBER(22)	Null	No	No	Number	
7	Dim_Financial_Elements_Attr	varchar_assign_value	VARCHAR2(1000)	Null	No	No	Text_Comments_Type2	
8	Dim_Financial_Elements_Attr	date_assign_value	DATE	Null	No	No	Datetime	
9	Dim_Financial_Elements_B	financial_elem_id	NUMBER(14)	Not Null	Yes	No	ID_NUMBER	
10	Dim_Financial_Elements_B	financial_elem_display_code	NUMBER(14)	Not Null	No	No	Number	
11	Dim_Financial_Elements_B	enabled_flag	VARCHAR2(1)	Not Null	No	No	Indicator	
12	Dim_Financial_Elements_B	leaf_only_flag	VARCHAR2(1)	Null	No	No	Indicator	
13	Dim_Financial_Elements_B	definition_language	VARCHAR2(10)	Not Null	No	No	Code_Alphanumeric_Medium	
14	Dim_Financial_Elements_B	created_by	VARCHAR2(30)	Not Null	No	No	Text_Short_Description	
15	Dim_Financial_Elements_B	creation_date	TIMESTAMP	Not Null	No	No	Datetime	
16	Dim_Financial_Elements_B	last_modified_by	VARCHAR2(30)	Not Null	No	No	Text_Short_Description	
17	Dim_Financial_Elements_B	last_modified_date	TIMESTAMP	Not Null	No	No	Datetime	
18	Dim_Financial_Elements_B	financial_elem_code	VARCHAR2(20)	Null	No	No	VARCHAR2	
19	Dim_Financial_Elements_Hier	hierarchy_id	NUMBER(10)	Not Null	Yes	No	Number_Medium	
20	Dim_Financial_Elements_Hier	parent_depth_num	NUMBER(22)	Not Null	No	No	Number_Generic	
21	Dim_Financial_Elements_Hier	parent_id	NUMBER(14)	Not Null	Yes	No	ID_NUMBER	
22	Dim_Financial_Elements_Hier	child_depth_num	NUMBER(22)	Not Null	No	No	Number	

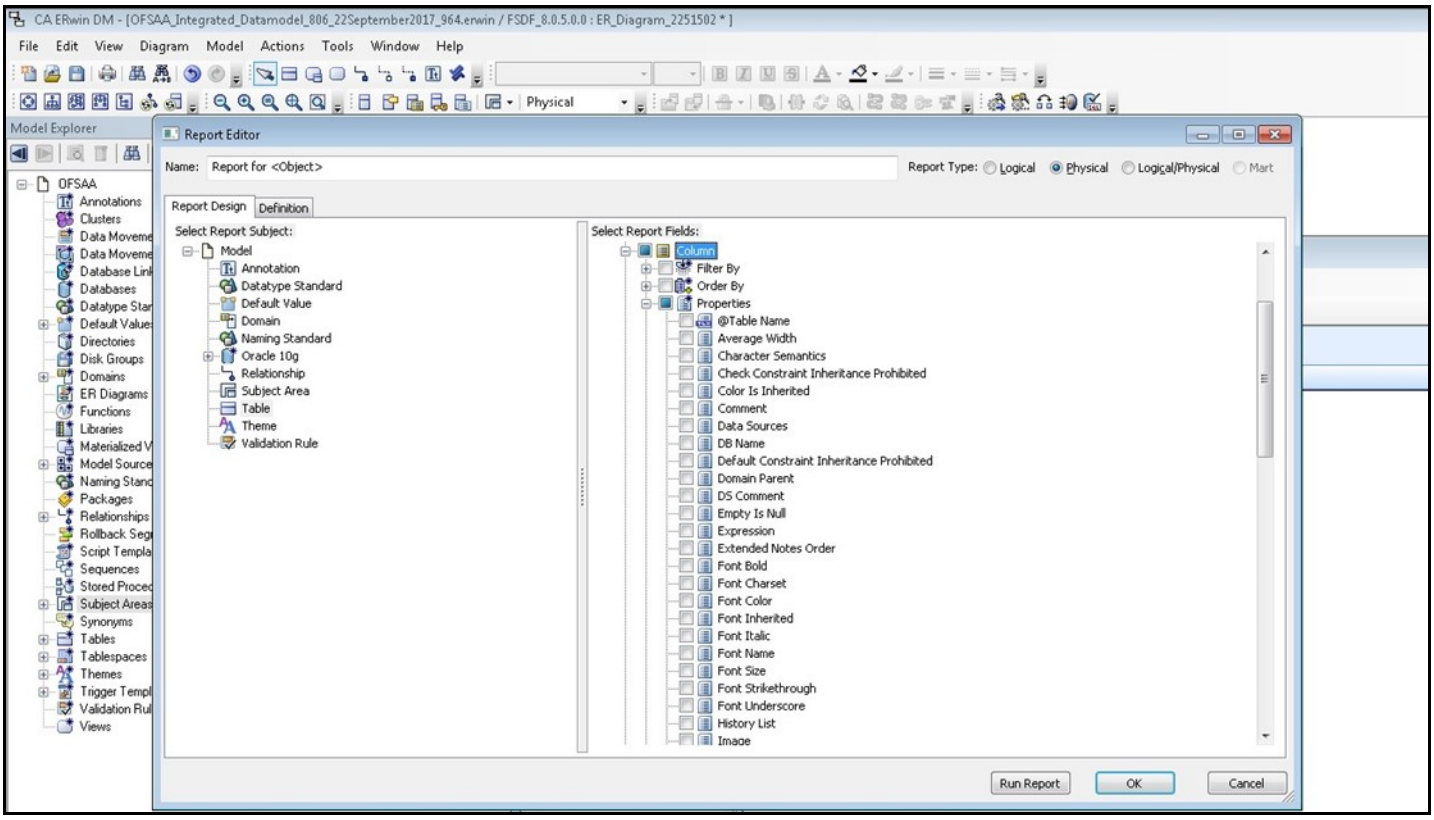
4. In the **Report Editor** page, in the Select Report Fields section, expand **Table**, expand **Properties**, and select **Physical Name (Entity/Table Physical_Name)**.

Figure 133: Select the Table details in the Report Editor page



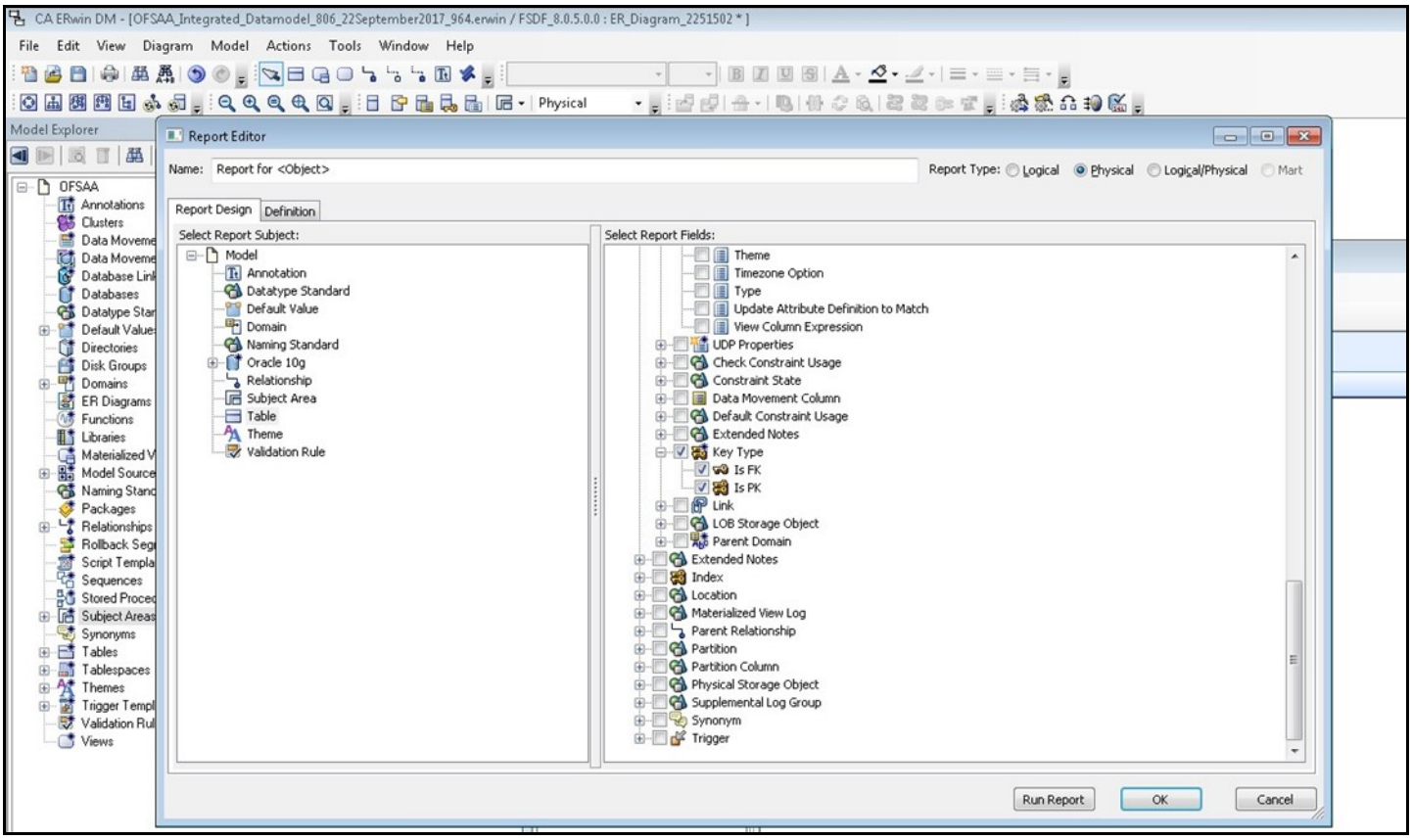
5. In the **Report Editor** page, in the Select Report Fields section, expand **Column**, expand **Properties**, select **Physical Name (Attribute/Column Physical_Name)**, **Physical Data Type**, **Null Option**, and **Domain Parent**.

Figure 134: Select the Column details in the Report Editor page



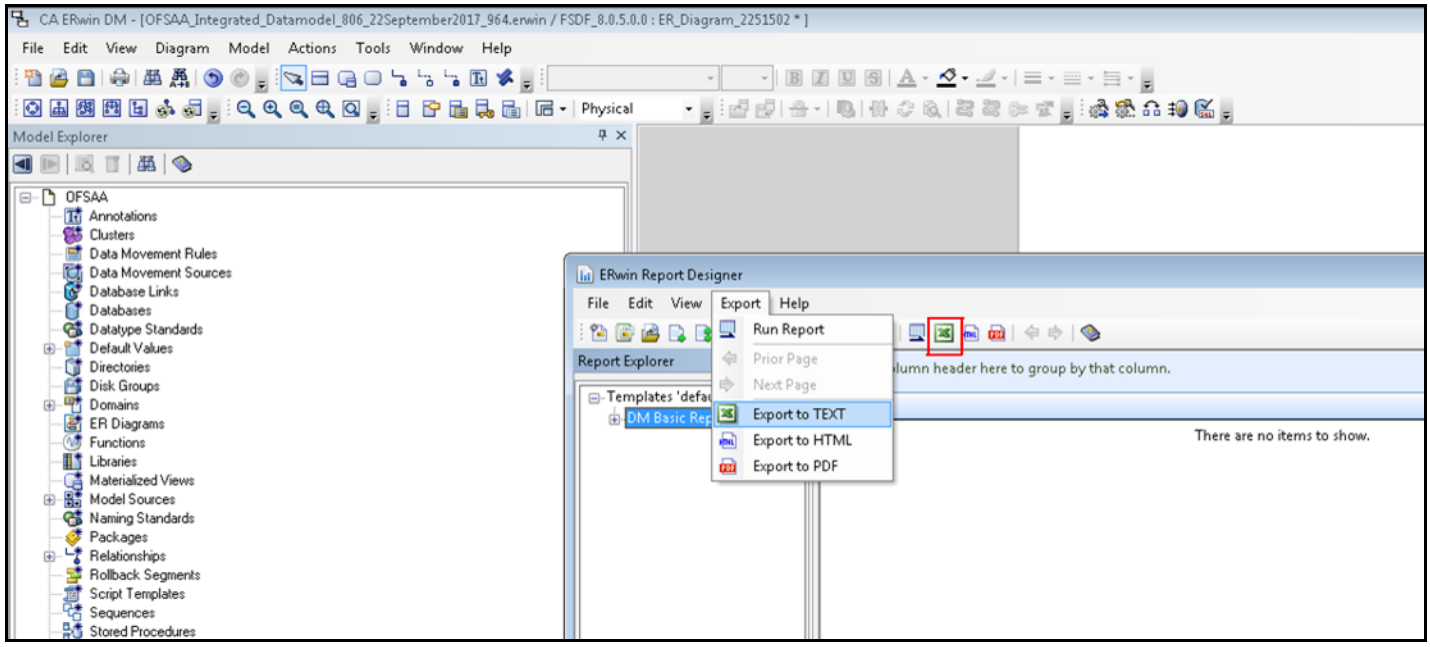
6. In the **Report Editor** page, in the Select Report Fields section, expand **Table**, expand **Key Type**, select **Is PK**, and **Is FK**. Click **OK**.

Figure 135: Select the PK and FK options in the Report Editor page



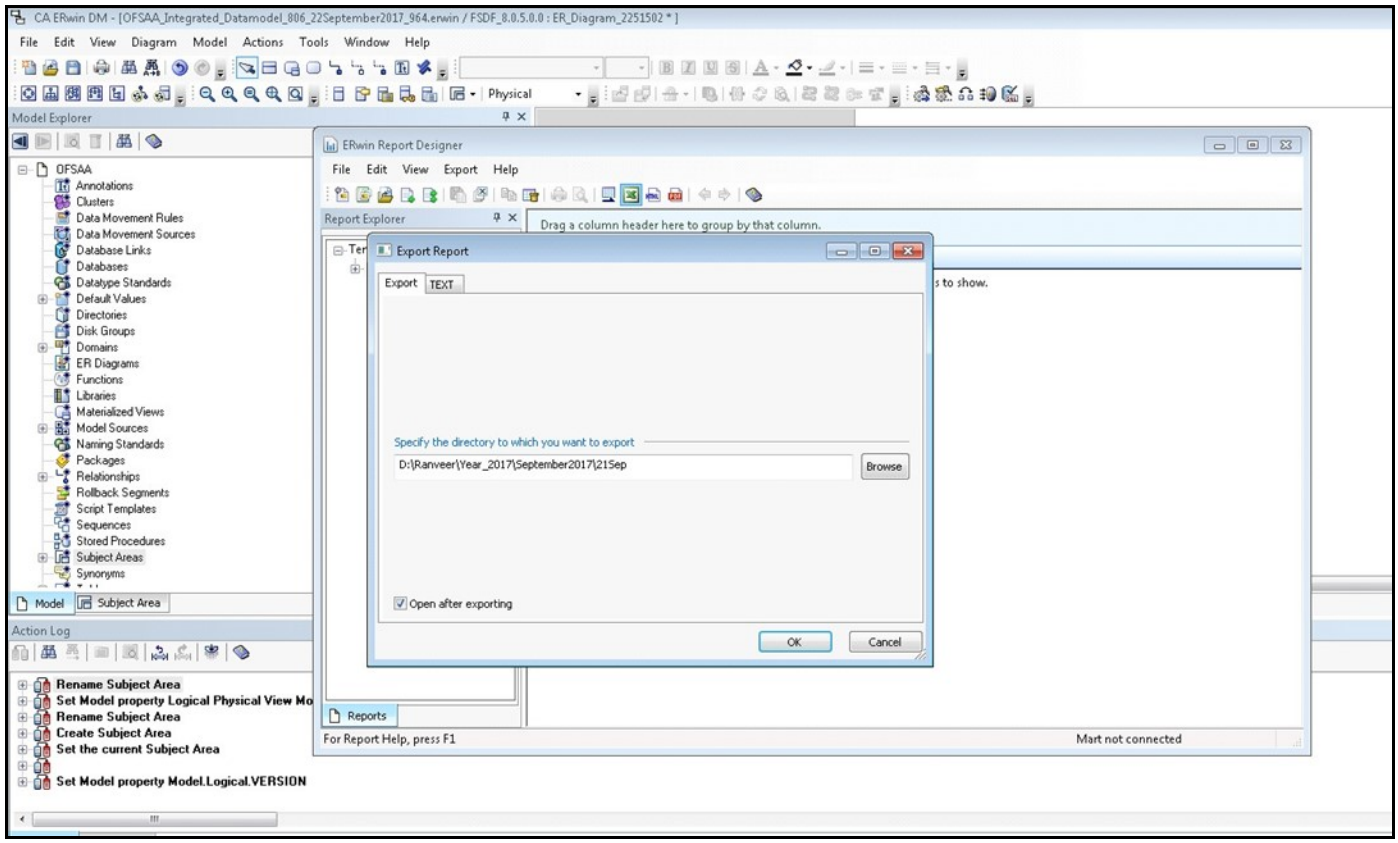
7. In the erwin **Report Designer** page, select **Export**, select **Export to TEXT** (Excel Format) or the Excel icon.

Figure 136: Select the report export option in the Report Editor page



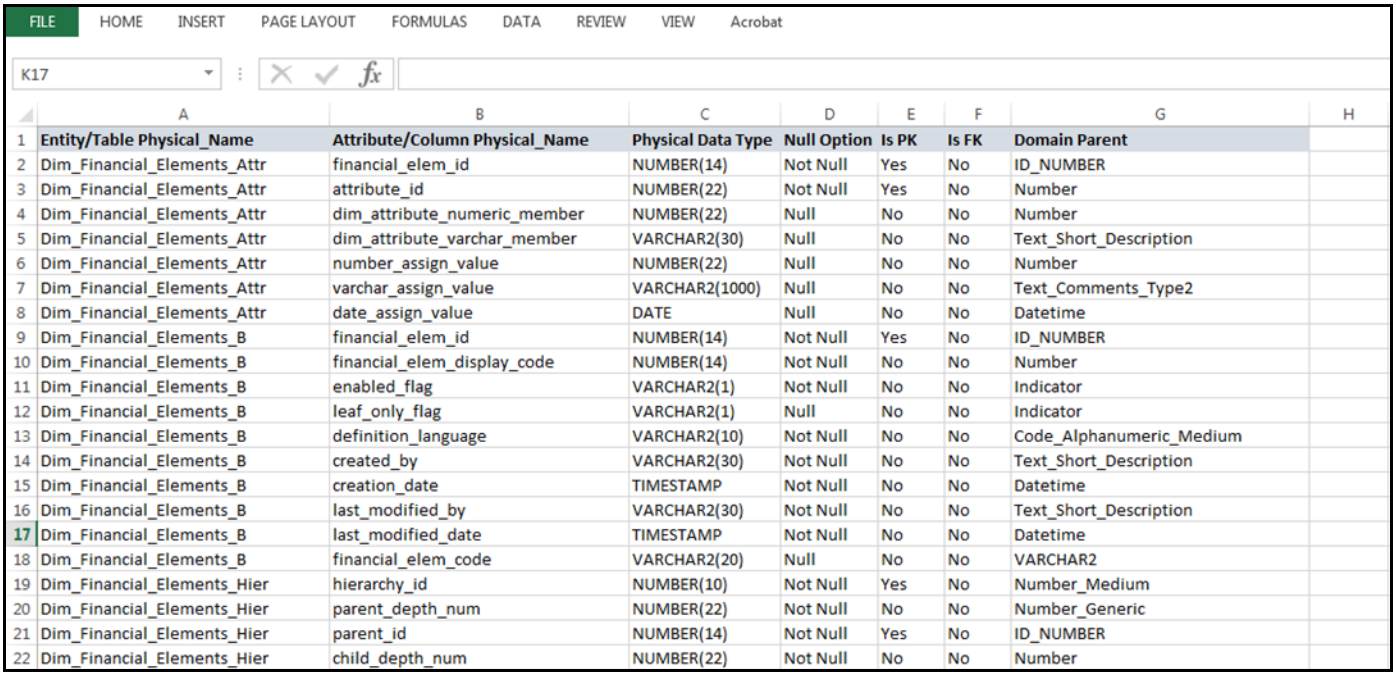
8. Enter or **Browse** the Export file path where you want to save the Data Model Report file.

Figure 137: Select the report export path



9. The generated Data Model Report file opens and must be in the following format.

Figure 138: Sample of the generated Data Model Report file



	A	B	C	D	E	F	G	H
1	Entity/Table Physical_Name	Attribute/Column Physical_Name	Physical Data Type	Null Option	Is PK	Is FK	Domain Parent	
2	Dim_Financial_Elements_Attr	financial_elem_id	NUMBER(14)	Not Null	Yes	No	ID_NUMBER	
3	Dim_Financial_Elements_Attr	attribute_id	NUMBER(22)	Not Null	Yes	No	Number	
4	Dim_Financial_Elements_Attr	dim_attribute_numeric_member	NUMBER(22)	Null	No	No	Number	
5	Dim_Financial_Elements_Attr	dim_attribute_varchar_member	VARCHAR2(30)	Null	No	No	Text_Short_Description	
6	Dim_Financial_Elements_Attr	number_assign_value	NUMBER(22)	Null	No	No	Number	
7	Dim_Financial_Elements_Attr	varchar_assign_value	VARCHAR2(1000)	Null	No	No	Text_Comments_Type2	
8	Dim_Financial_Elements_Attr	date_assign_value	DATE	Null	No	No	Datetime	
9	Dim_Financial_Elements_B	financial_elem_id	NUMBER(14)	Not Null	Yes	No	ID_NUMBER	
10	Dim_Financial_Elements_B	financial_elem_display_code	NUMBER(14)	Not Null	No	No	Number	
11	Dim_Financial_Elements_B	enabled_flag	VARCHAR2(1)	Not Null	No	No	Indicator	
12	Dim_Financial_Elements_B	leaf_only_flag	VARCHAR2(1)	Null	No	No	Indicator	
13	Dim_Financial_Elements_B	definition_language	VARCHAR2(10)	Not Null	No	No	Code_Alphanumeric_Medium	
14	Dim_Financial_Elements_B	created_by	VARCHAR2(30)	Not Null	No	No	Text_Short_Description	
15	Dim_Financial_Elements_B	creation_date	TIMESTAMP	Not Null	No	No	Datetime	
16	Dim_Financial_Elements_B	last_modified_by	VARCHAR2(30)	Not Null	No	No	Text_Short_Description	
17	Dim_Financial_Elements_B	last_modified_date	TIMESTAMP	Not Null	No	No	Datetime	
18	Dim_Financial_Elements_B	financial_elem_code	VARCHAR2(20)	Null	No	No	VARCHAR2	
19	Dim_Financial_Elements_Hier	hierarchy_id	NUMBER(10)	Not Null	Yes	No	Number_Medium	
20	Dim_Financial_Elements_Hier	parent_depth_num	NUMBER(22)	Not Null	No	No	Number_Generic	
21	Dim_Financial_Elements_Hier	parent_id	NUMBER(14)	Not Null	Yes	No	ID_NUMBER	
22	Dim_Financial_Elements_Hier	child_depth_num	NUMBER(22)	Not Null	No	No	Number	

NOTE Ensure that the file is saved in `.xlsx` format.

41.2 Extracting Data Model Report from erwin

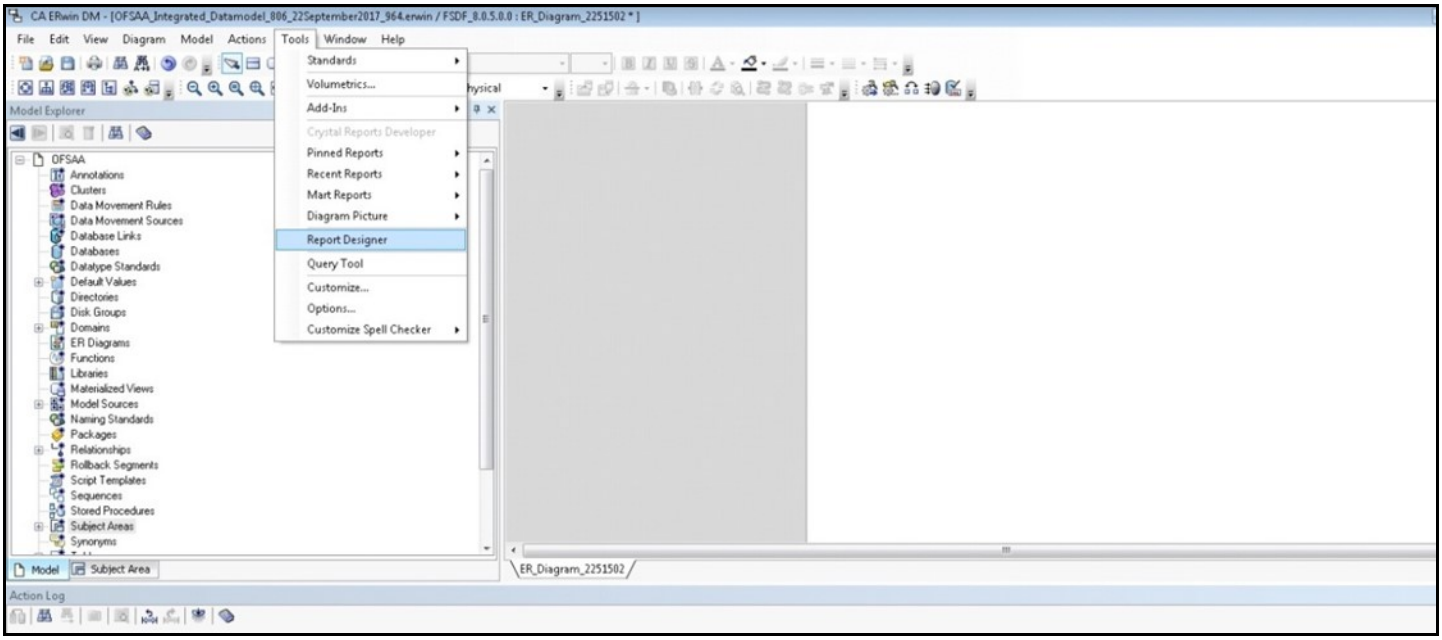
OIDF is a collection of data model artifacts delivered as erwin files or can be extracted as.XLS file from the erwin Data Modeler application. OIDF hence requires a license of the erwin Data Modeler application.

erwin is the current and only supported modeling tool to view and edit the model. Currently, the minimum version of erwin Data Modeler application supported is 2019R1 or a higher version for OIDF 8.1.2.3.0 and earlier versions. For OIDF 8.1.2.4.0 and later versions, the minimum version of erwin supported is 12.1.

To extract the Data Model Report from the erwin Data Modeler application, follow these steps:

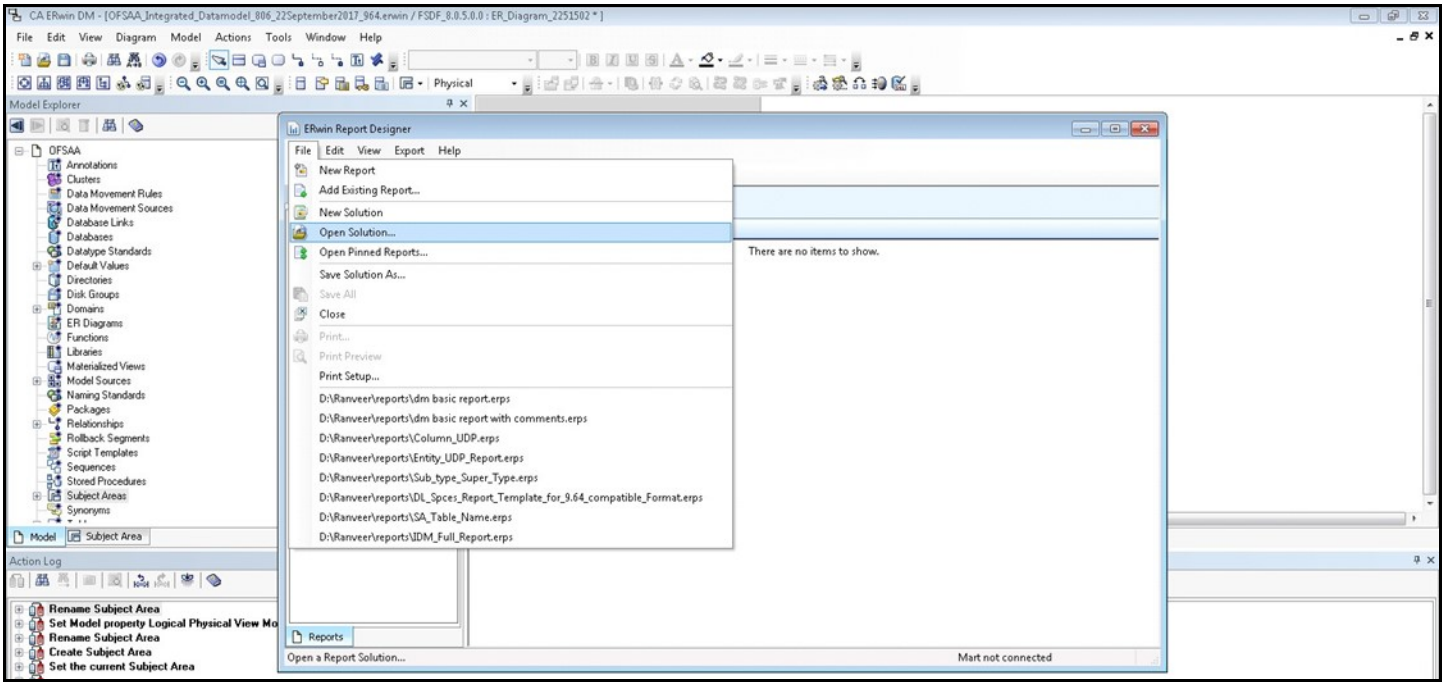
1. Open the Erwin Data Modeler application. Select the **Tools** menu and select the **Report Designer**.

Figure 139: Select Report Designer



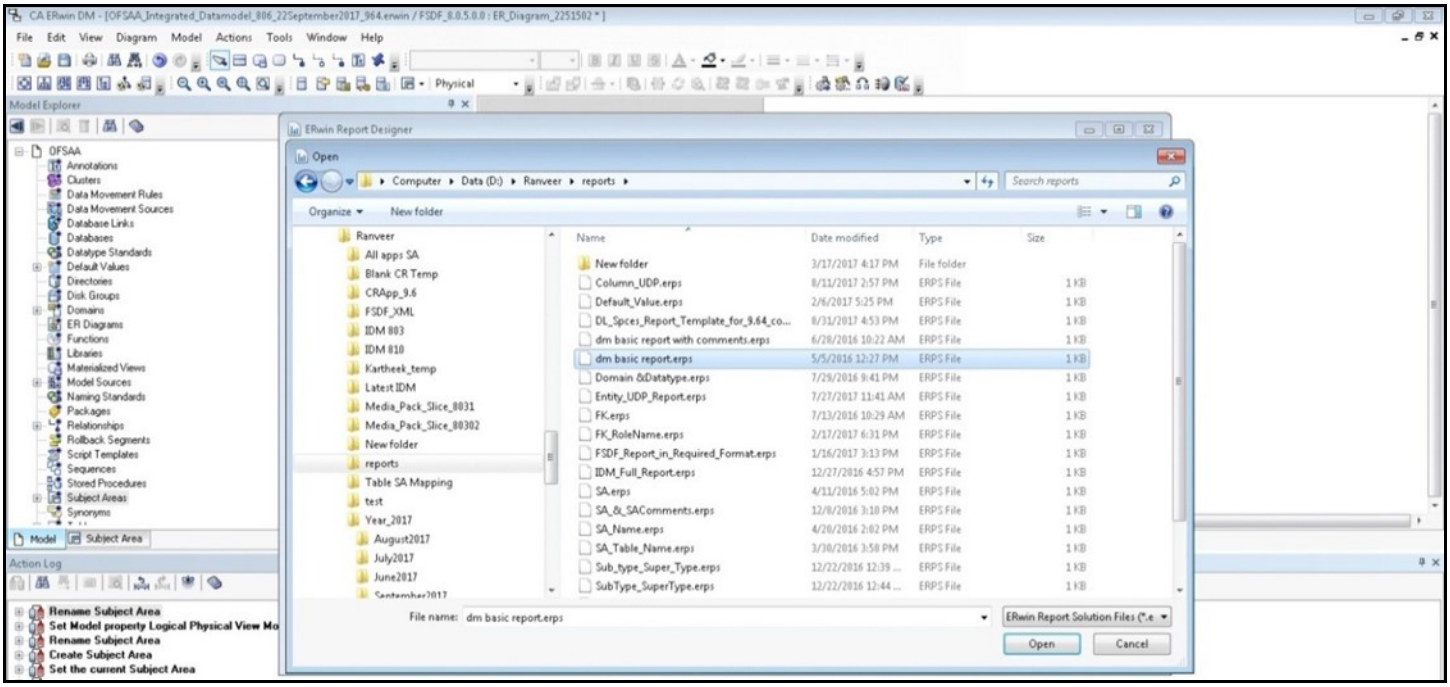
2. In the erwin **Report Designer** page, select **File**, select **Open Solution** (to extract the .erps report file).

Figure 140: Select Open Solution in the erwin Report Designer page



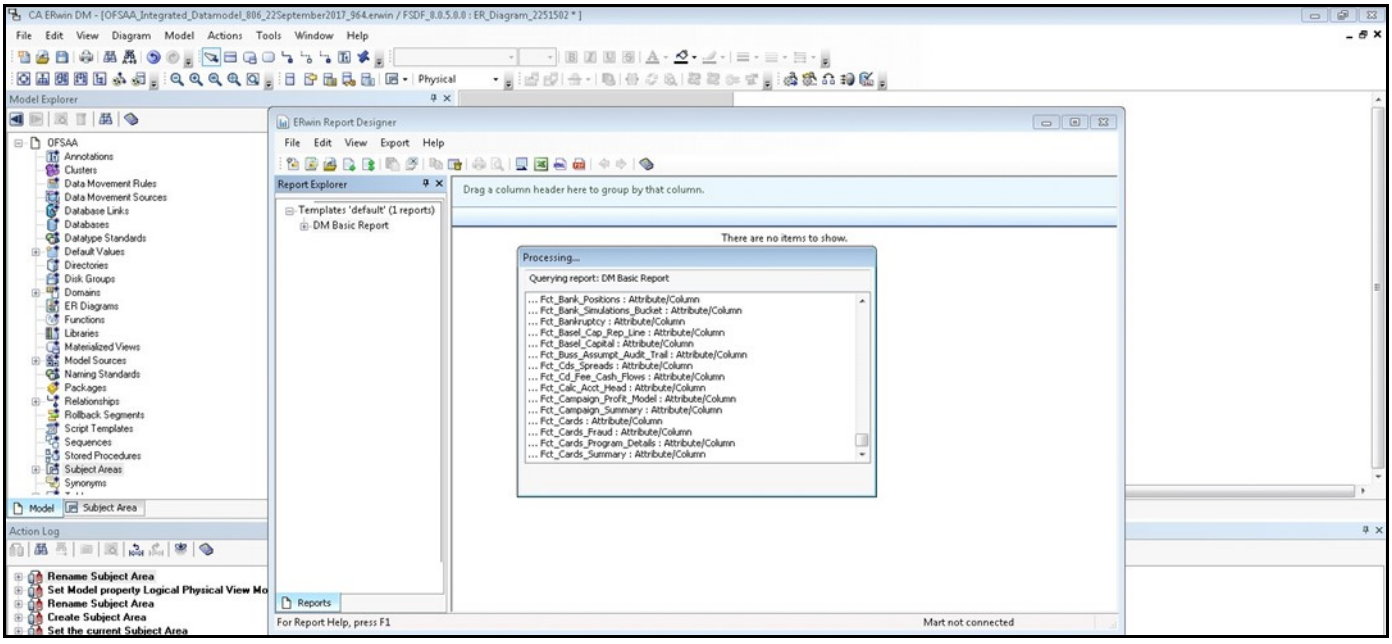
3. Browse the path of the `.erps` file, select the file, and click **Open**.

Figure 141: Browse and open the .erps file



4. The existing Data Model Report file is extracted and processed to export the file in the Excel format.

Figure 142: The Data Model Report file is extracted and processed



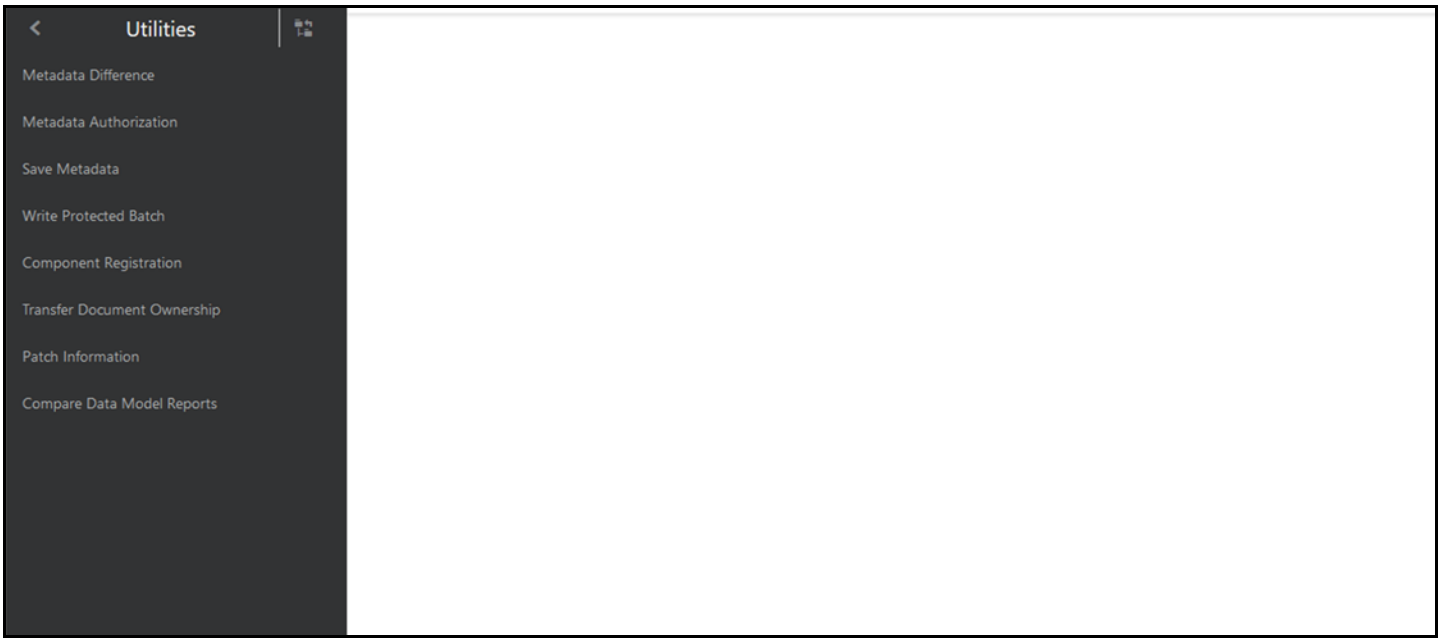
5. The generated Data Model Report file in the Excel format opens. **Save** the file in .xlsx format.

41.3 Compare Data Model Reports and Generate Data Model Difference Report

To extract the Data Model Report from the erwin Data Modeler application, follow these steps:

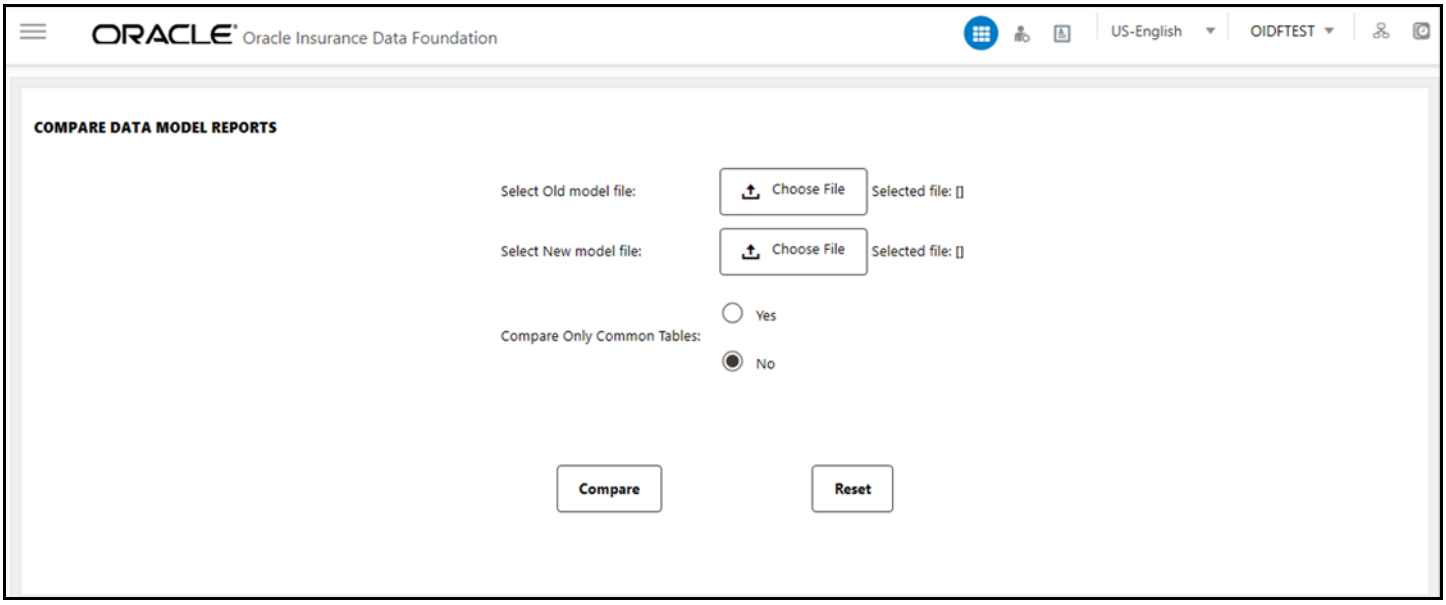
1. From **OFSAA Home**, select **Oracle Insurance Data Foundation**, select **Common Components**, select **Utilities**, and then select **Compare Data Model Reports**.

Figure 143: Navigation to the Compare Data Model Reports page



2. The **Compare Data Model Reports** page is displayed.

Figure 144: Compare Data Model Reports page

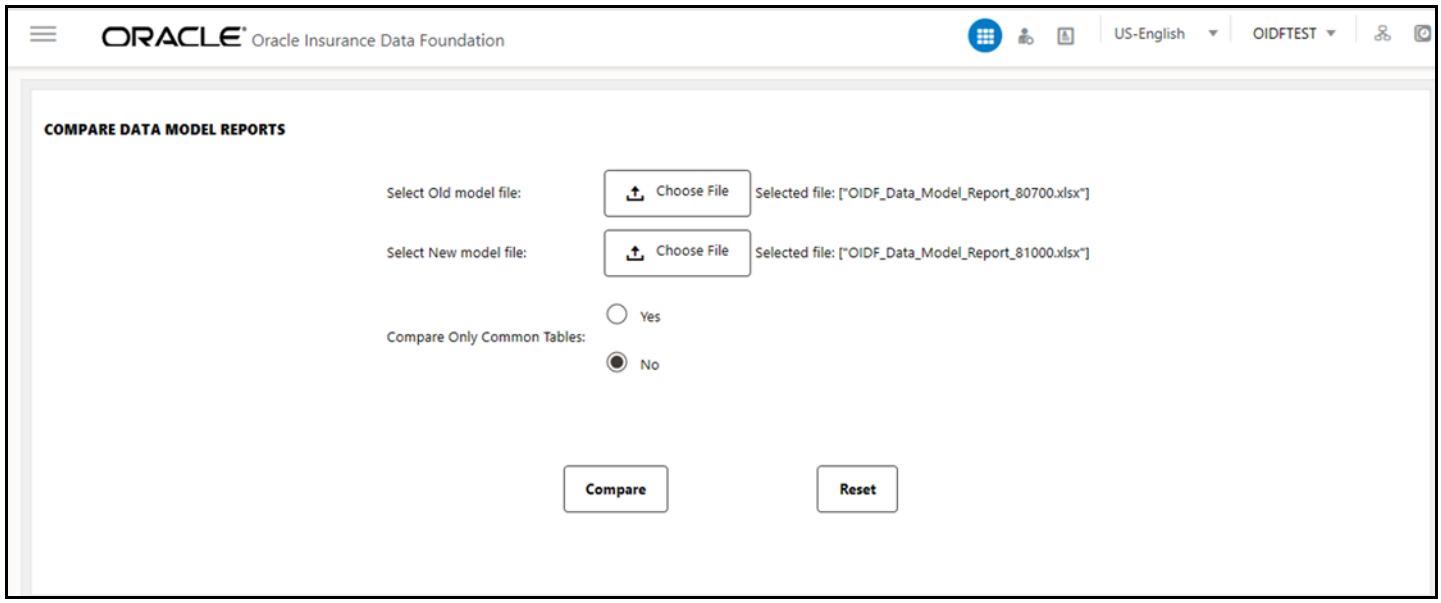


3. To browse and select the old data model report excel file, use **Choose File** associated with the **Select Old model file** label. To browse and select the new data model report excel file, use **Choose**

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 145: Select the old and new data model reports

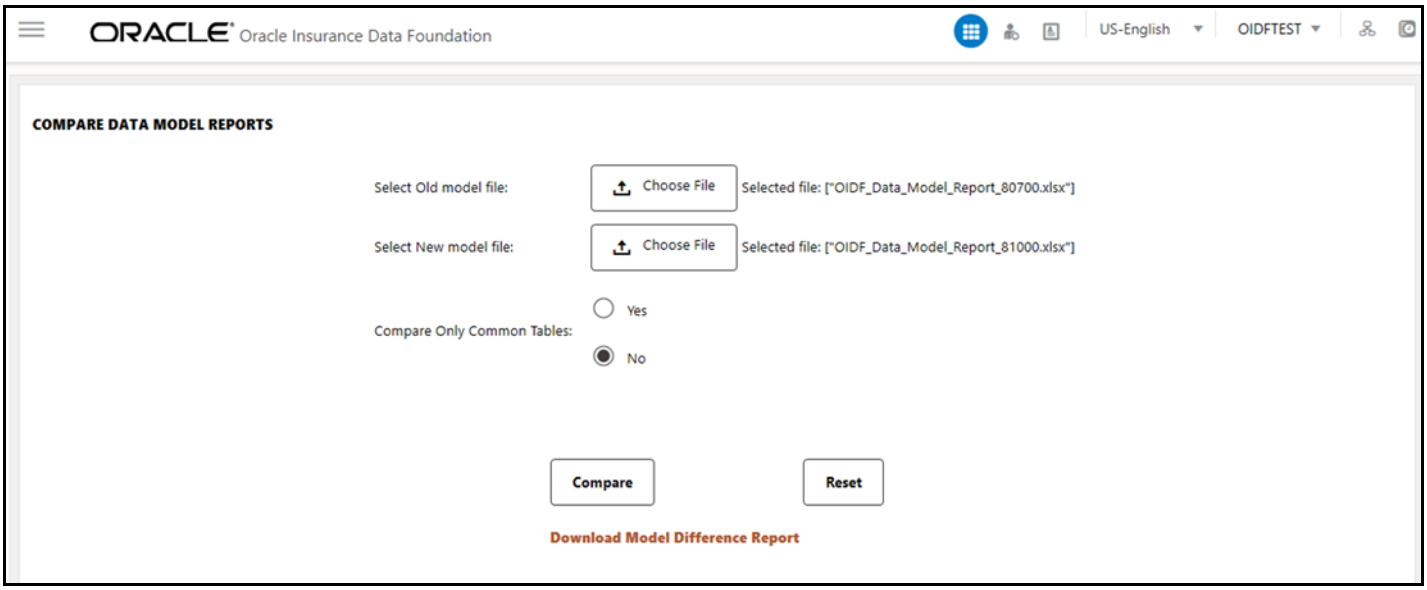


The **Compare Only Common Tables** option is **No** by default. Select **Yes** only if required.

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 146: Generate the Data Model Difference Report



5. **Save** the file.

42 Data Quality Framework

This chapter provides information about Data Quality Rules Execution in the Oracle Insurance Data Foundation application and step-by-step instructions to use this section.

Topics:

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 Rules
- Data Quality Groups

> **NOTE** For the latest information about Data Quality Framework, see the *Data Quality Framework* section in the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.

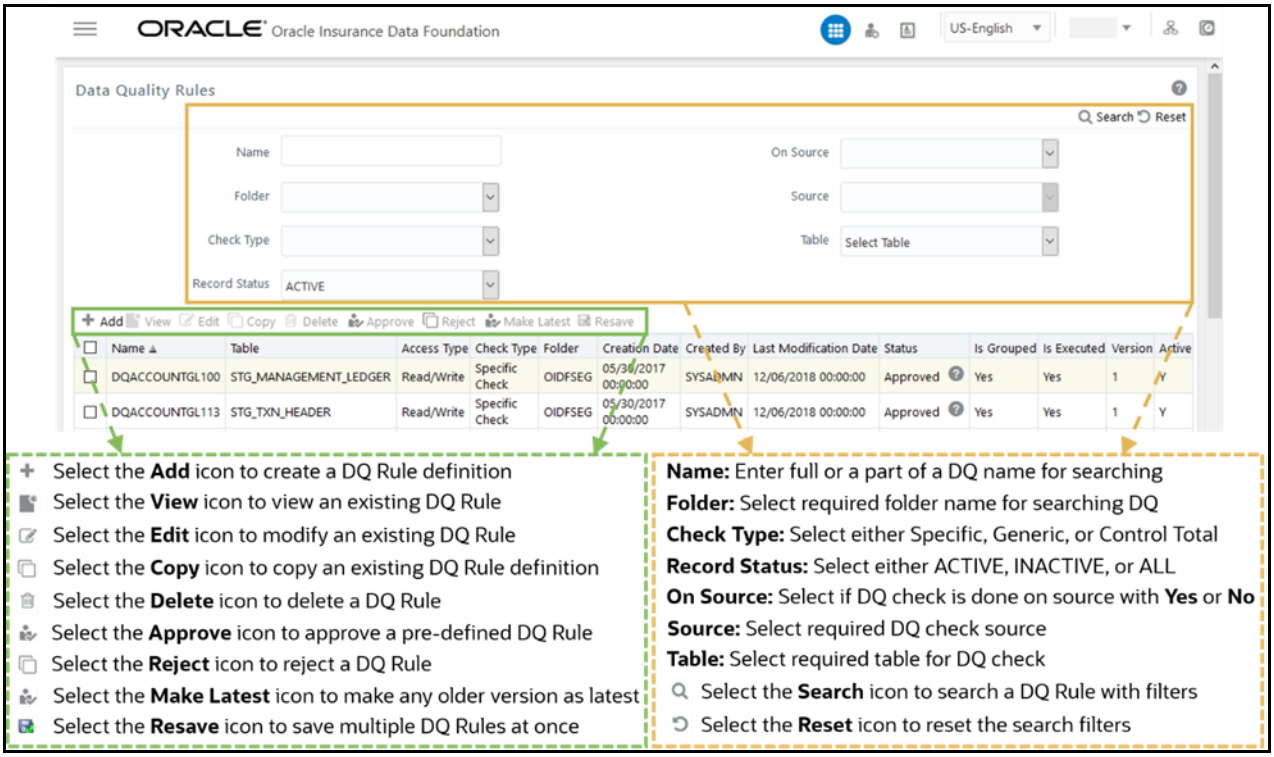
42.1 Data Quality Rules

Data Quality Rules allows you to create a DQ (Data Quality) definition and define 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.

For the roles mapped to the Data Quality Rules, see the *Data Quality Rules* section in the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.

To navigate to the **Data Quality Rules** page, from **OFSAA Home**, select **Oracle Insurance Data Foundation**, select **Data Management Framework**, select **Data Quality Framework**, and then select **Data Quality Rules**. The **Data Quality Rules** page is displayed.

Figure 147: Data Quality Rules page



A defined rule is displayed in Saved status until it is Approved or Rejected by the approver. The approved rules can be grouped further for execution and the rejected rules are sent back to the user with the Approver comments.

You can Add, View, Modify, Copy, Approve or Reject, Resave, or Delete the Data Quality Rules within the **Data Quality Rules** page. You can make any version of a Data Quality Rule as latest. You can also search for a Data Quality Rule based on Name, Folder, Check Type, On Source, Source, Table, or Record Status (Active, Inactive, and All).

To perform different actions related to the Data Quality Rules, see the following sub-sections in the *Data Quality Rules* section of the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0:

- *Creating a Data Quality Rule* section to create a Data Quality Rule
- *Viewing Data Quality Rule* section to view an existing Data Quality Rule
- *Modifying Data Quality Rule* section to edit an existing Data Quality Rule
- *Copying Data Quality Rule* section to copy an existing Data Quality Rule details
- *Deleting Data Quality Rule* section to delete a Data Quality Rule
- *Approving/ Rejecting Data Quality Rule* to approve a pre-defined DQ Rule or reject a DQ Rule
- *Versioning and Make Latest Feature* section to make any older version as latest
- *Resaving Data Quality Rule* section to save multiple Data Quality Rules simultaneously

To execute any Data Quality Rule, a Data Quality Group must be created and the rules for execution has to be mapped with this Data Quality Group. For more information, see the *Executing Data Quality Group*

section in the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.

For the latest information about Data Quality Rules, see the *Data Quality Rules* section in the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.

42.2Data Quality Groups

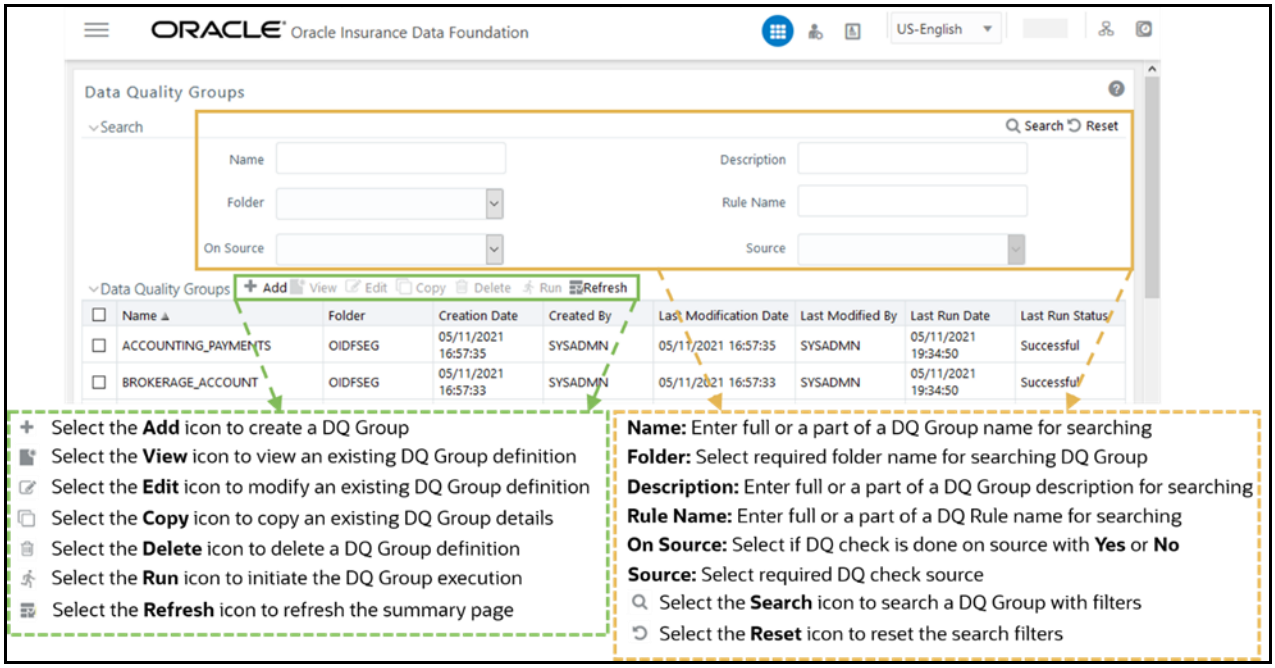
Data Quality Groups facilitates you to logically group the defined DQ definitions and schedule for execution. DQ definitions can be executed either through Data Quality Groups page of Data Management Tools framework or in Batch Execution page of Operations module.

When a Data Quality Group is executed for processing, the details of the execution are captured in a log file.

For the roles mapped to the Data Quality Groups, see the *Data Quality Groups* section in the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.

To navigate to the **Data Quality Groups** page, from **OFSAA Home**, select **Oracle Insurance Data Foundation**, select **Data Management Framework**, select **Data Quality Framework**, and then select **Data Quality Groups**.

Figure 148: Data Quality Groups Page



The **Data Quality Groups** Page displays the list of pre-defined Data Quality Groups with the other details such as Name, Folder, Creation Date, Created By, Last Modification Date, Last Modified By, Last Run Date, and Last Run Status.

You can create and execute DQ Group definitions and View, Modify, Copy, Refresh, or Delete DQ Group definitions within the **Data Quality Groups** Page.

To perform different actions related to the Data Quality Groups, see the following sub-sections in the *Data Quality Groups* section of the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0:

- *Creating Data Quality Group* section to create a Data Quality Group
- *Viewing Data Quality Group* section to view an existing Data Quality Group
- *Modifying Data Quality Group* section to edit an existing Data Quality Group
- *Copying Data Quality Group* section to copy an existing Data Quality Group details
- *Deleting Data Quality Group* section to delete a Data Quality Group
- *Executing Data Quality Group* section to execute a Data Quality Group definition
- *Viewing Data Quality Group Summary Log* section to view the execution log details of Data Quality Rules
- *Viewing Data Quality Report* section to view the execution summary report of Data Quality Rules

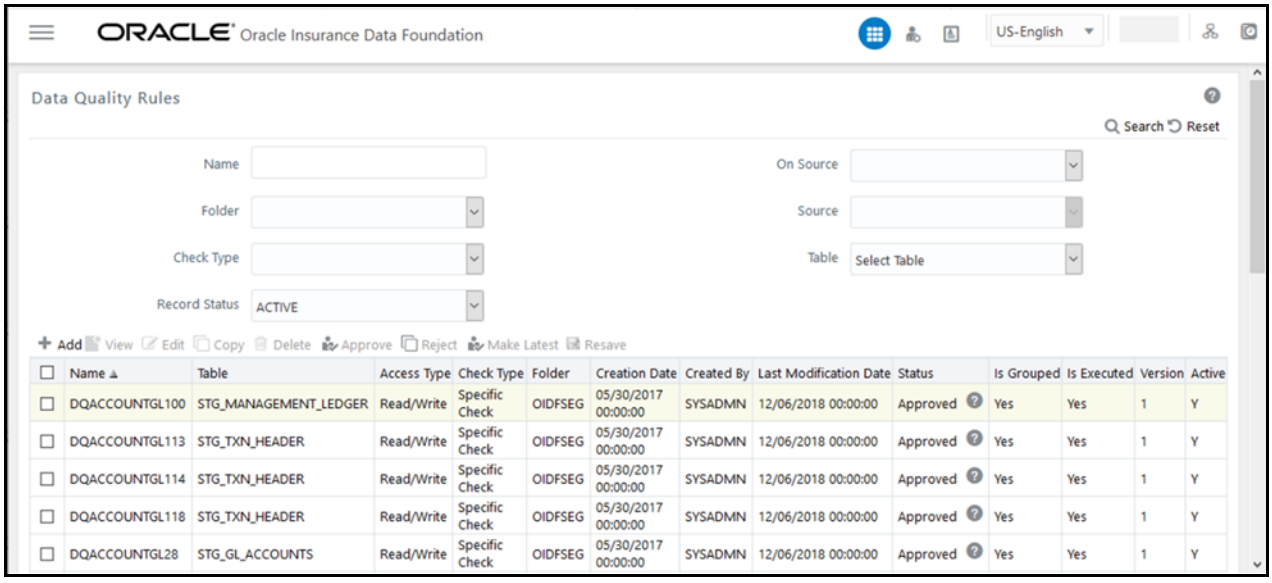
For the latest information about Data Quality Groups, see the *Data Quality Groups* section in the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.

42.3 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 spreadsheet.

In Specific Check, a particular column is checked based on rule's predefined checks, whereas 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.

Figure 149: Specified Data Quality Rules for the Staging tables

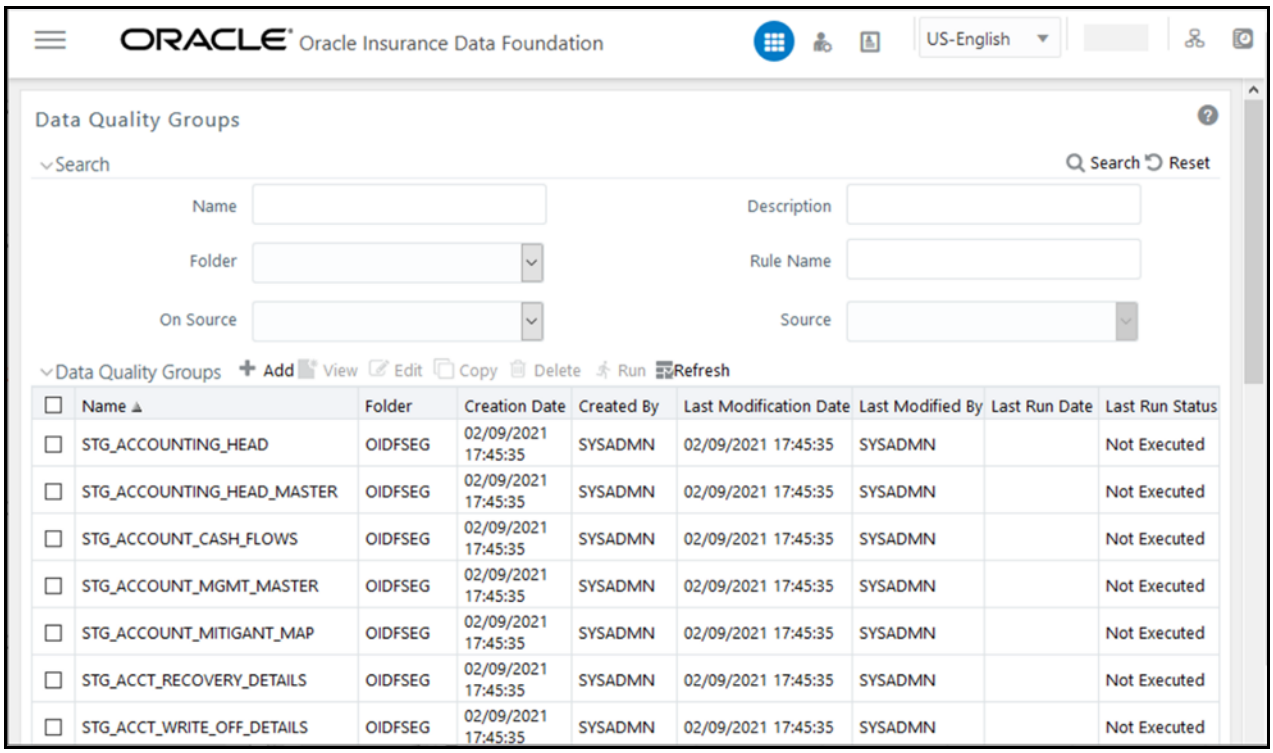


Name	Table	Access Type	Check Type	Folder	Creation Date	Created By	Last Modification Date	Status	Is Grouped	Is Executed	Version	Active
DQACCOUNTGL100	STG_MANAGEMENT_LEDGER	Read/Write	Specific Check	OIDFSEG	05/30/2017 00:00:00	SYSADMN	12/06/2018 00:00:00	Approved	Yes	Yes	1	Y
DQACCOUNTGL113	STG_TXN_HEADER	Read/Write	Specific Check	OIDFSEG	05/30/2017 00:00:00	SYSADMN	12/06/2018 00:00:00	Approved	Yes	Yes	1	Y
DQACCOUNTGL114	STG_TXN_HEADER	Read/Write	Specific Check	OIDFSEG	05/30/2017 00:00:00	SYSADMN	12/06/2018 00:00:00	Approved	Yes	Yes	1	Y
DQACCOUNTGL118	STG_TXN_HEADER	Read/Write	Specific Check	OIDFSEG	05/30/2017 00:00:00	SYSADMN	12/06/2018 00:00:00	Approved	Yes	Yes	1	Y
DQACCOUNTGL28	STG_GL_ACCOUNTS	Read/Write	Specific Check	OIDFSEG	05/30/2017 00:00:00	SYSADMN	12/06/2018 00:00:00	Approved	Yes	Yes	1	Y

42.4 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.

Figure 150: Specified Data Quality Groups for the Staging tables



For more information regarding Data Quality Group execution, see the *Executing Data Quality Group* section in the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.

The *DQ Group Mapping* sheet of the Oracle Insurance Data Foundation Application Pack Runchart document of this release, which displays the total groups and corresponding Rules mapped to that group.

42.5Batch 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 DQ Groups, simultaneously.

You can also use the Include or Exclude functionality to determine which all groups have to be executed. The following batch needs to be executed in OIDF:

<INFODOM>_DATA_QUALITY_BATCH_OIDF

See the RUN_CHART_SUMMARY sheet of this release Runchart document of this release.

For more information about the Batch Execution, see the *Operations* section of the Oracle Financial Services Advanced Analytical Applications Infrastructure User Guide Release 8.1.2.0.0.

43 APPENDIX A: Naming Conventions Used in OIDF Data Model

This Appendix chapter explains the various naming conventions used in OIDF Logical Data Model and Physical Data Model. In addition, the domains in PDM is also listed with description.

Topic:

43.1 Naming Conventions Used in OIDF 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 Insurance Data Foundation (OIDF) Logical Data Model (LDM) and Physical Data Model (PDM).

43.1.1 OIDF Physical Data Model Naming Conventions

The OIDF 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 OIDF 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 OIDF Physical Data Model.

43.1.1.1 PDM Logical View Mode Naming Conventions

In the logical view model, OIDF 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 25: 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 Reinsurance Contracts Held
	Transaction Tables		Transactions	Stage Reinsurance Issued Transactions

PDM Area	Table Type	Prefix	Suffix	Example
	Master Tables		Master	Stage Reinsurance Contract Type Master
Results (Datamart)	All Fact Tables	Fact		Fact Policy Transactions
	All Dimension Tables		Dimension	Reinsurance Contract Type Dimension

43.1.1.2 PDM Physical View Mode Naming Conventions

In the physical view model, OIDF 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 26: 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_REINSURANCE_CONTRACTS_HELD
	Transaction Tables		_TXNS	STG_REINSURANCE_ISSUED_TXNS
	Master Tables		_MASTER	STG_RI_CONTRACT_TYPE_MASTER
Results (Datamart)	All Fact Tables	FCT_		FCT_POLICY_TRANSACTIONS
	All Dimension Tables	DIM_		DIM_RI_CONTRACT_TYPE

The table below lists the prefix and/or suffix used for columns names in the physical view of the OIDF PDM. The prefix or suffix depends on the class and data type of the column.

Table 27: 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 28: 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 29: 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

Name	Abbreviated Form
End Of Period	eop
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

43.1.2 Domains (PDM and LDM)

Domains are Logical data types that are attached to each column within the model. The following table lists the domains and their descriptions.

Table 30: Domains and their descriptions

Domain Name	Domain Description
Date	DATE
Timestamp	TIMESTAMP
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)

44 APPENDIX B: Standard Data Expectations

This section provides information about the standard data expectations in the OIDF Application Pack.

44.1 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 OIDF considers 8.9 as the value in the column instead of 0.089, because 0.089 is not valid as interest rate value.

45 APPENDIX C: How to Define a Batch

This Appendix provides information about How to Define a Batch in the Oracle Insurance Data Foundation application and step-by-step instructions to use this section.

Topics:

45.1 Batch Definition

To create a batch using the OFSAAI Batch Maintenance page, follow these steps:

1. From the **OFSAA Home**, select **Operations**, select **Batch Maintenance**.
2. In the **Batch Maintenance** page, Select the + button from the Batch Name toolbar. The **New Batch Definition** page is displayed.
3. Enter the Batch Definition details as tabulated.

Table 31: Batch Definition fields and their description

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.
5. The new Batch definition details are displayed in the *Batch Name* section of the **Batch Maintenance** page with the specified **Batch ID**.

46 APPENDIX D Reporting Code Description

This section provides the Reporting Code description used in the OFSAA Data Model.

46.1 Reporting Code Description

Table 32: The list of Reporting Codes and their description

Report Code	Report Name
BS-C1	Balance sheet
BS-C1B	Off-balance sheet items
BS-C1D	Assets and liabilities by currency
Country - K1	Activity by country
Cover - A1A	Premiums, claims & expenses - Annual
Cover - A1Q	Premiums, claims & expenses - Quarterly
OF - B1A (solo)	Own funds (annual template - for solo entities)
OF - B1A (group)	Own funds (annual template - for groups)
OF - B1Q (solo)	Own funds (quarterly template - for solo entities)
OF - B1Q (group)	Own funds (quarterly template - for groups)
VA - C2A	Summary analysis of changes in BOF
VA - C2B	Analysis of changes in BOF due to investments
VA - C2C	Analysis of changes in BOF due to technical provisions
VA - C2D	Analysis of changes in BOF due to its own debt and other items
SCR - B2A	Solvency capital requirement (for the undertaking on standard formula or partial internal models)
SCR - B2B	Solvency capital requirement (for undertakings on partial internal models)
SCR - B2C	Solvency capital requirement (for the undertaking on full internal models)
SCR - B3A	Solvency capital requirement - market risk
SCR - B3B	Solvency capital requirement - counterparty default risk
SCR - B3C	Solvency capital requirement - life underwriting risk
SCR - B3D	Solvency capital requirement - health underwriting risk
SCR - B3E	Solvency capital requirement - non-life underwriting risk
SCR - B3F	Solvency capital requirement - non-life catastrophe risk
SCR - B3G	Solvency capital requirement - operational risk
MCR - B4A	Minimum capital requirement (except for composite undertakings)
MCR - B4B	Minimum capital requirement (for composite undertakings)

Report Code	Report Name
Assets - D1	Investments Data - Portfolio list (detailed list of investments) - Annual
Assets - D1Q	Investments Data – Quarterly (Portfolio list or Quarterly summary)
Assets - D1S	Structured products Data - Portfolio list
Assets - D2O	Derivatives data – open positions
Assets - D2T	Derivatives data - historical derivatives trades
Assets - D3	Return on investment assets (by asset category)
Assets - D4	Investment funds (look-through approach)
Assets - D5	Securities lending and repos
Assets - D6	Assets held as collateral
TP - F1	Life and Health SLT Technical Provisions - Annual
TP - F1Q	Life and Health SLT Technical Provisions - Quarterly
TP - F2	Projection of future cash flows (Best Estimate - Life)
TP - F3	Life obligations analysis
TP - F3A	Only for Variable Annuities - Description of guarantees by product
TP - F3B	Only for Variable Annuities - Hedging of guarantees
TP - F4	Information on annuities stemming from Non-Life insurance obligations
TP - E1	Non-Life Technical Provisions - Annual
TP - E1Q	Non-Life Technical Provisions - Quarterly
TP - E2	Projection of future cash flows (Best Estimate - Non-life)
TP - E3	Non-life Insurance Claims Information
TP - E4	Movements of RBNS claims
TP - E6	Loss distribution profile non-life
TP - E7A	Underwriting risks (peak risks)
TP - E7B	Underwriting risks (mass risks)
Re - J1	Facultative covers non-life & life
Re - J2	Outgoing Reinsurance Program in the next reporting year
Re - J3	Share of reinsurers
Re - SPV	Special Purpose Insurance Vehicles
G01	Entities in the scope of the group
G03	(Re)insurance Solo requirements
G04	Non-(re)insurance Solo requirements
G14	Contribution to group TP
G20	Contribution to Group SCR with D&A
IGT1	IGT - Equity type transactions, debt & asset transfer

Report Code	Report Name
IGT2	IGT - Derivatives
IGT3	IGT - Internal Reinsurance
IGT4	IGT - Cost sharing, contingent liabilities, off BS items and other IGT
RC	Risk Concentration

47 APPENDIX E Template to Generate Data Dictionary and Download Specification for erwin Data Modeler 2019R1 or a higher version

OFSAA data models for this release have been designed and released on 2019R1 or a higher version of erwin Data Modeler application for 8.1.2.3.0 and earlier versions. For 8.1.2.4.0 and later versions, the minimum supported version of erwin data model application is 12.1. 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 Data Modeler application 2019R1 or a higher version, for all OFSAA data models qualified on OIDF 8.1.2.0.0.

For the compatible version of the erwin Data Modeler with this OIDF Application Pack release, see the Oracle Financial Services Analytical Applications (OFSAA) Technology Matrix Release 8.1.2.0.0.

The prerequisite is 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 OIDF version 8.1.2.0.0 and erwin Data Modeler application 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 Data Modeler application 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 OIDF 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.

48 APPENDIX F About OIDF Documents

This section contains information about the OIDF Application Pack documentation resources.

The following are the types of available document resources:

- Product documents
- Technical documents

48.1 Product Documents

This resource consists of the product and data model end-user documents containing the functional, installation, and release details. The document types are as follows:

- **User Guide:**

  This document provides functional information about the features and components in the OIDF application pack including the OIDF data model. The following information is explained in detail:

  - OIDF Architecture and data model
  - OIDF Subject Areas
  - How to use OIDF
  - Data Domain Browser (DDB) implementation in OIDF
  - Mandatory Dimensions, T2T and SCD processes and execution, and loading multiple load Runs.
  - Executing OIDF processes using Process Modelling Framework (PMF), prior dated (backdated) processing using the Restatement Support feature, Metadata Browser, and the process of comparing the data model reports.
  - T2T and SCD metadata and the mapping related to Party, Product, Underwriting, Insurance Contracts, Telematics, Pandemic Data, Common Account Summary, Loan Account Summary, Transactions, General Ledger Data and Management Reporting, Insurance Claims, Actuarial, Common Account Summary, Credit Score Model and Probability of Default Model, and other tables.
  - Big Data implementation in OIDF
  - Data Quality Rules execution

- **Installation and Configuration Guide:**

  This document provides the prerequisites, installer, and data model patches, configuration, installation, and deployment information related to the OIDF application pack for both the fresh installation and upgrade installation scenarios. This document also contains detailed Big Data modes of installation and configuration information.

- **Release Notes:**

  This document provides OIDF application pack release related information. The release related information contains details about the release installer and data model patches, prerequisites, new features and enhancements, bug fixes, any known issues or limitations, and workarounds.

- **Data Model documents**:

The following are the OIDF data model documents:

- **OFSAA Data Model Extension Guidelines Document:**

  This document provides information about the extension process of the OFSAA data model and a sample, usage, and functions performed by each area of the model, the modelling methodology employed in each of the functional areas representing the usage pattern, data model structure, data model life cycle process, design guidelines for the staging, processing and reporting areas, configuring Entity UDPs for Application Data Interface of Data Integration Hub, creating the Subtype and Supertype relationship between two entities, deploying SQL Data Modeler configuration files and generating customized data model, data model merge guidelines and build guidelines.

- **OFSAA Data Model Naming Standards Guide:**

  This document provides details of the naming standards followed in the development process of the OFSAA data model for the tables, columns, PK and FK, keywords and domains information, table UDP and column UDP information.

- **Data Protection Implementation Guide:**

  This document provides GDPR related data protection methods implemented in the OFSAA applications such as Data Redaction, Right to be Forgotten, Data Portability, Pseudonymization, Notice and Consent, and Data Archival.

- **Cloning Reference Guide:**

  This document provides detailed steps to set up an OFSAA Instance Clone for the 8.1.x.x.x and 8.0.x.x.x releases in a faster and effective approach for further project developments. This document contains the approach to set up the OFSAA instances that are exact copies of the current OFSAA instance.

- **Security Guide:**

  This document contains information about the security parameters configuration provided by Oracle Financial Services Analytical Applications (OFSAA) and how to set it after the installation of the OIDF Application Pack.

You can download the OIDF Application Pack Release 8.1.x.x.x User Guide, Installation and Configuration Guide, Release Notes documents from the OHC Documentation Library.

You can download the OIDF Application Pack Release 8.1.x.x.x Data Model documents, Data Protection Implementation Guide, Cloning Reference Guide, and Security Guide from the OHC Documentation Library.

You can download the OIDF Application Pack Release 8.0.x.x.x Cloning Reference Guide from the OHC Documentation Library.

48.2 Technical Documents

This resource consists of the product and data model documents containing the technical details required for installing, upgrading and using the OIDF Application Pack, analyzing and using the data model, implementing data security, analyzing the technical metadata changes between the current release and the previous release. The document types are as follows:

- **Data Model Difference Report:**

This document consists of the changes between the current release version and the previous release versions of the data models. The details such as new tables and columns, dropped tables and columns, data type change, not Null change, domain change, PK and FK changes, and default value change are included in this document.

- **Run Chart:**

  This document consists of the following technical information required to set up, implement, and use the OIDF Application Pack:

  - Technical metadata batches and Run processes
  - Seeded data related table names, data load mode, seeded data type
  - Data Quality Rules and Data Quality Rule Groups
  - Dimension data loading
  - Exchange Rates loading
  - Mapper loading
  - Source Run data load processes
  - Execution Run data load processes

- **Changelog:**

  This document consists of the technical metadata changes between the current release and the previous release of the OIDF Application Pack, which is required for analysis before proceeding with the implementation process.

- **SCD Metadata:**

  This document consists of the SCD metadata technical details such as SCD map reference number, Dimension table, and column names and their corresponding Stage table and column names, column type and data type, and SCD type ID.

- **T2T Metadata:**

  This document consists of the T2T metadata technical details such as T2T name, Join and Filter conditions, source table and column names and corresponding target table and column names, and Expression.

- **Download Specifications Documents:**

  This document consists of the mapping from the Staging Data Model and provides an efficient way to manage the sourcing of data into the OIDF staging area. This document consists of the details of 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).

- **PII Datasheet:**

  This document consists of the Personally Identifiable Information (PII) dependent tables and columns on which the Data Redaction is applied in the current release of the OIDF Application Pack.

- **Upgrade Guide:**

  The document provides the approach for the upgrade of the OIDF Application Pack from the 8.0.x release versions to the v8.1 release containing the OOTB data model. Here, OIDF manages OFSAA common data model and associated metadata. This document consists of the different stages of the upgrade process, the assumptions made for the upgrade process, and the data and metadata

approach. A utility, called the ModelUpgrade is introduced in the OIDF Application Pack Release v8.1.0.0.0 to enable the upgrade installation using Incremental model upload. Using this utility, you can execute the packaged Prescripts before the upgrade and the Postscripts after the upgrade to handle the data and metadata.

- **OFSAA Licensing Information User Manual:**

  This is the Licensing Information document, which is a part of the product or program documentation under the terms of your Oracle license agreement and is intended to help you understand the program editions, entitlements, restrictions, prerequisites, special license rights, and (or) separately licensed third-party technology terms associated with the Oracle software program(s) covered by this document.

You can download the OIDF Application Pack Release 8.1.x.x.x technical documents from the My Oracle Support (MOS) Doc IDs 2099161.1 and 2197857.1.

OFSAA Support

Raise a Service Request (SR) in My Oracle Support (MOS) for queries related to OFSAA applications.

Send Us Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, indicate the title and part number of the documentation along with the chapter/section/page number (if available) and contact the Oracle Support.

Before sending us your comments, you might like to ensure that you have the latest version of the document wherein any of your concerns have already been addressed. You can access My Oracle Support site that has all the revised/recently released documents.