Oracle Insurance Data Foundation

User Guide

Release 8.0.7.0.0

May 2019

Oracle Insurance Data Foundation

Document Control

Version Number	Revision Date	Change Log
1.0	24-December-2018	• Launched OIDF on Hive with two Big Data deployment processes. They are, Stage and Results on Hive, and Stage on Hive and Results on RDBMS with the following functionalities. • The following enhancements are introduced: ▪ Party Producer Exam, Certification, and License ▪ Party Identification Document Sub Type ▪ Underwriting — Party Criminal Conviction — Party Driving Violation — Lifestyle Activity — Party Life Style Activity — Life Style Activity Aviation Experience Details — Life Style Activity Underwater Diving Experience Details — Life Style Activity Racing Experience Details — Life Style Activity Military Experience Details — Life Style Activity Foreign Travel Experience Details — Life Style Activity Competition Details — Life Style Activity Climbing Experience Details — Life Style Activity Air Sports Experience Details ▪ In Insurance Contracts: — Reinsurance Contract Held — Reinsurance Contract Issued — Other Reinsurance tables ▪ Policy Schedules • Added: ▪ New T2Ts for Fact Policy Transactions table — Capital and Borrowings — Placed Collateral — Legal Entity Population — Insurance Claims — Trading and Investments — Instruments Contracts — Metadata Browser — Time Dimension — Recommendation for Back dated Run — Compare Data Model Reports — Standard Data Expectations — How To Define a Batch — Updated Data Quality Rules Execution screenshots
2.0	17-May-2019	• The following enhancements are introduced:

Version Number	Revision Date	Change Log
		▪ Underwriting Application ▪ Underwriting Quotes ▪ Claim Details ▪ Claim Claimant Mapping ▪ Mapper Hierarchy in GL Data to Management Reporting ▪ New Subject Areas • Added new T2Ts for: ▪ Insurance Assumed Policy Cash Flow ▪ Party Insurance Policy Role Map • Added new SCDs for: ▪ Policy Dimension ▪ Document Submission Status Dimension ▪ Beneficiary Dimension • Added Seeded data for: ▪ Insurance License Agency Dimension

Table of Contents

1 Preface

This chapter discusses the following topics:

1.1 Intended Audience

Welcome to Release 8.0.7.0.0 of the Oracle Insurance Data Foundation User Guide.

This document is the user guide and reference guide for the Oracle Insurance Data Foundation (OIDF) release 8.0.7.0.0, and is intended for System Administrator and all users who are instrumental in configuring and administering OIDF with Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) platform. See Related Information Sources for more Oracle product information.

1.2 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

1.3 Access to Oracle Support

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

1.4 Related Information Sources

Refer to the below documents in OHC Documentation Library:

- Technical Whitepaper on Data Model Document Generation (OHC)
- Oracle Insurance Data Foundation Installation Guide release 8.0.7.0.0 (OHC)
- Oracle Insurance Data Foundation Installation Guide release 8.0.7.1.0 (OHC)
- OFSAAI Installation and Configuration Guide release 8.0.7.0.0 (OHC)
- Oracle Insurance Data Foundation Runchart release 8.0.7.1.0 (My Oracle Support)

2 Introduction to OIDF

This chapter 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.

This chapter includes the following 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 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 described in additional detail in the next chapter:

Criteria	Description
Logical Data Model (LDM)	LDM is a reference model of the Insurance domain for various types of Insurance. This model captures the business processes of a typical Insurance institution in detail. Logical Data Model means it cannot be readily used. The structures modeled in the OIDF LDM provide an abstract, graphical model of the Insurance industry domain using Entity-Relationship modeling. This can be considered as a detailed blueprint for organizing data within an Insurance institution. It provides a reference guide for institutions to understand the salient data related to a specific business process. To use the OIDF application, customers deploy OIDF Analytical Warehouse Model, which is derived from the LDM (blueprint).

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, unlike the OIDF LDM, meaning that it 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 Infrastructure).
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 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 (8.0.7.0.0) 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:

Component	Provider	Purpose
Oracle Financial Services Analytical Applications Infrastructure version 8.0.7.0.0	Oracle	OFSAAI is the platform on which the OIDF application is deployed and operated. It represents 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 11gR2 and 12cR1	Oracle	OIDF is certified on Oracle Database releases 11gR2 and higher.
ERwin data modeler version 9.64.01	Computer Associates (CA)	ERwin is a data modeling tool 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 chapter explains the background of OIDF, its functional architecture along with the differences from traditional warehouse architecture. OIDF product package consists of the Logical Data Model and Analytical Data Warehouse model. The individual components of Physical Data Model is explained in detail.

This chapter covers the following topics:

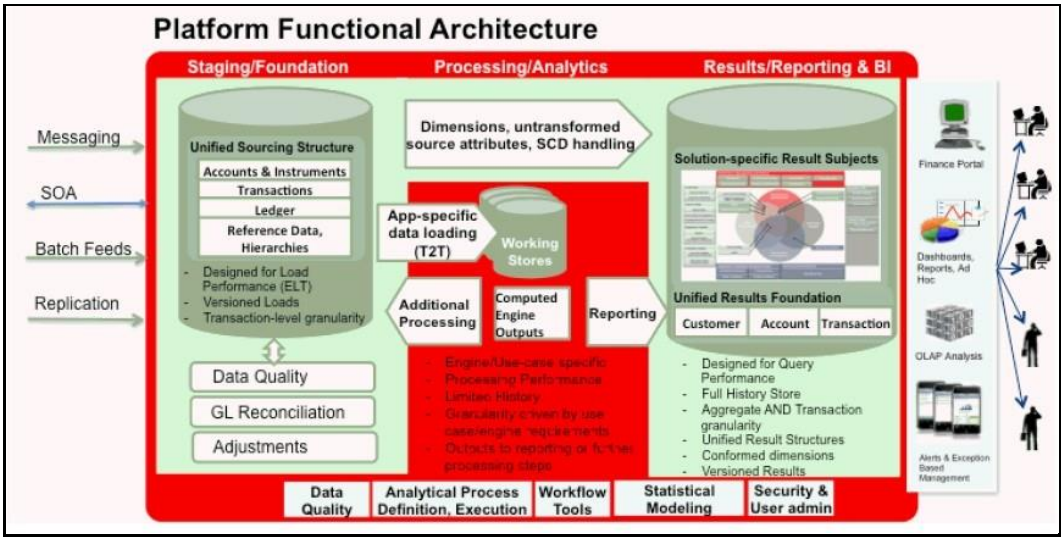
3.1 Background

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.

3.2 OIDF Architecture

The following figure depicts the OIDF Functional Architecture.



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

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

The architecture illustrates the following key concepts:

- A unified data-sourcing area for analytics: The OIDF Staging Data Model provides the basis for central, unified data-sourcing layer for a variety of analytical needs. The staging layer faces the operational (OLTP) and front office systems of an Insurance Company. It consists of tables to capture key business and operational data from these systems, which is then processed by one or more analytical engines.
- A unified reporting/consumption layer: Analytical results can be simple to complex, quantitative and qualitative measures of a provider's Risk, Compliance, Customer and Fund Performance. The OIDF Reporting data model is a dimensional data model spanning these key analytical functions. It forms the foundation of OFSAA Business Intelligence applications, but can clearly be used as the result data store for any equivalent engines and processes provided by other vendors, or custom-built solutions. By providing a single data repository for reporting needs, this layer provides a foundation for departmental and cross-departmental and cross-functional reporting.
- Additionally, the schema of the reporting area is specifically built for Insurance analytics. As an illustration, the reporting area has a 'Run dimension' that is shared across all BI/reporting use cases. Similarly, the aggregation of shared measures/reporting into a common set of output structures (Account Summary) allows for cross-functional reporting, which is increasingly the norm in Insurance institutions.
- Single point of control and operation: The Oracle Financial Services Analytical Applications Infrastructure is a separate Oracle product that offers a set of tools that are built on a common metadata foundation. These tools are used to control and manage the lifecycle of data from sourcing to reporting. There is a collection of frameworks to manage the following lifecycle steps of data within OIDF:
  - Metadata Management
  - Data Quality
  - Data Movement
  - Scheduling and runtime operations
  - Security/User management
  - Analytical Process Definition and Execution
- Processing Area: As explained earlier, the primary purpose of the OIDF is to serve as a container for analytical processing from sourcing to consumption. Such processing is usually delivered in the form of discrete units called analytical applications spanning different analytical use cases ranging from Policy to Claims to Fund Allocation.
- These applications consist of custom-built computational engines and numerical libraries, and may execute processes on the data that range from simple aggregations to complex, multi-step stochastic processes such as Monte-Carlo simulation.

- Hence, analytical applications place varying demands on the data infrastructure in terms of volumes and speed, and hence place different demands on data architecture. In practice, the normalized (3NF) design favored for
- Enterprise Data Warehouses often fails to be efficient or performant when it comes to analytical processing across a wide range of use cases.
- Therefore, the OIDF recognizes the need for distinct application-specific working stores, separate from the staging and reporting area.

ATTENTION

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

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

3.3 Differences from Traditional Warehouse Architecture

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

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

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

3.4 Subject Areas Organization

The Subject Areas are organized to support detailed level analysis related to set higher-level analytical reporting solutions.

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

Table: OIDF Subject Areas Organization

Subject Area	Description
OIDF Legal Entity Details	This subject area covers the legal entity or organization's structure related tables.
OIDF Party Contacts	This subject area covers the party contact details such as party address, party phone details and email details.
OIDF Party Definition	Party here refers to any person or organization who interacts with the company. All the parties associated with insurance company under different roles like policyholder, producer, agents, issuer etc. are loaded together in this entity.
OIDF Party Identification	This subject area covers the identification details of an individual Party or organization. Identification means the action or process of identifying someone or something or the fact of being identified.
OIDF Party Employment	The subject area covers the employment details of an individual Party. This data applies only to Parties who are individuals. A single Party may have multiple employment details.
OIDF Party Exam and Certificates	The subject are covers the details pertaining to the party's certification and related exams to it. An Exam is the formal verification or validation of information learned/known, typically used to satisfy a requirement for licenses, registrations or other firm education proof. Certification is formal procedure by which an accredited or authorized person or agency assesses and verifies (and attests in writing by issuing a certificate) the attributes, characteristics, quality, qualification, or status of individuals or organizations, goods or services, procedures or processes, or events or situations, in accordance with established requirements or standards.
OIDF Party Financials	This subject area covers the financial information (Balance-Sheet, Profit and Loss statement and Ratios) of the parties. Balance sheet is prepared as of a particular date (Balance sheet creation date).
OIDF Party Medical	This subject area covers all the medical tables related to party medical details such as party medical test details, family medical test data, and party disability details.
OIDF Party Results	This subject area covers the party result tables such as party ratings, party party relationship, party party account relationship.
OIDF Party Customer Results	This subject area covers OIDF Party Customer Results related tables.
OIDF Insurance Contracts	This subject area covers the insurance contract tables. Contract is defined as where 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
OIDF Trading and Investments	This subject area covers OIDF Trading and Investments related tables.

Subject Area	Description
OIDF Capital Borrowings	This subject area covers OIDF Capital Borrowings related tables. Capital Borrowings refers to borrowing done by Insurance company using debt instruments.
OIDF Capital Instruments	This subject area covers the capital Instrument details, it refers to an issue of capital instrument to raise funds from market. This primarily covers equity instruments. They may or not be traded in market.
OIDF Fixed Assets	This subject area covers OIDF Fixed Assets related tables.
OIDF Financial Assumptions	This subject area covers the financial assumptions details. Assumption is an estimate of an uncertain variable input into a financial model, normally for the purposes of calculating premiums or benefits. This subject area covers the tables related to Insurance Assumptions such as Fact Insurance Lapse Rate Assumptions.
OIDF Demographic Assumptions	This subject area covers the demographic assumptions details. For example, assumption relates to predicting a person's lifespan, given their age, gender, health conditions and other factors.
OIDF Market Data	This subject area covers OIDF Market Data related tables.
OIDF Payment Data	This subject area covers OIDF Payment Data related tables.
OIDF Accounting and General Ledger	This subject area covers the accounting and general ledger tables associated with the insurance company.
OIDF Insurance Transactions	This subject area covers OIDF Insurance Transactions related tables.
OIDF Insurance Claims	This subject area covers OIDF Insurance Claims related tables.
IFRS Accounting Outputs	This subject area covers the IFRS accounting related tables such as contractual service margin, homogenous risk group tables.
OIDF Actuarial Outputs	This subject area covers OIDF Actuarial Outputs related tables.
OIDF Solvency ll	This subject area covers OIDF Solvency ll related tables.
OIDF Sales and Marketing	This subject area covers OIDF Sales and Marketing related tables.
OIDF Underwriting Application	This subject area holds the applications processed in each period, for an analysis over time.
OIDF Underwriting Quotes	This subject area covers the insurance quotes details. Quotes are subject to change depending on information given at time of the quote.
OIDF Underwriting Crime Data	This subject are covers the criminal conviction details if the party is involved in any criminal activity. Criminal conviction is the outcome of a criminal prosecution which concludes in a judgment that the defendant is guilty of the crime charged.
OIDF Underwriting Driving Violation	This subject area covers the driving violation details related to the party identifier. Driving violation is any violation of vehicle laws that is committed by the driver of a vehicle, while the vehicle is moving.
OIDF Underwriting Life Style Activity	This subject area covers the information about a party's lifestyle activities. This section captures details surrounding the activities a party engages in that may be considered risky by insurance companies. In the Lifestyle Activity object, the more specific experience sub-objects (e.g. Aviation Experience, Racing Experience, Underwater Diving Experience, Climbing Experience, Air Sport Experience, and Foreign Travel) are optional, singly occurring and mutually exclusive.
OIDF Commonly Used Tables	This subject area covers all the common tables such as date, country, and location.

The list of solutions supported by Results only OIDF Subject Area are tabulated here.

Subject Area Name	Definition
ALM	Subject Areas corresponding to Asset Liability Management (ALM).
Regulatory Capital Calculation	Subject Area pertains to the Solvency II Regulatory Framework and its reporting requirements as specified in the framework.
PFT	Supports reporting related to Profitability analysis, part of the Enterprise Performance Management solution area.
OIPI	Supports reporting requirements of Insurance Analytics.
Economic Capital	Support for Aggregated Economic Capital Reporting based on risk assessments for Market, Credit and Operational Risk.
GL Reconciliation	Support for reporting specific to the Oracle GL reconciliation module that is part of the OFSAA product line. This allows a reporting view of the reconciliation processes and outputs.
CRM	Part of the Corporate Credit Risk Solution, allows reporting on Credit Limits.
Market Risk	Support for Market Risk analytics
IFRS 17	Supports the data requirement pertaining to the IFRS 17.
Centralized Reporting	Supports the centralized reporting for an insurance company covering insurance life cycle activities from underwriting to Claim settlement including risk management use cases as mentioned above.

3.5 OIDF Logical Data Model

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

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

Area	Logical Data Model	OIDF Physical Model
Purpose	Generic blueprint of the data reduced by the business processes of an Insurance Company.	Designed for analytical processing (IFRS17, Solvency, Customer Insight, Risk, Regulatory Reporting).
Format/Structure	Entity Relationship Diagrams organized into key Top Level subject areas.	Physical Tables/Columns Definitions.
Usage	Cannot be directly used – needs to be used within a physical database design step to arrive at a database schema.	Readily deployable.

Area	Logical Data Model	OIDF Physical Model
Scope	All the key business processes/activities and their supporting reference data requirements.	The business and reference data required for a set of analytical use cases. However, can be extended to serve other needs.

3.6 OIDF Physical Data Model

The OIDF Physical Data Model is the primary deployed structure in the OIDF. As detailed in the architecture section, it has the following key characteristics:

- Readily Deployable: The Physical Data Model is a readily deployable physical schema. It is provided as an ERwin data model file (for details on ERwin, see http://erwin.com/products/modeler/) and consists of tables grouped into distinct subject areas depending on function. The tables are either used to gather source data (Staging Area), or as containers of outputs/results from analytical processing and engines for reporting purposes (Reporting Area).
- Use-case Driven: The OIDF Physical Data model is driven by a set of clearly identified analytical use cases spanning Performance, Experience, and Compliance.
- Extensible: While the OIDF Physical Data Model satisfies a very large number of analytical use cases across Insurance Contracts, Claims, Underwriting, Actuarial Assumption, Financial Assumptions, Solvency, IFRS 17 subject areas, customers can find the need to customize the model for a specific installation.

These customizations can be done in accordance with guidelines published in the Chapter "Using OIDF," section of this manual.

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

- Staging Data Model
- Reporting Data Model

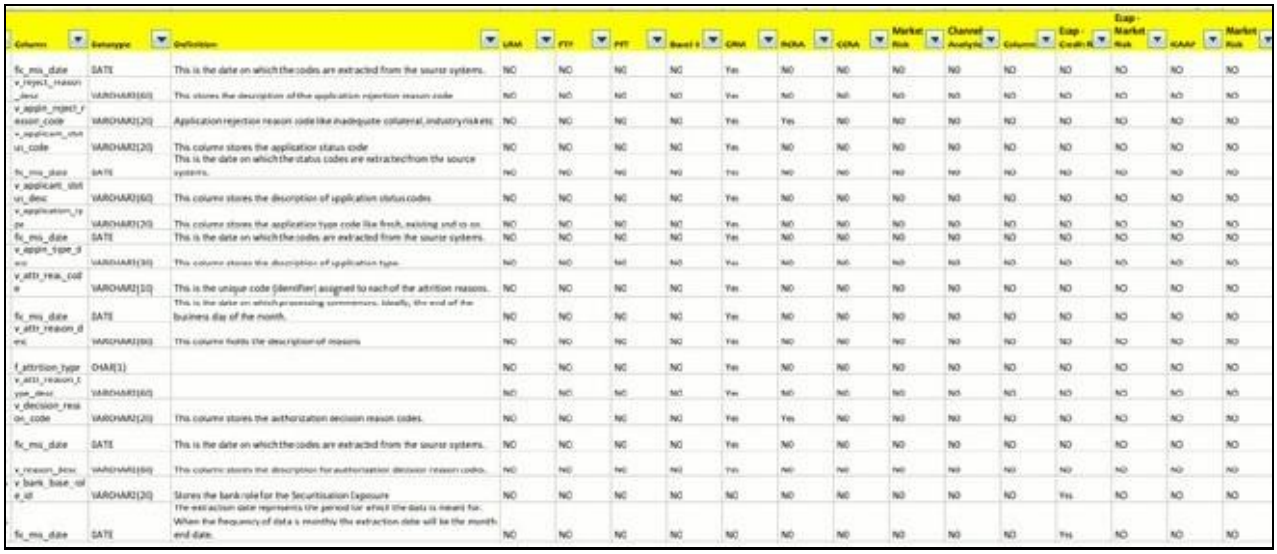
3.6.1 Staging Data Model

3.6.1.1 Overview/Design

The Common Staging Area Model (CSA) represents the point of entry of data into the OIDF. The CSA provides a simplified, unified data sourcing area for inputs required by analytical applications and engines. It consists of over 400 tables and nearly 9000 columns organized into distinct subjects. The salient features of the CSA are as follows:

- Mapping to Analytical Use Cases: Since the primary purpose of the OIDF is to be a data repository supporting analytics, each database object in the OIDF physical data model is necessarily mapped to a corresponding analytical use case.
- These mappings are captured in the data model, in the form of additional metadata called User-defined Properties (UDPs), and can be leveraged to reduce the scope of data gathering efforts by focusing on clearly-defined end use cases such as Market Risk Analytics, and others.
- These mappings can readily be extracted into a Download Specification, which lists the data demands for a specific analytical use case. An example is shown follows:

Figure 4. Download Specification



- The Mappings can be generated from the OIDF ERwin file using ERwin's reporting tools.
- Schema Design: The data model of the Common Staging Area is designed to facilitate loading efficiency. This means that it is denormalized.
- The primary focus of the staging area is to efficiently deliver data from operational systems such as Policy Administration, Claims Processing and Fund Management to multiple analytical applications that process this data.
- Typically, this data is extracted from source systems and loaded into OIDF directly, or alternatively into a pre-defined file-based operational image area from which it is subsequently loaded into the OIDF schema. In a large Insurance institution, it is not unusual to have 10s to 100s of millions of policies, and claims.
- Standard ETL (Extract, Transform, Load) approaches can fail to address this requirement adequately because of the complexity imposed by the target schema. If the target schema is increasingly normalized, then the ETL into this schema is correspondingly more complex, requiring careful load ordering to prevent integrity-related load failures, as well as integrity validation prior to loading. Such complex ETL processing is time-consuming and is prone to failure.
- To address this, the CSA is designed to support a simplified loading process. De-normalized tables allow efficient data loading, and subsequent transformations can be done to verify data integrity through a series of data quality checks. This represents an 'ELT (Extract Load Transform)' approach to data sourcing, which is far more suited for an analytical data repository.
- Application-managed Referential Integrity (RI): In conjunction with the database design of the staging schema, a key feature is the management of Referential Integrity primarily in the application tier, rather than within the database. Rather than imposing foreign key relationships in the database, which could cause complex loading order dependencies, relationships between staging tables are managed by the Data Quality (DQ) framework, a toolkit within the Analytical Application Infrastructure that captures the relevant data quality checks for a specific table in the staging model. These checks include:
  - Value checks (Nulls, Value ranges, business checks on numeric fields)
  - Referential Integrity checks (which are otherwise implemented as foreign key constraints)

ATTENTION This is also why the ER model of the staging area in ERwin does not contain any relationships – the staging area is a physical data model,

which is deployed using the Analytical Application Infrastructure, which manages it.

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

3.6.1.2 Details of the Staging Data Model

The CSA model consists of over 400 tables and nearly 9000 attributes. These tables are organized into multiple 'subjects', currently by analytical use case/solution need. However, it makes sense to first understand the staging area tables in terms of content before understanding how they map to analytical use cases.

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

- Business Data: This set of tables captures the actual business events and the resulting state of an Insurance Company from those business events. The OLTP systems (or Transactional Systems) capture this information resulting from the execution of the provider's different business processes. Broadly, this information can be categorized as:
  - Events: Business transactions, whether financial or non-financial, represent business happenings (events) that are relevant for analytical purposes. For example – a financial transaction by a customer on a current account is a specific event.
  - Events happen at a specific point in time, and are recorded by OLTP systems. In the staging area model, there are several transaction tables that capture this detail – for both financial and non-financial transactions.
  - State: The net effect of business transactions is to change the provider's overall financial/non-financial state. This state information is typically captured by product-specific systems in a Insurance Institution.
- Reference/Master Data: Events and state refer to 'business activities' of an Insurance institution. To provide more detail on these, Insurance institution need to capture additional data that provides context for these activities. This data may be variously called as 'reference data' or 'master data', and covers various business dimensions of a given transaction or account. For example – an Insurance institution has a master list of products that it sells to customers (Product Master). Similarly, it has a list of customers (Customer master). These and other lists provide context for each business transaction or account. Insurance Institution typically maintain 'Master' data for this purpose.
- With this background, the following lists the key categories of business data and reference data in the staging data model.

3.6.2 Reporting Data Model

The Reporting Data Model is the point where outputs of analytical processing are aggregated for reporting and Business Intelligence (BI) tools. Similar to the Common Staging Area being the foundation for data provisioning to analytical applications, engines and processes, the Reporting Data Model is the common data store for the outputs of these processes.

Outputs are computed quantitative measures and Key Performance Indicators that involve simple to complex, mathematical and statistical processing using the raw data, which is performed by specialized engines and computational models. In the OIDF design, the Reporting Data Model design ensures that the historical data is maintained.

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

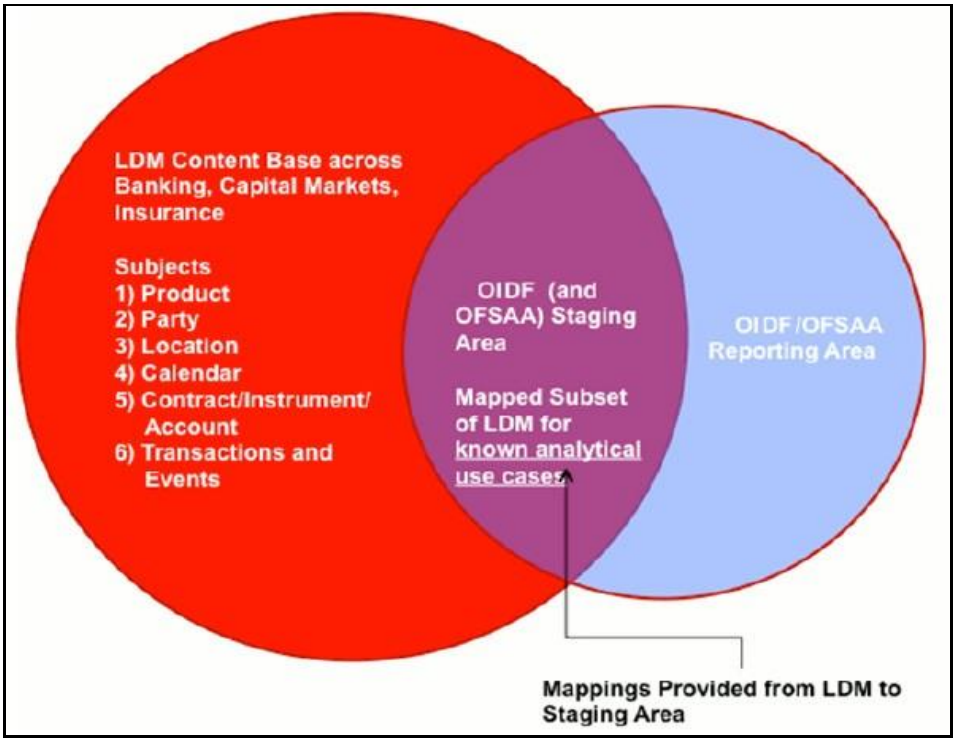
- Design: The Reporting Area data model is a dimensional data model. This means that it consists primarily of central fact tables (de-normalized), related to multiple dimension tables, also called a Star Schema. Additionally, the dimension tables are shared across the star schemas in the reporting mode, meaning they are Conformed Dimensions. This means that Drill-across reporting is naturally supported by the OIDF design.
- Additionally, in keeping with the key principle of the OIDF, the Reporting Model is organized by use cases to facilitate reporting and BI in a wide variety of areas.
- Support for multiple scenarios of analysis: As a result of the 2008 crisis, the Financial Services industry is moving towards scenario-based, forward-looking risk analysis instead of retroactive analysis. The reporting data model has been designed to support scenario analysis of the sort required by financial institutions that need to measure and report risk and performance under a variety of economic scenarios.
- To facilitate this, the Oracle Financial Services Advanced Analytical Infrastructure (OFSAAI) provides a Stress Testing framework, allowing risk analysis to be performed under a variety of known scenarios corresponding to different input parameter values to risk models.
- The reporting model provides support for this kind of analysis via a Run Dimension – it allows analytical engines to load multiple result sets identified by scenarios, and hence permits reporting related to baseline and stress conditions in economic terms.
- Support for Cross Functional Reporting: The third critical feature of the Reporting area design is the support for cross-functional reporting.
- Typically, Business Intelligence and Reporting Solutions work off a dedicated, purpose-specific data store called a data mart. Data marts are function-specific data stores typically star schemas (for example: Marketing Data Marts, Risk Data Marts, Customer Data Mart), that provide the necessary reporting and analytics relevant to a particular business function in the Insurance Institution.
- The trend is for cross-functional analytics and reporting. Majority of emerging needs relate to the analytical problems at the intersection of the distinct areas of Risk, Performance, Customer Insight, and Compliance.

3.7 LDM Relationship to the OIDF Physical Model

The coverage of the OIDF LDM in terms of core business can be applied to:

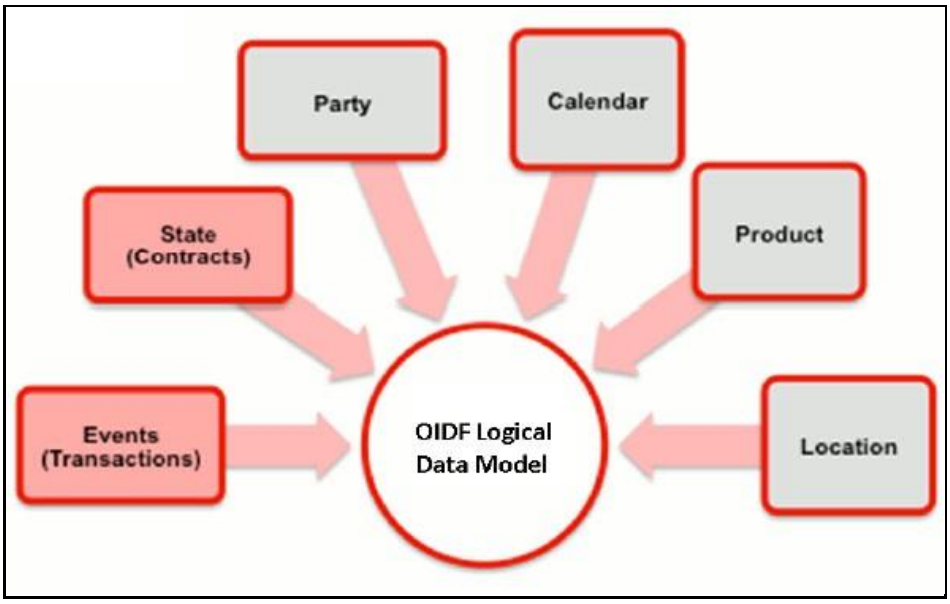
- Claims

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



3.8 OIDF LDM Content Details

The OFS LDM is organized into a small set of high-level subject areas, each of which represents a key category of data that is produced by the business processes in an Insurance institution. The following is the representation of Key Subjects in OIDF Logical Data Model.



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

3.9 Mapping from Logical Data Model to OIDF Staging Area

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

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

- LDM_Staging_mapping_Application
- LDM_Staging_mapping_Campaign
- LDM_Staging_mapping_Claim
- LDM_Staging_mapping_Collateral
- LDM_Staging_mapping_Collections and Recovery
- LDM_Staging_mapping_Contract
- LDM_Staging_mapping_Geography
- LDM_Staging_mapping_Instrument
- LDM_Staging_mapping_Organization
- LDM_Staging_mapping_Party
- LDM_Staging_mapping_Product
- LDM_Staging_mapping_Transaction

NOTE Column description updated as ‘deprecated’ is dropped from the model in the next major release.

4 Using OIDF

This chapter details on how the OIDF models are delivered and how they can be installed and configured into the required environment. The first two sections gives you an understanding of the Delivery Mechanism and OIDF Installation. The Data Dictionary and Download Specifications sections explains how the self-documenting ERwin file includes the data dictionary and Download Specifications within ERwin itself.

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

This chapter includes the following topics:

4.1 Delivery Mechanism

OIDF being a collection of data model artifacts, includes both a readily deployable model (the OIDF Physical Data Model) as well as a reference data model (the OIDF Logical Data Model). Both the data models (Physical and Logical) are delivered as ERwin files. The OIDF hence requires a license of the ERwin Data modeling tool.

ERwin is the current and only supported modeling tool to view and edit the model. Currently, the minimum version of ERwin supported is 7.1+.

> **NOTE** OFS AAI supports data model upload for data models generated using ERwin. For information on compatible versions, refer to the corresponding release Technology Matrix.

4.2 Installing OIDF

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

Please refer to the separate OIDF Install Guide 8.0.7.0.0 for step-wise instructions how to configure and install OIDF into an AAI instance.

4.3 OIDF Supporting Documentation

The preceding sections have provided an overview of the organization of the OIDF, and its various component data models. Appendix A, page A-1 explains the naming conventions used in 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, with another 800+ entities in the Logical Data Model.

Since it is delivered as an ERwin file, all the detailed metadata for the model (Table, Column, Entity, Attribute, Relationship) definitions are embedded in the file itself. The advantage of this approach is that any site-specific customizations to 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:

- Data Dictionary
- Download Specifications

For more information on Dimension Management and AMHM, refer to the Dimension Management chapter in Oracle Financial Services Analytical Applications Infrastructure 7.3 User Guide and Dimension Load Procedure section in Oracle Financial Services Analytical Applications Data Model Utilities 7.1 / 7.2 User Guide.

4.4 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 Technical Whitepaper on Data Model Document Generation, which details how to extract the data dictionary from ERwin section.

4.5 Download Specifications

As detailed in the staging area section, the mapping from the Staging Data Model to use cases, called a download specification provides an efficient way to manage the sourcing of data into the 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 Technical Whitepaper on Data Model Document Generation, which details how to extract the data dictionary from ERwin section.

4.6 Extending OIDF Physical Data Model

Oracle Insurance Data Foundation (OIDF) Physical Data Model (PDM) design evolves as the analytical use cases covered by the OIDF and enhanced as improvements are engineered as a part of the product lifecycle. While the model satisfies a very large number of analytical use cases across Risk, Finance, Marketing, and Compliance subject areas, customers may need to customize the model for a specific installation. These custom changes however may impact the ability of the OIDF installation to be upgraded to later versions of the product. The guidelines outlined in this section will help minimize the impact of custom changes to the model when the installation needs to be upgraded to a later version of OIDF.

This section consists of the following sub-sections:

- Customization Process Guidelines
- Staging Area Design Guidelines
- Results Area Design Guidelines
- Upgrading Data Model

4.6.1 Customization Process Guidelines

It is strongly recommended to consult OFSAA Support / Field Engineers / Consulting Staff before making any changes to the PDM for the following reasons:

- Tables in the PDM common Staging Area are designed to meet the complex needs of data sourcing for many different financial services analytical use cases and as such have a large number of columns, and the need for the modification should be reviewed with OFSAA consultants.

The Results Area star schemas have been designed with a set of common fact tables and dimension tables to support integration of results from multiple analytical applications and any customization should be reviewed in order to ensure that the unified reporting capabilities of the model are preserved.

After a review with OFSAA field consultants, an extension to the model should first be logged as a request for product enhancement via the standard support process. This allows:

1. Product support and product management teams to identify if a similar enhancement request was submitted on behalf of another customer so that a uniform Model Enhancement design recommendation can be provided to all customers.
2. OIDF product management to evaluate if the enhancement request is applicable more broadly to other customers and if the change should in fact is to be taken as a design requirement for subsequent releases.

> **NOTE** OFS AAI supports data model upload for data models generated using ERwin 7.1.x, 7.2.x, 7.3.x, 9.0.x, 9.2.x, 9.5.x, and 9.6.x versions.

4.6.2 Staging Area Design Guidelines

Following are the Staging Area Design Guidelines:

- Ensure that the naming conventions as detailed in Appendix A, page A-1 section are followed.
- Entity relationships and constraints are enforced through OFSAAI data management toolkit and are not enforced via database referential integrity checks.
- The model should not be changed to enforce referential integrity checks and other data quality checks via database definitions.
- All Staging Area tables must have a column that identifies the system from where data is sourced (source system ID).
- The code columns in master data tables and tables that contain dimension data should be designed to hold alphanumeric values.
- The Domain dictionary maintains the list of attribute domains. New columns must be identified with an existing domain instead of explicitly defining column data type and valid values. See guidelines in Appendix A, page A-1 section on the use of defined Domains.
- Tables (e.g. reference or look up tables with static data) needed for only a specific application or use case should be a part of the application specific processing area and should not be part of the common Staging Area in OIDF.
- OIDF download specifications identify the tables and columns for which data needs to be sourced for a specific analytical use case. Any new tables and/or column should have its "APPLICATION USAGE" UDP set with the appropriate application value so that the generated download specification includes the customized column and table. The master list of UDP's are maintained as a central dictionary in ERwin.

- All columns added or modified as a part of the customization should be marked as such:
  - The column level UDP named "CUSTOM" must be marked YES, identifying the column as a custom property.
  - The "Customization Reason" UDP should be specified. Valid values are provided as a drop down list and can be "Pending Enhancement Request" or "Specific to Customer".

The "Type of Change" UDP should be set to the appropriate type of change as provided in the drop down list (Length, Datatype, Logical Name, Description, and Addition).

4.6.3 Results Area Design Guidelines

The Results Area consists of a set of star schemas with conformed dimensions and common fact tables. Integration of results from multiple application use cases is achieved by having common fact tables for customer and account level measures. The design of the results area allows for drill-down and drill-across BI reporting, which should be preserved after customization.

Following are the Results Area Design Guidelines:

- Ensure that the naming convention for results tables and columns detailed in Appendix A, page A-1 section is followed.
- Dimensional conformance should be maintained: The same dimensional information should not be represented in different forms. In addition, dimension table design should be compatible with the slowly changing dimension process design and so should have the required columns.
- The common accounts summary fact table.
- (FCT_COMMON_POLICY_SUMMARY) consolidates measures at an account level granularity for all applications. Account level attributes captured from source systems in staging and those attributes that do not vary between runs should be part of the common accounts summary table. This enables integrated reporting of account information.

> **NOTE** Any account level application specific attributes and measures that are computed by applications should be part of the application specific account summary entities.

- The common customer summary fact table.
- (FCT_COMMON_CUSTOMER_SUMMARY) consolidates measures at a customer level granularity for all applications. Customer level attributes captured from source systems in staging and those attributes that do not vary between runs should be part of the common customer summary table. This enables integrated reporting of customer information.

> **NOTE** Any customer level application specific attributes and measures that are computed by applications should be part of the application specific customer summary entities.

- Aggregate Entities: Depending on performance requirements for each application, information can be reported out of aggregate entities. However, a drill through to the base entity from the aggregate entity is mandatory.

- Reporting and local currency support: Include additional attributes in the fact tables to store reporting and local currency equivalent of base measures. These attributes need to be computed by looking into the exchange rates.
- Support for full history: Any new tables in the Results area should be designed to support maintenance of full history.

4.6.4 Upgrading Data Model

The model upgrade process is achieved through the ERwin Model Compare and Merge utility. Refer to ERwin documentation for details on Menu options, process of comparing, and merging models.

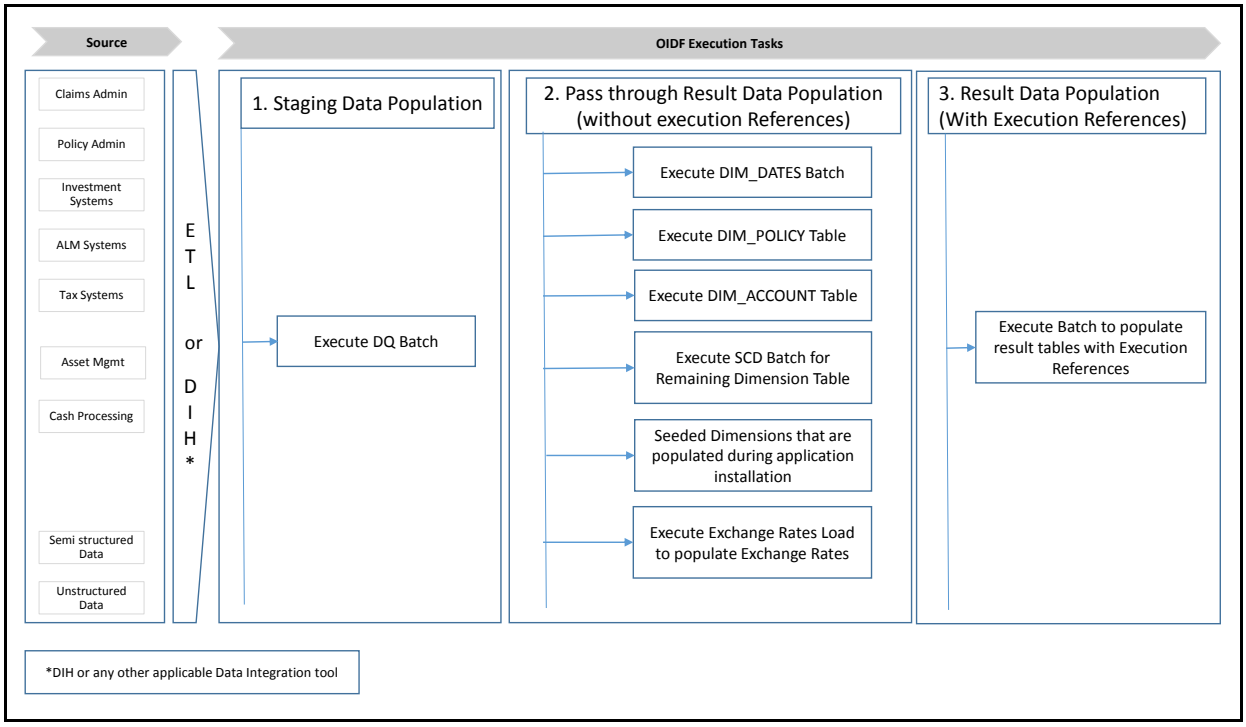
5 Data Flow Process for Data Foundation

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

This chapter includes these topics:

5.1 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.



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/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 are:
   a. Dimension Loading Process:
      i. Populate 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 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

5.2 Execution Processes

5.2.1 Prerequisites for the Execution Processes

After OIDF application installation is successfully complete, as a prerequisite to initiate the SCD process or T2T process, perform 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/Batch Name, and the order of execution, see the latest version of OIDF Run Chart.

1. Complete the OIDF application installation and post-installation configuration procedures successfully. For detailed procedures, see Oracle Insurance Data Foundation Application Pack Installation Guide Release 8.0.7.0.0.
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/DIH.

   > **NOTE** Data is already loaded into the OIDF Stage Master tables (see 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 Time Dimension Transformation.

4. Execute the Data Quality (DQ) batches.

   Data, 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.

5.2.2 Dimension Loading Process

Data into the Dimension tables can be loaded using one of the following processes:

- Seeded Process.
- Slowly Changing Dimension (SCD) Process.
- Loading Key Dimensions using AMHM.

5.2.3 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 OIDF Run Chart.

5.2.4 Slowly Changing Dimension (SCD) Process

> **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 Prerequisites for the Execution Processes section.
>
> Ensure you execute the batch corresponding to the required Dimension table. To access the Run Name/Batch Name, and the order of execution, see the latest version of OIDF 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 chapter About Dimension Loading 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.

To populate data in a Dimension table:

1. Execute the SCD Batch of the DIM_ACCOUNT table.
2. Execute the SCD Batch of the DIM_POLICY table.

3. Execute the <INFODOM>_DATA_FOUNDATION_SCD batch for the required Dimension table.
4. Verify log files and check error messages, if any.

5.2.4.1 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.

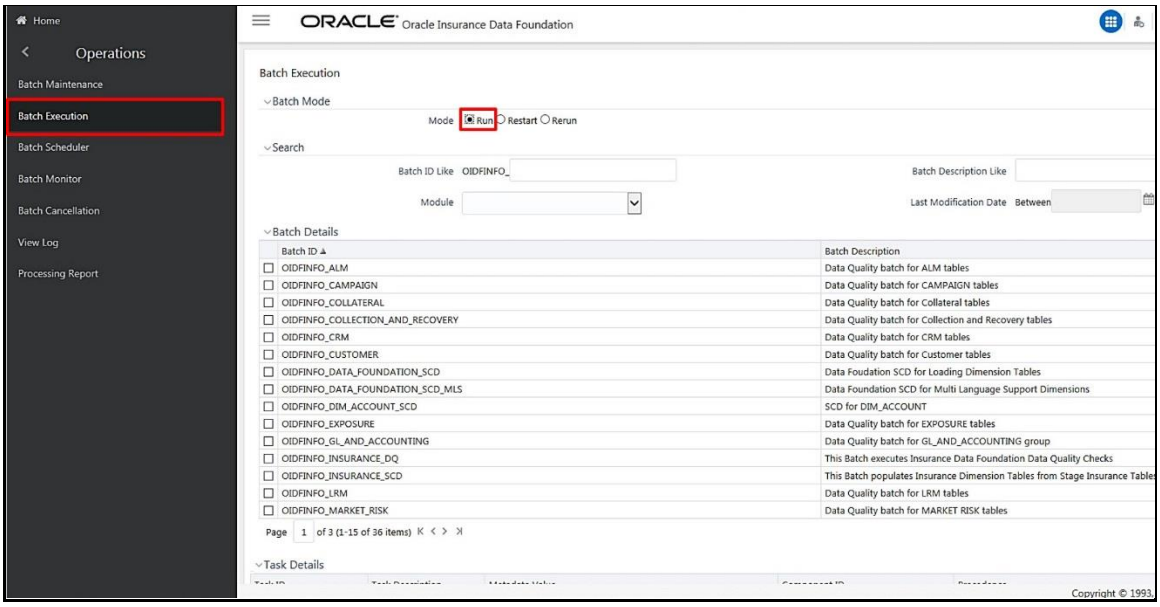
5.2.4.1.1 Execute the SCD Batch

To execute the SCD batch:

1. Navigate to the Batch Execution page.

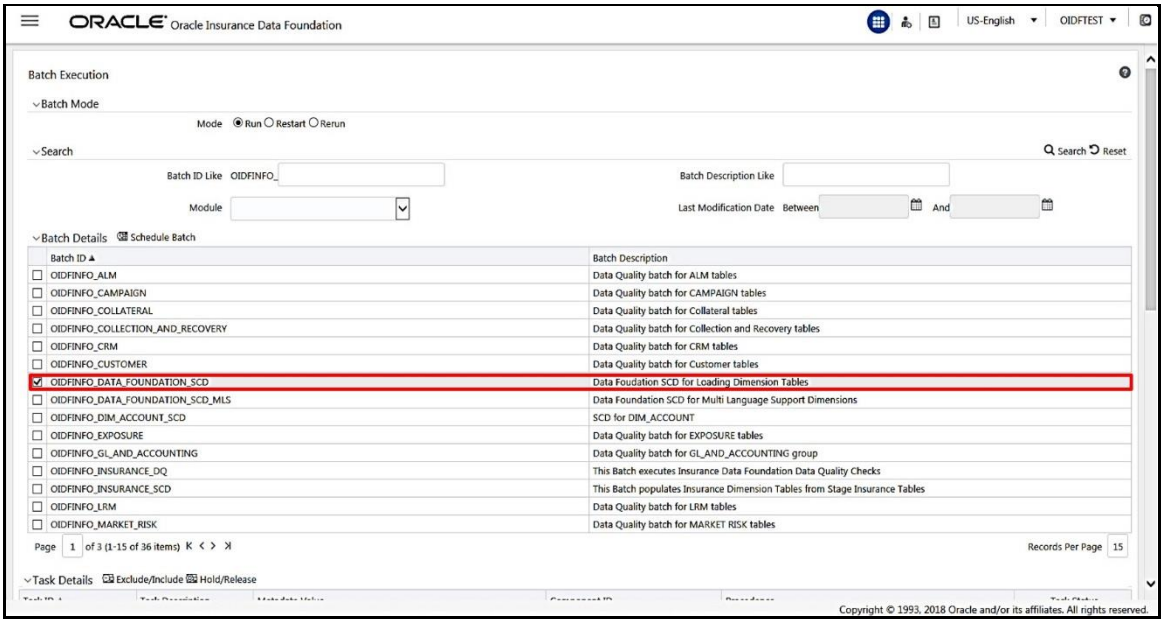
   Click **Oracle Insurance Data Foundation** > **Operations** > **Batch Execution**.

2. In the **Batch Mode** section of the *Batch Execution* page, select the **Run** option.



3. Select the Batch ID.
4. In the **Batch Details** section, search and select the required Batch ID:
   - For 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.

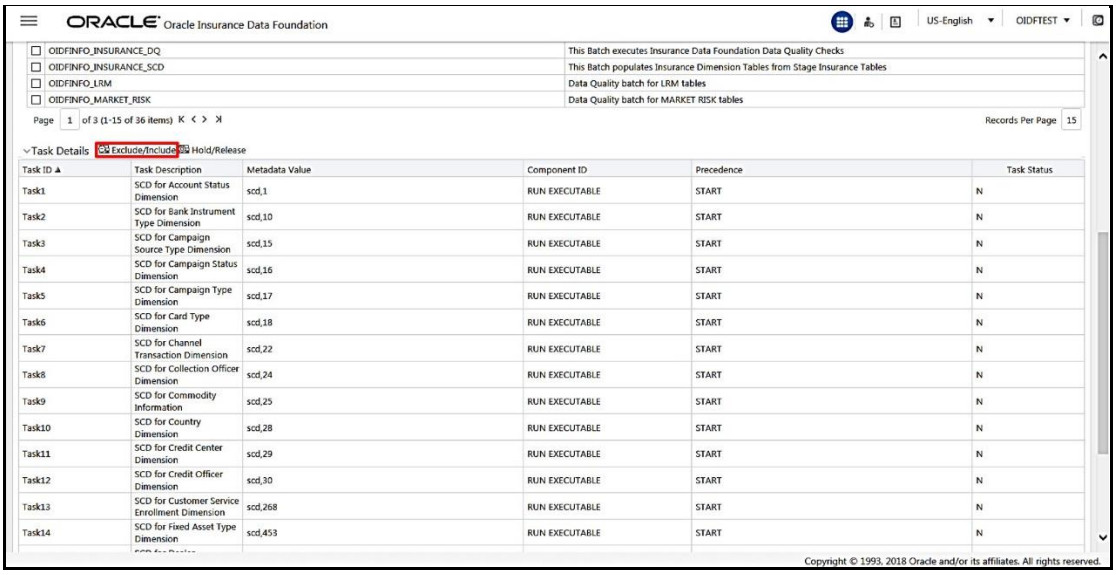


5. Exclude/Include tasks from/into the SCD batch.

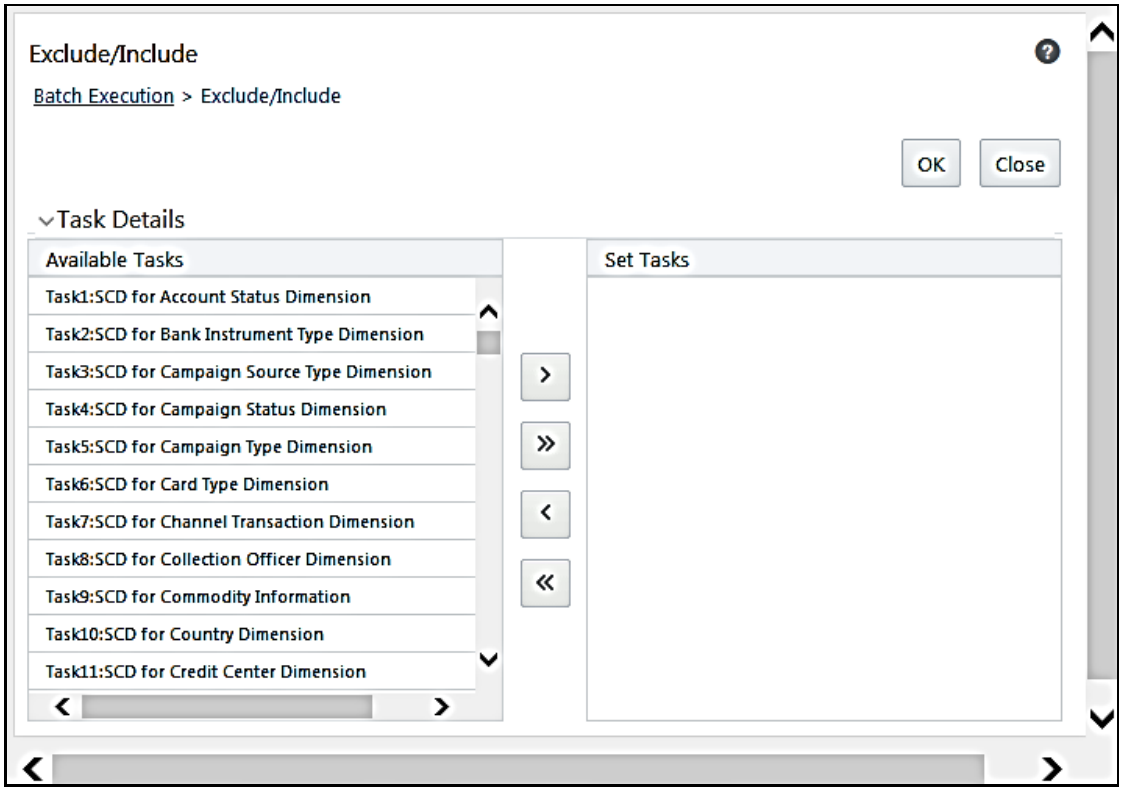
   a. Make a note of the Metadata Value (SCD map reference number) and the Task ID for the required Dimension table.

> **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.

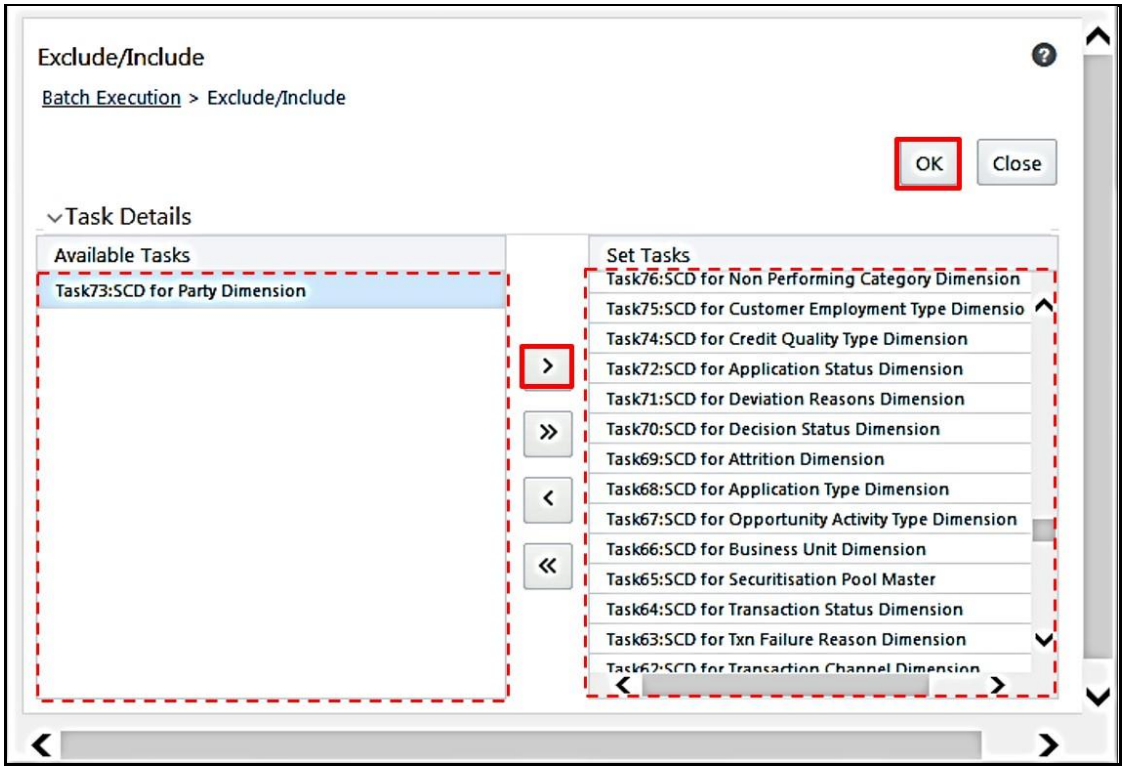


   c. The **Task Mapping – Exclude/Include** window appears.



d. Include/Exclude tasks.

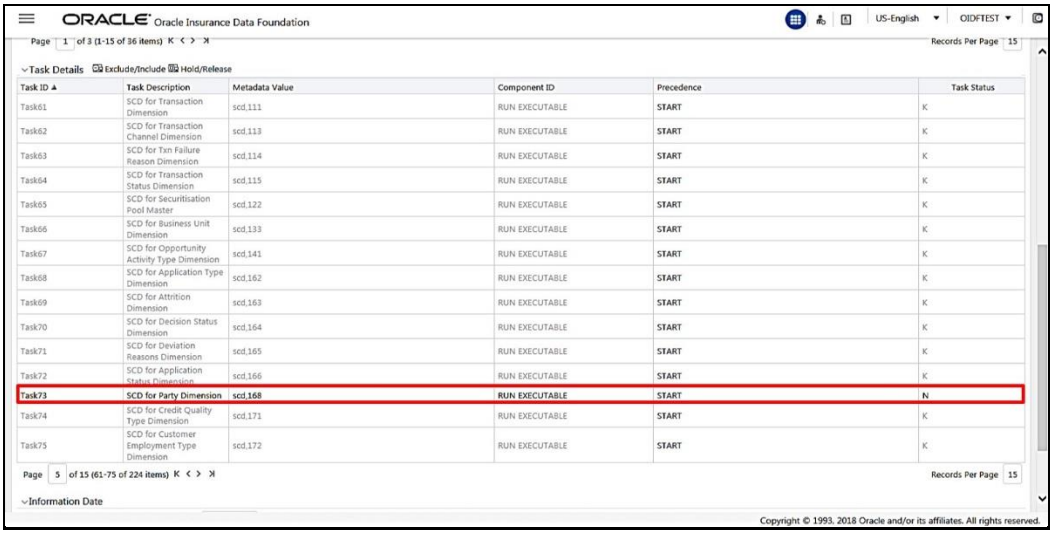
In the **Task Details** section, select the tasks that must be excluded from the batch execution procedure, and click > to move those tasks from the **Available Tasks** list to the **Set Tasks** list. Retain the task (Task ID noted in the earlier step) related to the currently required Dimension table in the **Available Tasks** list.



e. Confirm task inclusion/exclusion.

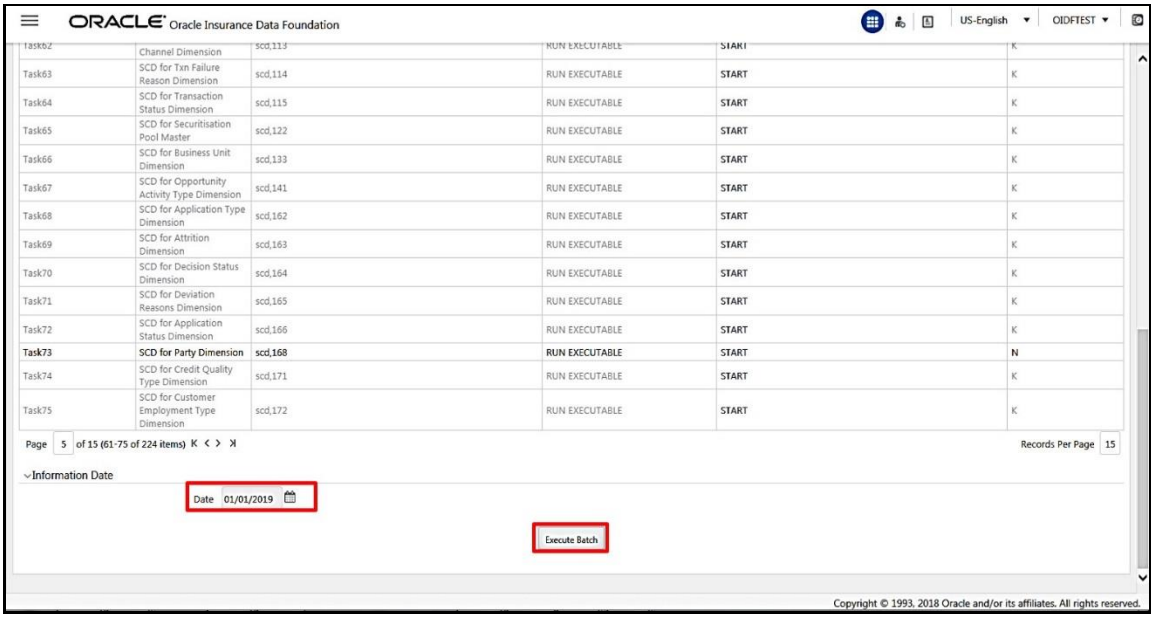
i. To save the changes, click **OK**. The warning message: *If you exclude a task, it will be skipped when executing the batch but, the precedence will not be altered. Do you want to exclude the selected task(s)?* appears.

ii. To proceed, click **OK**. The **Task Details** list consists only of tasks present in the **Available Tasks** list. For example, in the following screenshot, the highlighted task is executed during batch execution.



6. Select the FIC_MIS date.

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



7. Execute the SCD batch.

   Click **Execute Batch** to run the selected SCD batch with the selected Task:

   - For 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`.

8. Confirm the SCD batch execution.

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

   b. An acknowledgment message appears. Click **OK** to continue. The SCD batch is executed. As a result, data is populated in the target Dimension table.

5.2.4.2 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 section Execute SCD Batch.

5.2.4.3 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 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

To execute the <INFODOM>_DATA_FOUNDATION_SCD batch:

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

5.2.4.3.1 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).

5.2.4.3.2 Execute the SCD Batch

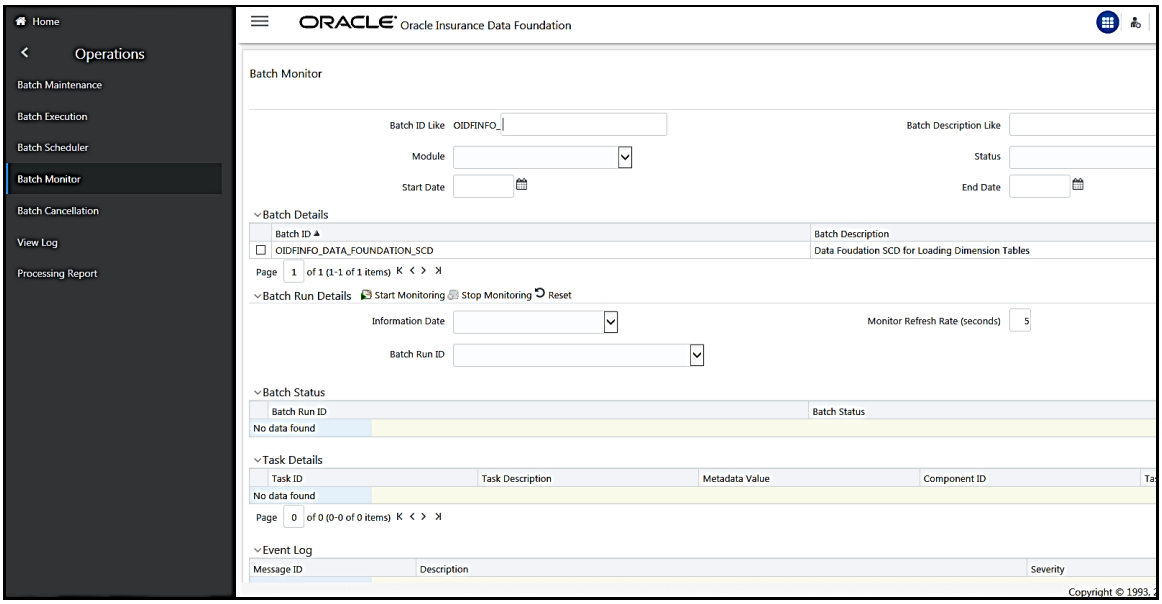
To execute the SCD batch, see the section Execute the SCD Batch.

5.2.4.3.3 Check the Execution Status of the SCD Batch

To check the execution status of the SCD batch:

1. Navigate to the Batch Monitor page.

   Click **Oracle Insurance Data Foundation** > **Operations** > **Batch Monitor**.



2. Select the Batch ID <INFODOM>_DATA_FOUNDATION_SCD.

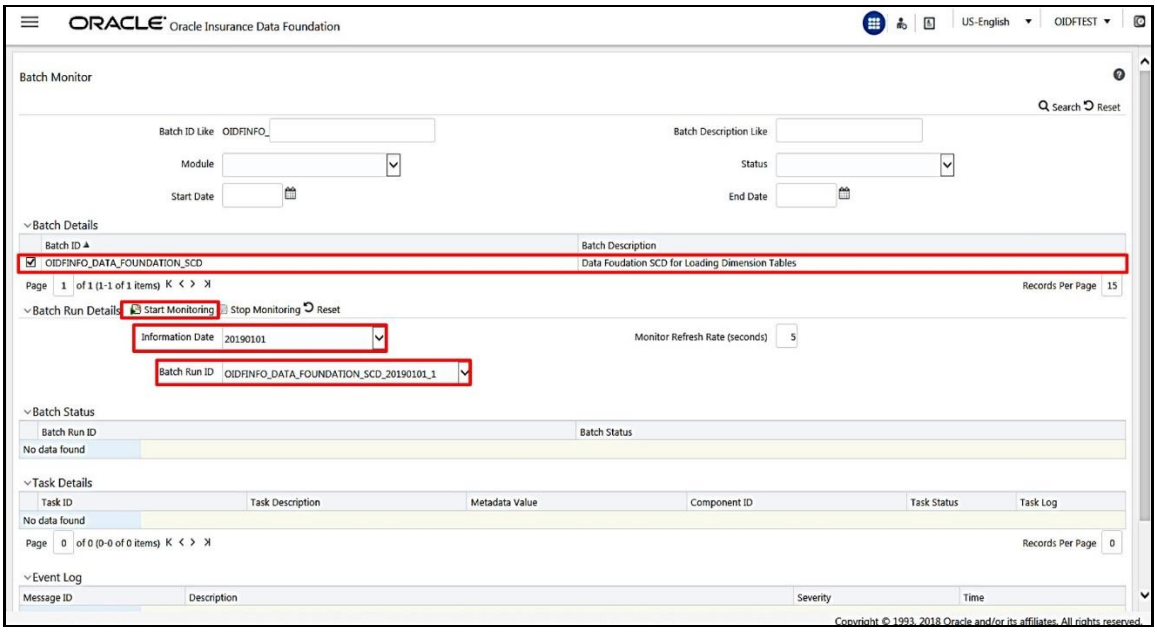
The *Batch Monitor* page appears on the right hand side. In the **Batch Details** section, select the Batch ID that was executed during the Execute the SCD batch step i.e., select the `<INFODOM>_DATA_FOUNDATION_SCD` Batch ID.

3. Select the FIC_MIS Date and Batch Run ID.

   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.

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

   Click the **Start Monitoring** icon.



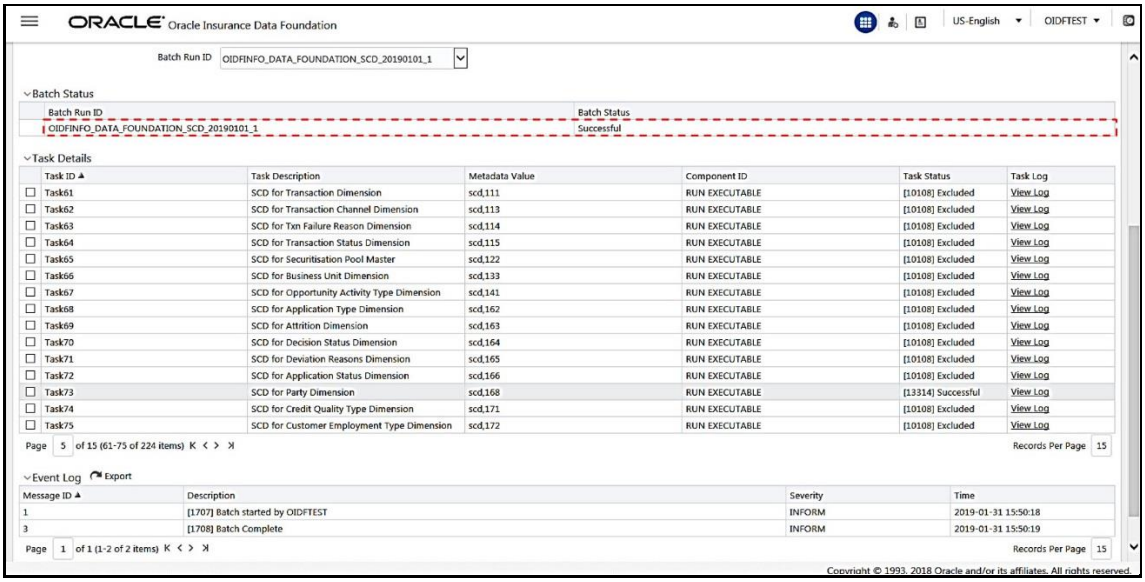
4. Check the Batch Status.

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

   The types of Status messages are:

   - Not Started
   - Ongoing
   - Failure
   - Successful

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



5.2.4.4 Verify Log Files and Check Error Messages

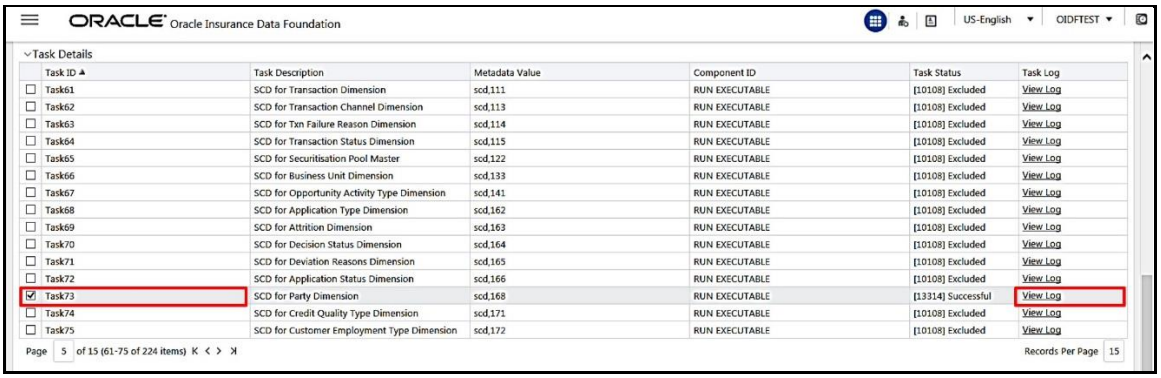
Use one of two methods to access the SCD batch execution log files in order to view the complete log generated for the SCD batch execution.

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

5.2.4.4.1 View and Download the Log Files from the Application UI

To view and download the log files from the application UI:

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



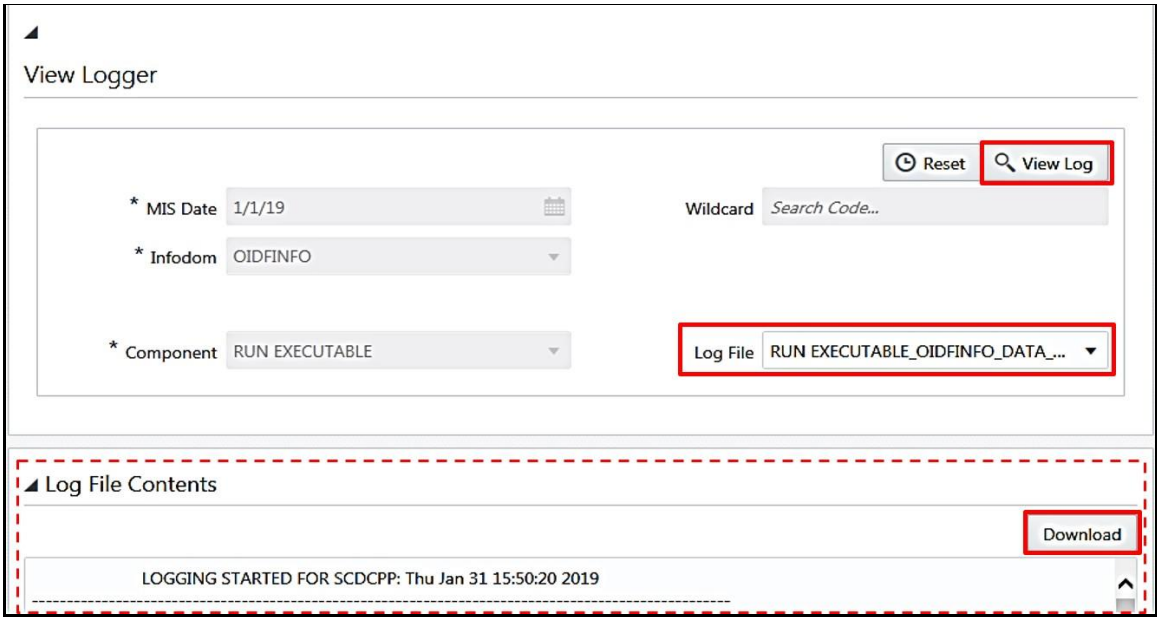
2. Select and view the log file:

   The *View Logger* pop-window appears.

   In the **Log File** box, select the required file, and then click **View Log**. The complete log file content appears 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**.

5.2.4.4.2 View Log Files in the Application Server

To view the execution log file on the application server:

- Go to the path `ftpshare/logs/<Run_Date>/<infodom>/RUN_EXECUTABLE`.

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

5.2.4.4.3 Check the Error Messages

To check the error messages:

- Open the log file present in the `ftpshare/logs/<Run_Date>/<infodom>/RUN_EXECUTABLE` folder for Dimension tables.

5.2.4.5 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 are 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 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.

5.2.5 Loading Key Dimensions using AMHM

To load key Dimensions using AMHM (Attributes, Members and Hierarchy Management), refer to the Dimension Management chapter in Oracle Financial Services Analytical Applications Infrastructure 7.3 User Guide and Dimension Load Procedure section in Oracle Financial Services Analytical Applications Data Model Utilities 7.1 / 7.2 User Guide.

5.2.6 Table to Table (T2T) Loading Process

T2T batch execution process is used to load data into the Result tables.

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

- Prerequisites for the Execution Processes
- Execute the SCD_Batch_of_the_DIM_ACCOUNT table
- Execute the SCD_Batch_of_the_DIM_POLICY table
- Execute the <INFODOM>_DATA_FOUNDATION_SCD batch for the required Dimension table

To access the Run Name/Batch Name, and their order of execution, see the latest version of OIDF Run Chart.

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

To load data into the Result tables using the T2T loading process:

1. Prerequisites for loading T2T.
2. Set the Run Parameters.
3. Save the Run Parameters.
4. Execute the Batch for T2T.
5. Check the Batch Execution Status of T2T.
6. Verify T2T Log Files and Check Error Messages.

5.2.6.1 Prerequisites for Loading T2T

To Resave hierarchies:

1. Navigate to the *Save Metadata* page.
2. To load values for the **Reporting Currency** parameter and the **Legal Entity** parameter (in the steps Set Run Default Parameters and Set Run Execution Parameters), save 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)

5.2.6.2 Set the Run Parameters

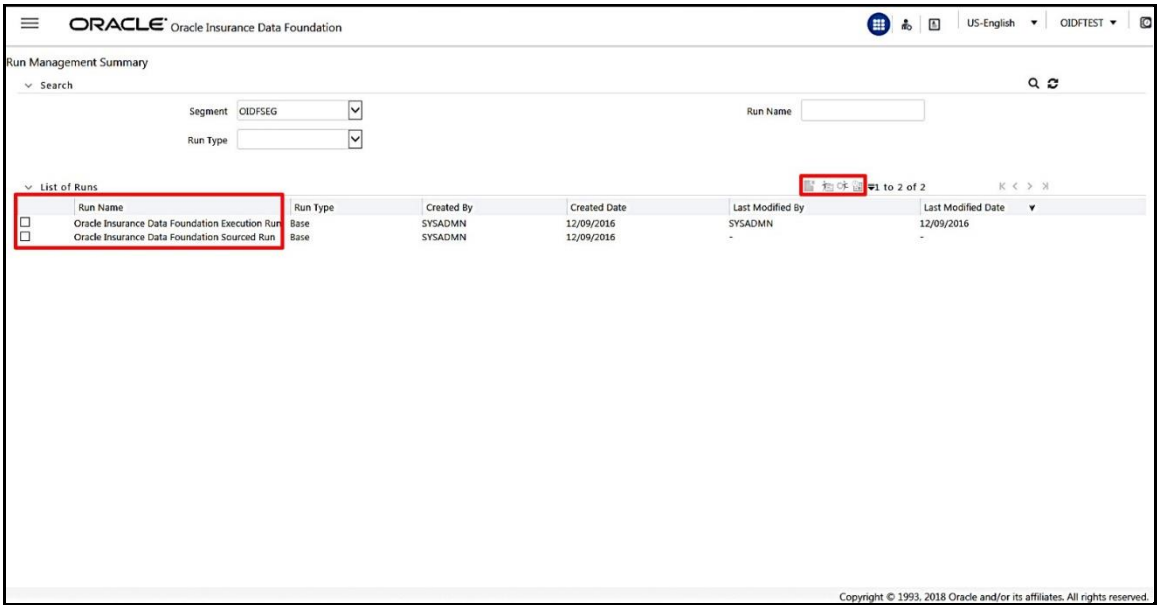
In the *Run Management* page, T2T Run parameters are setup. After creating the Run parameters, T2T batch can be executed either in the **Run Management** section or in the **Batch Execution** section.

> **NOTE** Run Management feature is used to execute Run with parameters during the T2T process. OIDF Run consists only of T2Ts and a DT. For detailed information about the Run Management, see the chapter About Executing Run through Run Management.

To set Run parameters for a T2T batch:

1. Navigate to the *Run Management Summary* page:

   Click **Oracle Insurance Data Foundation** > **Run Management** > **Run Management**. The **Run Management Summary** page appears.



2. Select the Run Name.

   In the **List of Runs** section, select the required **Run Name**. Select either **Oracle Insurance Data Foundation Execution Run** or **Oracle Insurance Data Foundation Sourced Run**.

3. OIDF application supports two types of Run processes.

   - Oracle Insurance Data Foundation Execution Run

     For data movement in Run enabled tables, this Run can be executed any number of times per day with each unique Run SKey.

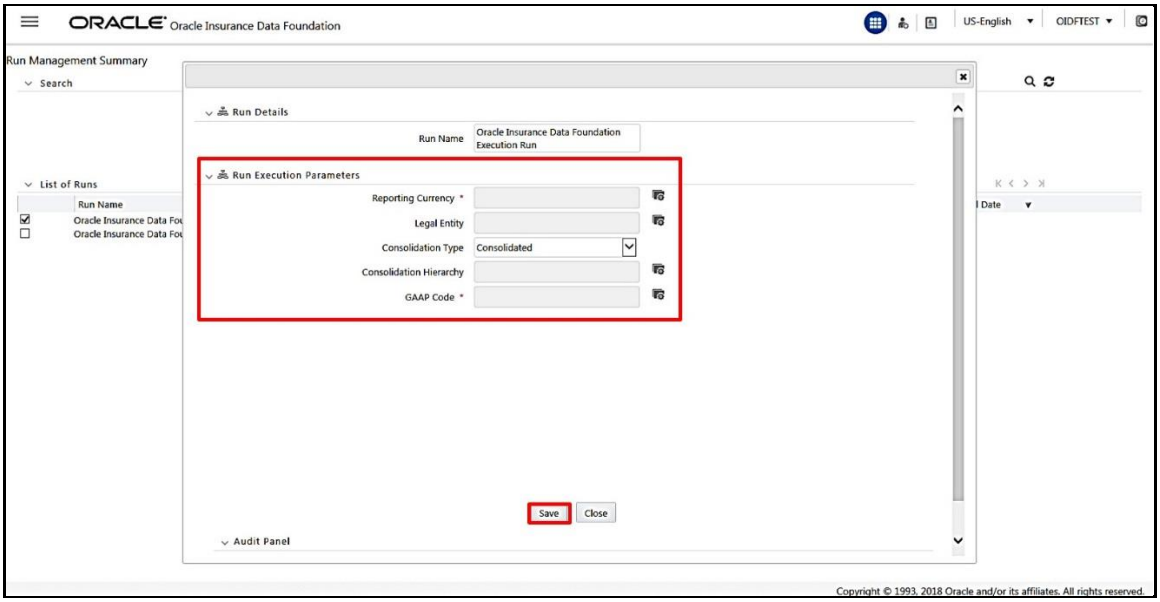
   - Oracle Insurance Data Foundation Sourced Run

     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.

4. Set Run Default Parameters.

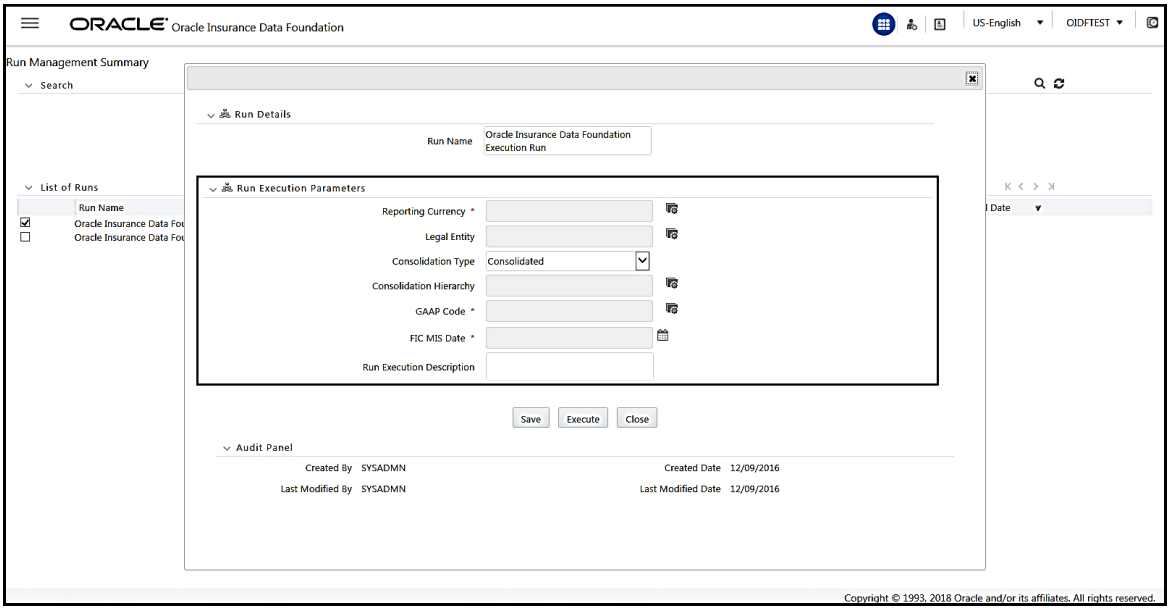
Click the **Run Default Parameters** icon. A pop-up window appears. Select required values for the following and click **Save**.

- Reporting Currency
- Legal Entity
- Consolidation Type
- Consolidation Hierarchy
- GAAP Code



5. Set Run Execution Parameters.

In the *Run Management Summary* page, for the selected **Run Name**, click the **Run Execution Parameters** icon. A pop-up window appears.



6. Select the FIC_MIS date.

   The values for the fields **Reporting Currency**, **Legal Entity**, **Consolidation Type**, **Consolidation Hierarchy**, and **GAAP Code** are populated in the earlier step.

   Select the required Run date using the **FIC MIS Date** calendar icon.

   Type a description in the **Run Execution Description** field. Use this is the description in the *Batch Execution* page to search for required batch.

   Save the Run parameters in the next procedure.

5.2.6.3 Save the Run Parameters

After configuring the required parameters, save the Run parameters.

To save the Run parameters:

1. In the *Run Management Summary* page pop-up window, click **Save**.
2. A batch is created consisting of all the defined **Run Execution Parameters**.

5.2.6.4 Execute the Batch for T2T

After saving the Run parameters, execute the batch in the *Run Management Summary* page pop-up window or in the *Batch Execution* window.

5.2.6.4.1 Execute the Batch for T2T in Run Management Summary

To execute the batch for T2T in the *Run Management Summary* page:

1. In the *Run Management Summary* page pop-up window, click **Execute**.
2. A batch consisting of all the defined Run execution parameters is created and executed immediately.
3. Check the Status of the executed Run in the Batch Monitor page or in the Run Execution Summary page. For detailed information, see Check the Batch Execution Status of T2T.

> **NOTE** When executing Run through Run Management, Run Skey auto-generates and stamps against each record. The DIM_RUN table stores Run SKey.

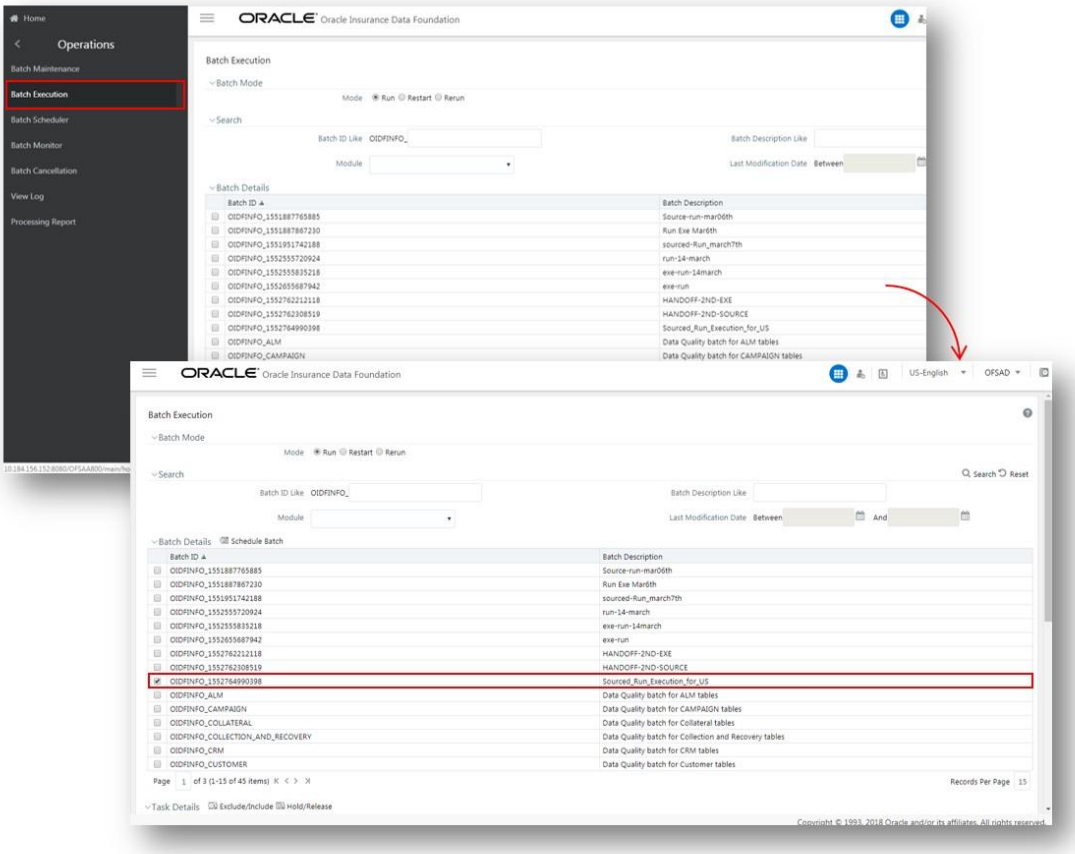
4. Make a note of the **Run ID** and **Batch ID**.
5. To verify the T2T log files, see Verify T2T Log Files and Check Error Messages.

5.2.6.4.2 Execute the Batch for T2T in Batch Execution

To execute the batch for T2T in the *Batch Execution* page.

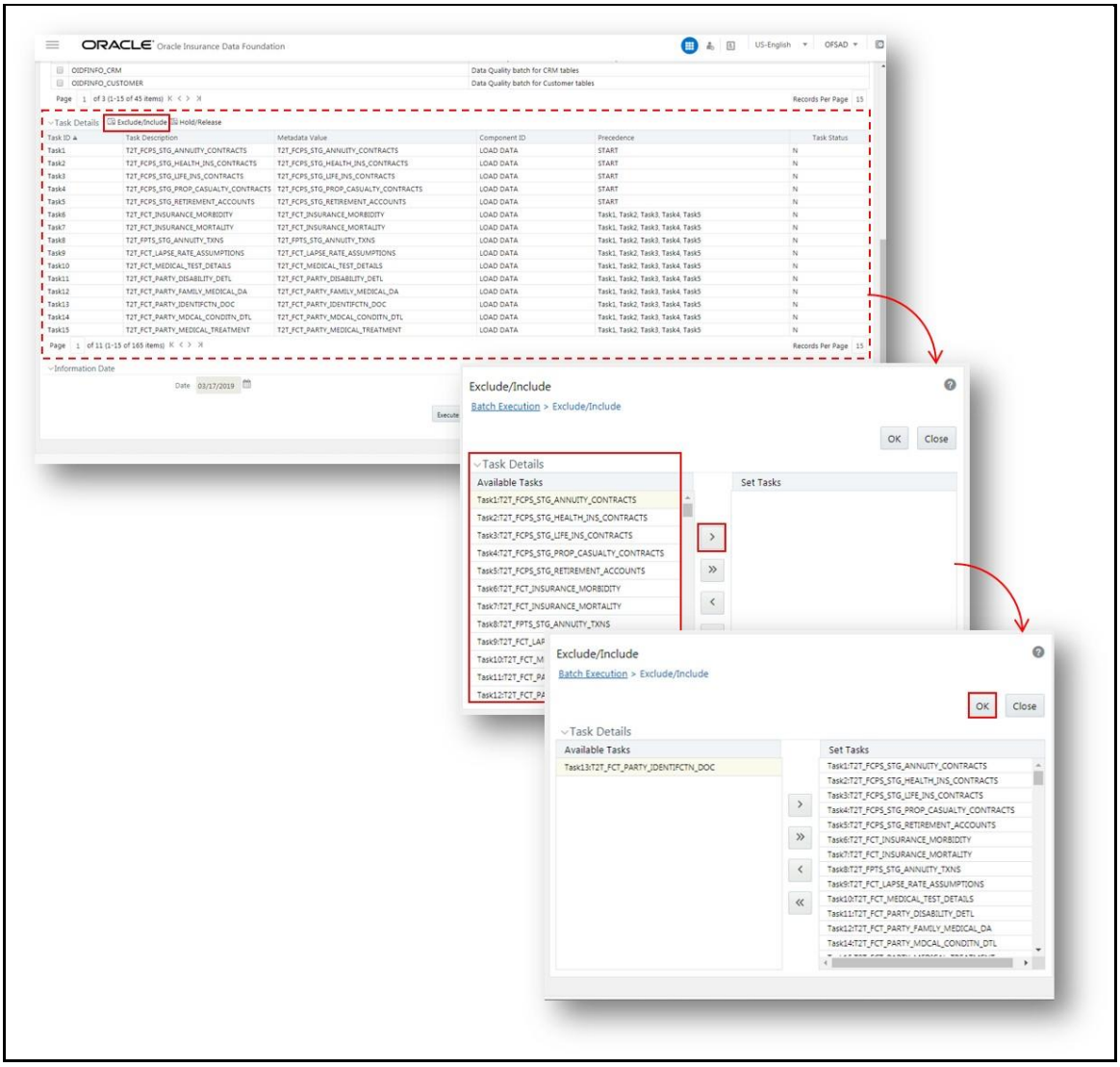
1. Navigate to the *Batch Execution* page and select the required Batch ID:
   - `<INFODOM>_EXE_RUN` for the Result table that consists of Run SKey.
   - `<INFODOM>_SOURCED_RUN` for the Result table that does not consist of Run SKey.

> **NOTE** For a detailed procedure, see Execute the SCD batch.



2. Exclude/Include tasks from/into the T2T batch.

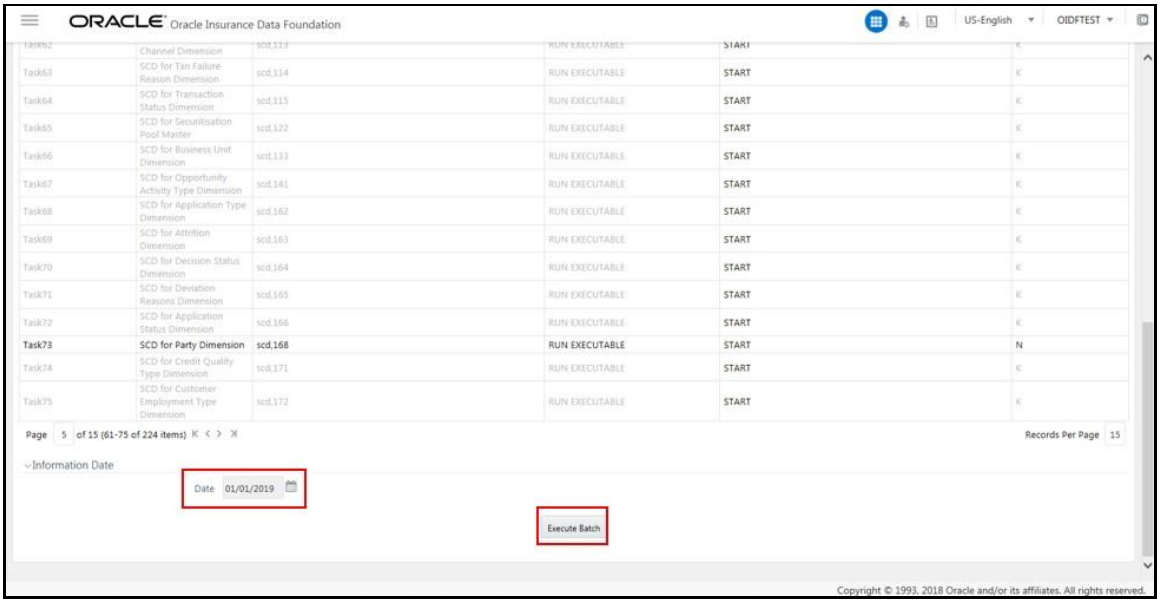
A summary of the steps is shown in the following screenshots. For a detailed procedure, see Exclude/Include tasks from/into the SCD batch.



3. Select the **FIC_MIS** date. To execute the T2T batch, click **Execute Batch**.

NOTE For detailed steps, see:

1. Select the FIC_MIS date
2. Execute the SCD batch
3. Confirm the SCD batch execution

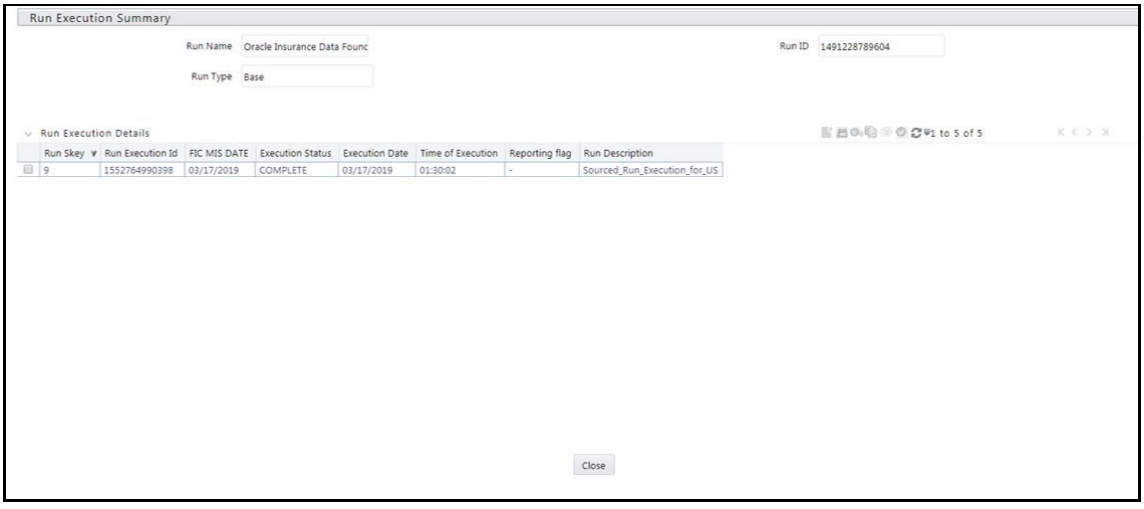


5.2.6.5 Check the Batch Execution Status of T2T

5.2.6.5.1 Check the Batch Execution Details for T2T in Run Execution Summary

To check the batch execution details for T2T that was executed in the *Run Management Summary* page:

1. Navigate to the *Run Management Summary* page.
2. Click the **Run Execution Summary** icon.
3. The **Run Execution Summary** pop-up window appears. This section lists data related to Run SKey, Run Execution ID, FIC_MIS_DATE, Execution Status, Execution Date, Time of Execution, Reporting Flag, and Run Description parameters.



5.2.6.5.2 Check the Batch Execution Details for T2T in Batch Monitor

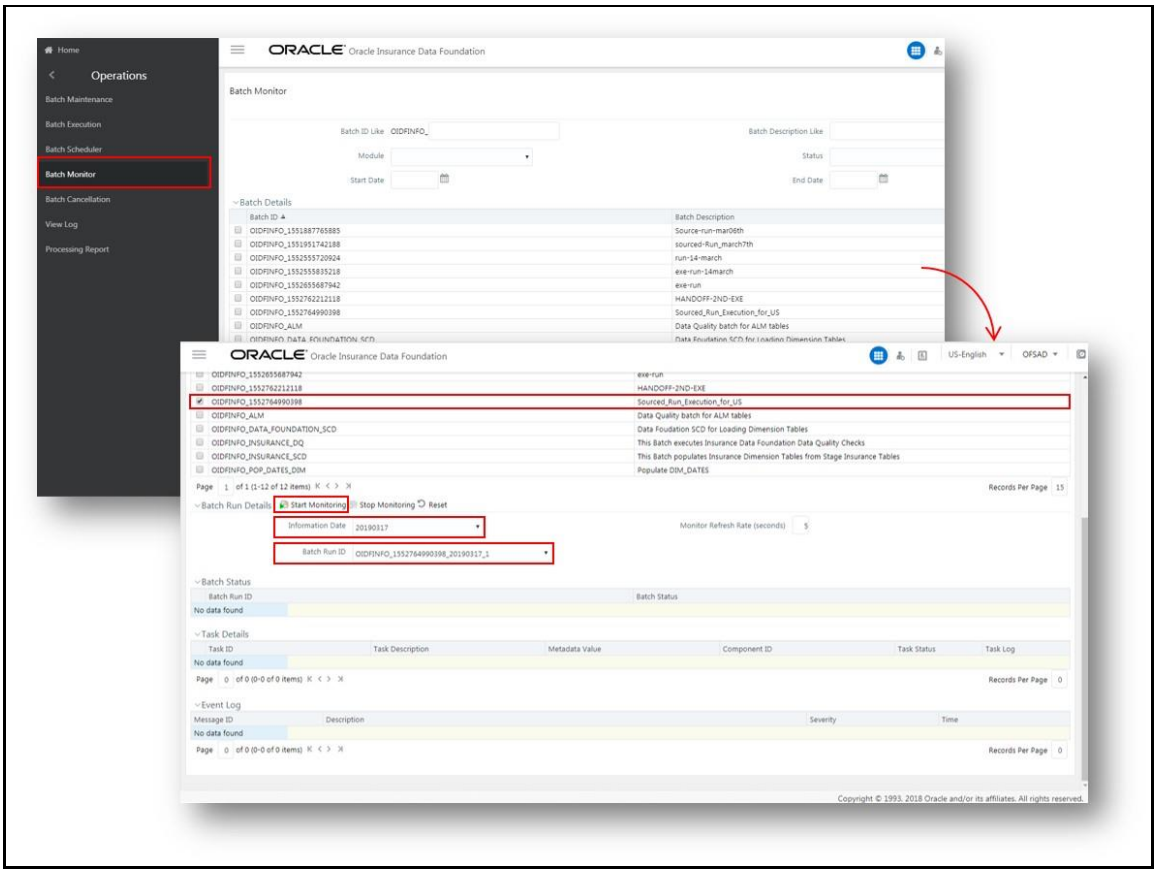
To check the batch execution details for T2T that was executed in the *Batch Execution* page:

1. Navigate to the *Batch Monitor* page and select the required Batch ID :

- `<INFODOM>_EXE_RUN` for Result table that consists of Run SKey.
- `<INFODOM>_SOURCED_RUN` for Result table that does not consist of Run SKey.

2. Select the required **Information Date**, **Batch Run ID**, and click the **Start Monitoring** icon.

   A summary of the steps is shown in the following screenshots. For a detailed procedure, see Check the execution status of the SCD batch.

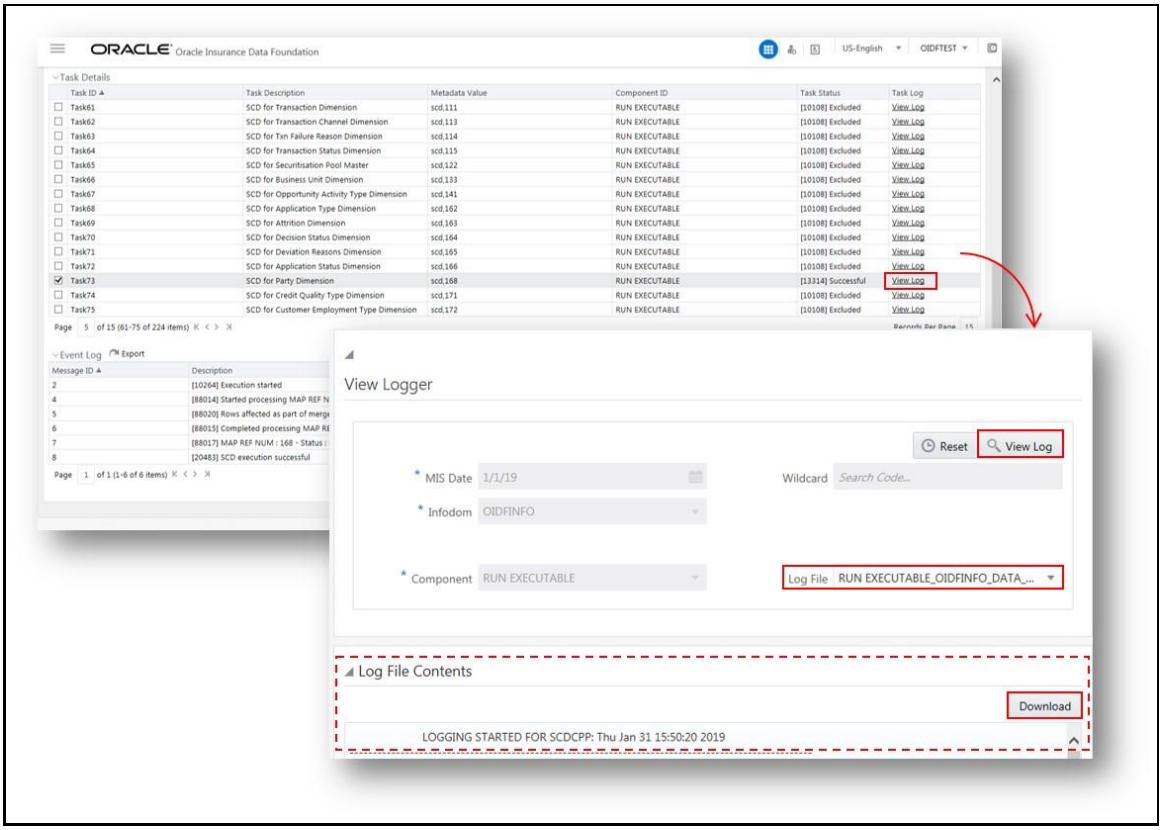


5.2.6.6 Verify T2T Log Files and Check Error Messages

A summary of the steps is shown in the following screenshots.

Two types of log files are generated:

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



- To access the execution log file on the application server:

  Go to the directory `ftpshare/logs/<Run_Date>/<infodom>/LOAD DATA`.

- To check the error messages:

  See the log file present in the `ftpshare/logs/<Run_Date>/<infodom>/LOAD DATA` folder for T2T.

5.2.6.7 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.

6 Loading Multiple Load Runs in OFSAA

This chapter 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.

This chapter includes the following topics:

6.1 Overview

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

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

6.2 Objective

To optimize the end-to-end data flow and the need for intra-day reporting, institutions could load intra-day records into OFSAA. Current application can only handle one set of records per date (incremental loads are not possible).

Users must adjust and reload data (either full or partial) for the current date.

Users must adjust and reload data (either full or partial) for any of past dates.

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

OIDF 8.0.7 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.

To leverage this design in 8.0.7 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** Ensure there is no blank line in the file.

6.3 Design Details

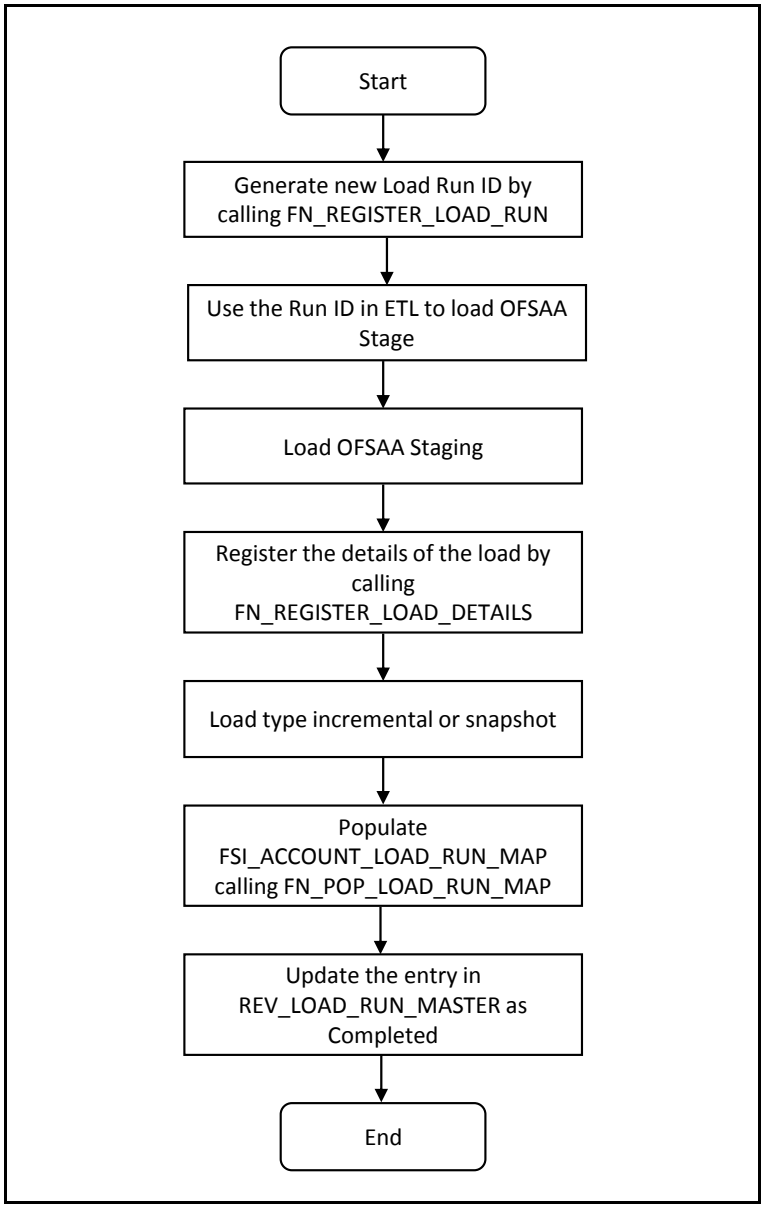
Loading of data into OFSAA can be in any of the following ways:

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

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

6.4 Implementation

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



NOTE Column N_LOAD_RUN_ID should always be populated only by the value returned by FN_REGISTER_LOAD_RUN.

Function: Register Load Run (FN_REGISTER_LOAD_RUN)

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

Parameters	Source of Values	Example Values
Batch ID	Auto generated if you are using OFSAA Framework	OFSOIDFINFO_20150101_1

Parameters	Source of Values	Example Values
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

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

6.5 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

6.5.1 Complete Snapshot Load Example

For example, if there are three Loan Contract accounts in an Insurance Company 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 appears as follows.

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

NOTE

After each load, you must 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 must call another function named FN_POP_LOAD_RUN_MAP. This is mandatory for incremental snapshot load scenario. This function populates an intermediate processing table that keep track of latest incoming record identifier.

Function: Populate Load Run Map

Parameters: Batch ID, MIS Date, Stage Table Name, Load Run ID, Load Run Name

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

Example:

```
Declare
Result number;
Begin
```

```
Result: =
fn_pop_load_run_map('OFSOIDFINFO_20150101_1','20150101','STG_CASA',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:

FIC_MIS_ DATE	V_ACCOUNT_ NUMBER	V_GAAP_ CODE	N_LOAD_RUN_ ID	F_LATEST_LOAD_RUN_ FLAG
1-Jan-15	LOAN1000	USGAAP	1	Y
1-Jan-15	LOAN1001	USGAAP	1	N
1-Jan-15	LOAN1002	USGAAP	1	N
1-Jan-15	LOAN1001	USGAAP	2	Y
1-Jan-15	LOAN1002	USGAAP	2	Y
1-Jan-15	LOAN1000	USGAAP	2	Y

There may be a requirement to reload a complete snapshot of data in the OFSAA staging again. This could either be to satisfy a intraday reporting requirement, or to load corrected source records in the OFSAA staging table. The earlier design forced users to truncate staging table to accommodate the new set of date. However with the introduction of Load Run identifier concept , you can retain both set of data in the staging area and allow the downstream application choose the correct set for processing. This 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 be loaded again staging with a different load run identifier . Additionally, strategy is to load the complete snapshot again to staging, all the records from the source such as, both changed and unchanged records will need to stamped with the new load run identifier

STG_LOAN_CONTRACTS after load appears as follows.

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.

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

6.5.2 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 appears as follows.

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

FIC_MIS_DA TE	V_ACCOUNT_NUMBE R	V_GAAP_COD E	N_LOAD_RUN_I D	N_EOP_BAL
01-JAN-15	LOAN1000	USGAAP	1	4066.213
01-JAN-15	LOAN1001	USGAAP	1	34538.905
01-JAN-15	LOAN1002	USGAAP	1	667.357
01-JAN-15	LOAN1001	USGAAP	2	34540.000
01-JAN-15	LOAN1002	USGAAP	2	670.000

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

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

Function - Populate Load Run Map

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

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

Example:

```
Declare
Result number;
Begin
Result: =
fn_pop_load_run_map('OFSOIDFINFO_20150101_1','20150101','STG_CASA',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.

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

6.5.3 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.

6.5.3.1 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.

6.5.3.2 Function - Register Load Run Details

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

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

Example:

```
Declare
Result number; Begin
Result: =
fn_register_load_details('OFSOIDFINFO_20150101_1','20150101',
'STG_CASA',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

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

6.5.4 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;
```

6.6 Loading data into OFSAA Results Tables from Staging Tables

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

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

6.6.1 Complete Snapshot Load Scenario

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

For example: 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 need to provide the Load Run ID which are supposed to use. Here in the following example, 1 is used.)
  - **d.** [DRCY]=USD,[LOADRUN]=1
  - **e.** Execute the batch after this change; it will load the Result table with the given Load Run Id Records.
  - **f.** Modify the Corresponding Process Task of a Run Each Time with Load Run Id
  - **g.** Select Process, Click Edit.
  - **h.** Choose Components, select the Object (T2T_FCPS_STG_ANNUITY_CONTRACTS) and click Components.
  - **i.** Click the Yellow Drop button near the T2T.
  - **j.** Add the Highlighted Condition with the text, and Save: "DRCY","USD","LOADRUN","1".
  - **k.** Fire the Run after this change. It loads the Result table with the given Load Run ID Records.

6.6.2 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_CASA.V_ACCOUNT_NUMBER

AND FSI_ACCOUNT_LOAD_RUN_MAP.N_LOAD_RUN_ID = STG_CASA.N_LOAD_RUN_ID AND
FSI_ACCOUNT_LOAD_RUN_MAP.FIC_MIS_DATE = STG_CASA.FIC_MIS_DATE AND
FSI_ACCOUNT_LOAD_RUN_MAP.V_GAAP_CODE = STG_CASA.V_GAAP_CODE AND
FSI_ACCOUNT_LOAD_RUN_MAP.F_LATEST_LOAD_RUN_FLAG = 'Y'
```

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

7 Time Dimension Transformation

This chapter 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 is used to specify how to rollup the child member balances (for example, 3 month rolling average).

This chapter includes the following topics:

7.1 Overview of Time Dimension Transformation

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 OFS Analytical Applications Infrastructure User Guide and the Oracle Insurance Data Foundation Installation Guide 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, refer to Oracle Financial Services Analytical Applications Infrastructure User Guide.
5. Batches will have to be created for executing the function. For more details, refer to section How to Define a Batch.

7.3 Tables Used by the Time Dimension Transformation

For more details on viewing the structure of earlier tables, refer to Oracle Financial Services Analytical Applications Data Model Data Dictionary or the OIDF Erwin Data Model.

7.4 Populating the Time Dimension Transformation

You can execute the function from the Operations (formerly Information Command Center (ICC) framework) module of OFSAAI.

This component for OIDF 8.0.7.0.0 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 window of OFSAAI. For more information on defining a new Batch, refer to section How to Define a Batch.

To define a new task for a Batch definition:

1. Select the check box adjacent to the newly created Batch Name in the Batch Maintenance window.
2. Click Add (+) button from the Task Details grid. The Task Definition window is displayed.
3. Enter the Task ID and Description.
4. Select Transform Data component from the drop down list.
5. Select the following from the Dynamic Parameters list:
   a. Datastore Type - Select the appropriate datastore type from the list.
   b. Datastore Name - Select the appropriate datastore name from the list.
   c. IP address - Select the IP address from the list.
   d. Rule Name - Select fn_DimDates from the drop down list of available transformations. (This is a seeded Data Transformation which is installed as part of the OIDF solution installer. If you don't see this in the list, contact Oracle support)
   e. Parameter List – Enter the Start Date and End Date.
      - Start Date – This is the starting date, from which the Transformation will populate DIM_DATES table. This date should be specified in 'YYYYMMDD' format.
      - For example, '20081131'.
      - End Date - This is the end date, to which the Transformation will populate DIM_DATES table. This date should also be specified in 'YYYYMMDD' format.

        For example, '20091231'.
6. Click Save. The Task definition is saved for the selected Batch.
7. Execute the batch.

You can execute a Batch definition from the Batch Execution section of OFSAAI Operations module. The function can also be executed directly on the database through SQLPLUS Details are:

Function Name: FN_DIM_DATES

Parameters: P_BATCH_RUN_ID, P_AS_OF_DATE, P_ST_DT, and P_ED_DT Sample Parameter Values: 'Batch1', '20091231', '20081131', and '20091231'

> **NOTE** This DT should be executed for each year for which data is present in the source table.

7.5 Checking the Batch Execution Status and Verifying Log Files

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 About Dimension Loading Process

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

This chapter includes the following topics:

8.1 About the 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 contain data that changes slowly, rather than changing on a time-based, regular schedule. SCD is considered and implemented as one of the most critical ETL (Extract Transform Load) task 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. They are:

- Type 1 SCDs - Overwriting
- Type 2 SCDs - Creating another Dimension Record
- Type 3 SCDs - Creating a Current Value Field

8.1.1 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.

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 until 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:

N_PRODUCT_SK EY	V_PRODUCT_ NAME	D_START_DATE	D_END_DAT E	F_LATEST_RECORD_ INDICATOR
1	Personal Loan	6/30/2010	12/31/9999	Y

8.1.2 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:

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	N
2	Personal Loan	6/30/2010	12/31/9999	Y

A new record is inserted into 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 is set to 'N'.

8.1.3 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 http://docs.oracle.com/cd/E14072_01/owb.112/e10935.pdf

Additional online sources include:

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

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

The SCDs used in Data Foundation solutions are listed in the Oracle Insurance Data Foundation - SCD Metadata spreadsheet under Technical Metadata for OIDF 8.0.7.0.0.

8.2 Populating Data in Dimension Table

Data Foundation solutions use the SCD component to handle dimensional data changes. For more details about the SCD component, see the chapter Slowly Changing Dimension (SCD). For SCD procedure, see Execute SCD Batch.

8.2.1 Prerequisites

The SCD executable should be present under `<installation home>/ficdb/bin` directory. The file name is `scd`.

The user executing the SCD component should have execute rights on the file mentioned as prerequisite in Executing the SCD Component.

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 can 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.

8.2.2 Tables Used by the SCD Component

The database tables used by the SCD component are:

- SYS_TBL_MASTER
- SYS_STG_JOIN_MASTER
- DIM_<dimensionname>_V

8.2.2.1 About SYS_TBL_MASTER Table

The solution installer populates one row per dimension for the seeded dimensions in the SYS_TBL_MASTER table.

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

Sample data:

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

Field Name	Value
MAP_REF_NUM	6
TBL_NM	DIM_LOB
STG_TBL_NM	STG_LOB_MASTER

Field Name	Value
SRC_PRTY	
SRC_PROC_SEQ	23
SRC_TYP	MASTER
DT_OFFSET	0
SRC_KEY	

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

8.2.2.2 About SYS_STG_JOIN_MASTER Table

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

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

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

Column Type	Description
PK	Primary Dimension Value (can be multiple for a given "Mapping Reference Number")

Column Type	Description
SK	Surrogate Key
DA	Dimensional Attribute (can be multiple for a given "Mapping Reference Number")
SD	Start Date
ED	End Date
LRI	Latest Record Indicator (Current Flag)
CSK	Current Surrogate Key
PSK	Previous Surrogate Key 9.SS Source Key
LUD	Last Updated Date / Time
LUB	Last Updated By

Sample Data:

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

Item	Description
MAP_REF_NUM	6
COL_NM	V_LOB_CODE
COL_TYP	PK
STG_COL_NM	V_LOB_CODE
SCD_TYP_ID	
PRTY_LOOKUP_REQD_FLG	N
COL_DATATYPE	VARCHAR
COL_FORMAT	61

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

8.2.2.3 About DIM_<dimensionname>_V

The database view which SCD uses as the source.

Example:

DIM_BANDS_V

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

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

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

Example:

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

8.2.3 Loading Dimension Tables

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

8.2.3.1 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 OIDF Run Chart.

8.2.3.2 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.

8.2.3.3 Loading Dimension Tables through Data Foundation SCD

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

> **NOTE** For a more comprehensive coverage of configuration and execution of a batch, see Oracle Financial Services Analytical Applications Infrastructure User Guide.

8.2.3.4 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, see Dimension Management chapter in Oracle Financial Services Analytical Applications Infrastructure 7.3 User Guide and Dimension Load Procedure section in Oracle Financial Services Analytical Applications Data Model Utilities 7.1 / 7.2 User Guide

8.2.3.5 Loading Data from STG_INTF Tables to DIM_INTF Tables

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

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

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

Staging tables are present for all key dimensions that are configured within the OFSAAI framework. The Dimension IDs used for key dimensions are:

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

8.2.3.6 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 details, refer to Dimension Loaders Section, in Chapter - Data Loaders, in the Oracle Financial Services Analytical Applications Data Model Utilities Guide.

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

8.2.4 Checking the SCD Batch Execution Status

To check the SCD batch execution status, see Verify log files and check error messages, if any.

8.3 Supplementary Information for Dimension Loading Process

Following are the supplementary information required for Dimension loading process.

8.3.1 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 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 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.
4. MERGE_HINT and SESSION_ENABLE_STATEMENT are in SYS_TBL_MASTER.

NOTE For illustration, Account Dimension is considered.

Table Name	Stage Table Name	Merge Hint	Session Enable Statement
DIM_ACCOUNT	STG_LOAN_CONTRACTS_V	/*+ parallel(DIM_ACCOUNT,1 0) */	"alter session enable parallel dml query", "alter table DIM_ACCOUNT nologging parallel 10"

5. 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.
6. Execute the SQL file with all the SESSION_DISABLE_STATEMENTs, after the successful completion of the SCD batch.
7. 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.
8. 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 new approach.
9. 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.
10. SCD execution occurs based on the GAAP code, which is configured in 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:

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.

8.3.2 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'.

8.3.3 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 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 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 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..

9 Account Dimension Table

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

This chapter includes the following topics:

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

9.1 About Account Dimension Table

The mapping details for the Account Dimension (DIM_ACCOUNT) tables are given here.

Map Reference Number	Source View Name	Logical Source View Name
190	STG_BORROWING_V	Stage Borrowing View
350	STG_CUSTODIAL_ACCOUNTS_V	Stage Custodial Accounts View
465	STG_FORWARDS_V	Stage Forwards View
195	STG_FUTURES_V	Stage Futures View
197	STG_GUARANTEES_V	Stage Guarantees View
198	STG_INVESTMENTS_V	Stage Investments View
201	STG_LOAN_CONTRACTS_V	Stage Loan Contracts View
351	STG_MANAGED_INV_ADV_V	Stage Managed Investment Account View
202	STG_MM_CONTRACTS_V	Stage Money Market Contracts View
203	STG_MUTUAL_FUNDS_V	Stage Mutual Funds View
205	STG_OPTION_CONTRACTS_V	Stage Option Contracts View
341	STG_PAYMENT_SETTLEMENT_ACCT_V	Stage Payment Settlement Account View
206	STG_REPO_CONTRACTS_V	Stage Repurchase Contracts View
354	STG_SPEND_OBLIGATIONS_V	Stage Spend Obligations View
208	STG_SWAPS_CONTRACTS_V	Stage Swaps Contracts View
210	STG_TRUSTS_V	Stage Trusts View

All the above-mentioned Product Processor tables are mutually exclusive in terms of functionality. Therefore, V_ACCOUNT_NUMBER or V_CONTRACT_CODE (Primary Key of each Product Processor)

must also have mutually exclusive values across all the tables. The same account number cannot be part of multiple Product Processor tables.

9.2 About SCD Process for Populating Account Dimension Table

The SCD population in DIM_ACCOUNT table generates individual numeric Surrogate Keys for every account number with an additional leg skey. Below are the columns that 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 DIM_ACCOUNT table. Here, the sources are the different product processor tables present in the solution, which are configured in FSI_DIM_ACCOUNT_SETUP_DETAILS table.

9.2.1 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 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.

9.3 Deploying Account Dimension Table on Hive

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

9.4 Populating Account Dimension Table

To populate data into Account Dimension table through 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 Dimension table, see the section Supplementary Information for Dimension Loading Process in the chapter About Dimension Loading Process.

10 Policy Dimension Table

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

This chapter includes the following topics:

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.

10.1 About Policy Dimension Table

The mapping details for the Policy Dimension (DIM_POLICY) tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name
224	STG_PROP_CASUALTY_CONTRACTS_V	Stage Property Casualty Contracts View
401	STG_LIFE_INS_CONTRACTS_V	Stage Life Insurance Contracts View
402	STG_HEALTH_INS_CONTRACTS_V	Stage Health Insurance Contracts View
403	STG_ANNUITY_CONTRACTS_V	Stage Annuity Contracts View
634	STG_RETIREMENT_ACCOUNTS_V	Stage Retirement Accounts View
674	STG_REINSURANCE_CNTRCTS_ISSUED	Stage Reinsurance Contracts Issued
675	STG_REINSURANCE_CONTRACTS_HELD	Stage Reinsurance Contracts Held

10.2 About SCD Process for Populating Policy Dimension Table

The SCD process for 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 DIM_POLICY table. Here, the sources are the different product processor tables present in the solution, which are configured in FSI_DIM_POLICY_SETUP_DETAILS table.

10.2.1 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 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.

10.3 Deploying Policy Dimension Table on Hive

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

10.4 Populating Policy Dimension Table

To populate data into 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 Dimension table, see the section Supplementary Information for Dimension Loading Process in the chapter About Dimension Loading Process.

11 About Executing Run through Run Management

11.1 Introduction

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

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

> **NOTE** To access the Run Name and Task details, see the latest version of OIDF Run Chart.

11.2 Summary and Details Page

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

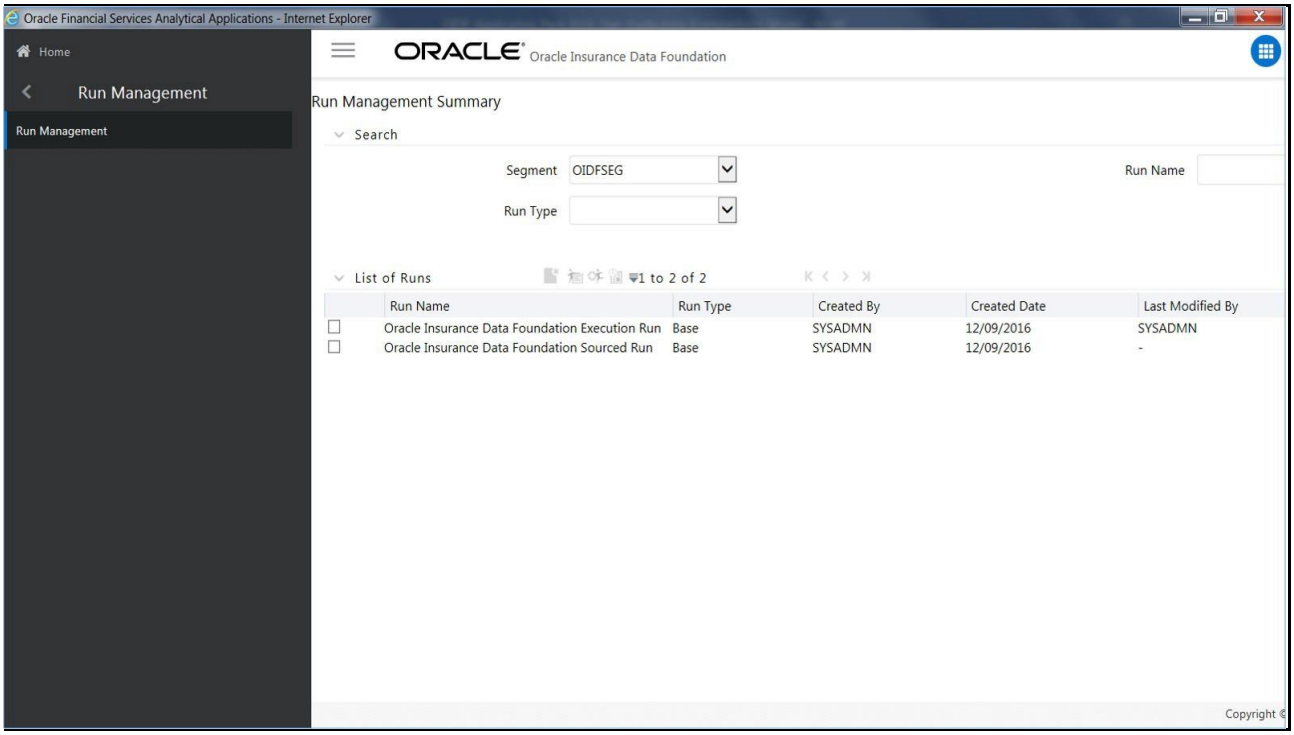
11.3 Navigation within the Summary Page

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

- Search
- List of Runs

11.3.1 Search Section

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



11.3.2 List of Runs Section

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

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

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

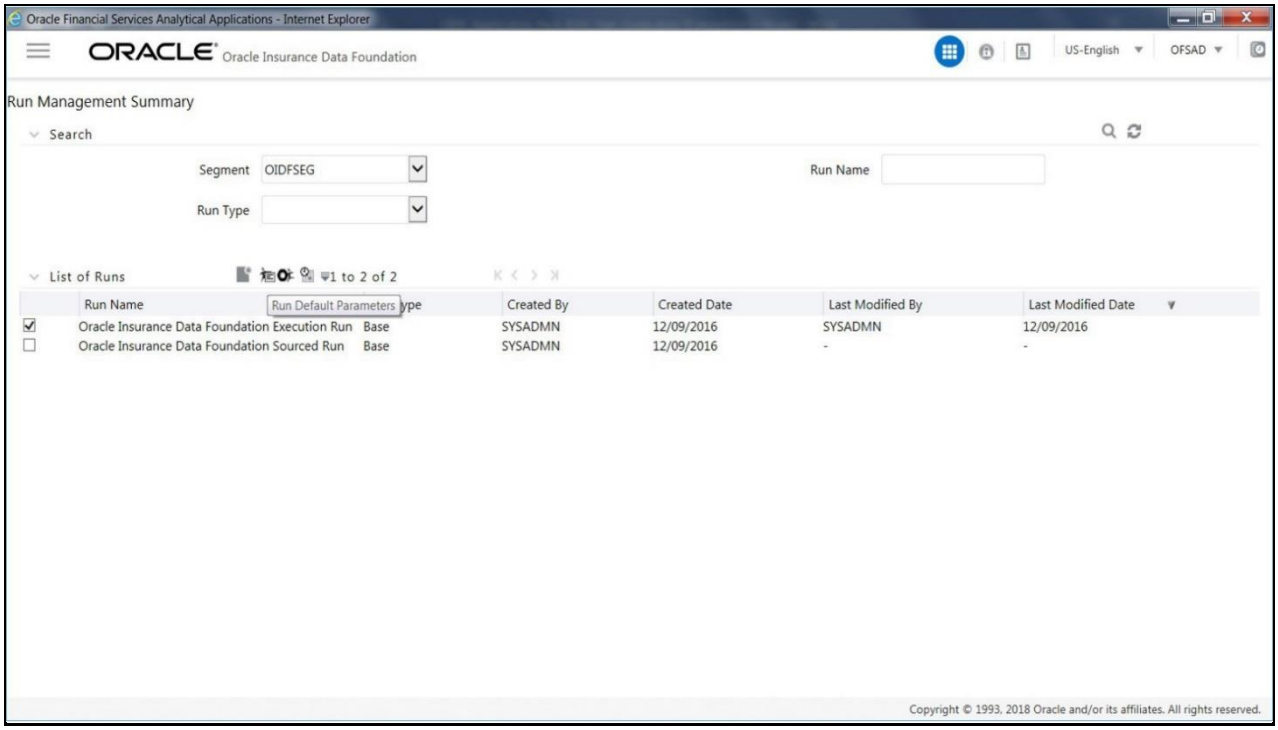
11.3.3 List of Runs Summary Grid

The following columns categorize each Run in the summary grid:

Field Name	Field Description
Run Name	Displays the short name of the Run.
Run Type	Displays the type of Run, Simulation or Baseline Run.
Created By	Displays the name of the User who defined the Run.
Creation Date	Displays the date on which the Run was created.
Last Modified By	Displays the name of the user who has performed any modifications to the Original Run details.
Last Modified Date	Displays the date on which the Original Run details were modified.

11.3.4 Navigation within Run Default Parameters Window

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



> **NOTE** To modify or view the parameters, the Modify Run Parameters role should be mapped to that relevant user profile.

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

11.3.4.1 Run Details Section

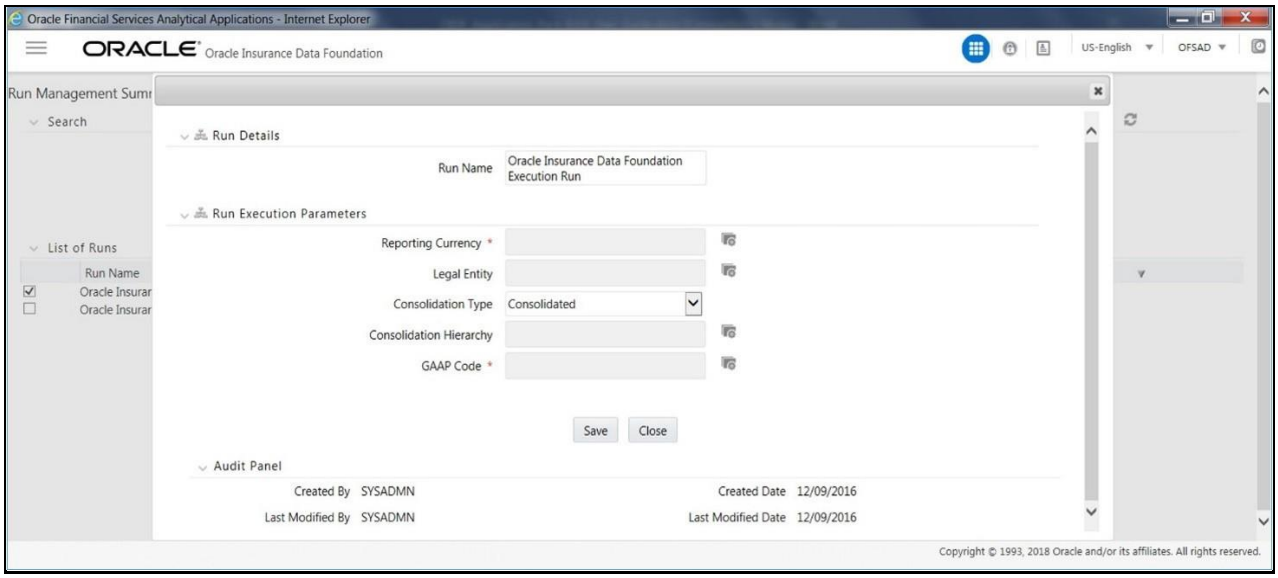
This section displays the name of the Run which is a read-only value.

11.3.4.2 Run Execution Parameters Section

In this section you can update the following:

Field Name	Field Description
Reporting Currency	Reporting Currency Code parameter is used for calculation of amounts in Reporting Currency during Data Population.
Legal Entity	Legal Entity Code parameter is used for identifying the legal entity, which is used for the Run.
Consolidation Type	Consolidation Type parameter is used for selecting legal entities on a solo or consolidation basis. In a solo run, only the selected legal entity will be used. In a consolidated run, along with the selected legal entity, all its child legal entities are also used.
Consolidation Hierarchy	Legal Entity Hierarchy is used for selecting the required hierarchy for the consolidated run. This parameter is not required for solo run.

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



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

- Legal Entity Code for Run (HOIDF001)
- Reporting Currency Code for Run (HOIDF002)
- Legal Entity Hierarchy for Run (HOIDF003)
- User Group Hierarchy (H_GROUP)

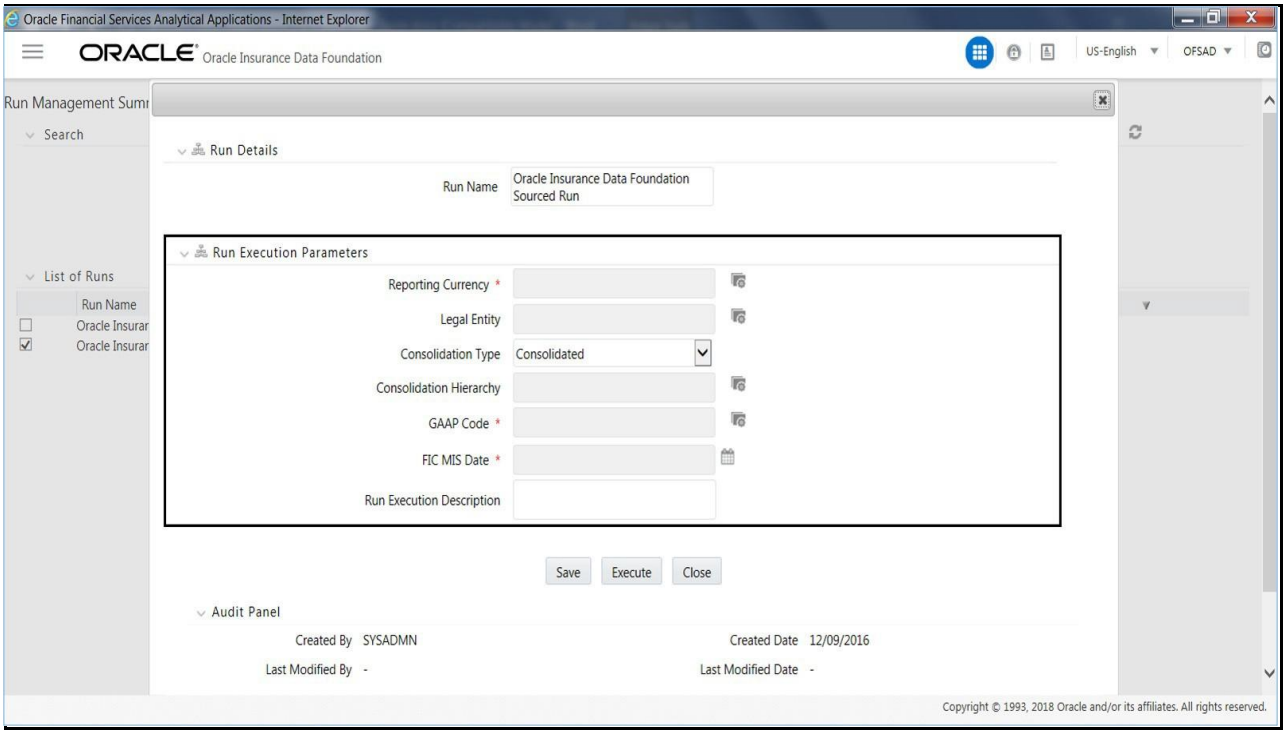
NOTE

For further details on Save Hierarchy, see Oracle Financial Services Advanced Analytical Applications Infrastructure Application Pack 8.0.7.0.0 on OHC.

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

11.3.5 Navigation within Run Execution Parameters Window

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



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

11.3.5.1 Run Details Section

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

11.3.5.2 Run Execution Parameters Section

The following Run execution parameters can be updated:

Field Name	Field Description
Reporting Currency	Reporting Currency Code parameter is used for calculation of amounts in Reporting Currency during Data Population.
Legal Entity	Legal Entity Code parameter is used for identifying the legal entity, which is used for the Run.
FIC MIS Date	Enter the extraction date in this field.
Run Execution Description	Enter a longer description of the Run.

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

Legal Entity Code for Run (HOIDF001)

Reporting Currency Code for Run (HOIDF002)

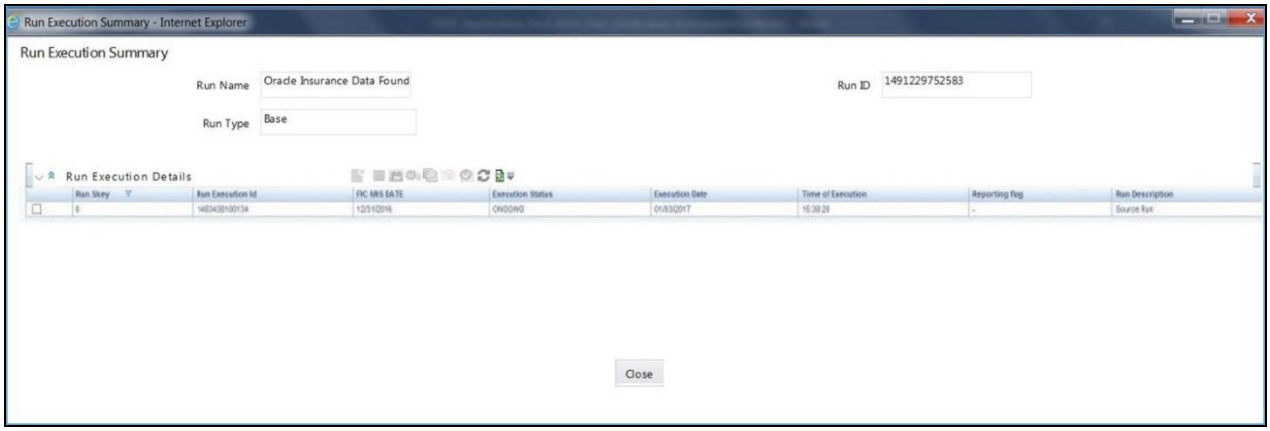
By clicking the Save button; a batch with the defined Run execution parameters is created. The batch created can be executed from the Batch Execution window.

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

NOTE For further details on Save Hierarchy and Batch Execution, see Oracle Financial Services Advanced Analytical Applications Infrastructure Application Pack 8.0.7.0.0 on OHC. To execute a Run, the execute run role should be mapped to your user profile. Currently, the users mapped under Admin or Operator User Groups automatically have this role.

11.3.6 Navigation within Run Execution Summary Page

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



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

11.3.6.1 Run Execution Summary Section

The Run Execution Summary displays the following details:

Field Name	Field Description
Run Name	Displays the name of the Run.
Run Type	Displays the type of Run, baseline or Simulation.
Run ID	Displays the Run Execution ID.

11.3.6.2 Run Execution Details Section

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

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

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

11.3.6.3 Run Execution Grid

The Run Execution Details displays the following details:

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

12 Legal Entity Consolidation

12.1 Introduction

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. User 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 setup using the rule ‘Capital Consolidation Level Selection’ in the process 'CAPITAL_CONSOLIDATION’.

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

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

12.1.1 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 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.

12.2 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 intra group assets and liabilities, equity, income, expenses and cash flows that relates to transactions between entities of the group (profits or losses resulting from intra group 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

12.3 Types of Consolidation

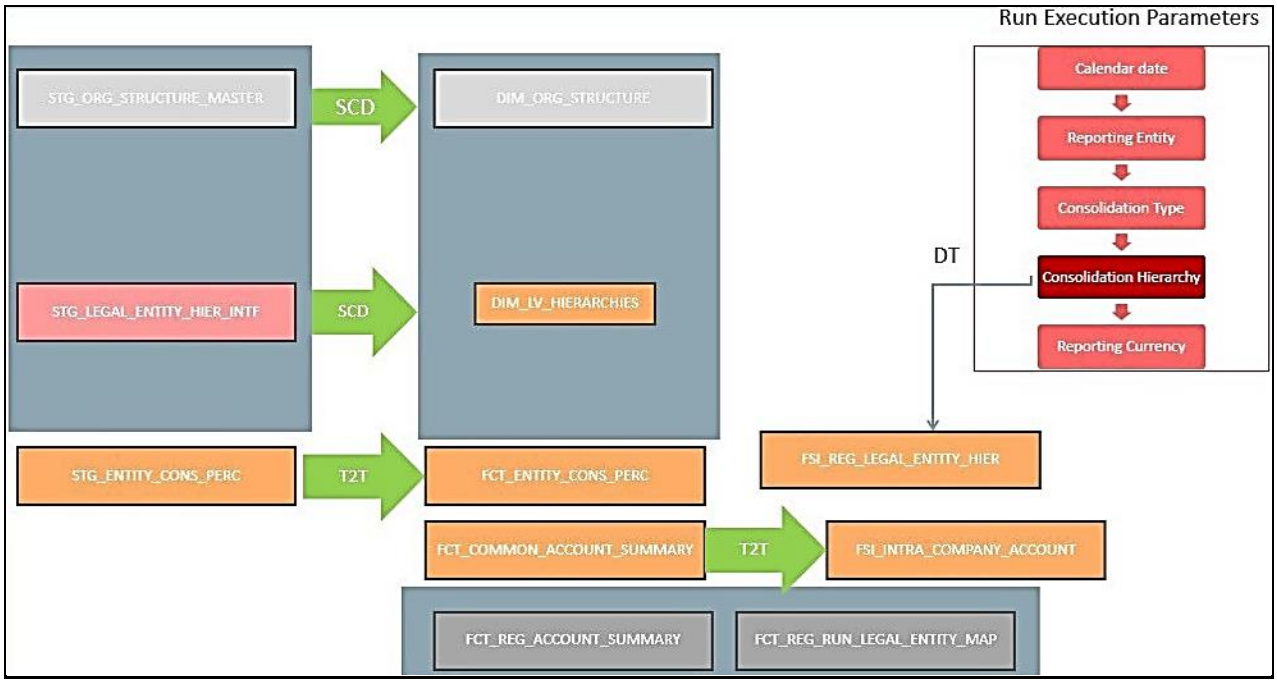
The types of Consolidation are as follows:

- 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 parent in subsidiary.

12.4 Consolidation Activities

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



12.5 Legal Entity Tables

This section provides information about Insurance Legal Entity population tables in the Oracle Insurance Data Foundation application and step-by step instructions to use this section.

This section includes the following topics:

- About Legal Entity Dimension Tables
- About Legal Entity T2T (Result Table)
- Deploying Legal Entity Tables on Hive
- Populating Legal Entity Dimension Tables
- Populating Legal Entity T2T Result Tables

12.5.1 About Legal Entity Dimension Tables

For information about the introduction to Legal Entity and the purpose of the Legal Entity tables, see Introduction.

Legal Entity Dimension table names and their description are given here.

Logical Dimension Table Name	Dimension Table Description
Legal Proceeding Status Dimension	This table stores the legal proceeding status codes for the customer along with the descriptions for each status code.
Legal Entity Group Dimension	This table stores the list of all the defined Legal Entity groups in an organization structure.
Legal Reporting Dimension	This table stores the legal reporting hierarchy in an organization. The lowest level of the hierarchy is the booking transit and the highest level is the whole Insurance Company.

The mapping details for the Legal Entity Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
269	STG_LEGAL_PROCEDING_STS_MASTER	Stage Legal Proceeding Status Master	DIM_LEGAL_PROCEDING_STATUS	Legal Proceeding Status Dimension
452	STG_LEGAL_ENTITY_GROUP_MASTER	Stage Legal Entity Group Master	DIM_LEGAL_ENTITY_GROUP	Legal Entity Group Dimension
58	STG_LEGAL_ENTITY_MASTER	Stage Legal Entity Master	DIM_LEGAL_REPORTING	Legal Reporting Dimension

12.5.2 About Legal Entity T2T (Result Table)

Legal Entity T2T and its description is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_LEGAL_ENTITY_DETAILS	Stage Legal Entity Details	FCT_LEGAL_ENTITY_DETAILS	Fact Legal Entity Details	T2T_FCT_LEGAL_ENTITY_DETAILS

12.6 Deploying Legal Entity Tables on Hive

All RDBMS related Legal Entity Dimension and Result tables can also be deployed on Hive (Stage and Results on Hive).

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.

12.7 Populating Legal Entity Dimension Tables

Follow this SCD process to populate data into any Legal Entity Dimension table:

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

1. To populate data into any Legal Entity Dimension table, execute the SCD batch. For a detailed procedure, see Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Legal Entity 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.

12.8 Populating Legal Entity T2T Result Table

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

> **NOTE** You can also follow this T2T process to populate data into any Hive-related Legal Entity T2T Result table.

1. To populate data into any Legal Entity T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Legal Entity Result table, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

13 Exchange Rates Tables

This chapter provides information about Exchange Rates tables in the Data Foundation application and step-by step instructions to use this section.

This chapter includes the following topics:

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

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

Handling Alternate Currency:

Exchange Rates is supplied by the customers in downloadable format. The applications may require the information in a different format. The current Exchange Rates population is enhanced to store the following variations:

- Inverse Rate: When the exchange rate information of two currencies are provided for a base and a counter currency, the inverse rates if not available is derived and populated.
- Triangulation Rate: When exchange rate information of two pairs are available as part of download, with a common currency in each of the pair, exchange rate for currencies not common as part of the download pair is derived and populated.

13.1 About Exchange Rates T2T (Result Table)

Exchange Rates T2T and its description is given here.

T2T Name	T2T Description
T2T_FSI_EXCHANGE_RATES	This T2T stores history of the exchange rates between two currencies.

The mapping details for the Exchange Rates T2T is given here.

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

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
VW_FSI_RATE_TRIANGULATION	FSI Rate Triangulation View	FSI_EXCHANGE_RATES	FSI Exchange Rates	T2T_FSI_EXCHANGE_RATES

13.2 Deploying Exchange Rates Tables on Hive

All RDBMS related Exchange Rates Result table can also be deployed on Hive (Stage and Results on Hive).

> **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.

13.3 Populating Exchange Rates T2T Result Table

Follow this T2T process to populate data into Exchange Rates T2T Result table:

> **NOTE** You can also follow this T2T process to populate data into any Hive related Exchange Rates T2T Result table.
>
> FSI_EXCHANGE_RATES table must be loaded prior loading any of the Account Summary tables.

1. To populate data into Exchange Rates T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Exchange Rates Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

14 Party Subject Area

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

This chapter includes the following topics:

14.1 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.

14.2 Party Dimension Table

Party Dimension table stores history of a Party. A Party here can be a customer, issuer, guarantor, etc.

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 various 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. SCD-32 is planned to be deprecated in a future release.

14.2.1 Using Stage Party Master to Load Party Dimension

The Party Master table contains information related to a person or an organization that is a party to Insurance Company. Party refers to customer, issuer, guarantor, etc. This table holds the master list of parties across all the roles and their details such as name, age, education, profession, gender, etc.

Party ID: Party identifier is a code that identifies whether the Party ID provided for a party is an organization identifier or a private identifier.

The mapping details to load to the Party Dimension table from the Stage Party Master table is given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
117	STG_PARTY_MASTER	Stage Party Master	DIM_PARTY	Party Dimension

14.2.2 Using Stage Party Type Master to Load Party Type Dimension

The Party Type Master table contains information related to type of the Party that may be an Individual, an Insurance Company, a Corporate - Small, a Corporate - Medium, a State Government, and a Sovereign etc.

Party ID: Party identifier is a code that identifies whether the Party ID provided for a party is an organization identifier or a private identifier.

The mapping details to load to the Party Dimension Type table from the Stage Party Type Master table is given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
180	STG_PARTY_TYPE_MASTER	Stage Party Type Master	DIM_PARTY_TYPE	Party Dimension Type

14.2.3 Using Stage Party Role Map to Populate Party Role Map T2T (Result Table)

NOTE Load STG_PARTY_ROLE_MAP with all customer records and Party Role must be ‘CUSTOMER’.

The Party Role Map table stores information related to different roles that are played by a Party. Roles here refers to Beneficiary, Producer, Agent, Issuer etc. If same party plays multiple roles, one needs to update party role map entity continuing with same record in stage party master with or without role specific data addition.

The mapping details to populate the Party Role Map T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_PARTY_ROLE_MAP	Stage Party Role Map	FCT_PARTY_ROLE_MAP	Fact Party Role Map	T2T_FCT_PARTY_ROLE_MAP

14.2.4 Using Stage Party Insurance Policy Role Map T2T (Result Table)

> **NOTE** Load STG_PARTY_INS_POLICY_ROLE_MAP with all customer records and Party Role must be 'CUSTOMER'.

The Party Insurance Policy Role Map table stores information related to different policies that are taken by a Party.

The mapping details to populate the Party Insurance Policy Role Map T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_PARTY_INS_POLICY_ROLE_MAP	Stage Party Insurance Policy Role Map	FCT_PARTY_INS_POLICY_ROLE_MAP	Fact Party Insurance Policy Role Map	T2T_FCT_PARTY_INS_POLICY_ROLE_MAP

14.3 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, etc.

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.

14.3.1 About Party Definition T2Ts (Result Tables)

Party Definition T2T consists of the Party and Party Probability of Default details.

Party Definition T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCT_PARTY_DETAILS	This T2T stores information related to Party such as Annual Income, and Primary Source of Wealth, etc.
T2T_FCT_PARTY_PD_DETAILS	This T2T stores the probability of default values as of given date for all relevant party.

The mapping details for the Party Definition T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_PARTY_DETAILS	Stage Party Details	FCT_PARTY_DETAILS	Fact Party Details	T2T_FCT_PARTY_DETAILS
STG_PARTY_PD_DETAILS	Stage Party Probability of Default Details	FCT_PARTY_PD_DETAILS	Fact Party Probability of Default Details	T2T_FCT_PARTY_PD_DETAILS

14.4 Party Contact Details Tables

Party Contact Details consists of details about phone, email, and address.

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

14.4.1 About Party Contact Details Dimension Tables

Party Contact Details Dimension table names and their description are given here.

Logical Dimension Table Name	Dimension Table Description
Phone Dimension	This table stores the master phone details.
Address Dimension	This table stores the master address details.
Email Dimension	This table stores the master email details.
Contact Dimension	This table stores the list of contacts imported by the Insurance Company.
Email Address Purpose Type Dimension	This table stores the email address purpose, or usage, of this address relative to this customer.
Phone Purpose Type Dimension	This table stores the phone purpose type, or usage, of this phone relative to this account.

The mapping details for the Party Contact Details Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
345	STG_PHONE_MASTER	Stage Phone Master	DIM_PHONE	Phone Dimension
346	STG_ADDRESS_MASTER	Stage Address Master	DIM_ADDRESS	Address Dimension
347	STG_EMAIL_MASTER	Stage Email Master	DIM_EMAIL	Email Dimension
144	STG_CONTACT_MASTER	Stage Contact Master	DIM_CONTACT	Contact Dimension
667	STG_EMAIL_ADRS_PRP_TYPE_MASTER	Stage Email Address Purpose Type Master	DIM_EMAIL_ADRS_PRP_TYPE	Email Address Purpose Type Dimension
668	STG_PHONE_PURPOSE_TYPE_MASTER	Stage Phone Purpose Type Master	DIM_PHONE_PURPOSE_TYPE	Phone Purpose Type Dimension

14.4.2 About Party Contact T2Ts (Result Tables)

Party Contact Details T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCT_PARTY_PHONE_MAP	This T2T stores the Party phone details.
T2T_FCT_PARTY_ADDRESS_MAP	This T2T stores the Party address details.
T2T_FCT_PARTY_EMAIL_MAP	This T2T stores the Party email details.

The mapping details for the Party Contact Details T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_PARTY_PHONE_MAP	Stage Party Phone Map	FCT_PARTY_PHONE_MAP	Fact Party Phone Map	T2T_FCT_PARTY_PHONE_MAP
STG_PARTY_ADDRESS_MAP	Stage Party Address Map	FCT_PARTY_ADDRESS_MAP	Fact Party Address Map	T2T_FCT_PARTY_ADDRESS_MAP
STG_PARTY_EMAIL_MAP	Stage Party Email Map	FCT_PARTY_EMAIL_MAP	Fact Party Email Map	T2T_FCT_PARTY_EMAIL_MAP

14.5 Party Identification Document Tables

The Party Identification Document contains 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.

14.5.1 About Party Identification Document Dimension Tables

Party Identification Document Dimension table names and their description are given here.

Logical Dimension Table Name	Dimension Table Description
Document Type Dimension	This table stores the list of all types of documents that are required by the insurance company for an account. The document types can be those specified by the Insurance Company or those required for the process in the Insurance Company.
Document Sub Type Dimension	The table stores the document sub type. Sample values: If the document type is visa then the sub type is student or tourist visa.

The mapping details for the Party Identification Document Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
457	STG_DOCUMENT_TYPE_MASTER	Stage Document Type Master	DIM_DOCUMENT_TYPE	Document Type Dimension
553	STG_DOCUMENT_SUB_TYPE_MASTER	Stage Document Sub Type Master	DIM_DOCUMENT_SUB_TYPE	Document Sub Type Dimension

14.5.2 About Party Identification Document T2T (Result Table)

Party Identification Document T2T and its description is given here.

T2T Name	T2T Description
T2T_FCT_PARTY_IDENTIFCTN_DOC	This T2T stores identification details of an individual Party. Identification means the action or process of identifying someone or something, or the fact of being identified.

The mapping details for the Party Identification Document T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_PARTY_IDENTIFICATION_DOC	Stage Party Identification Document	FCT_PARTY_IDENTIFICATION_DOC	Fact Party Identification Document	T2T_FCT_PARTY_IDENTIFCTN_DOC

14.6 Party Medical Attributes Tables

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

14.6.1 About Party Medical Attributes Dimension Tables

Party Medical Attributes Dimension table names and their description are given here.

Logical Dimension Table Name	Dimension Table Description
Medical Test Type Dimension	This table stores the medical test details that the Party has undergone. A medical test is a kind of medical procedure performed to detect, diagnose, or monitor diseases, disease processes, susceptibility, and determine a course of treatment. It is related to clinical chemistry and molecular diagnostics, and the procedures are typically performed in a medical laboratory.
Medical Treatment Type Dimension	This table stores details of the different types of medical treatment a Party has undergone due to illness, accidents or any medical emergency. Medical treatment means the management and care of a patient to combat disease or disorder.
Party Medical Condition Type Dimension	This table stores the medical condition type details of the Party if that Party is suffering from any health issues. The medical condition types are used to describe the condition of a patient (that is, their clinical status) in a hospital. Sample values: TUMORBENIGN, TUMORMALIGNANT, TUMOR, AAA, etc.
Medical Test Status Dimension	This table stores the medical test status details of a Party. Sample values: ACKNOWLEDGED, ADD, APPROVED, CANCELLED, COMPLETED, WARNING, etc.
Medical Test Result Type Dimension	This table stores the medical test result type details. Sample values: ANEG, APOS, ABNEG, ABPOS, ABNORMAL, AMBER, CLOTTED, etc.
Medical Test Reference Range Dimension	This table stores details about the medical test reference range codes. Sample values: MAXVALUE, MINVALUE.
Medical Condition Status Dimension	This table stores the medical condition status details. Sample values: COMPLETED, ONGOING, PENDING, RECOVERED, etc.
Medical Condition Cause Type Dimension	This table stores details about the medical condition cause type. Sample values: ANXIETY, CAFFEINE, FAMDEATH, etc.
Lab Test Remarks Dimension	This table stores the lab test remarks that are general classification of the remarks. Sample values: HIGH, LOW, NORMAL.
Laboratory Kit Type Dimension	This table stores the laboratory kit type details. Sample values: BLOOD, BLUR, ORAL, etc.

Logical Dimension Table Name	Dimension Table Description
Substance Usage Type Dimension	This table stores the substance type details if the Party is in usage of any substances.
Family Illness Type Dimension	This table stores the family illness type, if the Party has any illness that is connected to his/her family background. Sample Values: BLOOD PRESSURE, HEART ATTACK, DIABETES.
Disability Reason Dimension	This table stores the disability reason as to why the party is disabled. Sample values: DTA, FI.
Disability Type Dimension	This table stores the details of the type of disability caused to the Party due to the medical complaints or accidents or family illness. Sample values: LTC, LTD, STD, STC.
Individual Smoking History Dimension	This table stores the individual smoking history, which indicates the history of tobacco use of a client. Sample values: CURRENT, NEVER, PRIOR, OTHER, UNKNOWN.

The mapping details for the Party Medical Attributes Dimension tables are given here.

Map reference number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
477	STG_MEDICAL_TEST_TYPE_MASTER	Stage Medical Test Type Master	DIM_MEDICAL_TEST_TYPE	Medical Test Type Dimension
478	STG_MED_TREATMENT_TYPE_MASTER	Stage Medical Treatment Type Master	DIM_MEDICAL_TREATMENT_TYPE	Medical Treatment Type Dimension
479	STG_PARTY_MED_CNDN_TYPE_MASTER	Stage Party Medical Condition Type Master	DIM_PARTY_MEDICAL_CNDTN_TYPE	Party Medical Condition Type Dimension
482	STG_MEDICAL_TEST_STATUS_MASTER	Stage Medical Test Status Master	DIM_MEDICAL_TEST_STATUS	Medical Test Status Dimension
563	STG_MED_TEST_RSLT_TYPE_MASTER	Stage Medical Test Result Type Master	DIM_MED_TEST_RESULT_TYPE	Medical Test Result Type Dimension
564	STG_MED_TEST_REF_RANGE_MASTER	Stage Medical Test Reference Range Master	DIM_MED_TEST_REF_RANGE	Medical Test Reference Range Dimension
565	STG_MED_CONDTN_STATUS_MASTER	Stage Medical Condition Status Master	DIM_MED_CONDTN_STATUS	Medical Condition Status Dimension
566	STG_MED_CNDTN_CAUS_TYPE_MASTER	Stage Medical Condition Cause Type Master	DIM_MED_CONDTN_CAUSE_TYPE	Medical Condition Cause Type Dimension
481	STG_LAB_TEST_REMARKS_MASTER	Stage Lab Test Remarks Master	DIM_LAB_TEST_REMARKS	Lab Test Remarks Dimension

Map reference number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
561	STG_LABORATORY_KIT_TYPE_MASTER	Stage Laboratory Kit Type Master	DIM_LABORATORY_KIT_TYPE	Laboratory Kit Type Dimension
480	STG_SUBSTANC_USAGE_TYPE_MASTER	Stage Substance Usage Type Master	DIM_SUBSTANCE_USAGE_TYPE	Substance Usage Type Dimension
476	STG_FAMILY_ILLNESS_TYPE_MASTER	Stage Family Illness Type Master	DIM_FAMILY_ILLNESS_TYPE	Family Illness Type Dimension
473	STG_DISABILITY_REASON_MASTER	Stage Disability Reason Master	DIM_DISABILITY_REASON	Disability Reason Dimension
474	STG_DISABILITY_TYPE_MASTER	Stage Disability Type Master	DIM_DISABILITY_TYPE	Disability Type Dimension
562	STG_IND_SMOKING_HISTORY_MASTER	Stage Individual Smoking History Master	DIM_IND_SMOKING_HISTORY	Individual Smoking History Dimension

14.6.2 About Party Medical Attributes T2T (Result Tables)

Party Medical Attributes T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCT_MEDICAL_TEST_DETAILS	This T2T stores the medical test details of a Party.
T2T_FCT_PARTY_SUBS_USAGE_DTLS	This T2T stores the substance usage details of a Party, if the party is in usage of any of the tobacco, alcohol or harmful substance that leads to the medical complaints or conditions.
T2T_FCT_PARTY_FAMILY_MEDICAL_DA	This T2T stores the family medical data of a Party. The data is about medical history of the family members if any specific illness is present among most of the members.
T2T_FCT_PARTY_MDCAL_CONDITN_DTL	This T2T stores the medical enquiry details or medical condition details of a Party on a given date. These are the details from questionnaire asked by insurance agents mostly at time of underwriting or deemed necessary during life of policy as per policy terms and conditions.
T2T_FCT_PARTY_MEDICAL_TREATMENT	This T2T stores the medical treatment details of a Party, which informs if the Party is either undergoing or has undergone any medical treatment due to any medical conditions or accidents or family illness.
T2T_FCT_PARTY_DISABILITY_DETL	This T2T stores the Party disability details.

The mapping details for the Party Medical Attributes T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_MEDICAL_TEST_DETAILS	Stage Medical Test Details	FCT_MEDICAL_TEST_DETAILS	Fact Medical Test Details	T2T_FCT_MEDICAL_TEST_DETAILS
STG_PARTY_SUBSTANCE_USAGE_DTLS	Stage Party Substance Usage Details	FCT_PARTY_SUBSTANCE_USAGE_DTLS	Fact Party Substance Usage Details	T2T_FCT_PARTY_SUBS_USAGE_DTLS
STG_PARTY_FAMILY_MEDICAL_DATA	Stage Party Family Medical Data	FCT_PARTY_FAMILY_MEDICAL_DATA	Fact Party Family Medical Data	T2T_FCT_PARTY_FAMILY_MEDICAL_DA
STG_PARTY_MEDICAL_CONDITN_DTLS	Stage Party Medical Condition Details	FCT_PARTY_MEDICAL_CONDITN_DTLS	Fact Party Medical Condition Details	T2T_FCT_PARTY_MDCAL_CONDITN_DTL
STG_PARTY_MEDICAL_TRTMNTS_DTLS	Stage Party Medical Treatments Details	FCT_PARTY_MEDICAL_TREATMENTS	Fact Party Medical Treatments Details	T2T_FCT_PARTY_MEDICAL_TREATMENT
STG_PARTY_DISABILITY_DETAILS	Stage Party Disability Details	FCT_PARTY_DISABILITY_DETAILS	Fact Party Disability Details	T2T_FCT_PARTY_DISABILITY_DETL

14.7 Party Financial Data/Employment Attributes Tables

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

14.7.1 About Party Financial Data/Employment Attributes Dimension Tables

Party Financial Data/Employment Attributes Dimension table names and their description are given here.

Logical Dimension Table Name	Dimension Tables Description
Profession Dimension	This table stores the master list of all customer professions.
Profession Sub Type Dimension	This table stores the profession sub type. Sample value: Orthodontist.

The mapping details for the Party Financial Data/Employment Attributes Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
334	STG_PROFESSION_MASTER	Stage Profession Master	DIM_PROFESSION	Profession Dimension
676	STG_PROFESSION_SUB_TYPE_MASTER	Stage Profession Sub Type Master	DIM_PROFESSION_SUB_TYPE	Profession Sub Type Dimension

14.7.2 About Party Financial Data/Employment Attributes T2T (Result Table)

Party Financial Data/Employment Attributes T2T and its description is given here.

T2T Name	T2T Description
T2T_FCT_PARTY_EMPLOYMENT_DETAILS	This T2T stores the employment details of an individual Party. This data applies only to Parties who are individuals. A single Party may consist of multiple employment details.

The mapping details for the Party Financial Data/Employment Attributes T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_PARTY_EMPLOYMENT_DETAILS	Stage Party Employment Details	FCT_PARTY_EMPLOYMENT_DETAILS	Fact Party Employment Details	T2T_FCT_PARTY_EMPLOYMENT_DETAILS

14.8 Party Producer Exam, Certification, and License Tables

An Insurance Producer (also called an agent or insurance broker) means any person (individual, corporation, association, partnership, etc.) 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 eligibility of the Producer by evaluating the type of exam and its results, certifications completed, and license obtained by the Producer.

14.8.1 Using Stage Party Master to Load Producer Dimension

Populate the Party master where Party Type is Producer and/or Producer Agent. The Party Master table contains information related to a person or an organization that is a party to an Insurance Company. Party refers to producer, customer, issuer, guarantor, etc. This table holds the master list of parties across all the roles and their details such as name, age, education, profession, gender, etc.

Party ID: Party identifier is a code that identifies whether the Party ID provided for a party is an organization identifier or a private identifier.

The mapping details for the Producer Dimension table is given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
117	STG_PARTY_MASTER	Stage Party Master	DIM_PRODUCER	Producer Dimension

14.8.2 About Party Producer Exam, Certification, and License Dimension Tables

Party Producer Exam, Certification, and License Dimension table names and their description are given here.

Logical Dimension Table Name	Dimension Table Description
Producer Dimension	This table stores producer details.
Producer Status Dimension	This table stores the Producer status if the Party is a Producer.
Producer Agent Dimension	This table stores producer details who is an insurance agent.
Examination Dimension	This table stores the static fields of the Examination. An Exam is the formal verification or validation of information learned/known, typically used to satisfy a requirement for licenses, registrations or other firm education proof.
Exam Type Dimension	This table stores the exam type. Sample values: CEGP, LLQPB.
Insurance Exam Status Dimension	This table stores the status of the insurance license exam attempted by the Party.
Certification Dimension	This table stores the static fields of the Certification. Certification is formal procedure by which an accredited or authorized person or agency assesses and verifies (and attests in writing by issuing a certificate) the attributes, characteristics, quality, qualification, or status of individuals or organizations, goods or services, procedures or processes, or events or situations, in accordance with established requirements or standards.
Insurance License Dimension	This table stores the license related details of the Party as declared by the respective authorities of the insurance institutions. Here, license refers to the certificate or the document itself that confers permission to engage in insurance activity.
Insurance License Class Type Dimension	This table stores the Insurance License Class Type. Sample values: AGNCY, INDPART.
Insurance License Status Dimension	This table contains the license status of the Party who has applied for license or who has given exam.

The mapping details for the Party Producer Exam, Certification, and License Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
412	VW_STG_PARTY_MASTER_PRODUCER_V	Stage Party Master Producer View	DIM_PRODUCER	Producer Dimension
237	VW_STG_PARTY_MASTER_PROD_AGENT_V	Stage Party Master Producer Agent View	DIM_PRODUCER_AGENT	Producer Agent Dimension

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
559	STG_PRODUCER_STATUS_MASTER	Stage Producer Status Master	DIM_PRODUCER_STATUS	Producer Status Dimension
475	STG_CERTIFICATION_MASTER	Stage Certification Master	DIM_CERTIFICATION	Certification Dimension
554	STG_EXAMINATION_MASTER	Stage Examination Master	DIM_EXAMINATION	Examination Dimension
560	STG_EXAM_TYPE_MASTER	Stage Exam Type Master	DIM_EXAM_TYPE	Exam Type Dimension
555	STG_INS_EXAM_STATUS_MASTER	Stage Insurance Exam Status Master	DIM_INS_EXAM_STATUS	Insurance Exam Status Dimension
557	STG_INSURANCE_LICENSE_MASTER	Stage Insurance License Master	DIM_INSURANCE_LICENSE	Insurance License Dimension
556	STG_INS_LICENS_CLS_TYPE_MASTER	Stage Insurance License Class Type Master	DIM_INS_LICENSE_CLASS_TYPE	Insurance License Class Type Dimension
558	STG_INS_LICENSE_STATUS_MASTER	Stage Insurance License Status Master	DIM_INSURANCE_LICENSE_STATUS	Insurance License Status Dimension

14.8.3 About Party Producer Exam, Certification, and License T2T (Result Tables)

Party Producer Exam, Certification, and License T2T names and their description are given here.

T2T Name	T2T Description
T2T_FCT_PARTY_EXAM_DETAILS	This T2T stores the details pertaining to the certification and related exams of the Party. An Exam is the formal verification or validation of information learned/known, typically used to satisfy a requirement for licenses, registrations or other firm education proof.
T2T_FCT_PARTY_CERTIFICATE_DTLS	This T2T stores the certification details of the Party. Certification is formal procedure by which an accredited or authorized person or agency assesses and verifies (and attests in writing by issuing a certificate) the attributes, characteristics, quality, qualification, or status of individuals or organizations, goods or services, procedures or processes, or events or situations, in accordance with established requirements or standards. In case of insurance this refers to insurance licentiate and other accreditations.
T2T_FCT_PRDR_AGENT_LICENSE_DETAILS	This T2T stores the Producer License Details of the Party. There are different categories of insurance and a producer must be licensed in each category he or she wishes to transact business.

The mapping details for the Party Producer Exam, Certification, and License T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_PARTY_EXAM_DETAILS	Stage Party Exam Details	FCT_PARTY_EXAM_DETAILS	Fact Party Exam Details	T2T_FCT_PARTY_EXAM_DETAILS
STG_PARTY_CERTIFICATE_DTLS	Stage Party Certificate Details	FCT_PARTY_CERTIFICATE_DTLS	Fact Party Certificate Details	T2T_FCT_PARTY_CERTIFICATE_DTLS
STG_PRDR_AGENT_LICENSE_DETAILS	Stage Producer Agent License Details	FCT_PRDR_AGENT_LICENSE_DETAILS	Fact Producer Agent License Details	T2T_FCT_PRDR_AGENT_LICENSE_DETAILS

14.9 Party Results Tables

Party Results consists of the details related to Party such as Annual Income, and Primary Source of Wealth, etc.

14.9.1 About Party Results Dimension Tables

Party Results Dimension table names and their description are given here.

Logical Dimension Table Name	Remarks
Customer Type Dimension	This table stores the master list of customer type details. Sample values: OUR, OTH.
Credit Rating Dimension	This table stores credit rating information.

The mapping details for the Party Results Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
418	STG_PARTY_TYPE_MASTER	Stage Party Type Master	DIM_CUSTOMER_TYPE	Customer Type Dimension
466	STG_CREDIT_RATING_MASTER	Stage Credit Rating Master	DIM_CREDIT_RATING	Credit Rating Dimension

14.9.2 About Party Results T2T (Result Table)

Party Results T2T and its description is given here.

T2T Name	T2T Description
T2T_FCT_PARTY_DETAILS	This T2T stores information related to Party such as Annual Income, and Primary Source of Wealth, etc.

The mapping details for the Party Results T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_PARTY_DETAILS	Stage Party Details	FCT_PARTY_DETAILS	Fact Party Details	T2T_FCT_PARTY_DETAILS

14.10 Other Party Attributes Tables

This section consists of details about other Party attribute tables.

14.10.1 About Other Party Attributes Dimension Tables

Other Party Attributes Dimension table names and their description are given here.

Logical Dimension Table Name	Dimension Table Description
Party Type Dimension	This table stores Party type information. The Party may be an individual, Corporate - Small, Corporate - Medium, State Government, Sovereign etc.
Party Relationship Type Dimension	This table stores the relationship types defined by the Insurance Company. This table is used to determine the relationship type between two Parties. This can also be used for relationship type between an entity and a Party.
Relation Role Type Dimension	This table stores the Relation Role type details. Sample values: ORIGINATOR, PERFORMER, RECIPIENT, TARGET, etc.
Reinsurer Type Dimension	This table stores the list of types of reinsurers and generated from stage party type master which holds all the party types at one place.

The mapping details for the Other Party Attributes Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
180	STG_PARTY_TYPE_MASTER	Stage Party Type Master	DIM_PARTY_TYPE	Party Type Dimension
382	STG_PARTY_RELTNSHP_TYPE_MASTER	Stage Party Relationship Type Master	DIM_PARTY_RELATIONSHIP_TYPE	Party Relationship Type Dimension
483	STG_RELATION_ROLE_TYPE_MASTER	Stage Relation Role Type Master	DIM_RELATION_ROLE_TYPE	Relation Role Type Dimension

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
597	STG_REINSURER_TYPE_MASTER	Stage Reinsurer Type Master	DIM_REINSURER_TYPE	Reinsurer Type Dimension

14.10.2 About Other Party Attributes T2Ts (Result Tables)

Other Party Attributes T2T and its description is given here.

T2T Name	T2T Description
T2T_FCT_PARTY_DETAILS	This T2T stores the details pertaining to a Party.
T2T_FCT_PARTY_PARTY_RELATIONSHIP	This T2T stores the details about relationship between the Parties.
T2T_FCT_PARTY_RATING_DETAILS	This T2T stores the rating details of the customer or counterparty or guarantor, etc.
T2T_FCT_PARTY_ACCOUNT_ROLE_MAP	This T2T stores the details of role played by the Party against an account.
T2T_FCT_ACCOUNT_RATING_DETAILS	This T2T stores the account rating details from multiple sources.

The mapping details for the Other Party Attributes T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_PARTY_DETAILS	Stage Party Details	FCT_PARTY_DETAILS	Fact Party Details	T2T_FCT_PARTY_DETAILS
STG_PARTY_PARTY_RELATIONSHIP	Stage Party Party Relationship	FCT_PARTY_PARTY_RELATIONSHIP	Fact Party Party Relationship	T2T_FCT_PARTY_PARTY_RELATIONSHIP
STG_PARTY_RATING_DETAILS	Stage Party Rating Details	FCT_PARTY_RATING_DETAILS	Fact Party Rating Details	T2T_FCT_PARTY_RATING_DETAILS
STG_PARTY_ACCOUNT_ROLE_MAP	Stage Party Account Role Map	FCT_PARTY_ACCOUNT_ROLE_MAP	Fact Party Account Role Map	T2T_FCT_PARTY_ACCOUNT_ROLE_MAP
STG_ACCOUNT_RATING_DETAILS	Stage Account Rating Details	FCT_ACCOUNT_RATING_DETAILS	Fact Account Rating Details	T2T_FCT_ACCOUNT_RATING_DETAILS

14.11 Deploying Party Related Tables on Hive

All RDBMS related Party related Dimension and Result tables can also be deployed on Hive (Stage and Results on Hive).

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.

14.12 Populating Party Related Dimension Tables

Follow this SCD process to populate data into any Party related Dimension table:

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

1. To populate data into any Party related Dimension table, execute the SCD batch. For a detailed procedure, see Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Party related 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.

14.13 Populating Party Related T2T Result Tables

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

NOTE You can also follow this T2T process to populate data into any Hive related Party T2T Result table.

1. To populate data into any Party related T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Party related Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

15 Insurance Underwriting Entities

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

This chapter includes the following 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 organization. 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.

15.1 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 as Underwriting Application.

15.1.1 About Insurance Underwriting Application Dimension Tables

Insurance Underwriting Application Dimension table names and their remarks are given here.

Logical Dimension Table Name	Remarks
Underwriting Product Class Dimension	This table stores identification details of the underwriting class and its respective name or names. Underwriting class is carrier specific nomenclature representing the underwriting groups or products or risks in a given organization.
Application Dimension	This table stores the underwriting application details.
Application Type Dimension	This table stores the application type details such as fresh, existing, and enhancements.
Application Group Dimension	This table stores the underwriting application group details. Application group can contain one or more applications.
Application Status Dimension	This table stores the master list of application status such as processing, cancelled by customer, outstanding, outstanding from restructuring, etc.
Document Submission Status Dimension	This table stores the list of various status that a document can have. Document once submitted to an Insurance Company contains different status such as Pending, Received, Mailed for approval, Received Back, Under Processing, Approved, etc.
Rejection Reason Dimension	This table stores all the rejection reasons given by prospects for not buying a product or service.
Application Reject Reasons Dimension	This table stores the list of rejection reasons possible when processing an application.
Application Decision Event Dimension	This table stores the details of the outcome of all decision events for the individual application. During the process of underwriting, an application goes through various stages such as personal identity verification, address verification, employment verification, etc.
Underwriting Model Type Dimension	This table stores the list of all types of underwriting model types.
Underwriting Element Dimension	This table stores the classification details of insurance policy underwriting related to tobacco usage. Sample values: BLENDED, NONSMOKER, SMOKER, TOBACCO, etc.

The mapping details for the Insurance Underwriting Application Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
523	STG_UNW_PRODUCT_CLASS_MASTER	Stage Underwriting Product Class Master	DIM_UNW_PRODUCT_CLASS	Underwriting Product Class Dimension
664	STG_APPLICATIONS	Stage Applications	DIM_APPLICATION	Application Dimension
162	STG_APPL_TYPE_MASTER	Stage Application Type Master	DIM_APPLICATION_TYPE	Application Type Dimension
665	STG_APPLICATION_GROUP_MASTER	Stage Application Group Master	DIM_APPLICATION_GROUP	Application Group Dimension

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
166	STG_APPL_STATUS_MASTER	Stage Application Status Master	DIM_APPLICATION_STATUS	Application Status Dimension
673	STG_DOC_SUBMISSN_STATUS_MASTER	Stage Document Submission Status Master	DIM_DOC_SUBMISSION_STATUS	Document Submission Status Dimension
93	STG_REJECTION_REASON_MASTER	Stage Rejection Reason Master	DIM_REJECTION_REASON	Rejection Reason Dimension
3	STG_APPL_REJECT_REASON_MASTER	Stage Application Reject Reasons Master	DIM_APPLICATION_REJECT_REASONS	Application Reject Reasons Dimension
666	STG_APPLN_DECISION_EVNT_MASTER	Stage Application Decision Event Master	DIM_APPLN_DECISION_EVNT	Application Decision Event Dimension
409	STG_UNDRWRTNG_MDL_TYPE_MASTER	Stage Underwriting Model Type Master	DIM_UNDERWRITING_MODEL_TYPE	Underwriting Model Type Dimension
522	STG_UNDERWRITING_ELMNT_MASTER	Stage Underwriting Element Master	DIM_UNDERWRITING_ELEMENT	Underwriting Element Dimension
238	STG_UNDERWRITER_MASTER	Stage Underwriter Master	DIM_UNDERWRITER	Underwriter Dimension

15.1.2 About Insurance Underwriting Application T2Ts (Result Tables)

Insurance Underwriting Application T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCT_APPLICATION	This T2T stores the applications processed in each period, for an analysis over time.
T2T_FCT_APPLICATION_GROUP	This T2T stores the Application Group details. Application group can consist of one or more applications. In general, insurance underwriting team creates the application group.
T2T_FCT_APPLICATION_DOCUMENT	This T2T stores the documents submitted by the applicant. Underwriters may ask multiple verification documents from those seek to cover the risk with insurance company. All such documentary evidences are stored in this entity.
T2T_FCT_APPLICATION_DOC_MAP	This T2T stores the details of the document associated with the application and the status of the document such as a document is approved or rejected.
T2T_FCT_APPLICATION_EVENT_DEC	This T2T stores details of the outcome of all decision events for the individual application. During the process of underwriting, an application goes through various stages such as personal identity verification, address verification, employment verification, etc.
T2T_FCT_APPLICATIONS_DEVIATION	This T2T stores the application deviation details.

T2T Name	T2T Description
T2T_FCT_APPLICNS_DOC_PRINT_LOG	This T2T stores the history of documents printed at origination. Multiple documents can be printed for the loan at origination.
T2T_FCT_APPLN_GROUP_EVENT_DEC	This T2T stores details of the outcome of all decision events for an Application Group. During the process of underwriting, an application goes through various stages like personal identity verification, address verification, employment verification, etc.
T2T_FCT_APPLN_PARTY_ROLE_MAP	This T2T stores mapping details of an application to multiple roles played by a party. A party can be primary applicant, co-applicant, Guarantor, Nominee etc.

The mapping details for the Insurance Underwriting Application T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_APPLICATIONS	Stage Applications	FCT_APPLICATION	Fact Applications	T2T_FCT_APPLICATION
STG_APPLICATION_GROUP	Stage Application Group	FCT_APPLICATION_GROUP	Fact Application Group	T2T_FCT_APPLICATION_GROUP
STG_APPLICATION_DOCUMENT	Stage Application Document	FCT_APPLICATION_DOCUMENT	Fact Application Document	T2T_FCT_APPLICATION_DOCUMENT
STG_APPLICATION_DOCUMENT_MAP	Stage Application Document Map	FCT_APPLICATION_DOCUMENT_MAP	Fact Application Document Map	T2T_FCT_APPLICATION_DOC_MAP
STG_APPLICATION_EVENT_DECISION	Stage Application Event Decision	FCT_APPLICATION_EVENT_DECISION	Fact Application Event Decision	T2T_FCT_APPLICATION_EVENT_DEC
STG_APPLICATIONS_DEVIATION	Stage Applications Deviation	FCT_APPLICATIONS_DEVIATION	Fact Applications Deviation	T2T_FCT_APPLICATIONS_DEVIATION
STG_APPLICATIONS_DOC_PRINT_LOG	Stage Applications Document Print Log	FCT_APPLICATIONS_DOC_PRINT_LOG	Fact Applications Document Print Log	T2T_FCT_APPLICNS_DOC_PRINT_LOG
STG_APPLN_GROUP_EVENT_DECISION	Stage Application Group Event Decision	FCT_APPLN_GROUP_EVENT_DECISION	Fact Application Group Event Decision	T2T_FCT_APPLN_GROUP_EVENT_DEC
STG_APPLN_PARTY_ROLE_MAP	Stage Application Party Role Map	FCT_APPLN_PARTY_ROLE_MAP	Fact Application Party Role Map	T2T_FCT_APPLN_PARTY_ROLE_MAP

15.2 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.

15.2.1 About Insurance Underwriting Quotes Dimension Tables

Insurance Underwriting Quotes Dimension table names and their remarks are given here.

Logical Dimension Table Name	Remarks
Quote Source Dimension	This table stores the description of the source of the quote. For example, Producing Agency, Producing Agent, Customer, etc.
Quote Source Method Dimension	This table stores the description of the source method of the quotes. For example, Turbo rater, quick quote, etc.
Quote Submission Method Dimension	This table stores details of different methods from which the Insurance Company receives the quote. For example, mail, fax, internet, etc.
Quote Declination Type Dimension	This table stores all types of quote declinations. For example, Rates too high, Billing Plan, Fraud and Bankruptcy, Uninsurable Risks, Age of Building, etc.

The mapping details for the Insurance Underwriting Quotes Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
251	STG_QUOTE_SOURCE	Stage Quote Source	DIM_QUOTE_SOURCE	Quote Source Dimension
252	STG_QUOTE_SOURCE_METHOD	Stage Quote Source Method	DIM_QUOTE_SOURCE_METHOD	Quote Source Method Dimension
253	STG_QUOTE_SUBMISSION_METHOD	Stage Quote Submission Method	DIM_QUOTE_SUBMISSION_METHOD	Quote Submission Method Dimension
250	STG_QUOTE_DECLINATION_TYPE	Stage Quote Declination Type	DIM_QUOTE_DECLINATION_TYPE	Quote Declination Type Dimension

15.2.2 About Insurance Underwriting Quotes T2Ts (Result Tables)

Insurance Underwriting Quotes T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCT_INSURANCE_QUOTES	This T2T stores all the quote related attributes and measures.
T2T_FCT_INSURANCE_QUOTE_REPORTS	This T2T stores report related information for each quote.

The mapping details for the Insurance Underwriting Quotes T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_INSURANCE_QUOTES	Stage Insurance Quotes	FCT_INSURANCE_QUOTES	Fact Insurance Quotes	T2T_FCT_INSURANCE_QUOTES
STG_INSURANCE_QUOTE_REPORTS	Stage Insurance Quote Reports	FCT_INSURANCE_QUOTES_REPORTS	Fact Insurance Quote Reports	T2T_FCT_INSURANCE_QUOTE_REPORTS

15.3 Insurance Underwriting Tables for Risk Analysis

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

15.3.1 About Insurance Underwriting Dimension Tables for Risk Analysis

Underwriting Dimension table names (for Risk Analysis) and their remarks are given here.

Logical Dimension Table Name	Remarks
Underwriting Risk Sub Type Dimension	This table stores the underwriting sub risk type. Sample values: Preferred, Rated, Substandard, Standard, Uninsurable etc.
Underwriting Risk Type Dimension	This table stores the underwriting risk type. Sample values: best, better, worse, worst, other, unknown.

The mapping details for the Underwriting Dimension tables (for Risk Analysis) are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
524	STG_UNW_RISK_SUB_TYPE_MASTER	Stage Underwriting Risk Sub Type Master	DIM_UNW_RISK_SUB_TYPE	Underwriting Risk Sub Type Dimension
525	STG_UNW_RISK_TYPE_MASTER	Stage Underwriting Risk Type Master	DIM_UNW_RISK_TYPE	Underwriting Risk Type Dimension

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

Underwriting T2T and its description is given here.

T2T Name	T2T Description
T2T_FCT_UNW_PRODUCT_CLASS_DETAILS	This T2T stores the underwriting class product details. It provides life product allowed issue limitations based on the underwriting

T2T Name	T2T Description
	risk properties including gender, age, tobacco usage, underwriting class, and face/coverage amount.

The mapping details for the Underwriting T2T table is given here

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_UNW_PRODUCT_CLASS_DETAILS	Stage Underwriting Product Class Details	FCT_UNW_PRODUCT_CLASS_DETAILS	Fact Underwriting Product Class Details	T2T_FCT_UNW_PRODUCT_CLASS_DETAILS

15.3.3 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.

15.3.3.1 About Party Criminal Conviction Dimension Table

Party Criminal Conviction Dimension table name and its remarks is given here.

Logical Dimension Table Name	Remarks
Crime Type Dimension	This table stores the crime type if the Party is involved in any criminal activity.

The mapping details for the Party Criminal Conviction Dimension table is given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
550	STG_CRIME_TYPE_MASTER	Stage Crime Type Master	DIM_CRIME_TYPE	Crime Type Dimension

15.3.3.2 About Party Criminal Conviction T2T (Result Table)

Party Criminal Conviction T2T and its description is given here.

T2T Name	T2T Description
T2T_FCT_CRIMINAL_CONVICTION_DTLS	This T2T stores the criminal conviction details if the Party is involved in any criminal activity.

The mapping details for the Party Criminal Conviction T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_CRIMINAL_CONVICTION_DTLS	Stage Criminal Conviction Details	FCT_CRIMINAL_CONVICTION_DTLS	Fact Criminal Conviction Details	T2T_FCT_CRIMINAL_CONVICTION_DTLS

15.3.4 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 a suitable insurance coverage.

15.3.4.1 About Party Driving Violation Dimension Tables

Party Driving Violation Dimension table names and their remarks are given here.

Logical Dimension Table Name	Remarks
Driving violation type dimension	This table stores master list of driving violation type.
Driving Violation Charge Category Dimension	This table stores the driving violation charge category details. Sample values: Felony, Misdemeanor, Other, Unknown.

The mapping details for the Party Driving Violation Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
551	STG_DRIVIN_VIOLATN_TYPE_MASTER	Stage Driving Violation Type Master	DIM_DRIVING_VIOLATION_TYPE	Driving Violation Type Dimension
552	STG_DVIOLTN_CHRGE_CTGRY_MASTER	Stage Driving Violation Charge Category Master	DIM_DVIOLTN_CHRGE_CTGRY	Driving Violation Charge Category Dimension

15.3.4.2 About Party Driving Violation T2T (Result Table)

Party Driving Violation T2T and its description is given here.

T2T Name	T2T Description
T2T_FCT_DRIVING_VIOLATION_DTLS	This T2T stores the driving violation details related to the party identifier.

The mapping details for the Party Driving Violation T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_DRIVING_VIOLATION_DTLS	Stage Driving Violation Details	FCT_DRIVING_VIOLATION_DTLS	Fact Driving Violation Details	T2T_FCT_DRIVING_VIOLATION_DTLS

15.3.5 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 particular Lifestyle Activities of the Party.

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

15.3.5.1 About Party Lifestyle Activity Dimension Tables

Party Lifestyle Activity Dimension table names and their remarks are given here.

Logical Dimension Table Name	Remarks
Life Style Activity Type Dimension	This table stores the details of life style activity type. Sample values: AUTORACE, AVIATION, BOATRACE, BOXING, RUGBY, FORTRAVEL, MILITARY, ROCKCLIMB, DIVE, etc.
Life Style Activity Dimension	This table stores the details about lifestyle activity of a Party, its start date and end date.
Life Style Activity Violation Dimension	This table stores the information regarding life style activity violation.
Life Style Activity Violation Type Dimension	This table stores the Life Style Activity violation type. Sample values: FAA Equipment violation, license suspension, other.
Life Style Activity Aircraft Dimension	This table stores the aircraft model and its manufacturer details for the specified model and identifier.
Life Style Activity Aircraft Type Dimension	This table stores master list of type of the aircraft flown for the activity. Aircraft Type is used to specify the type of aircraft. If all the aircrafts are of the same Aircraft Type, then the type must be specified in the Aircraft Type on aviation experience. Sample vales: bomber, Attack, Experimental, fighter, etc.
Life Style Activity Air Sports Type Dimension	This table stores the different types of equipment used for the climbing activity. Sample values: Slings, Nuts, Tricams, SLCD=Spring-Loaded camming Devices, Bolts, Pitons, Skyhooks.
Life Style Activity Air Sports Balloon Type Dimension	This table stores the balloon type used for any air sports experience. Sample values: Air, Gas, Other.
Life Style Activity Aviation Highest Qualification Level Dimension	This table stores the highest qualification level achieved in aviation by the Party. Samples values: Commercial License, Instructors License, Instrument Rated, etc.
Life Style Activity Aviation Type Dimension	This table stores the Aviation Type details. It is used to specify the type of aviation when it applies to all the aircraft flown by the

Logical Dimension Table Name	Remarks
	client. If the Aviation Type varies by Aircraft, then the type must be specified in each individual Aircraft object instead. Sample values: Commercial, Military, other, private, unknown.
Life Style Activity Aviation Certificate Type Dimension	This table stores the aviation certificate type details. Sample values: single engine, sea, instrument rating, multi engine, land, other.
Life Style Activity Craft Certificate Type Dimension	This table stores the craft certificate type. Sample values: airplane, glider, lighter than air, other, powered lift, rotorcraft, unknown.
Life Style Activity Peril Type Dimension	This table stores the Life Style Activity Peril type details. Sample values: Dare-Devil Activities, Exhibitions, Exploration, Other, Unknown, Record Setting Activities, Rescue, Stunts.
Life Style Activity Climbing Type Dimension	This table stores the details of different types of climbing activity. Sample values: traditional, top-rope, sport, lead, bouldering, solo, aid.
Life Style Activity Medical Certificate Restriction Type Dimension	This table stores the details of medical certificate restriction type, defines the types of restrictions or constraints that may apply to a medical certificate. Sample values: other, restricted, special issue, unknown.
Life Style Activity Military Organization Type Dimension	This table stores the type of military organization the Party is part of. Sample values: air force, army, coast guard, marines, national guards, navy, others.
Life Style Activity Safety Equipment Used Type Dimension	This table stores the details of different types of safety equipment used during the activity. Sample values: Head and Neck Restraints, Onsite Medical Personnel, Other, Roll Cages, Unknown.
Life Style Activity Equipment Assembly Dimension	This table stores the details of equipment assembly. Sample values: Factory Assembled, home Assembled, Home build, other.
Life Style Activity Flying Purpose Dimension	This table stores the details of purpose for flying. Sample values: Advertising, Aerobatics, Agricultural, Combat, Flight surgeon, etc.
Life Style Activity Climbing Equipment Type Dimension	This table stores the details of different types of equipment used for the climbing activity. Sample values: Slings, Nuts, Tricams, SLCD=Spring-Loaded camming Devices, Bolts, Pitons, Skyhooks.
Life Style Activity Current Military Status Dimension	This table stores the details of current military status. Sample values: Active Duty, Active Reserve, Inactive, Retired, etc.
Life Style Activity Race Track Type Dimension	This table stores the details of different race tract type. Sample values: Closed Course Inland, Closed Course Ocean, Drag, Hill climb, etc.
Life Style Activity Racing Vehicle Type Dimension	This table stores the details of type of racing vehicles. Sample values: Automobile, Boat, Motorcycle, Other, Snowmobile, Unknown.
Life Style Activity Class Of Competition Type Dimension	This table stores the racing competitions class type details. Sub list values, where specified, correspond to the type of vehicle specified in the Vehicle Type property.

Logical Dimension Table Name	Remarks
	Sample Values: Formula 1, Cross country, Dirt track, Hill climbs, Ice racing, etc.
Life Style Activity Surface Type Dimension	This table stores the surface type details the race was primarily performed on. Sample values: Clay, dirt, ice, paved, sand, water, other, unknown.
Life Style Activity Travel Purpose Type Dimension	This table stores the travel type purpose details. Sample values: Business, miscellaneous, Education, Employment, Family Visit, Volunteer Work, etc.
Life Style Activity Travel Transportation Mode Dimension	This table stores the dominant mode of travel details while in the foreign country. Sample values: automobile, common carrier air, ship, etc.
Life Style Activity Underwater Diving Type Dimension	This table stores the diving type details. Samples values: Scuba, skin diving, snorkeling, hookah, hard hat.
Life Style Activity Underwater Dive Training Type Dimension	This table stores the master list of professionally recognized diving certifications and training courses.
Life Style Activity Underwater Dive Location Type Dimension	This table stores the details of dive location type, the environments, which apply to this Underwater Diving experience. Sample values: Coastal Waters, Deep Sea, Lakes, etc.
Life Style Activity Underwater Dive Purpose Type Dimension	This table stores the underwater dive purpose details. Sample values: Abalone, Cave, Cavern diving, Exploration, Free Diving, etc.
Life Style Activity Underwater Diving Equipment Type Dimension	This table stores the diving equipment type details. Sample values: closed circuit, wet suit, etc.
Season Type Dimension	This table stores the season type details. Sample values: fall, other, unknown, spring, summer, winter.
Terrain Type Dimension	This table stores the aviation certificate type details. Sample values: single engine, sea, instrument rating, multi engine, land, other.

The mapping details for the Party Lifestyle Activity Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
501	STG_LSA_TYPE_MASTER	Stage Life Style Activity Type Master	DIM_LIFESTYLE_ACTIVITY_TYPE	Life Style Activity Type Dimension
527	STG_LSA_AIRSPORTS_EXP_DETL_V	Stage Life Style Activity Airsports Experience Details View	DIM_LIFESTYLE_ACTIVITY	Life Style Activity Dimension
528	STG_LSA_AVIATION_EXP_DETAILS_V	Stage Life Style Activity Aviation Experience Details	DIM_LIFESTYLE_ACTIVITY	Life Style Activity Dimension
530	STG_LSA_COMPETITION_DTLS_V	Stage Life Style Activity Competition Details	DIM_LIFESTYLE_ACTIVITY	Life Style Activity Dimension

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
531	STG_LSA_MILITARY_EXP_DETAILS_V	Stage Life Style Activity Military Experience Details	DIM_LIFESTYLE_ACTIVITY	Life Style Activity Dimension
532	STG_LSA_RACING_EXPRNCE_DTLS_V	Stage Life Style Activity Racing Experience Details View	DIM_LIFESTYLE_ACTIVITY	Life Style Activity Dimension
533	STG_LSA_UW_DIVNG_EXP_DETAILS_V	Stage Life Style Activity Underwater Diving Experience Details View	DIM_LIFESTYLE_ACTIVITY	Life Style Activity Dimension
534	STG_LSA_FORGN_TRVEL_EXP_DTLS_V	Stage Life Style Activity Foreign Travel Experience Details View	DIM_LIFESTYLE_ACTIVITY	Life Style Activity Dimension
502	STG_LSA_VIOLATION_MASTER	Stage Life Style Activity Violation Master	DIM_LSA_VIOLATION	Life Style Activity Violation Dimension
503	STG_LSA_VIOLATION_TYPE_MASTER	Stage Life Style Activity Violation Type Master	DIM_LSA_VIOLATION_TYPE	Life Style Activity Violation Type Dimension
504	STG_LSA_AIRCRAFT_MASTER	Stage Life Style Activity Aircraft Master	DIM_LSA_AIRCRAFT	Life Style Activity Aircraft Dimension
505	STG_LSA_AIRCRAFT_TYPE_MASTER	Stage Life Style Activity Aircraft Type Master	DIM_LSA_AIRCRAFT_TYPE	Life Style Activity Aircraft Type Dimension
506	STG_LSA_AIRSPORTS_TYPE_MASTER	Stage Life Style Activity Air Sports Type Master	DIM_LSA_AIRSPORTS_TYPE	Life Style Activity Air Sports Type Dimension
507	STG_LSA_AS_BALLOON_TYPE_MASTER	Stage Life Style Activity Air Sports Balloon Type Master	DIM_LSA_AS_BALLOON_TYPE	Life Style Activity Air Sports Balloon Type Dimension
508	STG_LSA_AVIATION_HQL_MASTER	Stage Life Style Activity Aviation Highest Qualification Level Master	DIM_LSA_AVIATION_HQL	Life Style Activity Aviation Highest Qualification Level Dimension
509	STG_LSA_AVIATION_TYPE_MASTER	Stage Life Style Activity Aviation Type Master	DIM_LSA_AVIATION_TYPE	Life Style Activity Aviation Type Dimension
510	STG_LSA_AVITN_CRTF_TYPE_MASTER	Stage Life Style Activity Aviation Certificate Type Master	DIM_LSA_AVIATN_CERTF_TYPE	Life Style Activity Aviation Certificate Type Dimension
511	STG_LSA_CERT_CRAFT_TYPE_MASTER	Stage Life Style Activity Craft Certificate Type Master	DIM_LSA_CERT_CRAFT_TYPE	Life Style Activity Craft Certificate Type Dimension
512	STG_LSA_PERIL_TYPE_MASTER	Stage Life Style Activity Peril Type Master	DIM_LSA_PERIL_TYPE	Life Style Activity Peril Type Dimension
513	STG_LSA_CLIMBING_TYPE_MASTER	Stage Life Style Activity Climbing Type Master	DIM_LSA_CLIMBING_TYPE	Life Style Activity Climbing Type Dimension

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
515	STG_LSA_MED_CERTR_TYPE_MASTER	Stage Life Style Activity Medical Certificate Restriction Type Master	DIM_LSA_MED_CERTR_TYPE	Life Style Activity Medical Certificate Restriction Type Dimension
516	STG_LSA_MILTRY_ORG_TYPE_MASTER	Stage Life Style Activity Military Organization Type Master	DIM_LSA_MILITARY_ORG_TYPE	Life Style Activity Military Organization Type Dimension
517	STG_LSA_SAFETY_EQP_TYPE_MASTER	Stage Life Style Activity Safety Equipment Used Type Master	DIM_LSA_SAFETY_EQP_TYPE	Life Style Activity Safety Equipment Used Type Dimension
518	STG_LSA_EQUIPMT_ASSMBLY_MASTER	Stage Life Style Activity Equipment Assembly Master	DIM_LSA_EQUIPMENT_ASSEMBLY	Life Style Activity Equipment Assembly Dimension
519	STG_LSA_FLYING_PURPOSE_MASTER	Stage Life Style Activity Flying Purpose Master	DIM_LSA_FLYING_PURPOSE	Life Style Activity Flying Purpose Dimension
535	STG_LSA_CLMBNG_EQP_TYPE_MASTER	Stage Life Style Activity Climbing Equipment Type Master	DIM_LSA_CLMBNG_EQPMNT_TYPE	Life Style Activity Climbing Equipment Type Dimension
536	STG_LSA_CUR_MILITRY_STS_MASTER	Stage Life Style Activity Current Military Status Master	DIM_LSA_CUR_MILITRY_STS	Life Style Activity Current Military Status Dimension
537	STG_LSA_RACE_TRACK_TYPE_MASTER	Stage Life Style Activity Race Track Type Master	DIM_LSA_RACE_TRACK_TYPE	Life Style Activity Race Track Type Dimension
538	STG_LSA_RACE_VHCL_TYPE_MASTER	Stage Life Style Activity Racing Vehicle Type Master	DIM_LSA_RACE_VHCL_TYPE	Life Style Activity Racing Vehicle Type Dimension
539	STG_LSA_RACN_CLS_COMPTN_MASTER	Stage Life Style Activity Class Of Competition Type Master	DIM_LSA_RACN_CLS_OF_COMPTN	Life Style Activity Class Of Competition Type Dimension
540	STG_LSA_SURFACE_TYPE_MASTER	Stage Life Style Activity Surface Type Master	DIM_LSA_SURFACE_TYPE	Life Style Activity Surface Type Dimension
541	STG_LSA_TRAVEL_PUR_TYPE_MASTER	Stage Life Style Activity Travel Purpose Type Master	DIM_LSA_TRAVEL_PUR_TYPE	Life Style Activity Travel Purpose Type Dimension
542	STG_LSA_TRAVEL_TM_MASTER	Stage Life Style Activity Travel Transportation Mode Master	DIM_LSA_TRAVEL_TM	Life Style Activity Travel Transportation Mode Dimension
543	STG_LSA_UW_DIVING_TYPE_MASTER	Stage Life Style Activity Underwater Diving Type Master	DIM_LSA_UW_DIVING_TYPE	Life Style Activity Underwater Diving Type Dimension
544	STG_LSA_UW_DV_TRNG_TYPE_MASTER	Stage Life Style Activity Underwater Dive Training Type Master	DIM_LSA_UW_DV_TRNG_TYPE	Life Style Activity Underwater Dive Training Type Dimension
545	STG_LSA_UW_DVE_LOC_TYPE_MASTER	Stage Life Style Activity Underwater Dive Location Type Master	DIM_LSA_UW_DVE_LOC_TYPE	Life Style Activity Underwater Dive Location Type Dimension

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
546	STG_LSA_UWDIVE_PUR_TYPE_MASTER	Stage Life Style Activity Underwater Dive Purpose Type Master	DIM_LSA_UWDIVE_PUR_TYPE	Life Style Activity Underwater Dive Purpose Type Dimension
547	STG_LSA_UWDVNG_EQP_TYPE_MASTER	Stage Life Style Activity Underwater Diving Equipment Type Master	DIM_LSA_UWDVNG_EQP_TYPE	Life Style Activity Underwater Diving Equipment Type Dimension
515	STG_LSA_MED_CERTR_TYPE_MASTER	Stage Life Style Activity Medical Certificate Restriction Type Master	DIM_LSA_MED_CERTR_TYPE	Life Style Activity Medical Certificate Restriction Type Dimension
548	STG_SEASON_TYPE_MASTER	Stage Season Type Master	DIM_SEASON_TYPE	Season Type Dimension
549	STG_TERRIAN_TYPE_MASTER	Stage Terrain Type Master	DIM_TERRIAN_TYPE	Terrain Type Dimension

15.3.5.2 About Party Lifestyle Activity T2T (Result Table)

Party Lifestyle Activity T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCT_LIFESTYLE_ACTIVITY_DETAILS	This T2T stores the lifestyle activities details of a Party. This section captures details surrounding the activities a Party engages in, that may be considered risky by the Insurance Companies.
T2T_FCT_LSA_AIRCRAFT_DETAILS	This T2T stores the aircraft model details, which is used to specify the details of the types of aircraft that apply to this Aviation experience.
T2T_FCT_LSA_AIRSPORTS_EXP_DETAILS	This T2T stores the air sports experience details of a Party. It includes ballooning, hang gliding, parachuting and ultra lite experience.
T2T_FCT_LSA_COMPETITION_DTLS	This T2T stores the details of various life style activities competitions.
T2T_FCT_LSA_FORGN_TRAVEL_EXP_DTLS	This T2T stores foreign travel or foreign residence experience details of a Party.
T2T_FCT_LSA_MILITARY_EXP_DETAILS	This T2T stores the military experience details of a Party.
T2T_FCT_LSA_RACING_EXPERIENCE_DTLS	This T2T stores the racing experience details of a Party.
T2T_FCT_LSA_UW_DIVING_EXP_DETAILS	This T2T stores the underwater-diving experience details of a Party.
T2T_FCT_LSA_AVIATION_EXP_DETAILS	This T2T stores aviation experience details of a Party that may be flying or operating of aircraft.
T2T_FCT_LSA_CLIMIBING_EXP_DTLS	This T2T stores the climbing experience details of a Party.

The mapping details for the Party Lifestyle Activity T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_LIFESTYLE_ACTIVITY_DETAILS	Stage Lifestyle Activity Details	FCT_LIFESTYLE_ACTIVITY_DETAILS	Fact Lifestyle Activity Details	T2T_FCT_LIFESTYLE_ACTIVITY_DETAILS
STG_LSA_AIRCRAFT_DETAILS	Stage Life Style Activity Aircraft Details	FCT_LSA_AIRCRAFT_DETAILS	Fact Life Style Activity Aircraft Details	T2T_FCT_LSA_AIRCRAFT_DETAILS
STG_LSA_AIRSPORTS_EXP_DETAILS	Stage Life Style Activity Airsports Experience Details	FCT_LSA_AIRSPORTS_EXP_DETAILS	Fact Life Style Activity Airsports Experience Details	T2T_FCT_LSA_AIRSPORTS_EXP_DETAILS
STG_LSA_COMPETITION_DTLS	Stage Life Style Activity Competition Details	FCT_LSA_COMPETITION_DTLS	Fact Life Style Activity Competition Details	T2T_FCT_LSA_COMPETITION_DTLS
STG_LSA_FORGN_TRAVEL_EXP_DTLS	Stage Life Style Activity Foreign Travel Experience Details	FCT_LSA_FORGN_TRAVEL_EXP_DTLS	Fact Life Style Activity Foreign Travel Experience Details	T2T_FCT_LSA_FORGN_TRAVEL_EXP_DTLS
STG_LSA_MILITARY_EXP_DETAILS	Stage Life Style Activity Military Experience Details	FCT_LSA_MILITARY_EXP_DETAILS	Fact Life Style Activity Military Experience Details	T2T_FCT_LSA_MILITARY_EXP_DETAILS
STG_LSA_RACING_EXPERIENCE_DTLS	Stage Life Style Activity Racing Experience Details	FCT_LSA_RACING_EXPERIENCE_DTLS	Fact Life Style Activity Racing Experience Details	T2T_FCT_LSA_RACING_EXPERIENCE_DTLS
STG_LSA_UW_DIVING_EXP_DETAILS	Stage Life Style Activity Underwater Diving Experience Details	FCT_LSA_UW_DIVING_EXP_DETAILS	Fact Life Style Activity Underwater Diving Experience Details	T2T_FCT_LSA_UW_DIVING_EXP_DETAILS
STG_LSA_AVIATION_EXP_DETAILS	Stage Life Style Activity Aviation Experience Details	FCT_LSA_AVIATION_EXP_DETAILS	Fact Life Style Activity Aviation Experience Details	T2T_FCT_LSA_AVIATION_EXP_DETAILS
STG_LSA_CLIMBING_EXP_DTLS	Stage Life Style Activity Climbing Experience Details	FCT_LSA_CLIMBING_EXP_DTLS	Fact Life Style Activity Climbing Experience Details	T2T_FCT_LSA_CLIMIBING_EXP_DTLS

15.4 Deploying Insurance Underwriting Tables on Hive

All RDBMS related Insurance Underwriting Result tables can also be deployed on Hive (Stage and Results on Hive).

> **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.

15.5 Populating Insurance Underwriting Dimension Tables

Follow this SCD process to populate data into any Insurance Underwriting Dimension table:

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

1. To populate data into any Insurance Underwriting Dimension table, execute the SCD batch. For a detailed procedure, see Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of an Insurance Underwriting 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.

15.6 Populating Insurance Underwriting T2T Result Tables

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

> **NOTE** You can also follow this T2T process to populate data into any Hive related Insurance Underwriting T2T Result table.

1. To populate data into any Insurance Underwriting T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Insurance Underwriting Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

16 Insurance Contracts Tables

This chapter 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 and step-by step instructions to use this section.

This chapter includes the following topics:

Accounting standards defines 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.

In OIDF Insurance contract comprises of following important components:

- Contracts tables
- Coverage tables
- Supplemental Data
- Policy Funds tables
- Policy Loans tables

Party meaning beneficiary, producer and agent, underwriter etc., are expected to be created and available for reference in party set of tables as per information requirements for a related role. For details, see Party section.

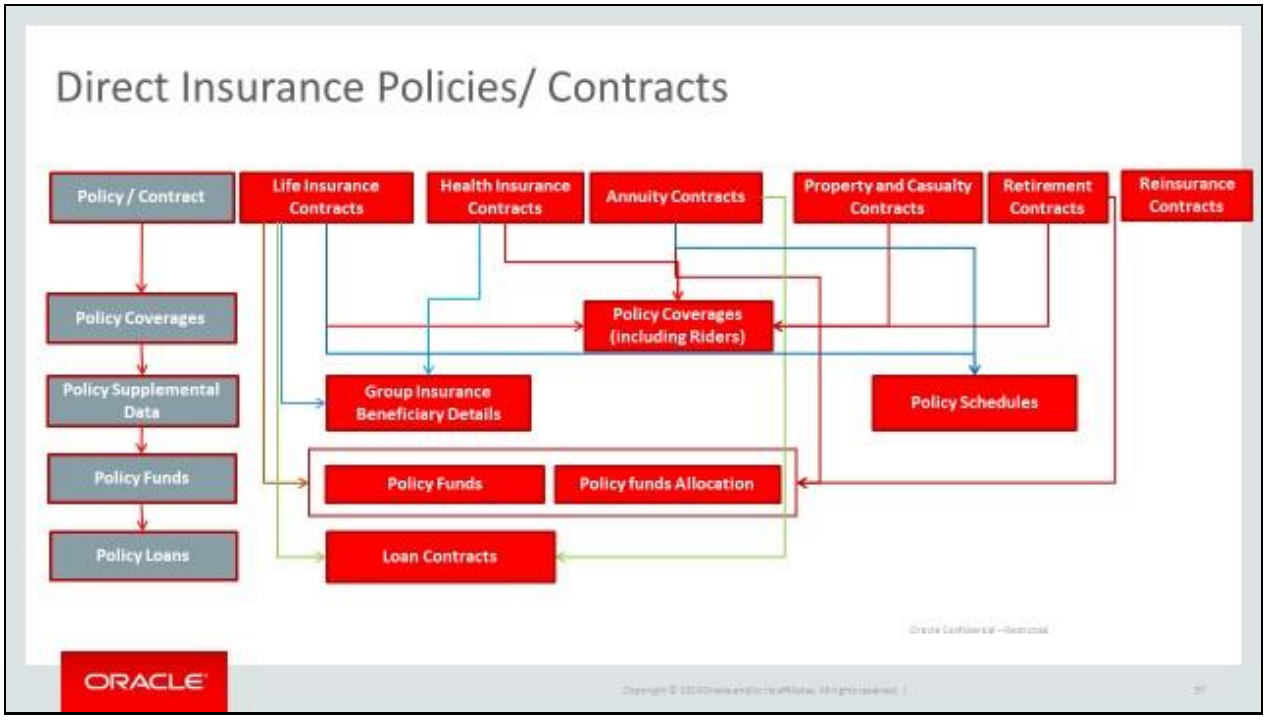
16.1 Contracts Tables

Contracts as defined earlier is an record in one of the following stage tables called as a product processors comprising of set of attributes relevant to a particular insurance contract. They are outputs sourced from policy administration systems.

This is the list of entities covered under direct and Indirect insurance or Reinsurance contracts to elaborate OIDF design for insurance contracts:

Source Logical Table Name	Source Table Name
Stage Annuity Contracts	STG_ANNUITY_CONTRACTS
Stage Health Insurance Contracts	STG_HEALTH_INS_CONTRACTS
Stage Life Insurance Contracts	STG_LIFE_INS_CONTRACTS
Stage Property and Casualty Contracts	STG_PROP_CASUALTY_CONTRACTS
Stage Retirement Accounts	STG_RETIREMENT_ACCOUNTS
Stage Reinsurance Contracts Held	STG_REINSURANCE_CONTRACTS_HELD
Stage Reinsurance Contracts Issued	STG_REINSURANCE_CNTRCTS_ISSUED

Following representative diagram elaborates the OIDF design for Direct Insurance Contracts. It conveys that base contract is loaded in contract table while coverages or riders are loaded to coverages tables. One policy can have more than one coverage associated with the same. Depending on the applicability, a policy can consist of a fund associated with it and/or a loan associated with it.



Contract attributes covers attributes like contract identifier, product identifier, sum insured, origination date, purchase date, maturity date, policy status etc. A group insurance contract should also be stored in of the above relevant table.

16.1.1 Common Policy Summary Tables

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

16.1.1.1 About Common Policy Summary Dimension Tables

Common Policy Summary Dimension table name and its remarks is given here. Common Policy Summary is also a part of Policy Dimension table (DIM_POLICY). For description about Policy Dimension, see the chapter Policy Dimension Table.

Logical Dimension Table Name	Remarks
Policy Dimension	This table stores the list of all policies issued by the insurer.

The mapping details for the Common Policy Summary Dimension tables are given here. Common Policy Summary is also a part of Policy Dimension table (DIM_POLICY). For mapping information about Policy Dimension, see the chapter Policy Dimension Table.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
616	VW_STG_LIFE_INS_CONTRACTS	Stage Life Insurance Contracts View	DIM_ASSUMED_POLICY	Assumed Policy Dimension
617	VW_STG_HEALTH_INS_CONTRACTS	Stage Health Insurance Contracts View	DIM_ASSUMED_POLICY	Assumed Policy Dimension
618	VW_STG_ANNUITY_CONTRACTS	Stage Annuity Contracts View	DIM_ASSUMED_POLICY	Assumed Policy Dimension
619	VW_STG_RETIREMENT_ACCOUNTS	Stage Retirement Accounts View	DIM_ASSUMED_POLICY	Assumed Policy Dimension
620	VW_STG_PROP_CASUALTY_CONTRACTS	Stage Property Casualty Contracts View	DIM_ASSUMED_POLICY	Assumed Policy Dimension

16.1.1.2 About Common Policy Summary T2T (Result Table)

Common Policy Summary T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCPS_STG_ANNUITY_CONTRACTS	This T2T captures 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.
T2T_FCPS_STG_HEALTH_INS_CONTRACTS	This T2T stores details of policy related to health insurance contracts.
T2T_FCPS_STG_LIFE_INS_CONTRACTS	This T2T stores details of policy related to life insurance contracts.

T2T Name	T2T Description
T2T_FCPS_STG_PROP_CASUALTY_CONTRACTS	This T2T stores details of policy related to property and casualty contracts.
T2T_FCPS_STG_RETIREMENT_ACCOUNTS	This T2T stores details of policy related to retirement accounts.

Common Policy Summary T2Ts and their description are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_ANNUITY_CONTRACTS	Stage Annuity Contracts	FCT_COMMON_POLICY_SUMMARY	Fact Common Policy Summary	T2T_FCPS_STG_ANNUITY_CONTRACTS
STG_HEALTH_INS_CONTRACTS	Stage Health Insurance Contracts	FCT_COMMON_POLICY_SUMMARY	Fact Common Policy Summary	T2T_FCPS_STG_HEALTH_INS_CONTRACTS
STG_LIFE_INS_CONTRACTS	Stage Life Insurance Contracts	FCT_COMMON_POLICY_SUMMARY	Fact Common Policy Summary	T2T_FCPS_STG_LIFE_INS_CONTRACTS
STG_PROP_CASUALTY_CONTRACTS	Stage Property Casualty Contracts	FCT_COMMON_POLICY_SUMMARY	Fact Common Policy Summary	T2T_FCPS_STG_PROP_CASUALTY_CONTRACTS
STG_RETIREMENT_ACCOUNTS	Stage Retirement Accounts	FCT_COMMON_POLICY_SUMMARY	Fact Common Policy Summary	T2T_FCPS_STG_RETIREMENT_ACCOUNTS

16.1.1.3 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.

16.1.1.3.1 Reinsurance Data Model

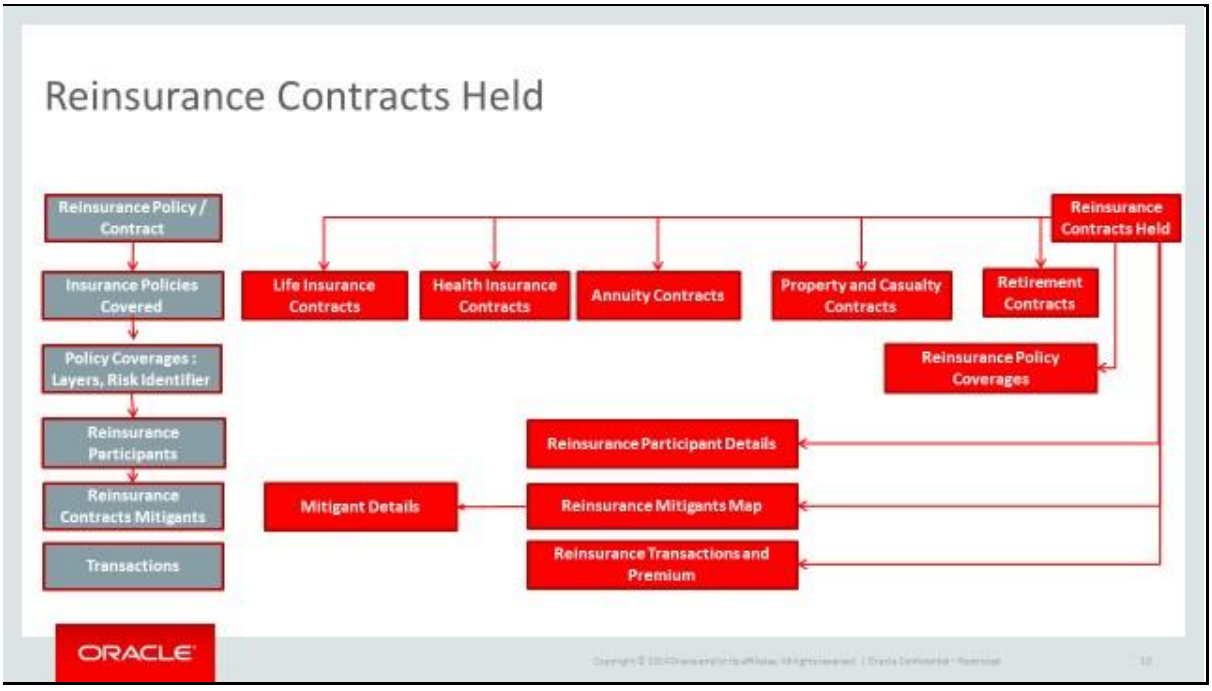
This is the list of entities covered under reinsurance contracts and representative diagram to elaborate OIDF design for reinsurance contracts:

Reinsurance Held				
Contract Type	**Stage Entity**	**Result Entity**	**Key Contract Dimension**	**Differentiator**
Direct Written Policies	Stage Life Insurance Contracts	Fact Common Policy Summary	Policy Dimension	Direct Or Assumed Insurance Indicator = Direct
	Stage Property and Casulaty Contracts			
	Stage Health insurance Contracts			
	Stage Annuity Contracts			
	Stage Retirement Contracts			
	Stage Policy Coverages	Fact Policy Coverages	Coverage Dimension	
Reinsurance Contract	Stage Reinsurance Contract Held	Fact Reinsurance Contract Held	Policy Dimension	
	Stage Reinsurance Held Mitigant Map	Fact Reinsurance Held Mitigant Map	Policy and Mitigant Dimension	
Participation Details	Stage Insurance Participation Details	Fact Insurance Participation Details		
Cash flow - Direct Policies	Stage Insurance Policy Cash Flow	Fact Insurance Policy Cash Flow		
Cash flow - RI Held	Stage Insurance Policy Cash Flow	Fact Insurance Policy Cash Flow		
Reinsurance Issued				
	Stage Entity	**Result Entity**	**Key Contract Dimension**	**Differentiator**
Assumed Policies	Stage Life Insurance Contracts	Fact Assumed Policy Summary	Assumed Policy Dimension	Direct Or Assumed Insurance Indicator = Assumed
	Stage Property and Casulaty Contracts			
	Stage Health insurance Contracts			
	Stage Annuity Contracts			
	Stage Retirement Contracts			
	Stage Policy Coverages	Fact Policy Coverages	Coverage Dimension	
Reinsurance Contract	Stage Reinsurance Contract Issued	Fact Reinsurance Contract Issued	Policy Dimension	
	Stage Reinsurance Issued Placed Collateral Map	Fact Reinsurance Issued Placed Collateral Map	Policy and Placed Collateral Dimension	
Participation Details	Stage Insurance Participation Details	Fact Insurance Participation Details		
Cash flow - Assumed Policies*	Stage Assumed Policy Cash Flow	Fact Assumed Policy Cash Flow		
Cash flow - RI Issued	Stage Insurance Policy Cash Flow	Fact Insurance Policy Cash Flow		

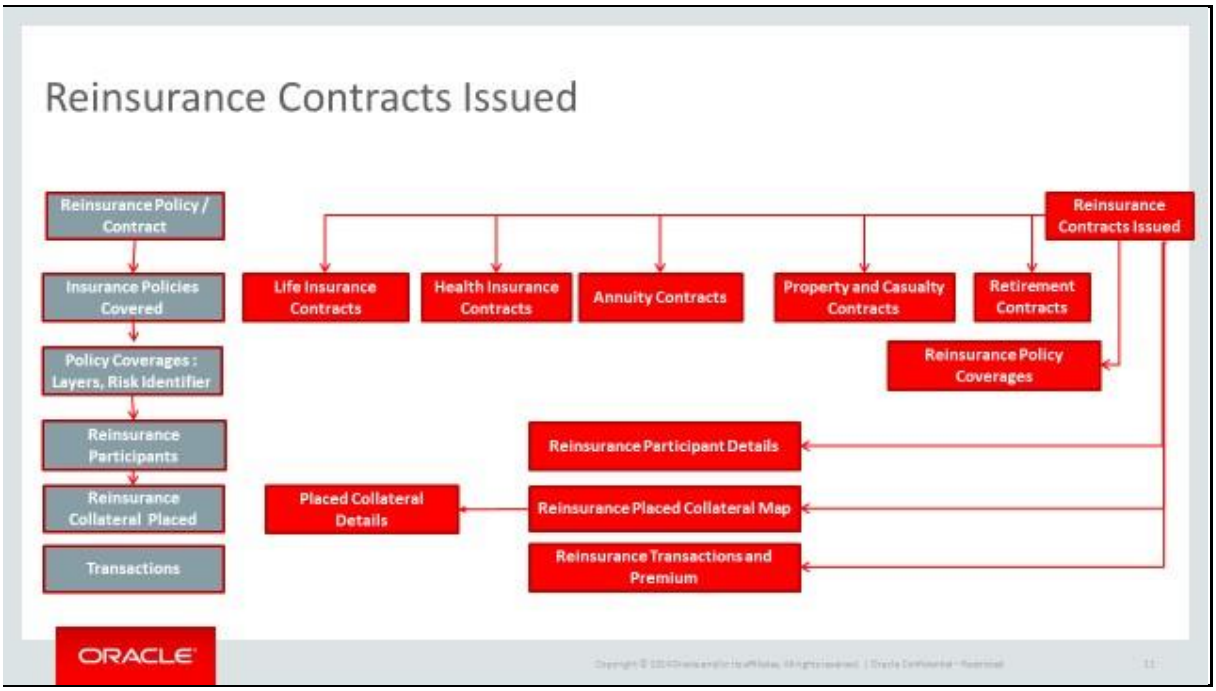
Reinsurance Contracts consists of two types of contracts:

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

This is the representative diagram to elaborate OIDF design for Reinsurance Contracts Held



This is the representative diagram to elaborate OIDF design for Reinsurance Contracts Issued



16.1.1.3.2 About Reinsurance Contracts Dimension Tables

Reinsurance Contracts Dimension table names and their remarks are given here.

Logical Dimension Table Name	Remarks
Insurance Reserve Function Method Dimension	This table stores the method used to calculate reserves and defines the mortality function used to calculate reserves and net premium for valuation. Sample values: OTHER, UNKNOWN, RESERVFUNCFCL, RESERVFUNCFUL.
Consequence Of Loss Dimension	This table stores the code that identifies consequences of losses. Sample values: Acid burns, Amputation, Blindness, etc.
Tax Reimbursement Calculation Method Dimension	This table stores the Tax Reimbursement Calculation Method Code details and the information about the method for calculating the portion of the Premium Taxes reimbursed to the direct insurer. Sample values: OTHER, UNKNOWN, REINPREMTXRM12.
Insurance Reserve Calculation Basis Dimension	This table stores the Insurance Reserve Calculation Basis Code details, in which, basis is used in calculating the standard GAAP reserve. Sample value: OTHER.
Reinsurance Contract Type Dimension	This table stores the Reinsurance Contract Type details. Sample values: DIP is Direct Insurance Policy , RC is Reinsurance Contract , FC is Facultative Contract , etc.
Insurance Sub Standard Policy Reserve Method Dimension	This table stores the Insurance Sub Standard Policy Reserve Method Code details, a code list from Life Reinsurance Activity Report LREACT UGP, UPP, etc. Describes the actuarial

Logical Dimension Table Name	Remarks
	methodology used to calculate the substandard policy reserves. For example, Net Level.
Reinsurance Accounting Basis Dimension	This table stores the Reinsurance accounting basis details, which specifies the basis of accounting for the contract. Sample values: Accident Year, Accounting Year, Calendar Year, Underwriting Year, etc.
Insurance Reserve Calculation Method Dimension	This table stores the information about the reserves calculation method on this coverage and is used by valuation programs to find the correct reserves for this coverage. Sample values: OTHER, UNKNOWN, RESVMETHYT is 1 or 5 Year Preliminary Term.
Insurance Sub Standard Reserve Calculation Basis Dimension	This table stores the Insurance Sub Standard Reserve Calculation Basis Code details. The substandard GAAP reserve factor is applied to the amount indicated to calculate the substandard GAAP reserve amount. Sample values: OTHER, UNKNOWN, etc.
Contract Limit Retention Basis Dimension	This table stores the code identification details of the basis of the contract limits and retentions applicable to the claim that is reported. Sample values: Per accident, Per event, Per occurrence, Per location, etc.
Insurance Reserve Valuation Type Dimension	This table stores Reserve Valuation Type details. Sample values: OTHER, UNKNOWN, RESVALTPCAM is Canadian Modified CANMOD, etc.
Insurance Retention Level Type Dimension	This table stores the Insurance Retention Level Type details and the level of retention on the coverage. Sample values: OTHER, UNKNOWN, RETNLEVRNF is Full proportional, RETNLEVRFR is Full retention, etc.
Reinsurance Business Origin Dimension	This table stores the Reinsurance Business Origin Code details, in which, codes specify whether the contract was directly or indirectly ceded. Sample values: Cession, Retrocession, Coinsurance, Direct insurance.

The mapping details for the Reinsurance Contracts Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
621	STG_INS_RES_FUNC_MTHD_MASTER	Stage Insurance Reserve Function Method Master	DIM_INS_RES_FUNC_MTHD	Insurance Reserve Function Method Dimension
622	STG_CONSEQUENCE_OF_LOSS_MASTER	Stage Consequence Of Loss Master	DIM_CONSEQUENCE_OF_LOSS	Consequence Of Loss Dimension
623	STG_TAX_REIMB_CAL_MTHD_MASTER	Stage Tax Reimbursement Calculation Method Master	DIM_TAX_REIMB_CAL_MTHD	Tax Reimbursement Calculation Method Dimension

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
624	STG_INS_RES_CAL_BASIS_MASTER	Stage Insurance Reserve Calculation Basis Master	DIM_INS_RES_CAL_BASIS	Insurance Reserve Calculation Basis Dimension
625	STG_RI_CONTRACT_TYPE_MASTER	Stage Reinsurance Contract Type Master	DIM_RI_CONTRACT_TYPE	Reinsurance Contract Type Dimension
626	STG_INS_SS_POL_RES_MTHD_MASTER	Stage Insurance Sub Standard Policy Reserve Method Master	DIM_INS_SS_POL_RES_MTHD	Insurance Sub Standard Policy Reserve Method Dimension
627	STG_RI_ACCOUNTING_BASIS_MASTER	Stage Reinsurance Accounting Basis Master	DIM_RI_ACCOUNTING_BASIS	Reinsurance Accounting Basis Dimension
628	STG_INS_RES_CAL_MTHD_MASTER	Stage Insurance Reserve Calculation Method Master	DIM_INS_RES_CAL_MTHD	Insurance Reserve Calculation Method Dimension
629	STG_INS_SS_RES_BASIS_MASTER	Stage Insurance Sub Standard Reserve Calculation Basis Master	DIM_INS_SS_RES_BASIS	Insurance Sub Standard Reserve Calculation Basis Dimension
630	STG_CL_RETENTN_BASIS_MASTER	Stage Contract Limit Retention Basis Master	DIM_CL_RETENTN_BASIS	Contract Limit Retention Basis Dimension
631	STG_INS_RES_VALN_TYPE_MASTER	Stage Insurance Reserve Valuation Type Master	DIM_INS_RES_VALN_TYPE	Insurance Reserve Valuation Type Dimension
632	STG_INS_RETN_LEVEL_TYPE_MASTER	Stage Insurance Retention Level Type Master	DIM_INS_RETN_LEVEL_TYPE	Insurance Retention Level Type Dimension
633	STG_RI_BUSINESS_ORIGIN_MASTER	Stage Reinsurance Business Origin Master	DIM_RI_BUSINESS_ORIGIN	Reinsurance Business Origin Dimension

16.1.1.3.3 About Reinsurance Contracts T2Ts (Result Tables)

Reinsurance Contracts T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCT_REINSURANCE_CONTRTS_HLD	This T2T stores all the attributes of a reinsurance contract held by the entity.
T2T_FCT_REINSURANCE_CONTRTS_ISS	This T2T stores all the attributes of a reinsurance contract issued by the entity.
T2T_FCT_RI_ISSUED_PLCD_COLL_MAP	This T2T stores the mapping between Placed Collateral and Reinsurance Contract. Reinsurers provide security to Insurance Company to make sure that they are adequately covered.

The mapping details for the Reinsurance Contract T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Definition Name
STG_REINSURANCE_CONTRACTS_HELD	Stage Reinsurance Contracts Held	FCT_REINSURANCE_CONTRACTS_HELD	Fact Reinsurance Contracts Held	T2T_FCT_REINSURANCE_CONTRTS_HLD
STG_REINSURANCE_CNTRCTS_ISSUED	Stage Reinsurance Contracts Issued	FCT_REINSURANCE_CNTRCTS_ISSUED	Fact Reinsurance Contracts Issued	T2T_FCT_REINSURANCE_CONTRTS_ISS
STG_RI_ISSUED_PLACED_COLL_MAP	Stage Reinsurance Issued Placed Collateral Map	FCT_RI_ISSUED_PLACED_COLL_MAP	Fact Reinsurance Issued Placed Collateral Map	T2T_FCT_RI_ISSUED_PLCD_COLL_MAP

16.2 Coverage Tables

Coverage is an individual contractual obligation incurred by contract provider. Aggregation of all such obligations within the terms and conditions defined should be equal to contractual liability of a contract. Let us take the following of the two examples

- A Life Insurance contract offering a normal death benefit, accidental death benefit and critical illness benefits to the insured.
- An Auto Insurance policy covering damages to vehicle, engine coverage, vehicle replacement, zero depreciation cover, roadside assistance and towing.

In the above example a single row is created in life insurance contract entity and property and casualty insurance contract entity while policy coverage entity will store three rows for life insurance contract while 5 rows for auto insurance contracts. In many insurance contracts, certain coverages are optional known as riders while some are bundled as a part of contract itself. All such coverages are to be stored in stage policy coverages. Different coverages are effective for different period and having different terms and conditions within main contract. Stage Policy Coverage entity allows end user to handle this flexibility. In case of group insurance policy coverages refers to the single benefit or rider bundled as a part of product.

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

16.2.1 About Policy Coverage T2T (Result Table)

Policy Coverage T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCT_POLICY_COVERAGES	This T2T stores policy coverages details. A policy can have multiple coverages.
T2T_FCT_POLICY_COVERG_BASIS_DTL	This T2T stores the coverage basis value for a given policy.

The mapping details for the Policy Coverage T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_POLICY_COVERAGES	Stage Policy Coverages	FCT_POLICY_COVERAGES	Fact Policy Coverages	T2T_FCT_POLICY_COVERAGES
STG_POLICY_COVERG_BASIS_DTL	Stage Policy Coverage Basis Detail	FCT_POLICY_COVERG_BASIS_DTL	Fact Policy Coverage Basis Detail	T2T_FCT_POLICY_COVERG_BASIS_DTL

16.3 Supplemental Data

This 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.

16.3.1 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. A group insurance contract is one which is agreed between insurance company and entity which sponsors the policy. This contract is should be stored in one of the seven product processors above. However, in general stage life insurance and stage health insurance contracts will be referred in most of the use cases of group insurance.

Details of members of group insurance are stored in stage group insurance beneficiary details. This entity holds details of each member like individual limits, individual usage, and identifiers like card numbers, annual and per condition deductibles. In cases where such details are not available, aggregated numbers should be loaded in main contract entities itself.

16.3.1.1 About Group Beneficiary Dimension Table

Reinsurance Contracts Dimension table names and their remarks are given here.

Logical Dimension Table Name	Remarks
Beneficiary Dimension	This table stores beneficiary details.

The mapping details for the Reinsurance Contracts Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
672	STG_PARTY_MASTER	Stage Party Master	DIM_BENEFICIARY	Beneficiary Dimension

16.3.1.2 About Group Policy Beneficiary T2T (Result Table)

Group Policy Beneficiary T2T and its description is given here.

T2T Name	T2T Description
T2T_FCT_GRPINS_POL_BENFCIARY_DTLS	This T2T stores group insurance policy beneficiary details.

The mapping details for the Group Policy Beneficiary T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_GRPINS_POL_BENFCIARY_DTLS	Stage Groupings Policy Beneficiary Details	FCT_GRPINS_POL_BENFCIARY_DTLS	Fact Groupings Policy Beneficiary Details	T2T_FCT_GRPINS_POL_BENFCIARY_DTLS

16.3.2 Policy Schedules Table

Policy Schedule refers to the agreed non-linear premium and benefit schedules associated with the insurance contracts. Step Up annuity contracts, Term life Insurance where benefits are increasing or decreasing with remaining term of contract are few examples of such contracts. Agreed schedules can be stored in this entity.

16.3.2.1 About Policy Schedules T2T (Result Table)

Policy Schedules T2T and its description is given here.

T2T Name	T2T Description
T2T_FCT_POLICY_SCHEDULES	This T2T stores the details of policy schedules. Policy Schedules refer to the premium and sum insured for each policy year until policy maturity. This is used in case of increasing or decreasing the benefit or Premium payments associated with a policy.

The mapping details for the Policy Schedules T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_POLICY_SCHEDULES	Stage Policy Schedules	FCT_POLICY_SCHEDULES	Fact Policy Schedules	T2T_FCT_POLICY_SCHEDULES

16.4 Policy and Fund Tables

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

There are two entities namely policy fund details and policy fund allocation. Policy funds provide funds wise investment snapshot in time while policy fund allocation provide changes over time. While there is technically no restriction as these entities are connected by policy code to the main contract table, they are most commonly applicable to annuity contracts, life insurance contracts and retirement accounts.

16.4.1 About Policy and Fund T2T (Result Table)

Policy and Fund T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCT_POLICY_FUNDS	This T2T stores the policy wise fund details. This covers units invested for each fund, total amount invested (current value of investment as of today), etc.
T2T_FCT_POLICY_FUND_ALLOCATION	This T2T stores the investment allocation for a given policy, fund and investment bucket combination. Investment amount is allocated to different funds and during different time period, which is stored in this entity. This covers premium amount invested in each bucket, running total premium, and cash surrender value for each bucket.

The mapping details for the Policy and Fund T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_POLICY_FUNDS	Stage Policy Funds	FCT_POLICY_FUNDS	Fact Policy Funds	T2T_FCT_POLICY_FUNDS
STG_POLICY_FUND_ALLOCATION	Stage Policy Fund Allocation	FCT_POLICY_FUND_ALLOCATION	Fact Policy Fund Allocation	T2T_FCT_POLICY_FUND_ALLOCATION

16.5 Policy Loans Tables

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

Ecosystem of loan entities cover Stage Loan contracts, Stage Mitigants, Stage Account Mitigant Map, Stage Account Write off Details, Stage Account Recovery details.

16.5.1 About Policy Loans T2T (Result Table)

Policy Loans T2T and its description is given here.

T2T Name	T2T Description
T2T_FLAS_STG_LOAN_CONTRACTS	This T2T stores the details of loans. This table includes mortgage, and vehicle loans.

The mapping details for the Policy Loans is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_LOAN_CONTRACTS	Stage Loan Contracts	FCT_LOAN_ACCOUNT_SUMMARY	Fact Loan Account Summary	T2T_FLAS_STG_LOAN_CONTRACTS

16.6 Deploying Insurance Contracts Tables on Hive

All RDBMS related Insurance Contracts Result tables can also be deployed on Hive (Stage and Results on Hive).

> **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.

16.7 Populating Insurance Contracts Dimension Tables

Follow this SCD process to populate data into any Insurance Contracts Dimension table:

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

1. To populate data into any Insurance Contracts Dimension table, execute the SCD batch. For a detailed procedure, see Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of an Insurance Contracts 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.

16.8 Populating Insurance Contracts T2T Result Tables

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

NOTE You can also follow this T2T process to populate data into any Hive related Insurance Contracts T2T Result table.

1. To populate data into any Insurance Contracts T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Insurance Contracts Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

17 Common Account Summary Tables

This chapter 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.

This chapter includes the following topics:

17.1 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 expectation of profit from sale or holding of such investment.
- Instrument: Tradable investments use instrument with ISIN/CUSIP/Other known identifiers. Instrument definition most commonly is published by issuer of instrument, which can be another public or private or government entity or stock exchange.

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 expectation of profit from sale or holding of such investment. Capital Instruments also make use of same relevant instrument set of tables.

Contracts

OIDF has different entities known as product processors to hold variety of instrument types. Given below is the list of trading and investment product processors.

Instruments

Instrument refers to an issue or derivative contract which is tradable in market. Over the counter contracts are considered as instruments. Thus Government bonds, municipal bonds, future contracts, option contracts

and excludes forwards, swap contracts etc. Multiple contracts in product processor entities mentioned above can have same instrument traded.

Following table explains need of creating an instrument before populating the contract for a given product processor.

Product Processor	A record in instrument contract master and dimension is required?
Stage Investments	Yes for Exchange Traded Contracts.
Stage MM Contracts	Yes for Exchange Traded Contracts.
Stage Futures	Yes
Stage Option Contracts	Yes
Stage Credit Derivatives	Yes for Exchange Traded Contracts.
Stage Fx Contracts	No
Stage Forwards	No
Stage Mutual Funds	Yes
Stage Commodities	Yes for Exchange Traded Contracts.

17.2 Trading and Investments Contract Tables

17.2.1 About Trading and Investments Contract T2Ts (Result Tables)

Trading and Investments Contract T2Ts and their description is given here.

T2T Name	T2T Description
T2T_STG_INVESTMENTS_CAS	This T2T stores the details of investment contract. For example: Government and Municipal Bonds, Covered/Secured Bonds, Corporate Bonds, etc.
T2T_STG_MM_CAS	This T2T stores the details of Money Market contract. For example: Certificates of Deposit (CDs), U.S. Treasury bills, commercial paper, municipal notes, federal funds, etc. The money market is used by a wide array of participants, from a company raising money by selling commercial paper into the market to an investor purchasing CDs as a safe place to park money for the short term.
T2T_STG_MUTUAL_FUNDS_CAS	This T2T stores the details of Mutual Funds contract.
T2T_STG_OPTIONS_CAS	This T2T stores records of call and put options that are bought (long) or sold (short) by the customer.
T2T_STG_FUTURES_CAS	This T2T stores the details of Futures contract.
T2T_STG_FORWARDS_CAS	This T2T stores the details of Forwards contract.
T2T_STG_CREDIT_DERIVATIVES_CAS	This T2T stores the details of Credit Derivatives contract.
T2T_STG_COMMODITIES_CAS	This T2T stores the details of Commodities contract.
T2T_STG_SWAPS_CONTRACTS_CAS	This T2T stores the details of swaps contract.

T2T Name	T2T Description
T2T_STG_FX_CONTRACTS_CAS	This T2T stores the details of Foreign Exchange contracts.
T2T_FCT_FUND_CIS_COMPOSITION	This T2T stores the composition details of the Investment Funds.

The mapping details for the Trading and Investments Contract T2Ts are given here.

SOURCE TABLE NAME	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_INVESTMENTS	Stage Investments	FCT_COMMON_ACCOUNT_SUMMARY	Fact Common Account Summary	T2T_STG_INVESTMENTS_CAS
STG_MM_CONTRACTS	Stage Money market Contracts	FCT_COMMON_ACCOUNT_SUMMARY	Fact Common Account Summary	T2T_STG_MM_CAS
STG_MUTUAL_FUNDS	Stage Mutual Funds	FCT_COMMON_ACCOUNT_SUMMARY	Fact Common Account Summary	T2T_STG_MUTUAL_FUNDS_CAS
STG_OPTION_CONTRACTS	Stage Option contracts	FCT_COMMON_ACCOUNT_SUMMARY	Fact Common Account Summary	T2T_STG_OPTIONS_CAS
STG_FUTURES	Stage Futures	FCT_COMMON_ACCOUNT_SUMMARY	Fact Common Account Summary	T2T_STG_FUTURES_CAS
STG_FORWARDS	Stage Forwards	FCT_COMMON_ACCOUNT_SUMMARY	Fact Common Account Summary	T2T_STG_FORWARDS_CAS
STG_CREDIT_DERIVATIVES	Stage Credit Derivatives	FCT_COMMON_ACCOUNT_SUMMARY	Fact Common Account Summary	T2T_STG_CREDIT_DERIVATIVES_CAS
STG_COMMODITIES	Stage Commodities	FCT_COMMON_ACCOUNT_SUMMARY	Fact Common Account Summary	T2T_STG_COMMODITIES_CAS
STG_SWAPS_CONTRACTS	Stage Swap Contracts	FCT_COMMON_ACCOUNT_SUMMARY	Fact Common Account Summary	T2T_STG_SWAPS_CONTRACTS_CAS
STG_FX_CONTRACTS	Stage FX (Foreign Exchange) Contracts	FCT_COMMON_ACCOUNT_SUMMARY	Fact Common Account Summary	T2T_STG_FX_CONTRACTS_CAS
STG_FUND_CIS_COMPOSITION	Stage Fund CIS Composition	FCT_FUND_CIS_COMPOSITION	Fact Fund CIS Composition	T2T_FCT_FUND_CIS_COMPOSITION

17.3 Instruments Tables

17.3.1 About Instruments Dimension Tables

Instruments Dimension table names and their description are given here.

Logical Dimension Table Name	Dimension Table Description
Instruments Contracts Dimension	This table stores data related to contracts and instruments in the market and their details such as Effective Date, Maturity Date, Face Value, Day Convention, Strike, etc.

Logical Dimension Table Name	Dimension Table Description
Geography Dimension	This table stores the distinct list of all geographical locations, where any of the transaction channels of the Insurance Company are located.

The mapping details for the Instruments Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
69	STG_INSTRUMENT_CONTRACT_MASTER	Stage Instrument Contract Master	DIM_INSTRUMENT_CONTRACT	Instruments Contracts Dimension
64	STG_GEOGRAPHY_MASTER	Stage Geography Master	DIM_GEOGRAPHY	Geography Dimension

17.3.2 About Instruments T2T (Result Table)

Instruments Contracts T2T and its description is given here.

T2T Name	T2T Description
T2T_FCT_INSTRUMENT_RATING_DETAILS	This T2T stores the rating of trading instruments including Mitigant that are tradable.

The mapping details for the Instruments T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_INSTRUMENT_RATING_DETAILS	Stage Instrument Rating Details	FCT_INSTRUMENT_RATING_DETAILS	Fact Instrument Rating Details	T2T_FCT_INSTRUMENT_RATING_DETAILS

17.4 Capital and Borrowings Tables

Capital and Borrowings pertains to the capital and borrowings part of balance sheet of a financial institution. It covers capital instruments such as Equity and Debt along with non-retail borrowings done by financial institutions.

17.4.1 About Capital and Borrowings T2Ts (Result Tables)

Capital and Borrowings T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCT_CAP_INSTR_POSITIONS	This T2T stores the regulatory position of capital instruments and details of treatment to capital instrument under I and III regulations.
T2T_FCT_CAP_INSTR_TXNS	This T2T stores the transactions on the capital instruments. The entity stores settled and unsettled transactions differentiated by mnemonic codes. It also stores the transactions that are settled with cash or with other means such as securities.
T2T_FCT_INSTR_PROPOSED_TXNS	This T2T stores the proposed set of instruments that will be transacted by the Insurance Company.

The mapping details for the Capital and Borrowings T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_CAP_INSTR_POSITIONS	Stage Capital Instrument Positions	FCT_CAP_INSTR_POSITIONS	Fact Capital Instrument Positions	T2T_FCT_CAP_INSTR_POSITIONS
STG_CAP_INSTR_TXNS	Stage Capital Instrument Transactions	FCT_CAP_INSTR_TXNS	Fact Capital Instrument Transactions	T2T_FCT_CAP_INSTR_TXNS
STG_INSTR_PROPOSED_TXNS	Stage Instrument Proposed Transactions	FCT_INSTR_PROPOSED_TXNS	Fact Instrument Proposed Transactions	T2T_FCT_INSTR_PROPOSED_TXNS

17.5 Other Common Account Summary Tables

Other Common Account Summary T2Ts and their description, and T2T mapping details are given here.

The target table is Fact Common Account Summary (FCT_COMMON_ACCOUNT_SUMMARY).

Source Table Name	Logical Stage Table Name	T2T Name	T2T Description
STG_BORROWING_COMMITMENT_TXNS	Stage Borrowing Commitment Transactions	T2T_STG_BORROWING_COMMITMENT_TXNS	This T2T stores all the transaction records of Borrowing Commitment Contracts
STG_BORROWING_COMMITMENTS	Stage Borrowing Commitments	T2T_STG_BORROWING_COMMITMENTS_CAS	This T2T stores the details of borrowing commitment contract.
STG_GUARANTEES	Stage Guarantees	T2T_STG_GUARANTEES_CAS	This T2T stores the guarantees issued by the Insurance Company to its customers.
STG_LOAN_CONTRACTS	Stage Loan Contracts	T2T_STG_LOANS_CAS	This T2T stores the details of loans lent by the Insurance Company to its customers.

Source Table Name	Logical Stage Table Name	T2T Name	T2T Description
STG_REPO_CONTRACTS	Stage Reporting Contracts	T2T_STG_REPO_CONTRACTS_CAS	This T2T stores the details of repurchase contracts.
STG_CUSTODIAL_ACCOUNTS	Stage Custodial Accounts	T2T_STG_CUSTODIAL_ACCOUNTS_CAS	This T2T stores the details of all accounts held in a custodial capacity.
STG_MANAGED_INV_ADV	Stage Managed Investment Advances	T2T_STG_MANAGED_INV_ADV_CAS	This T2T stores the details of all managed investment account and other services.

17.6 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 application subject area.

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

17.7 Deploying Common Account Summary Tables on Hive

All RDBMS related Common Account Summary Dimension and Result tables can also be deployed on Hive (Stage and Results on Hive).

> **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.

17.8 Populating Common Account Summary Dimension Tables

Follow this SCD process to populate data into any Common Account Summary Dimension table:

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

1. To populate data into any Common Account Summary Dimension table, execute the SCD batch. For a detailed procedure, see Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of an Common Account Summary 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.

17.9 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 batch, the RUNSkey is defaulted to -1.

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

> **NOTE** You can also follow this T2T process to populate data into any Hive related Common Account Summary T2T Result table.

1. To populate data into any Common Account Summary T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Common Account Summary Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

18 Loan Account Summary Tables

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

This chapter includes the following topics:

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

Loan Account in Insurance is used, when an insured takes a loan against his/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.

18.1 About Loan Account Summary T2T (Result Table)

Loan Account Summary T2T and its description is given here.

T2T Name	T2T Description
T2T_FLAS_STG_LOAN_CONTRACTS	This T2T stores the details of loan contracts. This table includes mortgage, and vehicle loans.

The mapping details for the Loan Account Summary T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_LOAN_CONTRACTS	Stage Loan Contracts	FCT_LOAN_ACCOUNT_SUMMARY	Fact Loan Account Summary	T2T_FLAS_STG_LOAN_CONTRACTS

18.2 Deploying Loan Account Summary Tables on Hive

All RDBMS related Loan Account Summary Result can also be deployed on Hive (Stage and Results on Hive).

> **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.

18.3 Populating Loan Account Summary T2T Result Tables

Follow this T2T process to populate data into any Loan Account Summary T2T Result tables:

NOTE You can also follow this T2T process to populate data into any Hive related Loan Account Summary T2T Result table.

1. To populate data into any Loan Account Summary T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Loan Account Summary Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

19 Fiduciary Services Investment Summary Population

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

This chapter includes the following 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, etc. 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.

19.1 About Fiduciary Services Investment Summary T2T (Result Table)

Fiduciary Services Investment Summary T2T and its description is given here.

T2T Name	T2T Description
T2T_FFSIS_STG_CUSTODIAL_ACCOUNTS	This T2T stores the details of all accounts held in a custodial capacity.
T2T_FFSIS_STG_INVESTMENTS	This T2T stores the details of investment contract.
T2T_FFSIS_STG_MANAGED_INV_ADV	This T2T stores details of all managed investment account and other services.
T2T_FFSIS_STG_MUTUAL_FUNDS	This T2T stores details of all mutual funds contracts.
T2T_FFSIS_STG_TRUSTS	This T2T stores the details of the trusts account.

The mapping details for the Fiduciary Services Investment Summary T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_CUSTODIAL_ACCOUNTS	Stage Custodial Accounts	FCT_FIDUCIARY_SERV_INVST_SUMM	Fact Fiduciary Services Investment Summary	T2T_FFSIS_STG_CUSTODIAL_ACCOUNTS
STG_INVESTMENTS	Stage Investments	FCT_FIDUCIARY_SERV_INVST_SUMM	Fact Fiduciary Services Investment Summary	T2T_FFSIS_STG_INVESTMENTS
STG_MANAGED_INV_ADV	Stage Managed Investment Advances	FCT_FIDUCIARY_SERV_INVST_SUMM	Fact Fiduciary Services Investment Summary	T2T_FFSIS_STG_MANAGED_INV_ADV

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_MUTUAL_FUNDS	Stage Mutual Funds	FCT_FIDUCIARY_SERV_INVST_SUMM	Fact Fiduciary Services Investment Summary	T2T_FFSIS_STG_MUTUAL_FUNDS
STG_TRUSTS	Stage Trusts	FCT_FIDUCIARY_SERV_INVST_SUMM	Fact Fiduciary Services Investment Summary	T2T_FFSIS_STG_TRUSTS

19.2 Deploying Fiduciary Services Investment Summary Result Tables on Hive

All RDBMS related Fiduciary Services Investment Summary Result tables can also be deployed on Hive (Stage and Results on Hive).

> **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.

19.3 Populating Fiduciary Services Investment Summary T2T Result Tables

Follow this T2T process to populate data into Fiduciary Services Investment Summary T2T Result tables:

> **NOTE** You can also follow this T2T process to populate data into any Hive related Fiduciary Services Investment Summary T2T Result table.

1. To populate data into any Fiduciary Services Investment Summary T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Fiduciary Services Investment Summary Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

20 Mitigants Results Tables

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

This chapter includes the following topics:

Mitigant means to reduce loss of life and property by reducing the impact of undesirable events.

The purpose of Mitigants Results table is to store the list of actions to be taken to reduce the impact of undesirable events.

20.1 About Mitigants Results T2Ts (Result Tables)

Mitigants Results T2Ts and its description is given here.

T2T Name	T2T Description
T2T_FCT_MITIGANTS	This T2T stores consolidated data of all Mitigants and their details.
T2T_FCT_ACCOUNT_MITIGANT_MAP	This T2T stores account to Mitigant mapping details. This T2T stores more than one Mitigant to be mapped to an account.
T2T_FCT_RI_HELD_MITIGANT_MAP	This T2T stores the mapping between Mitigant and Reinsurance contract. Reinsurers provide security to Insurance Company to ensure that they are adequately covered.

The mapping details for the Mitigants Results T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_MITIGANTS	Stage Mitigants	FCT_MITIGANTS	Fact Mitigants	T2T_FCT_MITIGANTS
STG_ACCOUNT_MITIGANT_MAP	Stage Account Mitigant Map	FCT_ACCOUNT_MITIGANT_MAP	Fact Account Mitigant Map	T2T_FCT_ACCOUNT_MITIGANT_MAP
STG_RI_HELD_MITIGANT_MAP	Stage Reinsurance Held Mitigant Map	FCT_RI_HELD_MITIGANT_MAP	Fact Reinsurance Held Mitigant Map	T2T_FCT_RI_HELD_MITIGANT_MAP

20.2 Deploying Mitigants Results Tables on Hive

All RDBMS related Mitigants Results can also be deployed on Hive (Stage and Results on Hive).

> **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 Mitigants Results T2T Result Tables

Follow this T2T process to populate data into any Mitigants Results T2T Result tables:

> **NOTE** You can also follow this T2T process to populate data into any Hive related Mitigants Results T2T Result table.

1. To populate data into any Mitigants Results T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Mitigants Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

21 Placed Collateral Tables

This chapter provides information about Placed Collateral Result tables in the Oracle Insurance Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

Financial Institutions place collateral in derivative transactions, reinsurance issued or similar use cases where 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 book and collateral placed are treated differently. For example, assume a use case where Insurance Company has purchased a lot of Government securities with market value of 10 Million USD. When they write a reinsurance contract, Insurance Company has sought a collateral comprising of a government securities worth 4 million USD. In this case, stage investments holds a data reflecting 6 million USD, whereas stage placed collateral holds data of 4 million USD.

21.1 About Placed Collateral Dimension Tables

Placed Collateral Dimension table names and their description are given here.

Logical Dimension Table Name	Remarks
Placed Collateral Dimension	This table stores the details of master collaterals that are placed by the Insurance entity with other Insurance entities in order to secure its borrowings.

The mapping details for the Placed Collateral Dimension table is given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
202	STG_PLACED_COLLATERAL_MASTER	Stage Placed Collateral Master	DIM_PLACED_COLLATERAL	Placed Collateral Dimension

21.2 About Placed Collateral T2Ts (Result Tables)

Placed Collateral T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCT_PLACED_COLLATERAL	This T2T stores the details of collateral which are placed against an account.
T2T_FCT_ACCT_PLACED_COLL_MAP	This T2T stores account to placed collateral mapping details. It is an intersection table to denote that a placed collateral can be used in multiple accounts and an account contains multiple collaterals.
T2T_FCT_RI_ISSUED_PLCD_COLL_MAP	This T2T stores the mapping between Placed Collateral and Reinsurance Contract. Reinsurers provide security to Insurance Company to make sure that they are adequately covered.

The mapping details for the Placed Collateral T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_PLACED_COLLATERAL	Stage Placed Collateral	FCT_PLACED_COLLATERAL	Fact Placed Collateral	T2T_FCT_PLACED_COLLATERAL
STG_ACCT_PLACED_COLL_MAP	Stage Account Placed Collateral Map	FCT_ACCT_PLACED_COLL_MAP	Fact Account Placed Collateral Map	T2T_FCT_ACCT_PLACED_COLL_MAP
STG_RI_ISSUED_PLACED_COLL_MAP	Stage Reinsurance Issued Placed Collateral Map	FCT_RI_ISSUED_PLACED_COLL_MAP	Fact Reinsurance Issued Placed Collateral Map	T2T_FCT_RI_ISSUED_PLCD_COLL_MAP

21.3 Deploying Placed Collateral Tables on Hive

All RDBMS related Placed Collateral Dimension and Result tables can also be deployed on Hive (Stage and Results on Hive).

> **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.

21.4 Populating Placed Collateral Dimension Tables

Follow this SCD process to populate data into any Placed Collateral Dimension table:

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

1. To populate data into any Placed Collateral Dimension table, execute the SCD batch. For a detailed procedure, see Slowly Changing Dimension (SCD) Process.

2. To check the SCD batch execution status of a Placed Collateral 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.

21.5 Populating Placed Collateral T2T Result Tables

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

> **NOTE** You can also follow this T2T process to populate data into any Hive related Placed Collateral T2T Result table.

1. To populate data into any Placed Collateral T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Placed Collateral Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

22 Insurance Policy Transactions Tables

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

This chapter includes the following 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.

22.1 About Insurance Policy Transactions T2Ts (Result Tables)

Insurance Policy Transactions T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCT_POLICY_TRANSACTIONS_HLD	This T2T stores the reinsurance held transaction details.
T2T_FCT_POLICY_TRANSACTIONS_ISS	This T2T stores the reinsurance issued transaction details.
T2T_FPT_STG_RETIREMENT_ACCOUNTS_TXNS	This T2T stores retirement accounts transactions.
T2T_FPT_STG_PROP_CASU_POLICY_TXNS	This T2T stores the transaction details for property and casualty policy.
T2T_FPT_STG_LIFE_INS_POLICY	This T2T stores the transaction details for life insurance policy.
T2T_FPT_STG_HEALTH_INS_TXNS	This T2T stores the transaction details for a health insurance policy.
T2T_FPT_STG_ANNUITY_TXNS	This T2T stores transactions details for annuity contracts.

The mapping details for the Insurance Policy Transactions T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_REINSURANCE_HELD_TXNS	Stage Reinsurance Held Transactions	FCT_POLICY_TRANSACTIONS	Fact Policy Transactions	T2T_FCT_POLICY_TRANSACTIONS_HLD
STG_REINSURANCE_ISSUED_TXNS	Stage Reinsurance Issued Transactions	FCT_POLICY_TRANSACTIONS	Fact Policy Transactions	T2T_FCT_POLICY_TRANSACTIONS_ISS
STG_RETIREMENT_ACCOUNTS_TXNS	Stage Retirement Accounts Transactions	FCT_POLICY_TRANSACTIONS	Fact Policy Transactions	T2T_FPT_STG_RETIREMENT_ACCOUNTS_TXNS

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_PROP_CASU_POLICY_TXNS	Stage Property Casualty Policy Transactions	FCT_POLICY_TRANSACTIONS	Fact Policy Transactions	T2T_FPT_STG_PROP_CASU_POLICY_TXNS
STG_LIFE_INS_POLICY_TXNS	Stage Life Insurance Policy Transactions	FCT_POLICY_TRANSACTIONS	Fact Policy Transactions	T2T_FPT_STG_LIFE_INS_POLICY
STG_HEALTH_INS_POLICY_TXNS	Stage Health Insurance Policy Transactions	FCT_POLICY_TRANSACTIONS	Fact Policy Transactions	T2T_FPT_STG_HEALTH_INS_TXNS
STG_ANNUITY_TXNS	Stage Annuity Transactions	FCT_POLICY_TRANSACTIONS	Fact Policy Transactions	T2T_FPT_STG_ANNUITY_TXNS

22.2 Deploying Insurance Policy Transactions Tables on Hive

All RDBMS related Insurance Policy Transactions Result tables can also be deployed on Hive (Stage and Results on Hive).

> **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.

22.3 Populating Insurance Policy Transactions T2T Result Tables

> **NOTE** As a prerequisite, ensure to load DIM_STD_TRANSACTION_TYPE seeded table.

Follow this T2T process to populate data into Insurance Policy Transactions T2T Result tables:

> **NOTE** You can also follow this T2T process to populate data into any Hive related Insurance Policy Transactions T2T Result table.

1. To populate data into Insurance Policy Transactions T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.

2. To check the T2T batch execution status of the Insurance Policy Transactions Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

23 GL to Management Reporting Table

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

This chapter includes these topics:

A general ledger account is an account or record used to sort, store and summarize the transactions of a company.

The purpose of GL to Management Reporting tables for reporting that provides 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.

23.1 About GL to Management Reporting T2T (Result Table)

GL to Management Reporting T2T and its description is given here.

T2T Name	T2T Description
T2T_FCT_MGMT_REPORTING_STG_GL_DATA	This T2T stores management reporting data related to organization and product profitability, income statement and balance sheet.

The mapping details for the GL to Management Reporting T2T is given here.

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

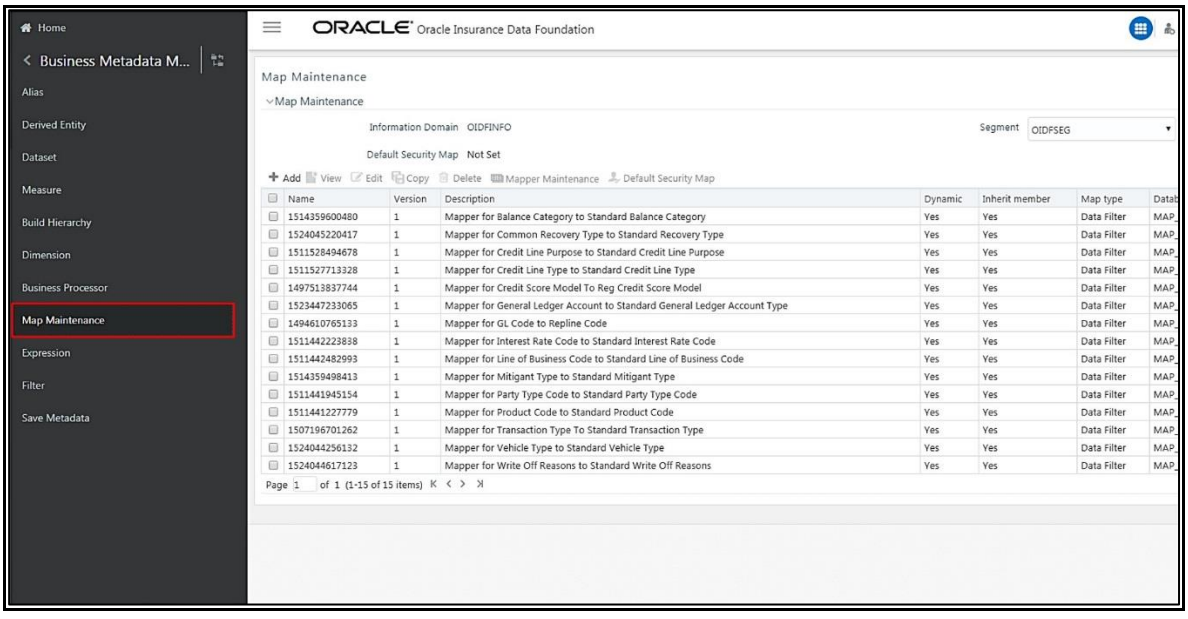
23.2 About Mapper for GL to Management Reporting Result Table

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.

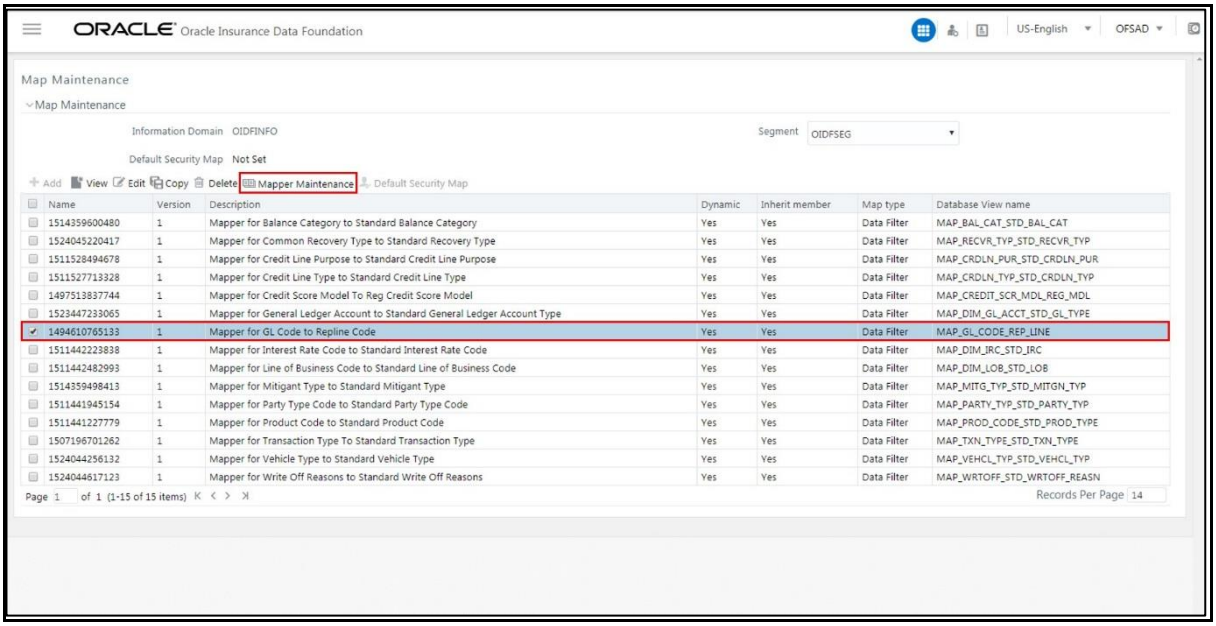
23.3 Maintaining Mappers for GL to Management Reporting Result Table

To maintain Mappers through the Map Maintenance component of OFSAAI:

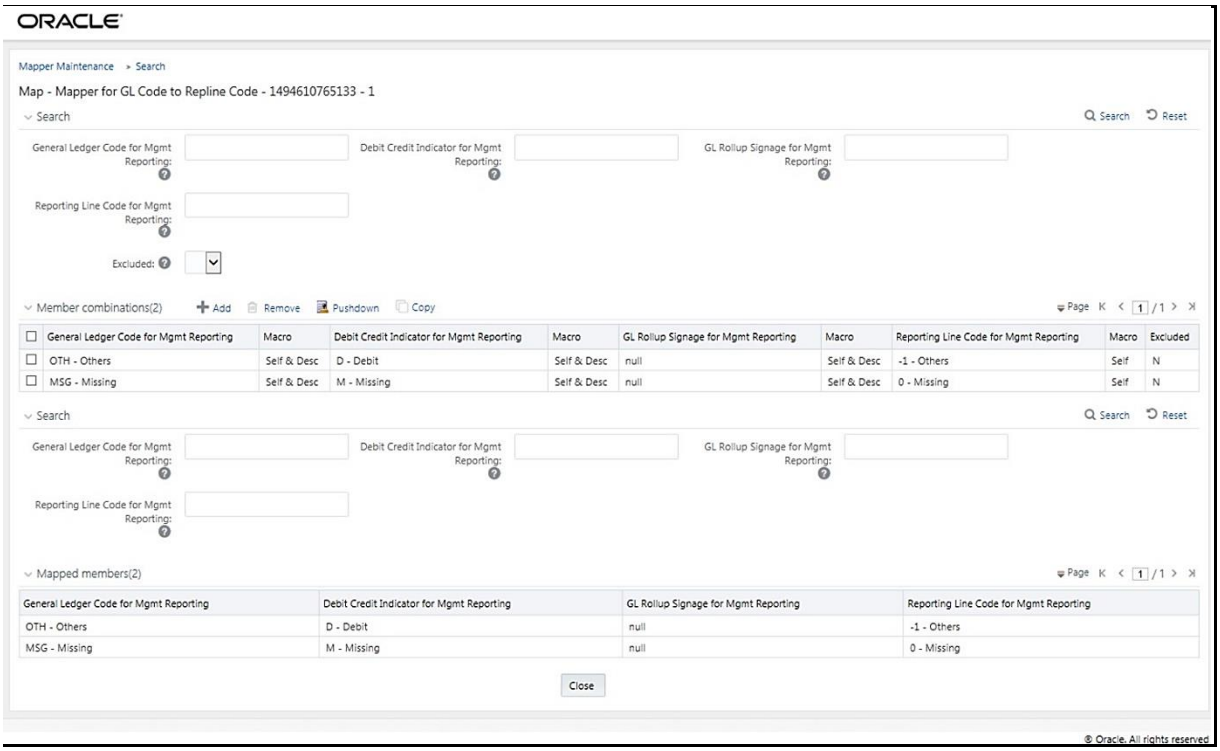
1. Go to OFSAAI > Oracle Insurance Data Foundation > Unified Analytical Metadata > Business Metadata Management > Map Maintenance. The Map Maintenance page appears.



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



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



23.3.1 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

23.3.2 Possible Mapping Combinations

There are four types of mapping combinations. They are:

- Mapping combinations at child hierarchy level.
- Mapping combinations at parent and child hierarchy level.
- Mapping combinations at parent hierarchy level without Descendants.
- Mapping combinations at parent hierarchy level by removing one or more Descendants.

23.3.2.1 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:

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

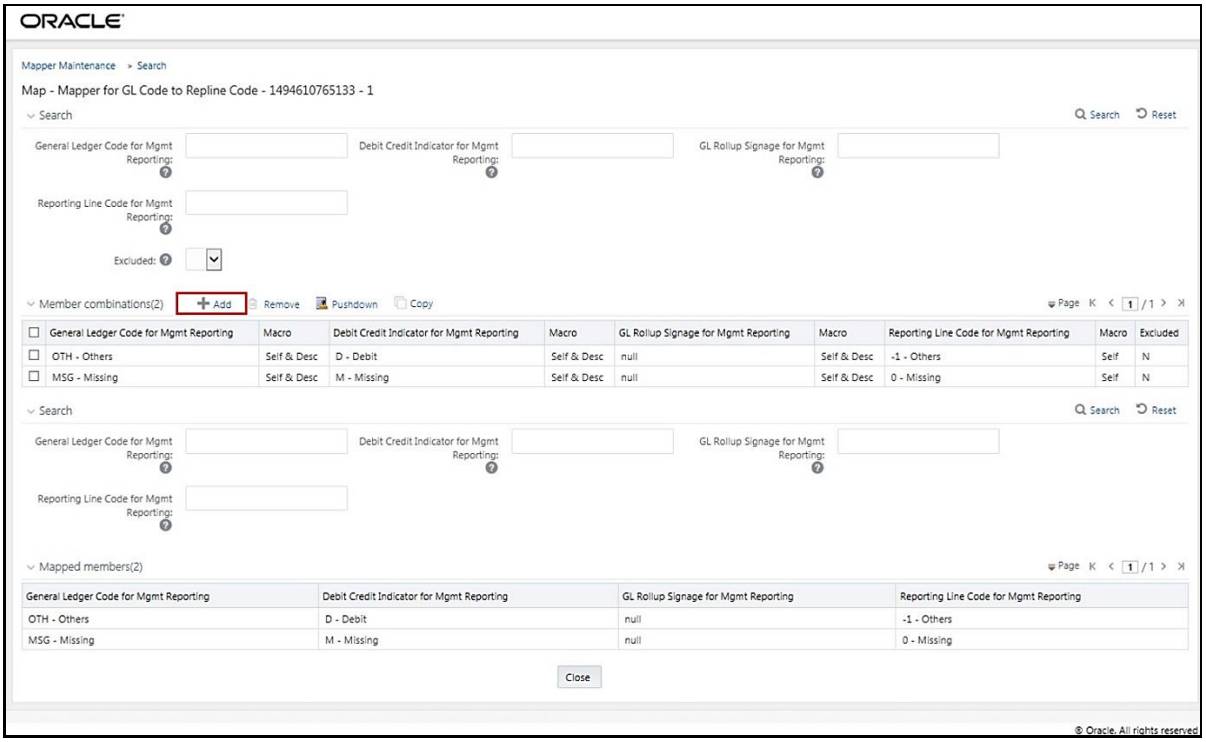
To map one GL Code to one Reporting Line Code, use the Mapper Maintenance utility. In this utility, select one member in the General Ledger Code for Mgmt Reporting hierarchy, select one member in the Debit Credit Indicator hierarchy with the value Debit or Credit or Missing, and select one member in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

NOTE The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:

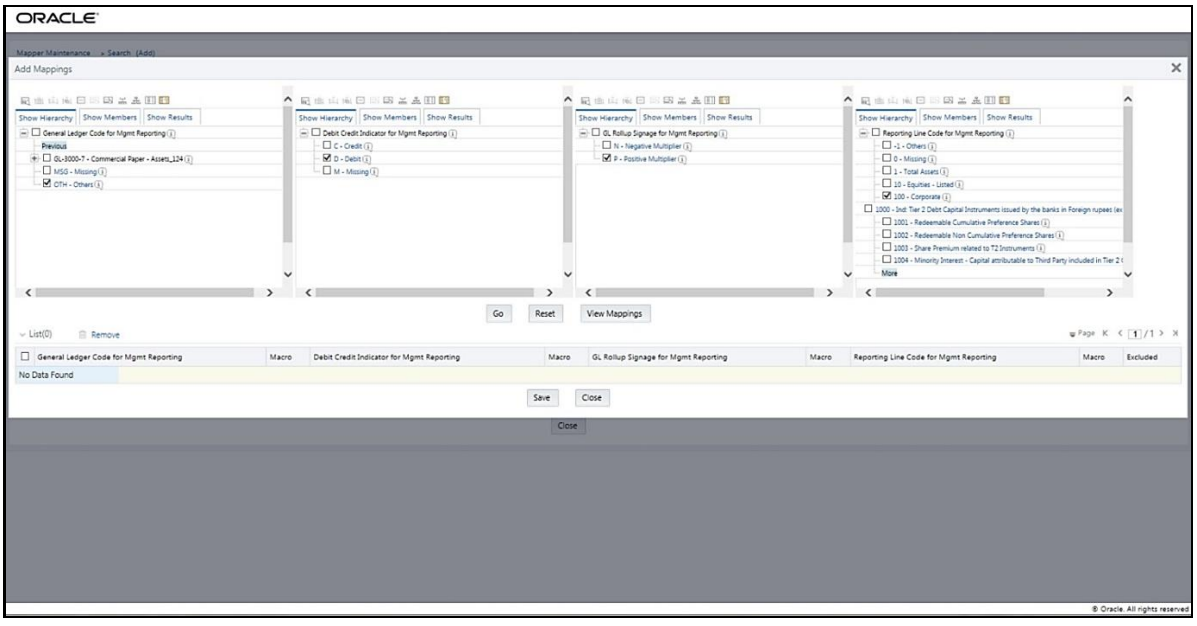
- The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
- The value must be D, when you map GL Code to the Debit member.
- The value must be C, when you map GL Code to the Credit member.

To perform One-to-One mapping with or without Debit Credit Indicator:

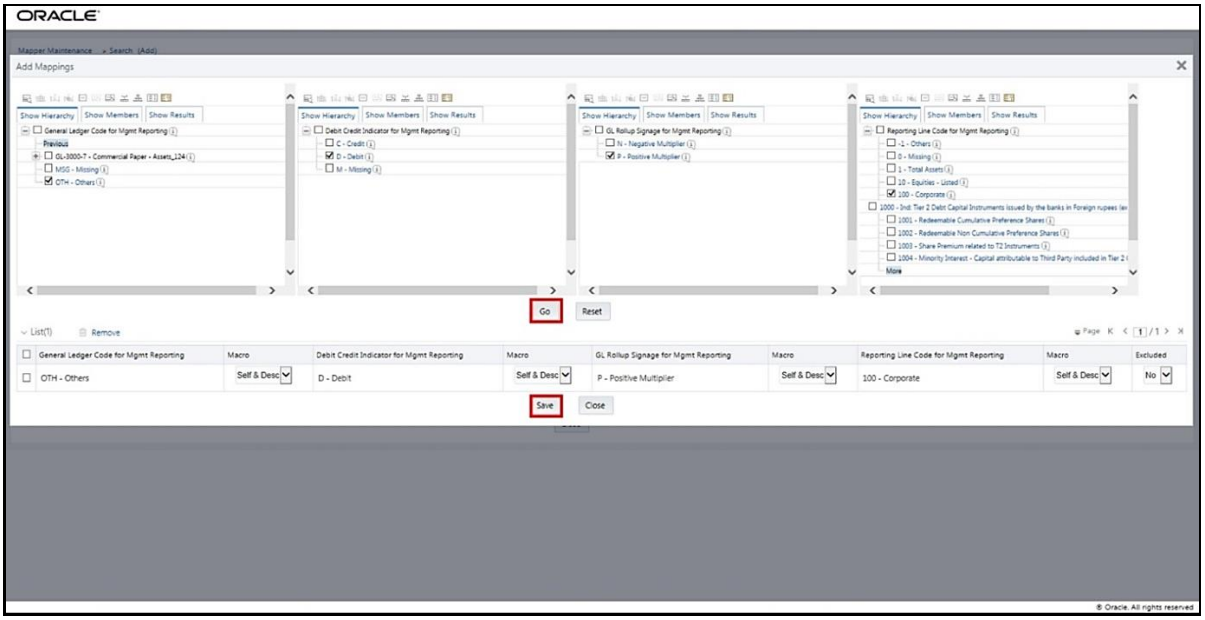
1. In the Mapper Maintenance page, in the Member combinations section, click Add.



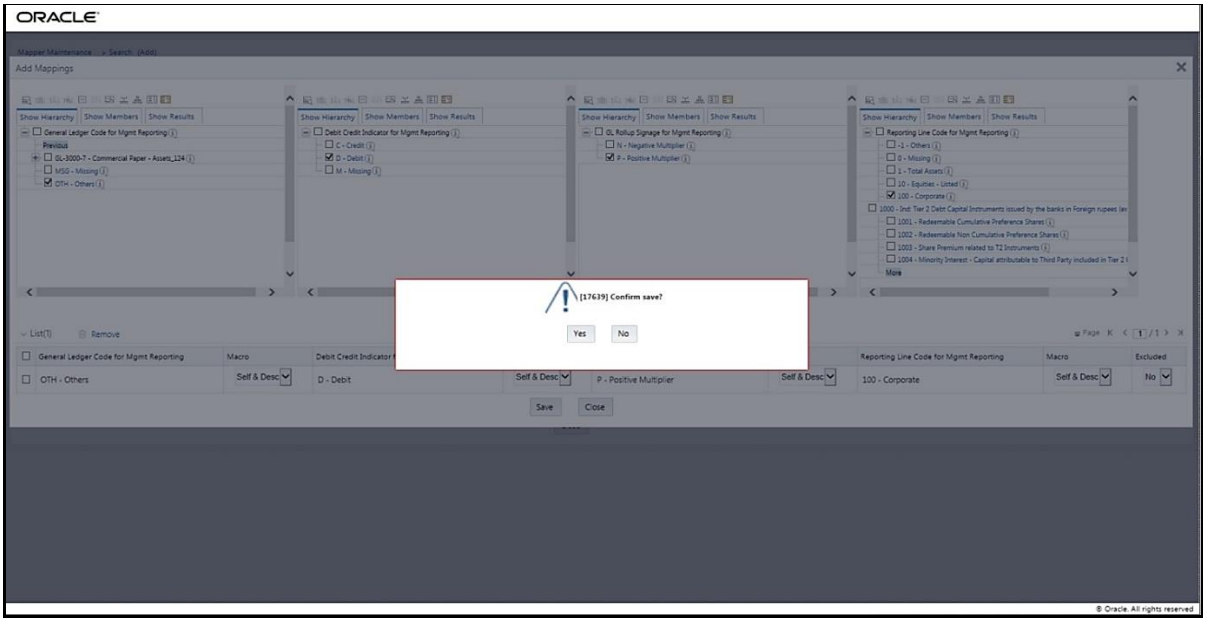
2. The Add Mappings pop-up page appears. For illustration, select the member of the General Ledger Code for Mgmt Reporting hierarchy OTH - Others, and the member of the Debit Credit Indicator for Mgmt Reporting hierarchy D - Debit to map to the member of the Reporting Line Code for Mgmt Reporting hierarchy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



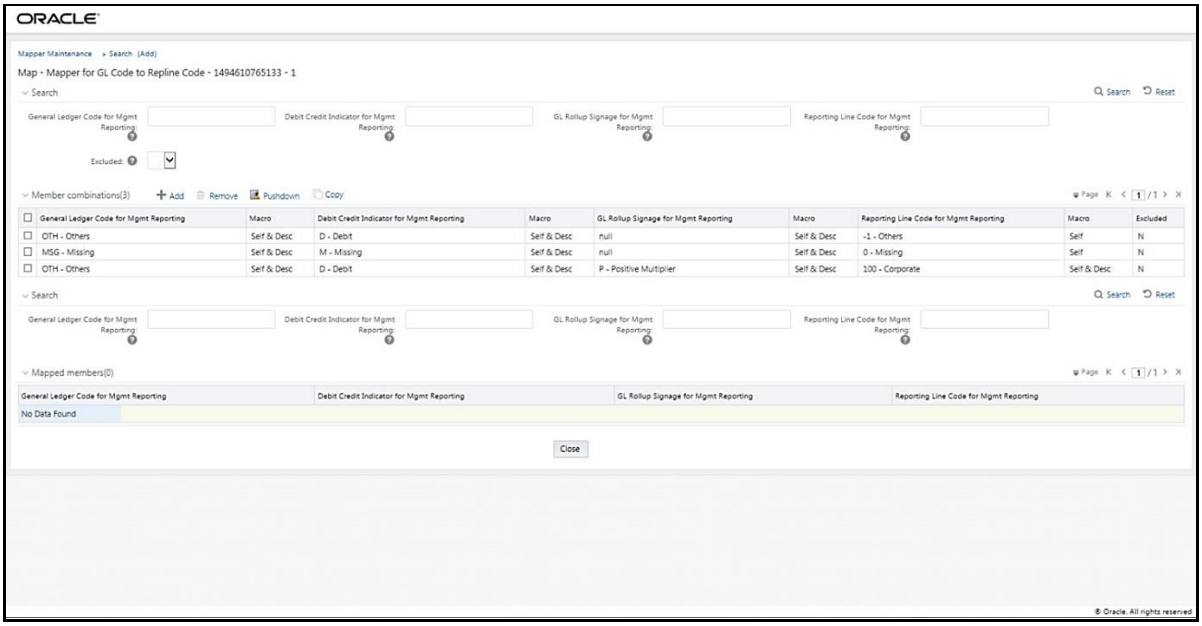
3. To map the members, click Go. The list of mapped members appear at the bottom. To save the mappings, click Save.



4. An acknowledgement pop-up message appears. To confirm saving the mapping, click Yes.



5. The mapped member combinations are listed in the Mapper Maintenance page.



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

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

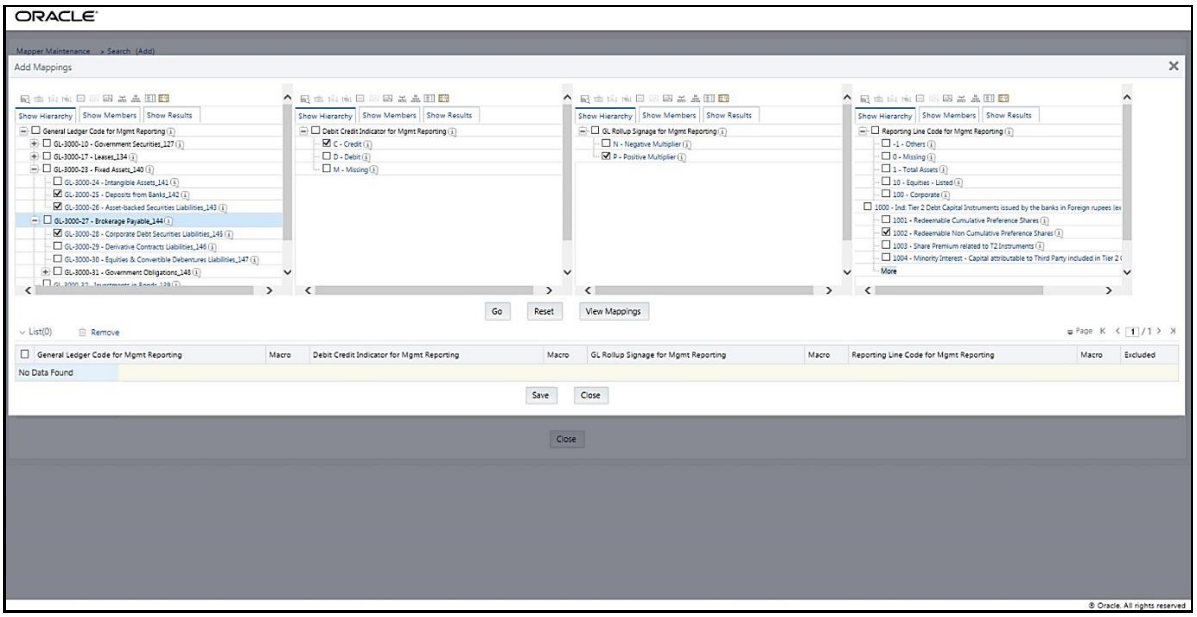
NOTE In the T2T process, in the Stage GL Data table, the GL Code aggregation takes place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:

- The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
- The value must be D, when you map GL Code to the Debit member.
- The value must be C, when you map GL Code to the Credit member.

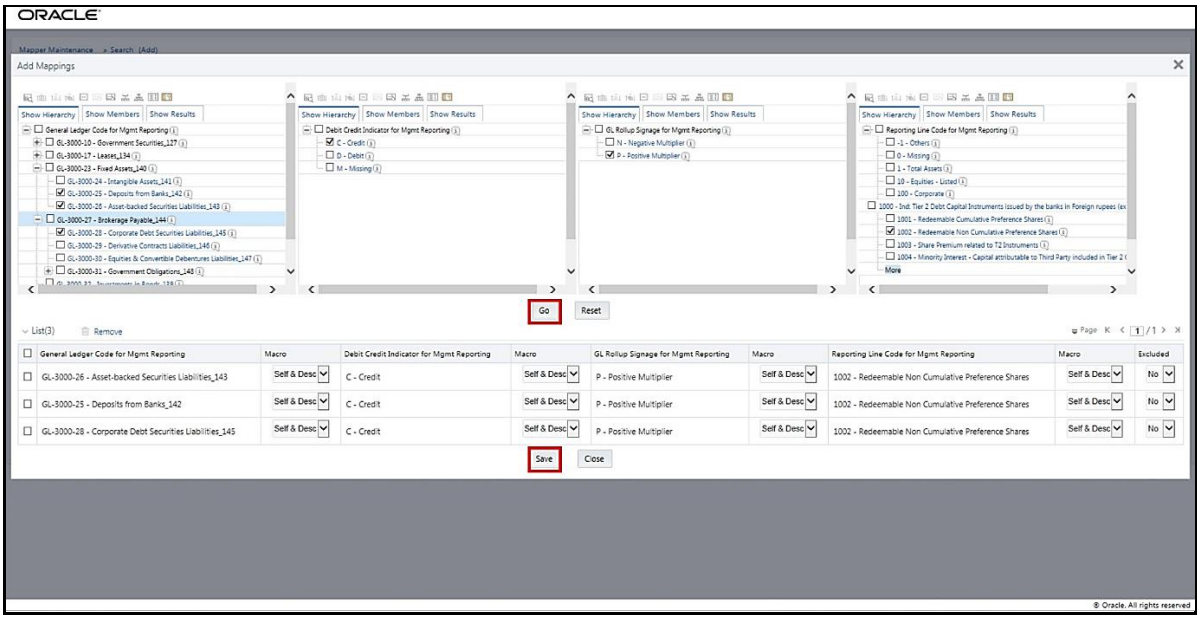
To perform Many-to-One mapping with or without Debit Credit Indicator:

1. In the Mapper Maintenance page, in the Member combinations section, click Add.
2. The Add Mappings pop-up page appears. 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.

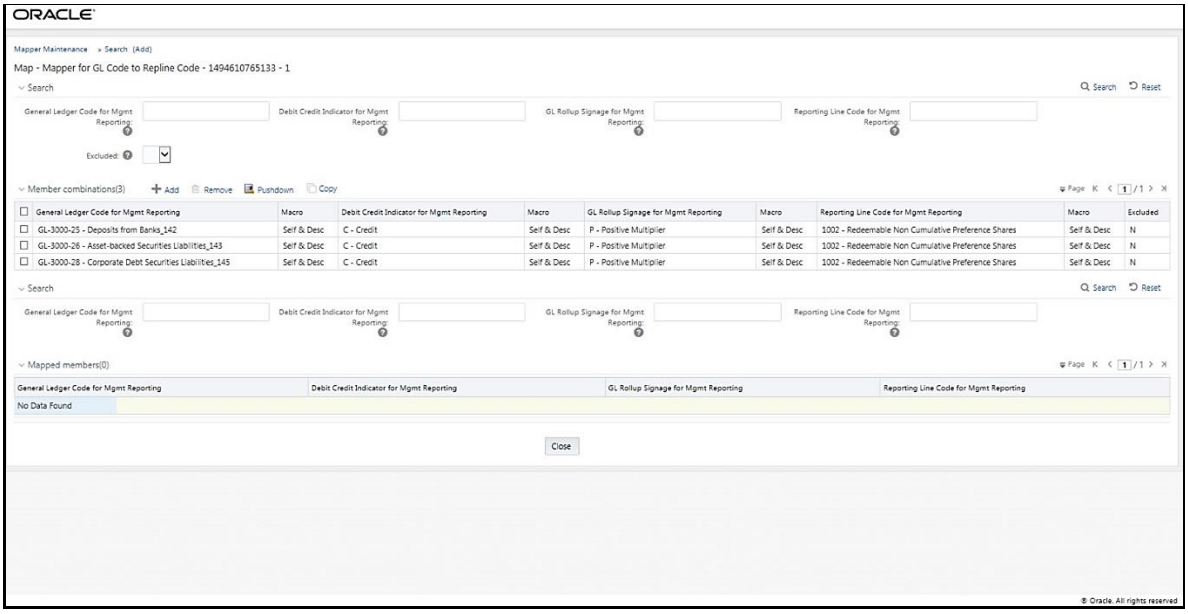


3. To map the members, click Go. The list of mapped members appear at the bottom. To save the mappings, click Save.



4. An acknowledgement pop-up message appears. To confirm saving the mappings, click Yes.

5. The mapped member combinations are listed in the Mapper Maintenance page.



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

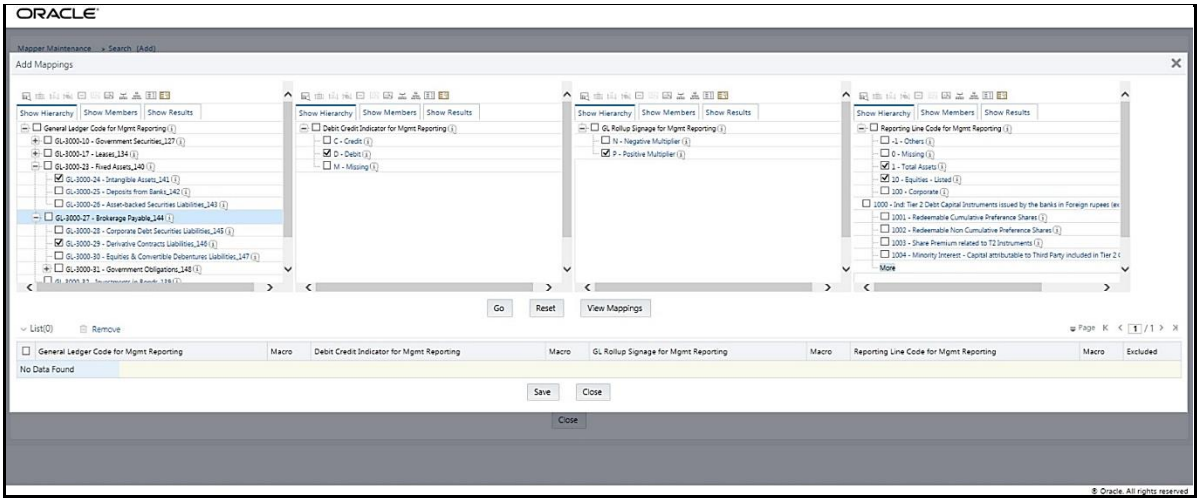
To map multiple GL Codes to multiple Reporting Line Codes, use the Mapper Maintenance utility. In this utility, select two or more members in the General Ledger Code for Mgmt Reporting hierarchy, select one member in the Debit Credit Indicator hierarchy with the value Debit or Credit or Missing, and 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:

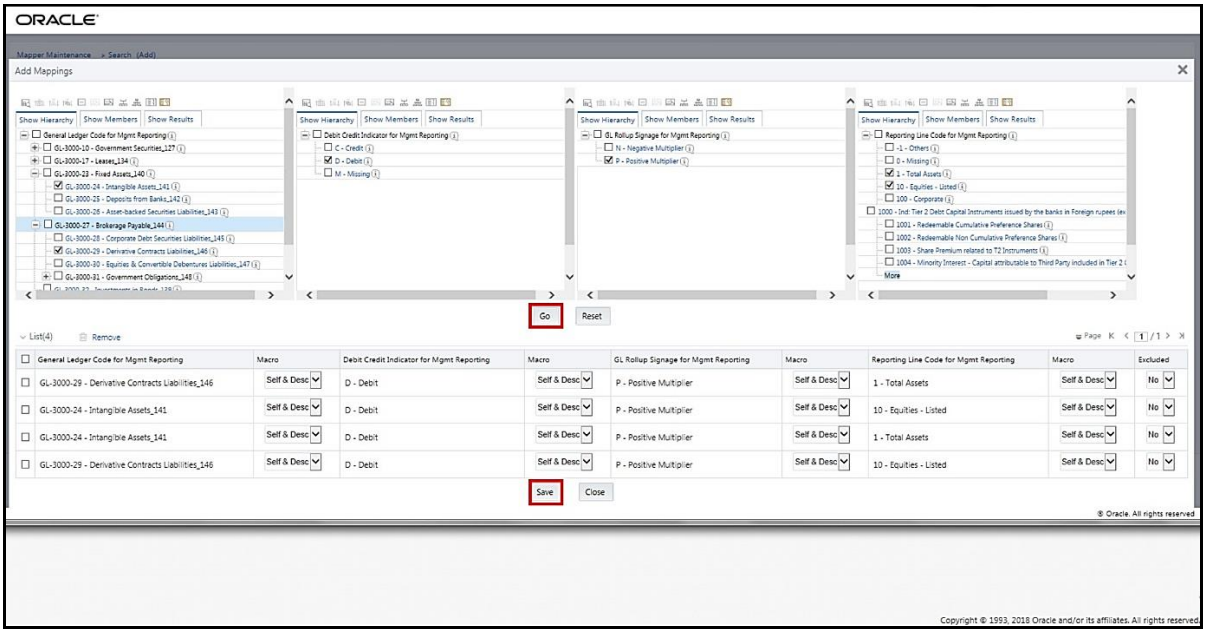
- The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
- The value must be D, when you map GL Code to the Debit member.
- The value must be C, when you map GL Code to the Credit member.

To perform Many-to-Many mapping with or without Debit Credit Indicator:

1. In the Mapper Maintenance page, in the Member combinations section, click Add.
2. The Add Mappings pop-up page appears. 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.

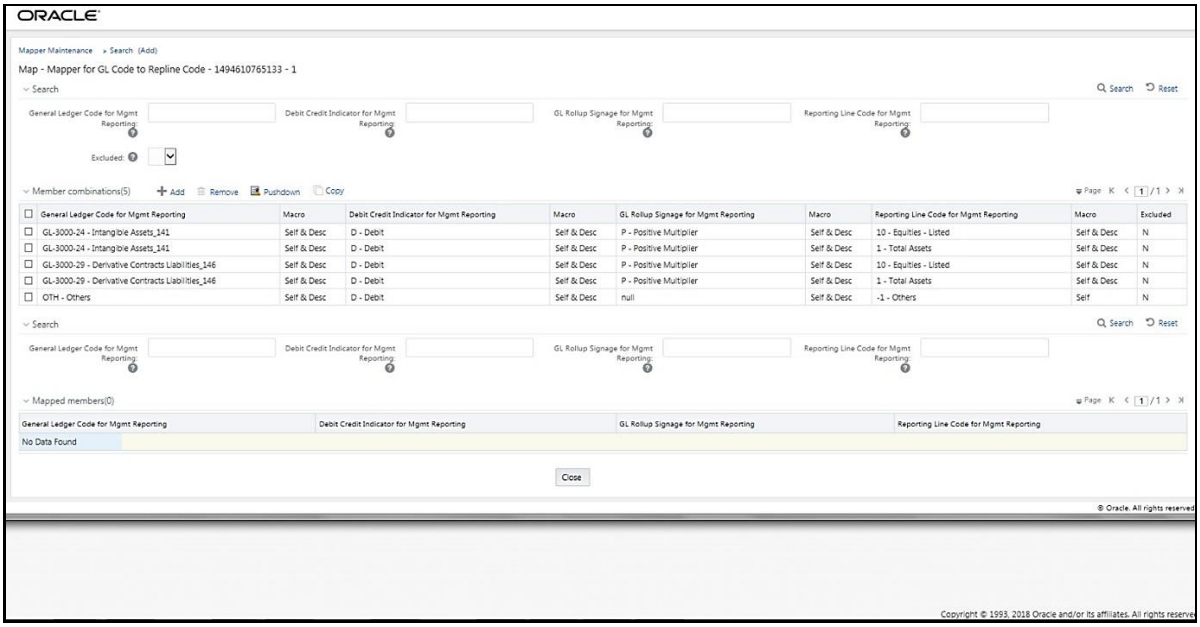


3. To map the members, click Go. The list of mapped members appear at the bottom. To save the mappings, click Save.



4. An acknowledgement pop-up message appears. To confirm saving the mappings, click Yes.

5. The mapped member combinations are listed in the Mapper Maintenance page.



23.3.2.2 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:

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

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

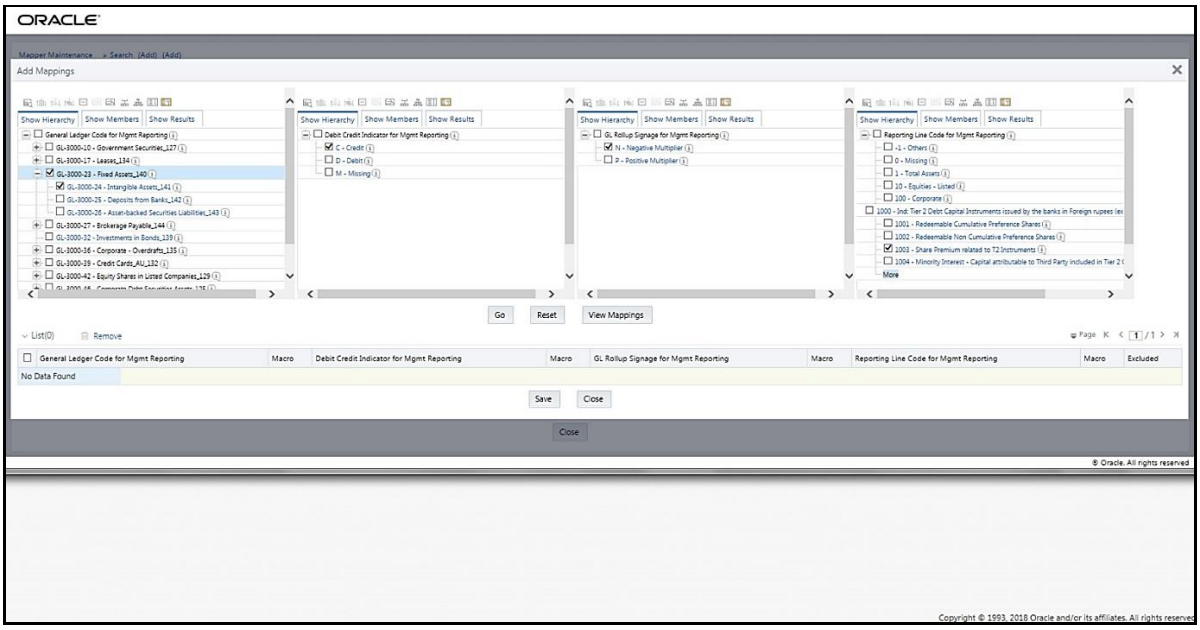
NOTE In the T2T process, in the Stage GL Data table, the aggregation of parent GL Codes and their descendants take place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:

- The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
- The value must be D, when you map GL Code to the Debit member.

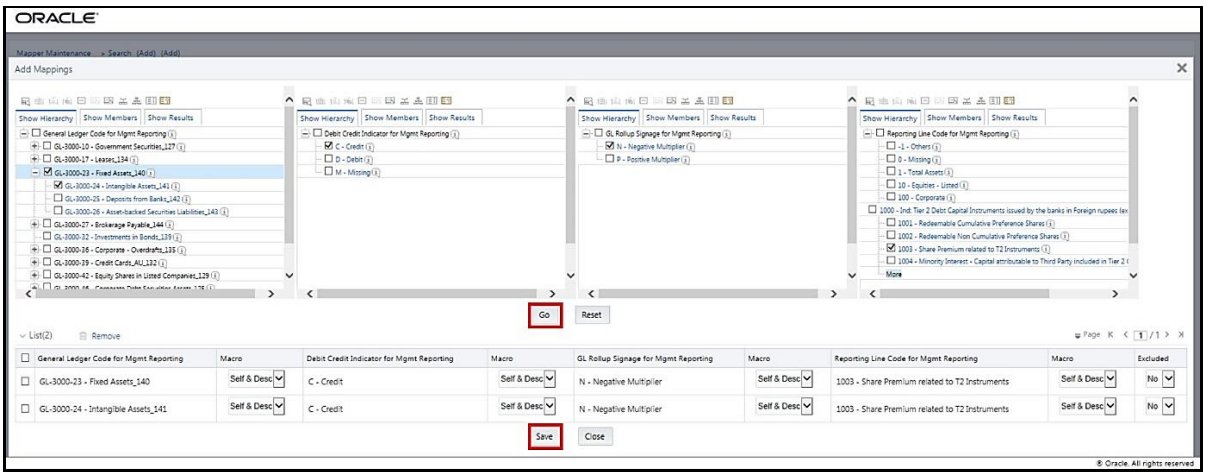
- The value must be C, when you map GL Code to the Credit member.

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

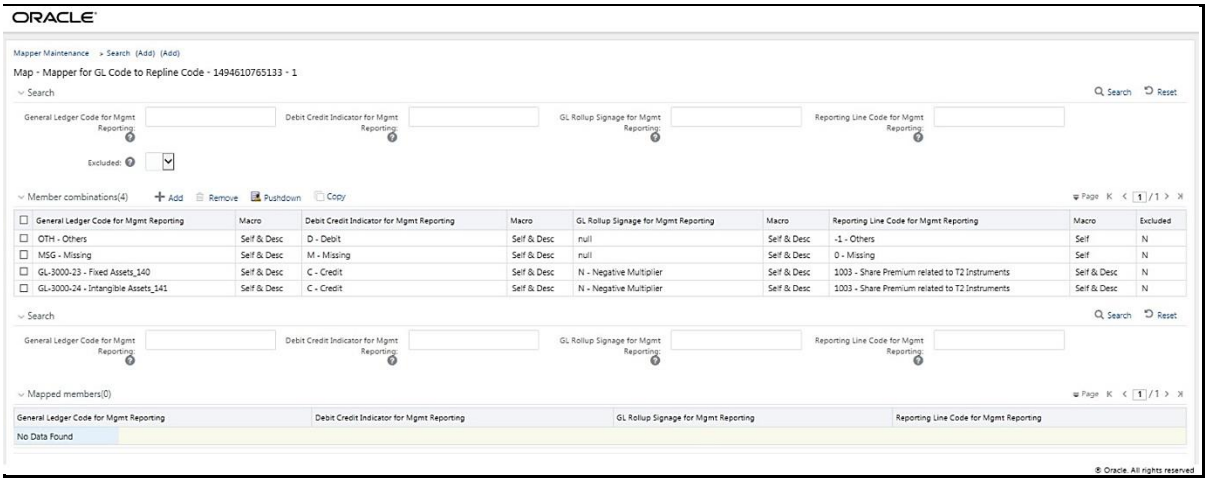
1. In the Mapper Maintenance page, in the Member combinations section, click Add.
2. The Add Mappings pop-up page appears. For illustration, select one parent member of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-23 – Fixed Assets_140 and its child member GL-3000-24 – Intangible Assets_141, 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.



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



4. An acknowledgement pop-up message appears. To confirm saving the mappings, click Yes.
5. The mapped member combinations are listed in the Mapper Maintenance page.



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

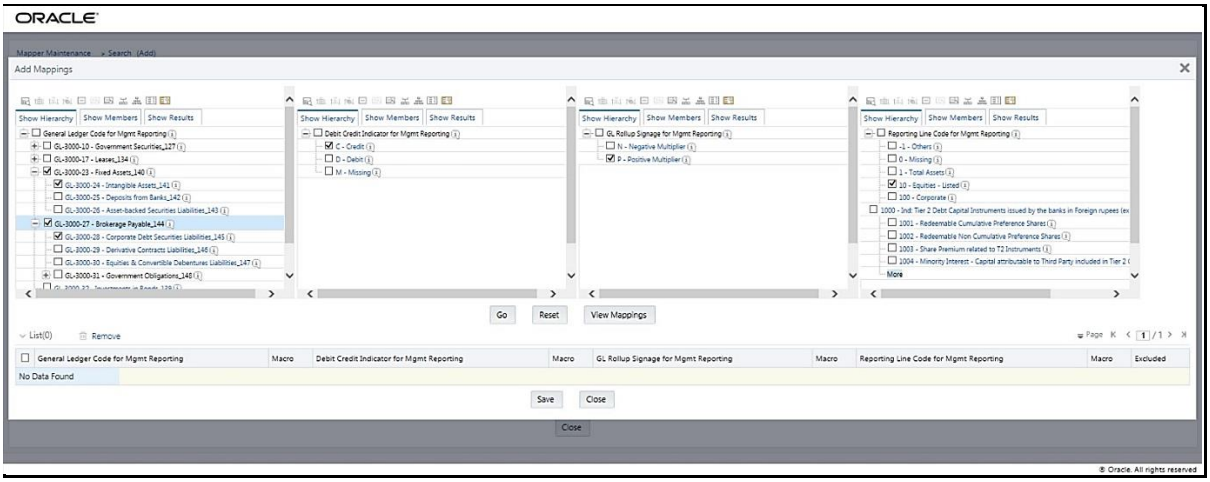
To map multiple parent GL Codes to one Reporting Line Code, use the Mapper Maintenance utility. In this utility, select two or more parent GL Codes, select one member in the Debit Credit Indicator hierarchy with the value Debit or Credit or Missing, and select one member in the Reporting Line Code hierarchy.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

NOTE In the T2T process, in the Stage GL Data table, the aggregation of parent GL Codes and their descendants take place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:

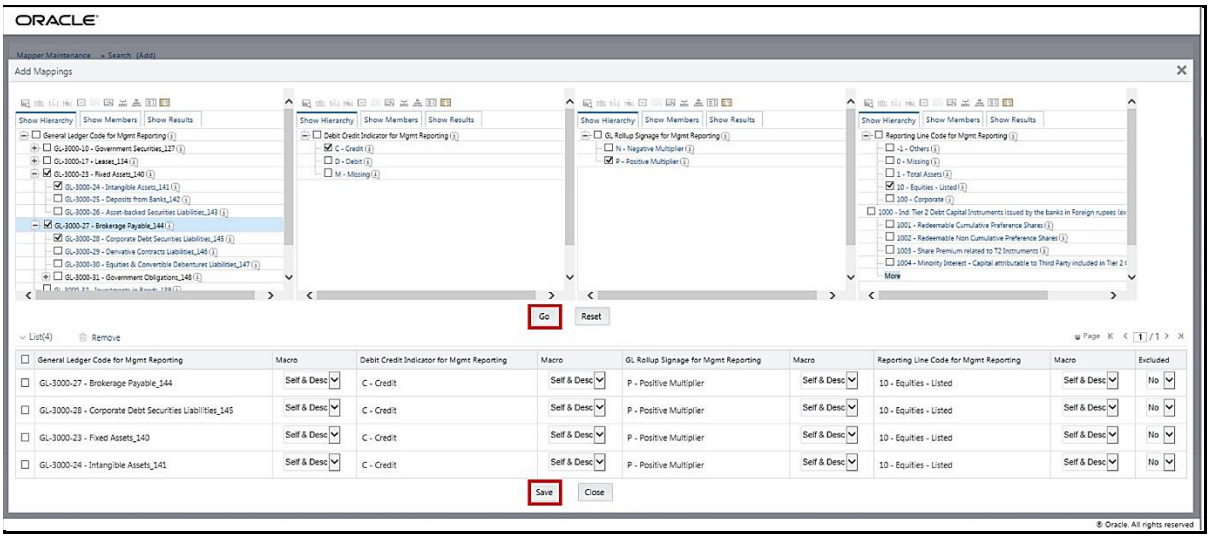
- The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
- The value must be D, when you map GL Code to the Debit member.
- The value must be C, when you map GL Code to the Credit member.

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

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

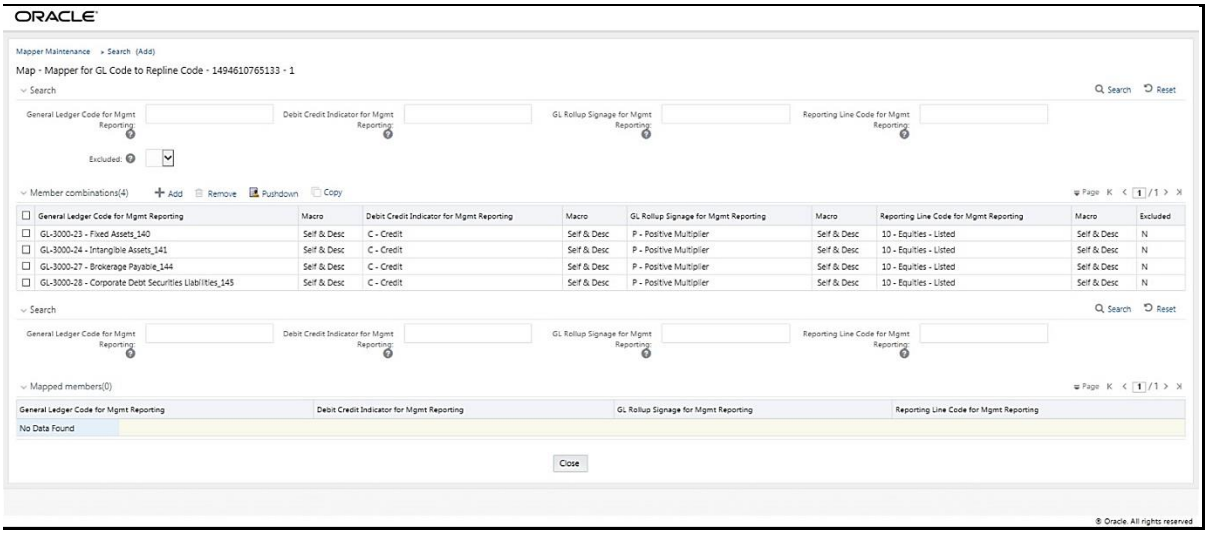


3. To map the members, click Go. The list of mapped members appear at the bottom. To save the mappings, click Save.



4. An acknowledgement pop-up message appears. To confirm saving the mappings, click Yes.

5. The mapped member combinations are listed in the Mapper Maintenance page.



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

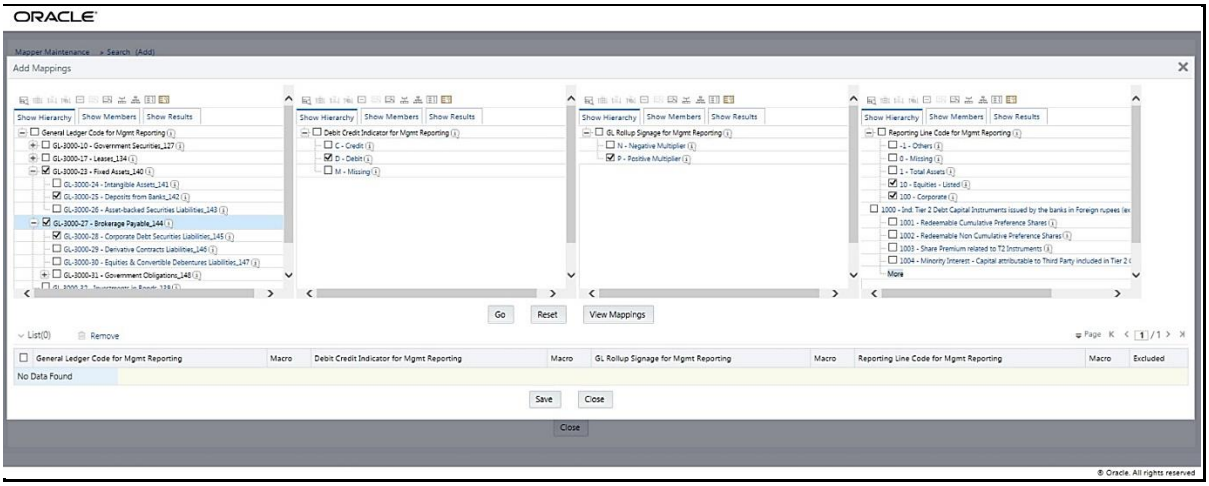
To map multiple parent GL Codes to multiple Reporting Line Codes, use the Mapper Maintenance utility. In this utility, select two or more parent GL Codes, select one member in the Debit Credit Indicator hierarchy with the value Debit or Credit or Missing, and select two or more members in the Reporting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

NOTE In the T2T process, in the Stage GL Data table, the aggregation of parent GL Codes and their descendants take place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:

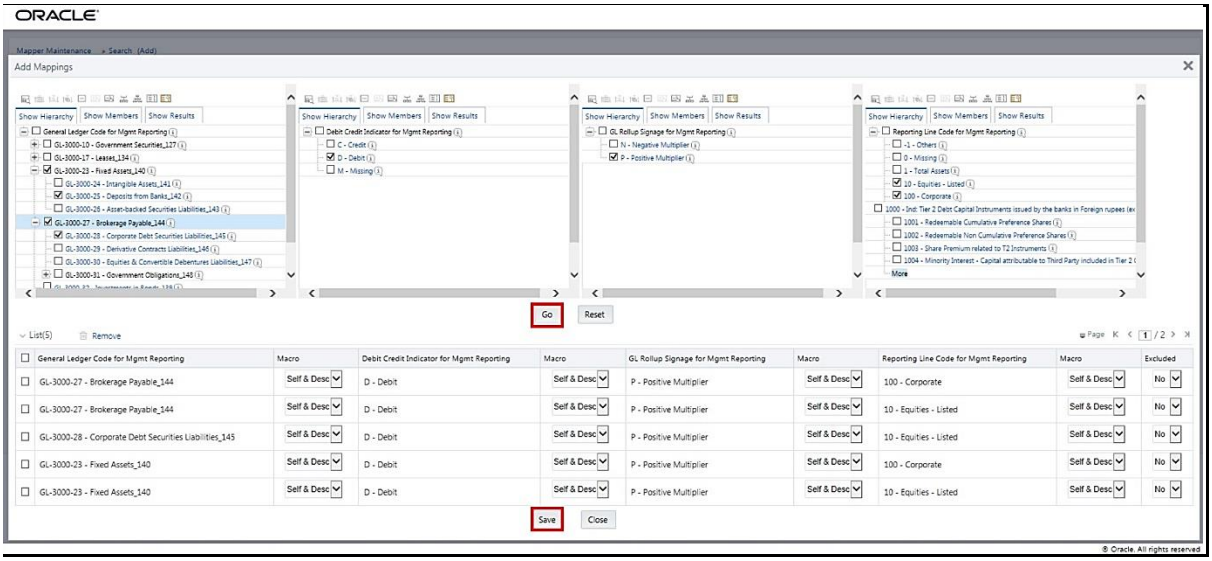
- The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
- The value must be D, when you map GL Code to the Debit member.
- The value must be C, when you map GL Code to the Credit member.

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

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

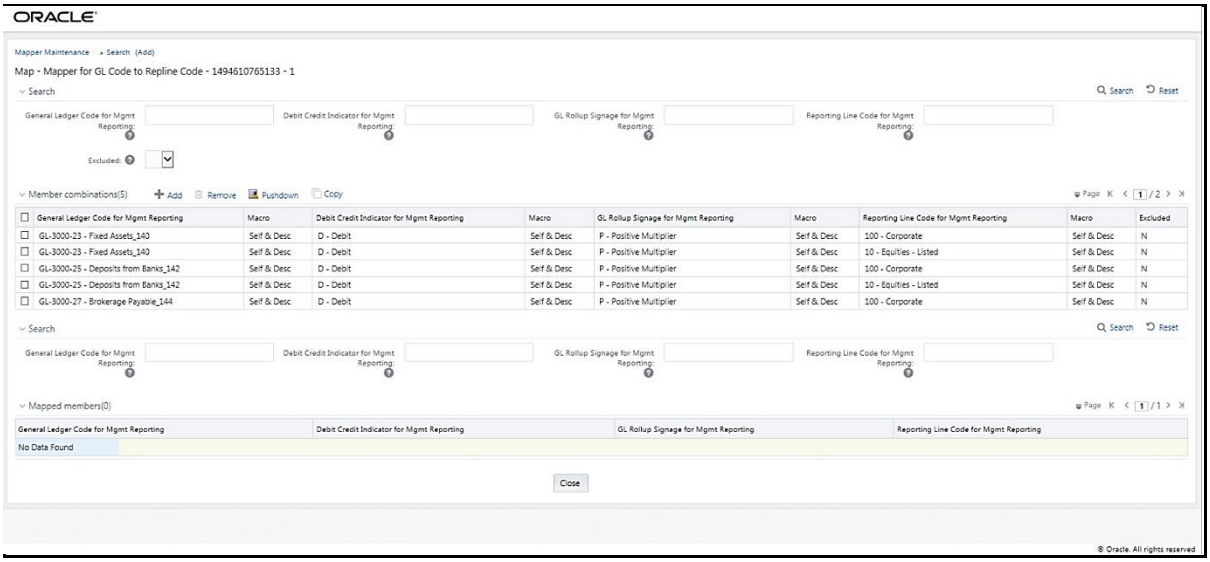


3. To map the members, click Go. The list of mapped members appear at the bottom. To save the mappings, click Save.



4. An acknowledgement pop-up message appears. To confirm saving the mappings, click Yes.

5. The mapped member combinations are listed in the Mapper Maintenance page.



23.3.2.3 Mapping Combinations at Parent Hierarchy Level without Descendants

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

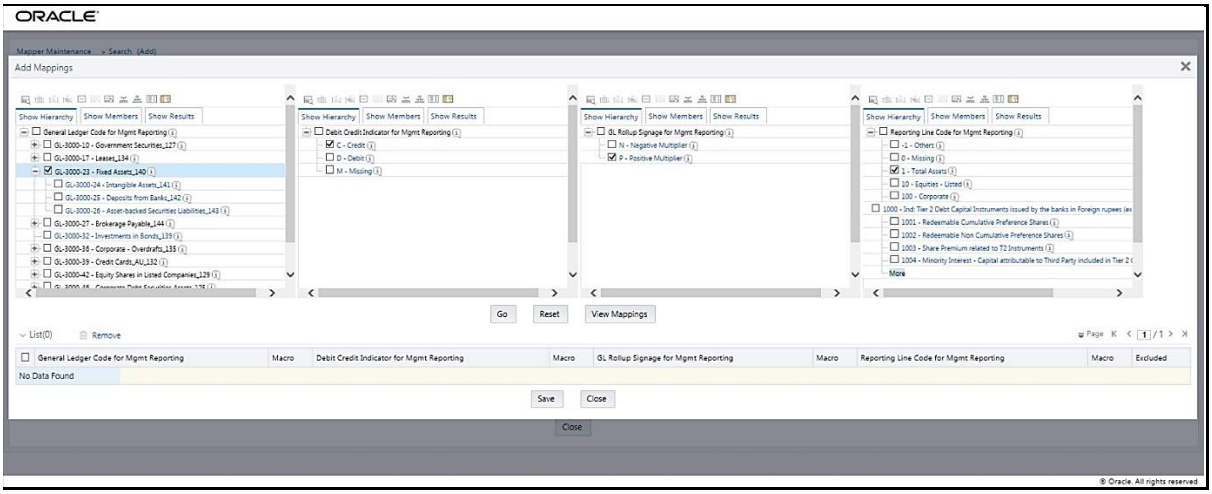
To map one parent GL Code without Descendants to one or more Reporting Line Codes, use the Mapper Maintenance utility. In this utility, select one parent GL Code, select one member in the Debit Credit Indicator hierarchy with the value Debit or Credit or Missing, and select one or more members in the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

NOTE In the T2T process, in the Stage GL Data table, the aggregation of parent GL Code takes place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:

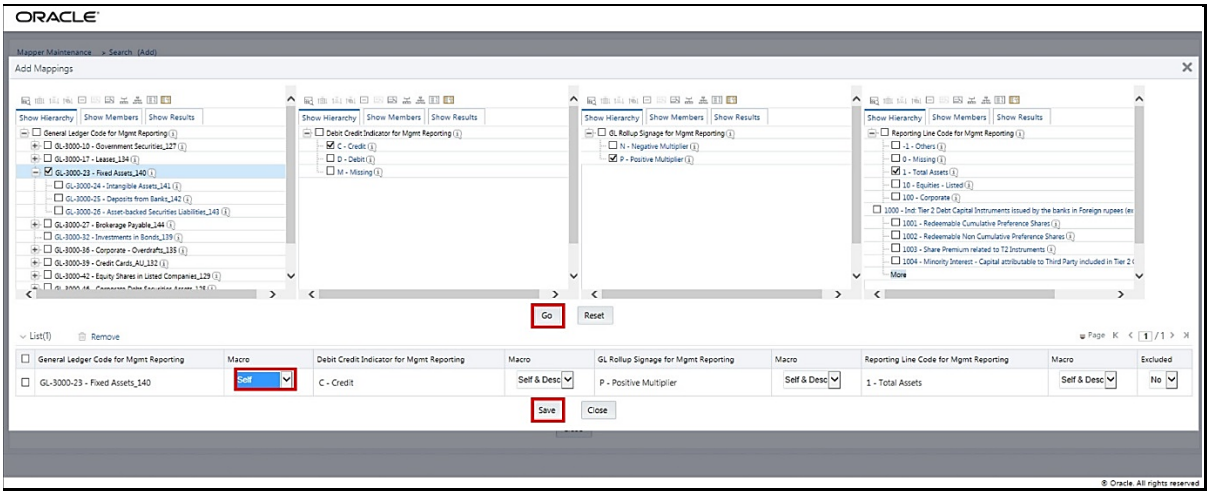
- The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
- The value must be D, when you map GL Code to the Debit member.
- The value must be C, when you map GL Code to the Credit member.

To perform One Parent to One or Many Reporting Line Codes mappings without Descendants:

1. In the Mapper Maintenance page, in the Member combinations section, click Add.
2. The Add Mappings pop-up page appears. 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.

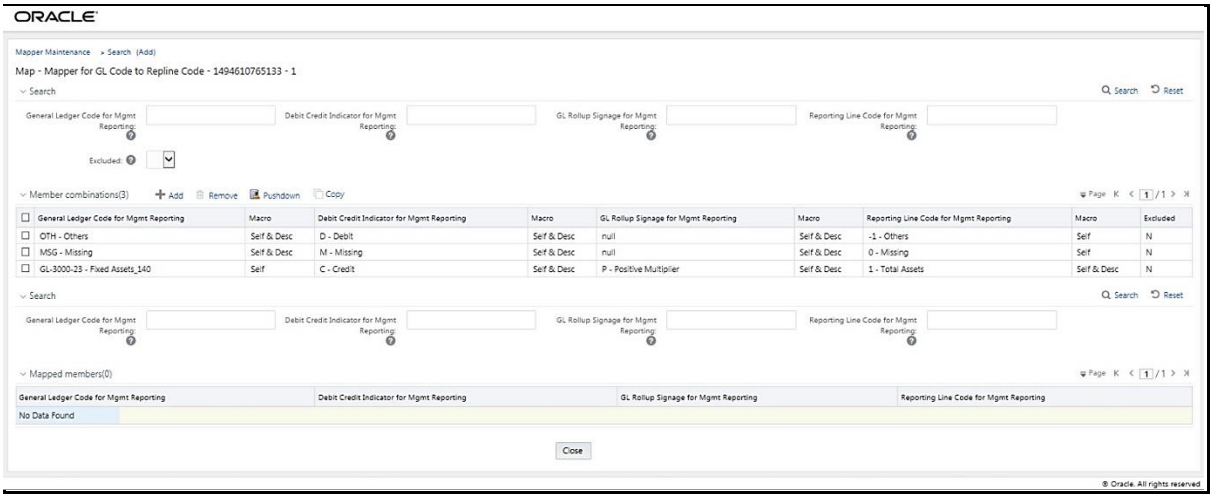


3. To map the members, click Go. The list of mapped member appears at the bottom. To exclude the Descendants of the GL Code from the mapping, in the Macro column, select Self. To save the mappings, click Save.



4. An acknowledgement 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.



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

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

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

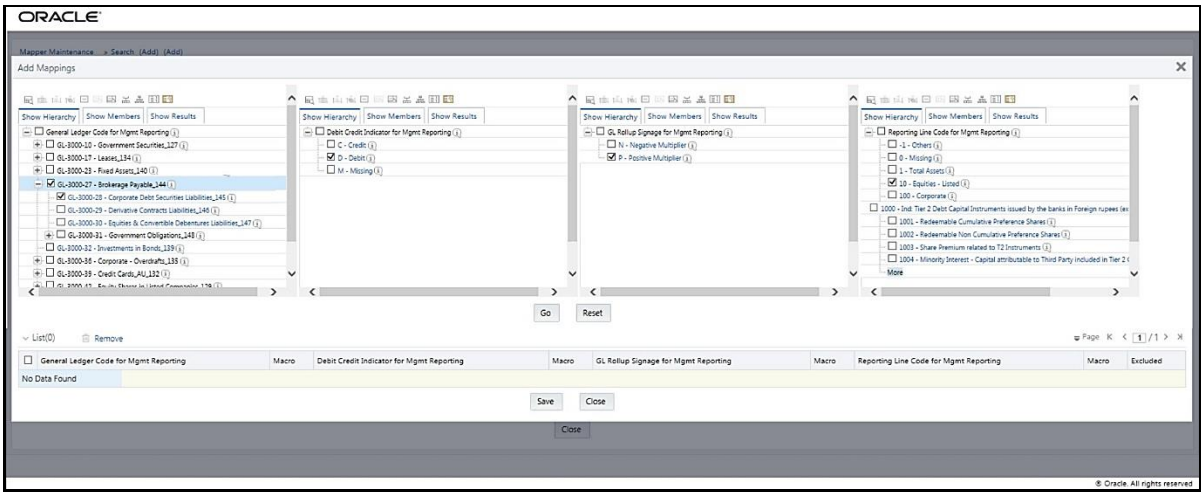
NOTE In the T2T process, in the Stage GL Data table, the aggregation of parent GL Code and its descendants take place in all the Measure columns that are associated with the Primary Keys. The conditions for the Debit Credit Indicator column values in the Stage GL Data table are:

- The value must be M, when you map GL Code to the Missing (MSG) member. If nothing is mapped, the system automatically sets value to M.
- The value must be D, when you map GL Code to the Debit member.
- The value must be C, when you map GL Code to the Credit member.

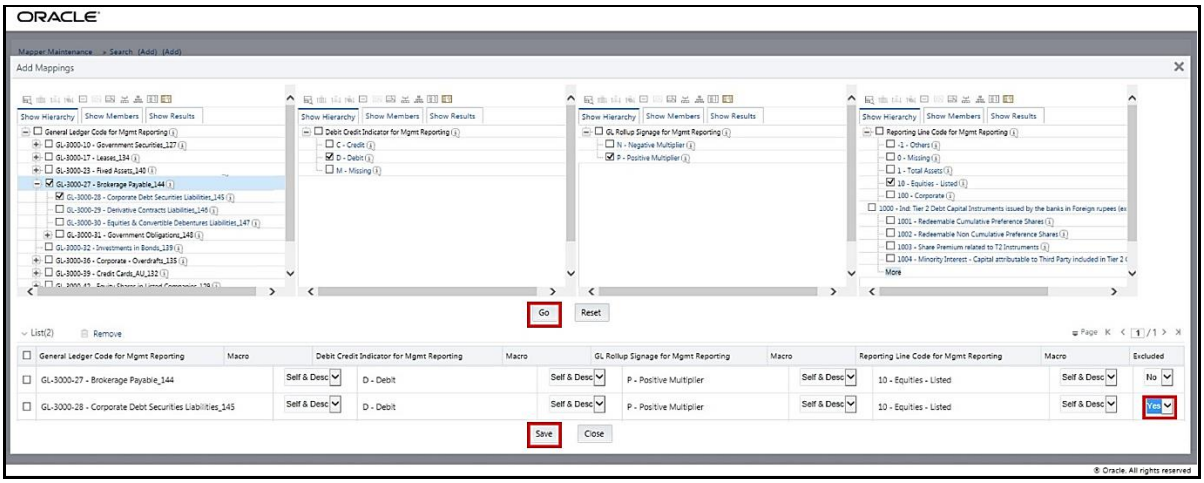
To perform One Parent to One or Many Reporting Line Codes mapping by removing one or more GL Code Descendants:

1. In the Mapper Maintenance page, in the Member combinations section, click Add.
2. The Add Mappings pop-up page appears. 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

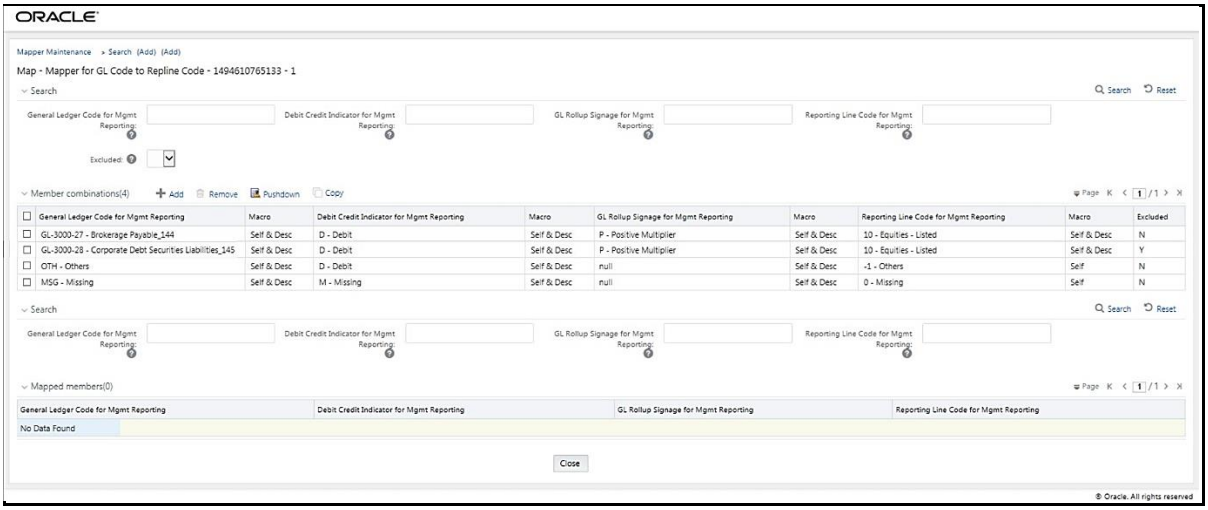
the Debit Credit Indicator for Mgmt Reporting hierarchy D - Debit to map to the member of the Reporting Line Code for Mgmt Reporting hierarchy 10 – Equities - Listed with the member of the GL Rollup Signage for Mgmt Reporting hierarchy P - Positive Multiplier.



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



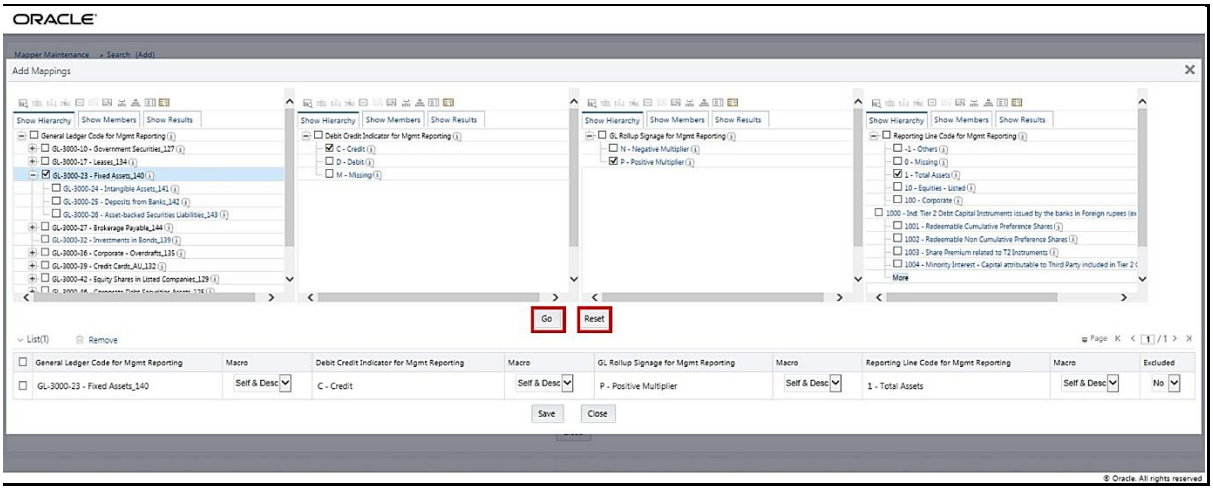
4. An acknowledgement pop-up message appears. To confirm saving the mappings, click Yes.
5. The mapped member combinations are listed in the Mapper Maintenance page with the Macro value of GL Code as Self.



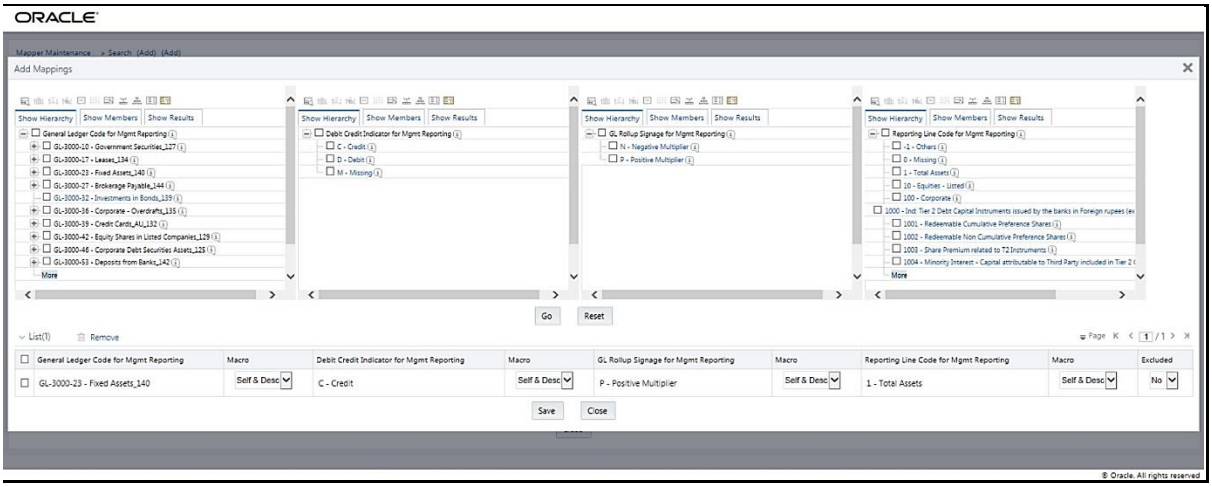
23.3.3 Performing Multiple Sets of Mapping Combinations

To perform mapping more than one time in the Add Mappings page, use the Reset functionality. To perform this procedure:

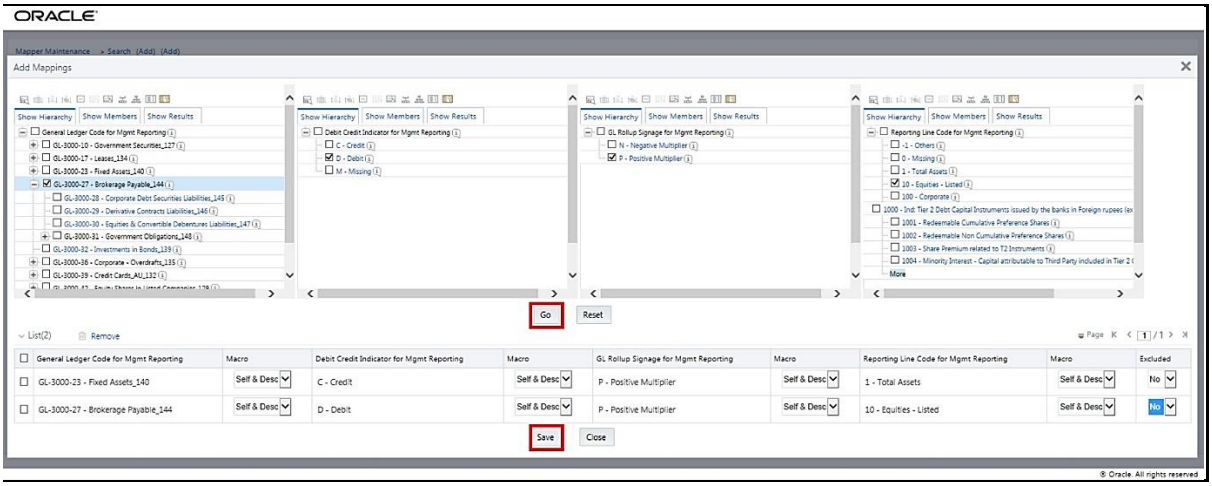
1. In the Mapper Maintenance page, in the Member combinations section, click Add.
2. The Add Mappings pop-up page appears. For illustration, select one parent member of the General Ledger Code for Mgmt Reporting hierarchy, GL-3000-23 – Fixed Assets_140, and the member of the Debit Credit Indicator for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



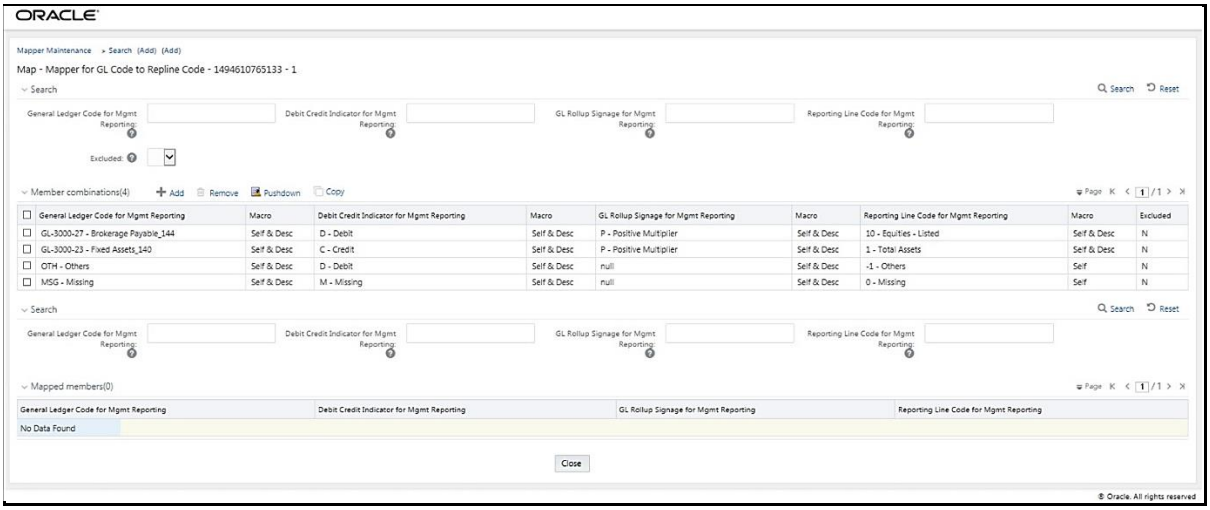
3. The mapping selections clear.



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



5. An acknowledgement pop-up message appears. To confirm saving the mappings, click Yes.
6. Both set of mapped member combinations are listed in the Mapper Maintenance page.



23.4 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 should be C or D or M).
- V_MEMBER_3 = Reporting Line Code (values from DIM_REP_LINE.N_REP_LINE_CD).

```
SELECT * FROM MAP_GL_CODE_REP_LINE
```

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

NOTE These values are real business key columns and not display codes.

23.5 Deploying GL to Management Reporting Result Tables on Hive

All RDBMS related GL to Management Reporting Result table can also be deployed on Hive (Stage and Results on Hive).

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.

23.6 Populating GL to Management Reporting T2T Result Tables

Follow this T2T process to populate data into GL to Management Reporting T2T Result table:

> **NOTE** You can also follow this T2T process to populate data into any Hive related GL to Management Reporting T2T Result table.

1. To populate data into any GL to Management Reporting T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the GL to Management Reporting Result table, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

24 Insurance Claims Tables

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

This section includes the following topics:

An insurance claim is a formal request to an Insurance Company for coverage or compensation for a covered loss or policy event. The Insurance Company validates the claim and, once approved, issues payment to the insured or an approved interested party on behalf of the insured.

OIDF sources data from claim systems, meaning, once claim is generated.

Claim Tables holds the summary of claim process so far covering various claims dates, latest status of claim, and various claim amounts.

24.1.1 About Insurance Claims Dimension Tables

Insurance Claims Dimension table names and their description are given here.

Logical Dimension Table Name	Dimension Table Description
Dimension Claim Refusal Reason	This table stores the list of all the reasons for which a claim can be refused by the entity.
Dimension Claim Status	This table stores the list of all status codes and descriptions, which are applicable for a claim transaction.
Claim Dimension	This table stores the list of all claims.
Claim Referral Reason Dimension	This table stores different referral reasons for a claim.
Claim Feature Dimension	This tables stores feature details that describes an aspect of a claim for injuries to a person.
Claim Feature Status Dimension	This table stores the status of a claim feature, based on which various calculations changes.
Claim Reopening Reason Type Dimension	This table stores the reopening reason type. Sample values: R1 – Mathematical or computational mistake, R2 – Inaccurate data entry, D0 – Changes in service date, E0 – Change in patient status, etc.

The mapping details for the Insurance Claims Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
233	STG_CLAIM_REFUSAL_REAS_MASTER	Stage Claim Refusal Reason Master	DIM_CLAIM_REFUSAL_REASON	Dimension Claim Refusal Reason
234	STG_CLAIM_STATUS_MASTER	Stage Claim Status Master	DIM_CLAIM_STATUS	Dimension Claim Status
239	STG_CLAIM_DETAILS	Stage Claim Details	DIM_CLAIM	Claim Dimension
242	STG_CLAIM_REFERRAL_REAS_MASTER	Stage Claim Referral Reason	DIM_CLAIM_REFERRAL_REASON	Claim Referral Reason Dimension
599	STG_CLAIM_FEATURE_MASTER	Stage Claim Feature Master	DIM_CLAIM_FEATURE	Claim Feature Dimension
605	STG_CLAIM_FEATURE_STTS_MASTER	Stage Claim Feature Status Master	DIM_CLAIM_FEATURE_STATUS	Claim Feature Status Dimension
648	STG_CLM_REOPEN_RSN_TYPE_MASTER	Stage Claim Reopening Reason Type Master	DIM_CLM_REOPEN_RSN_TYPE	Claim Reopening Reason Type Dimension

24.1.2 About Insurance Claims T2T (Result Table)

Insurance Claims T2T and its description is given here.

T2T Name	T2T Description
T2T_FCT_CLAIM_DETAILS	This T2T stores all the attributes of a policy claims paid until date along with the summary of transactions.
T2T_FCT_CLAIM_CLAIMANT_MAPPING	This T2T stores the details of all the claimants involved in a claim process.
T2T_FCT_CLAIM_TRANSACTIONS	This T2T stores all the transactions related to claims reported with the entity.

The mapping details for the Insurance Claims T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_CLAIM_DETAILS	Stage Claim Details	FCT_CLAIM_DETAILS	Fact Claim Details	T2T_FCT_CLAIM_DETAILS
STG_CLAIM_CLAIMANT	Stage Claim Claimant	FCT_CLAIM_CLAIMANT_MAPPING	Fact Claim Claimant Mapping	T2T_FCT_CLAIM_CLAIMANT_MAPPING
STG_CLAIM_TXNS	Stage Claim Transactions	FCT_CLAIM_TRANSACTION	Fact Claim Transaction	T2T_FCT_CLAIM_TRANSACTIONS

24.2 Deploying Insurance Claims Tables on Hive

All RDBMS related Insurance Claims Dimension and Result tables can also be deployed on Hive (Stage and Results on Hive).

> **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.

24.3 Populating Insurance Claims Dimension Tables

Follow this SCD process to populate data into any Insurance Claims Dimension table:

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

1. To populate data into any Insurance Claims Dimension table, execute the SCD batch. For a detailed procedure, see Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of an Insurance Claims 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.

24.4 Populating Insurance Claims T2T Result Tables

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

> **NOTE** You can also follow this T2T process to populate data into any Hive related Insurance Claims T2T Result table.

1. To populate data into any Insurance Claims T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Insurance Claims Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

25 Actuarial Assumptions Tables

This chapter provides information about Insurance Actuarial Policy Summary, Mortality and Morbidity tables, and various Insurance Lapse Rate Assumptions Result tables in the Oracle Insurance Data Foundation Application and step-by step instructions to use this section.

This chapter includes the following 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.

There are two types of Insurance Actuarial Assumptions tables:

- Financial Assumptions tables
- Demographic Assumptions tables

25.1 Financial Assumptions Tables

Financial assumptions evaluates the projected benefits of Party or organization in a certain plan. These financial assumptions include assumptions about interest rates, and lapse rate assumptions.

- **Insurance Lapse Rate Assumptions**

  Insurance Lapse Rate is the rate at which insurance policy terminates because of the failure in premium payment by the policyholder. The Insurance Lapse Rate Assumptions table stores the insurance lapse rates to be used for assessment of insurance policies. The lapse rate group code binds multiple lapse rates under one heading.

  The purpose of the Insurance Lapse Rate Assumptions table is that the Insurers can set premiums which results in high or low-priced product.

- **Interest Rate Curve (IRC)**

  Interest Rate Curve is a line that plots the interest rates, at a set point in time, of bonds having equal credit quality but differing maturity dates.

Financial Assumptions includes Insurance Lapse Rate Assumptions and Interest Rate Curve (IRC).

25.1.1 About Interest Rate Curve Dimension Table

Interest Rate Curve Dimension table name and its description is given here.

Logical Dimension Table Name	Remarks
Interest Rate Curve Dimension	This table stores the interest rate curve definitions.

The mapping details for the Interest Rate Curve Dimension tables is given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
246	STG_IRCS	Stage Interest Rate Curve	DIM_IRC	Interest Rate Curve Dimension

25.1.2 About Insurance Lapse Rate Assumptions T2T (Result Table)

Insurance Lapse Rate Assumptions T2T and its description is given here.

T2T name	T2T Description
T2T_FCT_LAPSE_RATE_ASSUMPTIONS	This T2T stores the insurance lapse rates, which is used for valuation of insurance policies. This T2T stores the lapse rate group code that binds multiple lapse rates under one heading.

The mapping details for the Insurance Lapse Rate Assumptions T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_INSURANCE_LAPSE_RATES	Stage Insurance Lapse Rates	FCT_LAPSE_RATE_ASSUMPTIONS	Fact Lapse Rate Assumptions	T2T_FCT_LAPSE_RATE_ASSUMPTIONS

25.2 Demographic Assumptions Tables

Demographic assumptions evaluates the projected benefits of all the party or organization in a certain plan. These demographic assumptions include assumptions about mortality, disability, termination of employment, and retirement.

Demographic Assumptions includes:

- Insurance Mortality

  The Insurance Mortality is the amount charged by the Insurer for delivering the guaranteed Sum Assured on premature death of the policyholder.

  The purpose of the Insurance Mortality table is to store the sum charged every year by the Insurer to the policyholder to deliver the life cover.

- Insurance for Morbidity

Morbidity on insurance terms is the insurance coverage provided by the insurer to the policyholder for the specified illness of the policyholder.

25.2.1 About Insurance Mortality and Morbidity T2Ts (Result Tables)

Insurance Mortality and Morbidity T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCT_INSURANCE_MORTALITY	This T2T stores the sum charged every year by the Insurer to the policyholder to deliver the life cover.
T2T_FCT_INSURANCE_MORBIDITY	This T2T stores details of morbidity rates, which is actual age, survival probabilities etc., related to particular mortality tables.

The mapping details for the Insurance Mortality and Morbidity T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_INSURANCE_MORTALITY	Stage Insurance Mortality	FCT_INSURANCE_MORTALITY	Fact Insurance Mortality	T2T_FCT_INSURANCE_MORTALITY
STG_INSURANCE_MORBIDITY	Stage Insurance Morbidity	FCT_INSURANCE_MORBIDITY	Fact Insurance Morbidity	T2T_FCT_INSURANCE_MORBIDITY

25.3 Deploying Actuarial Assumptions Tables on Hive

All RDBMS related Actuarial Assumptions Result tables (Insurance Lapse Rate Assumptions, Mortality, and Morbidity Result tables) can also be deployed on Hive (Stage and Results on Hive).

> **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.

25.4 Populating Actuarial Assumptions Dimension Tables

Follow this SCD process to populate data into any Actuarial Assumptions Dimension table:

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

1. To populate data into any Actuarial Assumptions Dimension table, execute the SCD batch. For a detailed procedure, see Slowly Changing Dimension (SCD) Process.

2. To check the SCD batch execution status of an Actuarial Assumptions 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.

25.5 Populating Actuarial Assumptions T2T Result Tables

Follow this T2T process to populate data into any Actuarial Assumptions T2T Result table (Insurance Lapse Rate Assumptions, Mortality, and Morbidity Result tables):

> **NOTE** You can also follow this T2T process to populate data into any Hive related Actuarial Assumptions T2T Result table.

1. To populate data into any Actuarial Assumptions T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Actuarial Assumptions Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

26 Actuarial Output Tables

This chapter provides information about Actuarial Outputs such as Actuarial Cash Flows, Calculations, and Result Areas in the Oracle Insurance Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

26.1 About Actuarial Cash Flows, Calculations, and Result Areas Summary Tables

Actuarial Cash Flows, Calculations, and Result Areas Summary tables store attributes pertaining to insurance cash flows, calculations, and Result Areas related data on an 'as-is' basis received from the source system. The data is populated in the following Dimension tables through Seeded Data. Dimension table names and their description are given here.

Dimension Table Name	Logical Dimension Table Name	Remarks
DIM_ONEROUS_CLASSIFICATION	Onerous Classification Dimension	This is the Seeded dimension table for the onerous classification of the contract at Homogenous risk level. Sample values: ONEROUS, POSO, REMANINGCONTRACTS.
DIM_INS_ACCTG_VAL_APPROACH	Insurance Accounting Valuation Approach Dimension	This is the Seeded dimension table for Insurance Accounting Valuation Approach. Sample values: BBA, VFA, PAA.
DIM_INS_RISK_EXPIRY_STATUS	Insurance Risk Expiry Status Dimension	This is the Seeded dimension table for Insurance Risk Expiry Status. Samples values: EXPIRED, UNEXPIRED.

The following tables are direct results populated from processing or actuarial engines:

Fact Table Name	Logical Fact Table Name	Fact Table Description
FCT_ACTUARIAL_POLICY_SUMMARY	Fact Actuarial Policy Summary	This table stores the cash flow of all the individual policy in an aggregated level for

Fact Table Name	Logical Fact Table Name	Fact Table Description
		reporting the summary. These are the resulting tables, which contains calculations.
FCT_INS_HRG_CASH_FLOW	Fact Insurance Homogeneous Risk Group Cash Flow	This table stores the cash flow estimates at Homogeneous Risk Group level for insurance policy group.
FCT_INS_HRG_CASH_FLOW_SUMMARY	Fact Insurance Homogeneous Risk Group Cash Flow Summary	This table stores the cash flow estimates at Homogeneous Risk Group summary level of Homogenous Risk Group.

26.1.1 About Financial Element Dimension Table

Insurance Financial Element Dimension table name and its description is given here.

Logical Dimension Table Name	Remarks
DIM_INS_FINANCIAL_ELEMENT	This table stores Insurance financial elements details. Sample values: Outflow Sum Insured, Outflow Expected Benefits, Outflow Expected Expense, etc.

The mapping details for the Insurance Financial Element Dimension table is given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
487	STG_INS_FINANCIAL_ELMNT_MASTER	Stage Insurance Financial Element Master	DIM_INS_FINANCIAL_ELEMENT	Insurance Financial Element Dimension

26.1.2 About Insurance Scenario Dimension Table

Insurance Financial Element Dimension table name and its description is given here.

Logical Dimension Table Name	Remarks
DIM_INSURANCE_SCENARIO	This table stores different scenario or assumption that can affect the calculation of cash flow. Sample values: Adverse, Severely Adverse, etc.

The mapping details for the Insurance Financial Element Dimension table is given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
486	STG_INS_SCENARIO_MASTER	Stage Insurance Scenario Master	DIM_INSURANCE_SCENARIO	Insurance Scenario Dimension

26.1.3 About Insurance Vintage Dimension Table

Insurance Vintage Dimension table name and its description is given here.

Logical Dimension Table Name	Remarks
DIM_INSURANCE_VINTAGE	This table stores Insurance Vintage associated with each process. Vintage refers to period of origination or first premium paid or major change on contract terms and conditions. This is a custom defined table. The end user can store monthly, yearly, half-yearly, quarterly values or as required. Sample values: Q1YYYY, Q2YYYY, M1YYYY, etc.

The mapping details for the Insurance Financial Element Dimension table is given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
485	STG_INSURANCE_VINTAGE_MASTER	Stage Insurance Vintage Master	DIM_INSURANCE_VINTAGE	Insurance Vintage Dimension

26.1.4 About Fact Insurance Policy Cash Flow T2T (Result Table)

Fact Insurance Policy Cash Flow T2T and its description is given here.

T2T Name	T2T Description
T2T_FCT_INS_POLICY_CASH_FLOW	This T2T stores the cash flow estimates at a policy level.
T2T_FCT_INS_ASSUM_POLICY_CASH_FLOW	This T2T stores the cash flow estimates at a policy level. It captures the details for assumed policies.

The mapping details for the Fact Insurance Policy Cash Flow T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_INS_POLICY_CASH_FLOW	Stage Insurance Policy Cash Flow	FCT_INS_POLICY_CASH_FLOW	Fact Insurance Policy Cash Flow	T2T_FCT_INS_POLICY_CASH_FLOW
STG_INS_ASSUM_POLICY_CASH_FLOW	Stage Insurance Assumed Policy Cash Flow	FCT_INS_ASSUM_POLICY_CASH_FLOW	Fact Insurance Assumed Policy Cash Flow	T2T_FCT_INS_ASSUM_POLICY_CASH_FLOW

26.2 Deploying Actuarial Cash Flows, Calculations, and Result Areas Summary Tables on Hive

All RDBMS related Actuarial Cash Flows, Calculations, and Result Areas Summary Tables Dimension and Result tables can also be deployed on Hive (Stage and Results on Hive).

> **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.

26.3 Populating Actuarial Cash Flows, Calculations, and Result Areas Summary Dimension Tables

Follow this SCD process to populate data into any Actuarial Cash Flows, Calculations, and Result Areas Summary Tables Dimension table:

> **NOTE** You can also follow this SCD process to populate data into any Hive related Actuarial Cash Flows, Calculations, and Result Areas Summary Dimension table.

1. To populate data into any Actuarial Cash Flows, Calculations, and Result Areas Summary Tables Dimension table, execute the SCD batch. For a detailed procedure, see Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of any Actuarial Cash Flows, Calculations, and Result Areas Summary Tables 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.

26.4 Populating Actuarial Cash Flows, Calculations, and Result Areas Summary T2T Result Tables

Follow this T2T process to populate data into any Actuarial Cash Flows, Calculations, and Result Areas Summary T2T Result table:

NOTE You can also follow this T2T process to populate data into any Hive related Actuarial Cash Flows, Calculations, and Result Areas Summary T2T Result table.

1. To populate data into any Actuarial Cash Flows, Calculations, and Result Areas Summary T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Actuarial Cash Flows, Calculations, and Result Areas Summary Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

27 Common Customer Summary Tables

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

This chapter includes the following topics:

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

27.1 About Dimension Tables that Loads Common Customer Summary T2T

Common Customer Summary Dimension table names and their description are given here.

Logical Dimension Table Name	Dimension Table Description
Channel Dimension	This table stores the master list of all unique codes that denote channels through which customers can be acquired.
Education Dimension	This table stores the details of education master information of a customer.
Geography Dimension	This table stores the distinct list of all geographical locations, where any of the transaction channels of the Insurance Company are located.
Industry Dimension	This table stores industry information.
Account Management Dimension	This table stores the organization hierarchy details across the management.
Migration Reasons Dimension	This table stores reasons for deviation.
Vintage Dimension	This table stores vintage definitions.
Customer Dimension	This table stores the list of the customers and counterparties, and their attributes of an organization
Profession Dimension	This table stores the master list of all customer professions.
Marital Status Dimension	This table stores customer marital status details.
Customer Type Dimension	This table stores the master list of customer types.
Credit Rating Dimension	This table stores credit rating information.

The mapping details for the Common Customer Summary Dimension tables are given here.

Map Reference Number	Source Table Name	Logical Stage Table Name	Dimension Table Name	Logical Dimension Table Name
21	STG_SALES_CHANNEL_MASTER	Stage Sales Channel Master	DIM_CHANNEL	Channel Dimension
40	STG_CUST_EDUCATION_MASTER	Stage Customer Education Master	DIM_EDUCATION	Education Dimension
47	STG_GEOGRAPHY_MASTER	Stage Geography Master	DIM_GEOGRAPHY	Geography Dimension
51	STG_INDUSTRY_MASTER	Stage Industry Master	DIM_INDUSTRY	Industry Dimension
62	STG_ACCOUNT_MGMT_MASTER	Stage Account Management Master	DIM_MANAGEMENT	Account Management Dimension
68	STG_MIGRATION_REASON_MASTER	Stage Migration Reason Master	DIM_MIGRATION_REASONS	Migration Reasons Dimension
116	STG_VINTAGE_MASTER	Stage Vintage Master	DIM_VINTAGE	Vintage Dimension
335	VW_STG_PARTY_MASTER_CUST	Stage Party Master Customer View	DIM_CUSTOMER	Customer Dimension
344	STG_PROFESSION_MASTER	Stage Profession Master	DIM_PROFESSION	Profession Dimension
389	STG_MARITAL_STATUS_MASTER	Stage Marital Status Master	DIM_MARITAL_STATUS	Marital Status Dimension
418	STG_PARTY_TYPE_MASTER	Stage Party Type Master	DIM_CUSTOMER_TYPE	Customer Type Dimension
466	STG_CREDIT_RATING_MASTER	Stage Credit Rating Master	DIM_CREDIT_RATING	Credit Rating Dimension

27.2 About T2Ts (Result Tables) that Loads Common Customer Summary T2T

T2T for Common Customer Summary and its description is given here.

T2T Name	T2T Description
T2T_FCT_PARTY_RATING_DETAILS	This T2T stores the rating details of the Customer or Counterparty or Guarantor etc.
T2T_FCT_PARTY_FINANCIALS	This T2T stores the financial information (Balance Sheet, Profit and Loss statement, and Ratios) of the parties like Customer and Guarantor.

The mapping details for T2Ts to load Common Customer Summary is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_PARTY_RATING_DETAILS	Stage Party Rating Details	FCT_PARTY_RATING_DETAILS	Fact Party Rating Details	T2T_FCT_PARTY_RATING_DETAILS
STG_PARTY_FINANCIALS	Stage Party Financials	FCT_PARTY_FINANCIALS	Fact Party Financials	T2T_FCT_PARTY_FINANCIALS

27.3 About Common Customer Summary T2T (Result Table)

Common Customer Summary T2T and its description is given here.

T2T Name	T2T Description
T2T_FCT_COMMON_CUSTOMER	This T2T stores different attributes pertaining to raw customer data received from the source system.

The mapping details for the Common Customer Summary T2T is given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_CUSTOMER_DETAILS	Stage Customer Details	FCT_COMMON_CUSTOMER_SUMMARY	Fact Common Customer Summary	T2T_FCT_COMMON_CUSTOMER

27.4 Deploying Common Customer Summary Tables on Hive

All RDBMS related Common Customer Summary Result table can also be deployed on Hive (Stage and Results on Hive).

> **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.

27.5 Populating Common Customer Summary T2T Result Tables

Follow these steps to populate data in Common Customer Summary T2T Result tables:

1. Prerequisites.
2. Populating Common Customer Summary T2T Result Tables.

27.5.1 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.

27.5.1.1 Populating Dimension Tables to Load Common Customer Summary T2T

For Dimension mapping details for this section, see About Dimension Tables that Loads Common Customer Summary T2T. 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 Common Customer Summary Dimension table.

1. To populate data into any Dimension table (required to load Common Customer Summary T2T), execute the SCD batch. For a detailed procedure, see Slowly Changing Dimension (SCD) Process.
2. To check the SCD batch execution status of a Dimension table (required to load Common Customer Summary T2T), 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.

27.5.1.2 Populating T2Ts to Load Common Customer Summary T2T

For T2T mapping details for this section, see About T2Ts (Result Tables) that Loads Common Customer Summary T2T. Follow this T2T process to populate data into any T2T that is required to load data into Common Customer Summary T2T:

1. To populate data into any Common Customer Summary T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Common Customer Summary Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

27.5.1.3 Populating Other Tables to Load Common Customer Summary T2T

Load these tables with data:

- DIM_BANDS

  To follow Dimension loading process, see Populating Dimension Tables to Load Common Customer Summary T2T.

- DIM_GENDER

  To follow 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 Common Policy Summary T2T Tables.

  To follow T2T process, see Populating T2Ts to Load Common Customer Summary T2T.

27.5.2 Populating Common Customer Summary T2T Result Tables

Follow T2T process 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).

To follow the T2T process, see Populating T2Ts to Load Common Customer Summary T2T.

28 Credit Score Model and Probability of Default Model Tables

This chapter provides information about Credit Score Model and Probability of Default Model tables in the Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

28.1 About Credit Score Model and Probability of Default Model T2Ts (Result Tables)

Credit Score Model and Probability of Default Model T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCT_ACCT_CREDIT_SCORE_DETAILS	This T2T stores the details of the credit score of account throughout its lifetime.
T2T_FCT_SERV_ACCT_CREDIT_SCORE_DTL	This T2T stores the details of the credit score of serviced account throughout its lifetime.
T2T_FCT_PARTY_PD_DETAILS	This T2T stores the probability of default values as of given date for all relevant Party.
T2T_FCT_INSTRUMENT_PD_DETAILS	This T2T stores the probability of default values as of given date for all relevant instruments.

The mapping details for the Credit Score Model and Probability of Default Model T2Ts are given here.

Source Table Name	Logical Stage Table Name	Target Table Name	Logical Fact Table Name	T2T Definition Name
STG_ACCT_CREDIT_SCORE_DETAILS	Stage Account Credit Score details	FCT_ACCT_CREDIT_SCORE_DETAILS	Fact Account Credit Score details	T2T_FCT_ACCT_CREDIT_SCORE_DETAILS
STG_SERV_ACCT_CREDIT_SCORE_DTL	Stage Service Account Credit Score Detail	FCT_SERV_ACCT_CREDIT_SCORE_DTL	Fact Service Account Credit Score Detail	T2T_FCT_SERV_ACCT_CREDIT_SCORE_DTL
STG_PARTY_PD_DETAILS	Stage Party Probability of Default Details	FCT_PARTY_PD_DETAILS	Fact Party Probability of Default Details	T2T_FCT_PARTY_PD_DETAILS
STG_INSTRUMENT_PD_DETAILS	Stage Instrument Probability of Default Details	FCT_INSTRUMENT_PD_DETAILS	Fact Instrument Probability of Default Details	T2T_FCT_INSTRUMENT_PD_DETAILS

28.1.1 Staging Data Expectation for Credit Score Model

Stage Account Credit Score Details and Stage Service Account Credit Score Details tables expect data incrementally and not in snapshot mode.

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

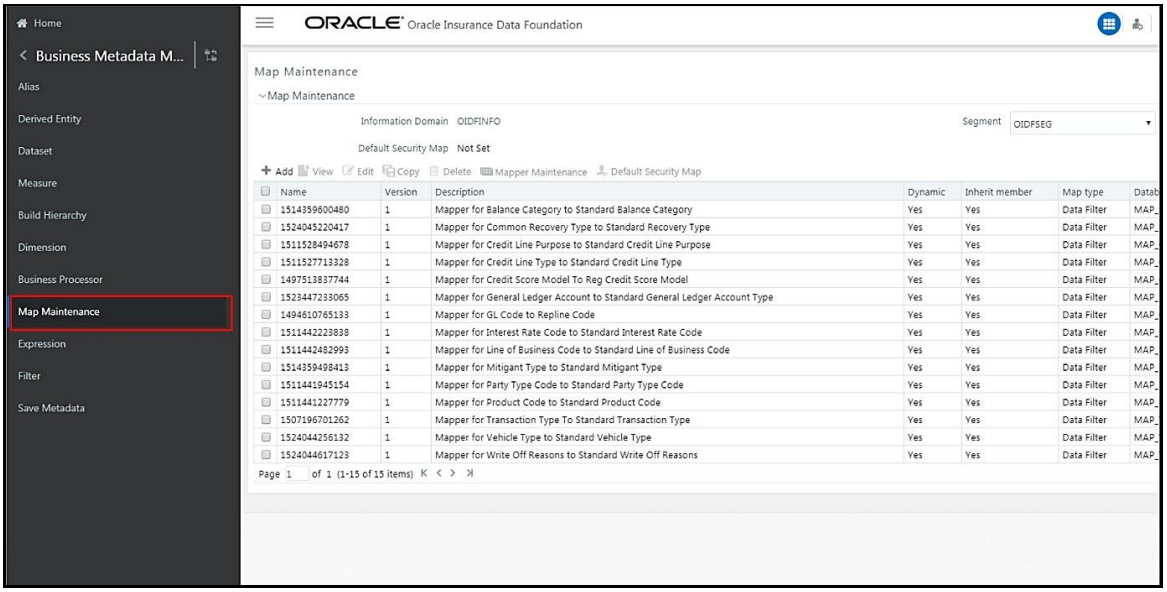
28.2 About Mapper for Credit Score Model to Regulatory Credit Score Model

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 is 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 Mapper Maintenance Component of OFSAAI.

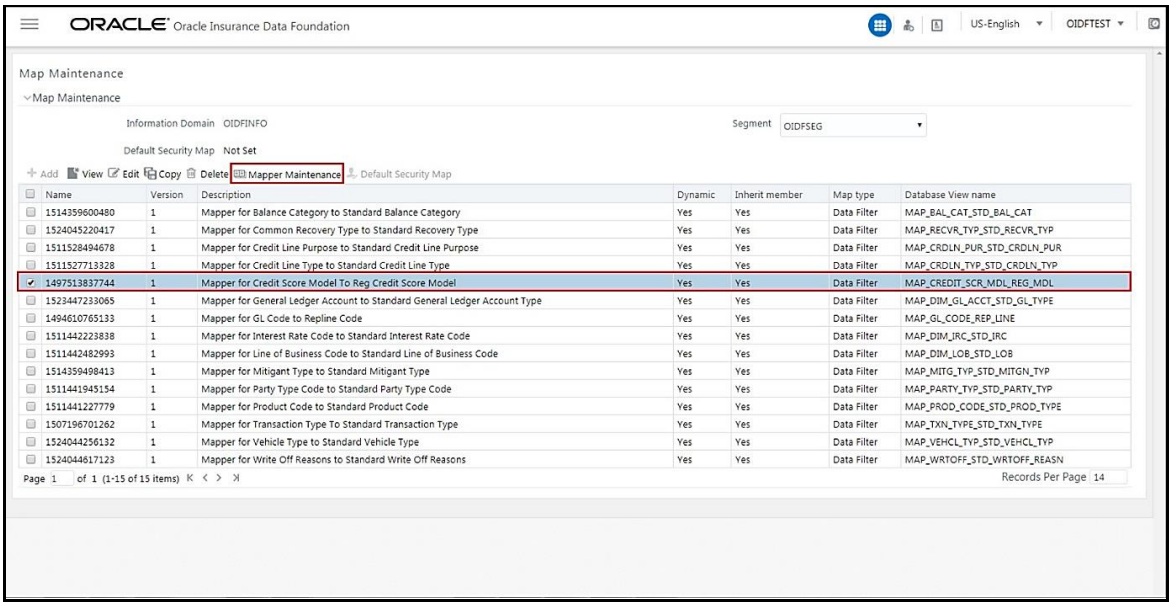
28.3 Maintenance of Mapper for Credit Score Model to Regulatory Credit Score Model

To maintain Mapper for Credit Score Model to Regulatory Credit Score Model, perform these steps.

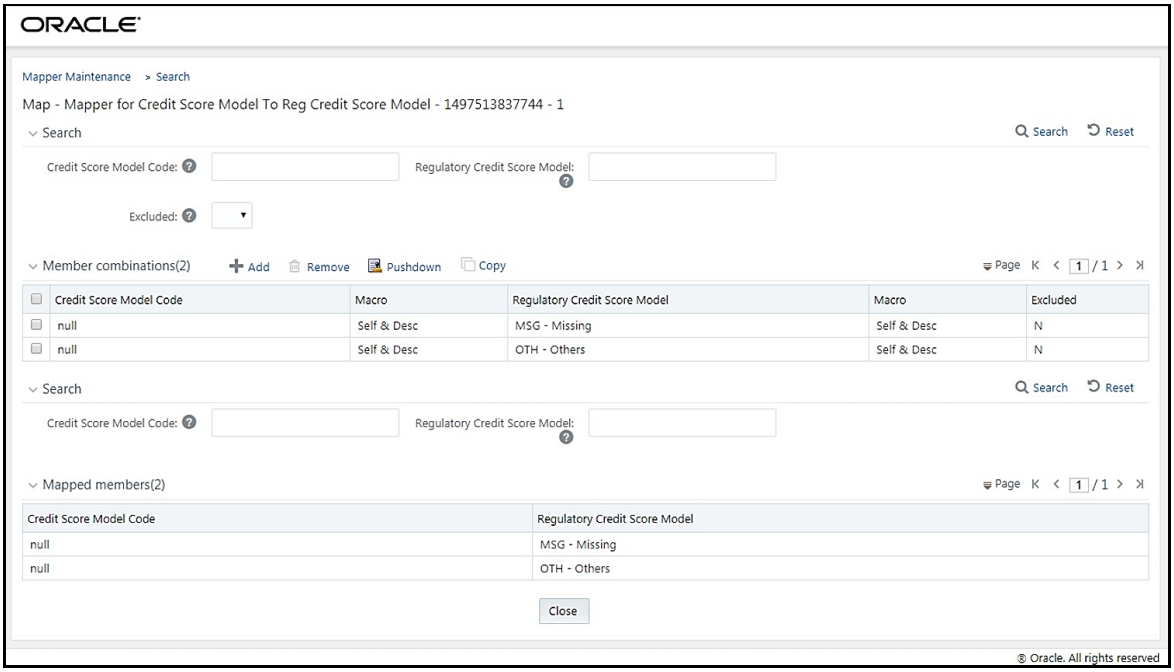
1. Go to OFSAAI > Oracle Insurance Data Foundation > Unified Analytical Metadata > Business Metadata Management > Map Maintenance.



2. The Map Maintenance page appears. Select Mapper for Credit Score Model to Regulatory Credit Score Model. Click Mapper Maintenance icon.



3. The Mapper Maintenance window appears. 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.



28.3.1 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

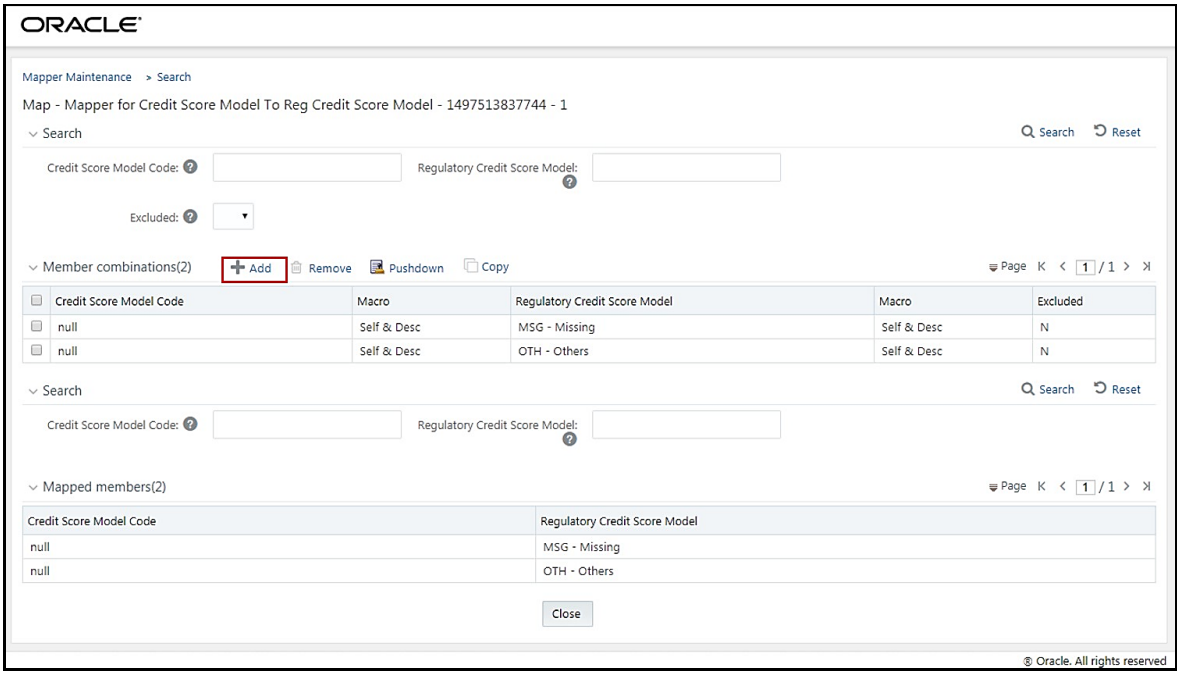
28.3.2 Possible Mapping Combinations

One Credit Score Model in source can be mapped only to one Regulatory Credit Score Model. One to Many or Many to Many mapping will lead to error in T2T as the records 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.

28.3.2.1 One-to-One Mapping

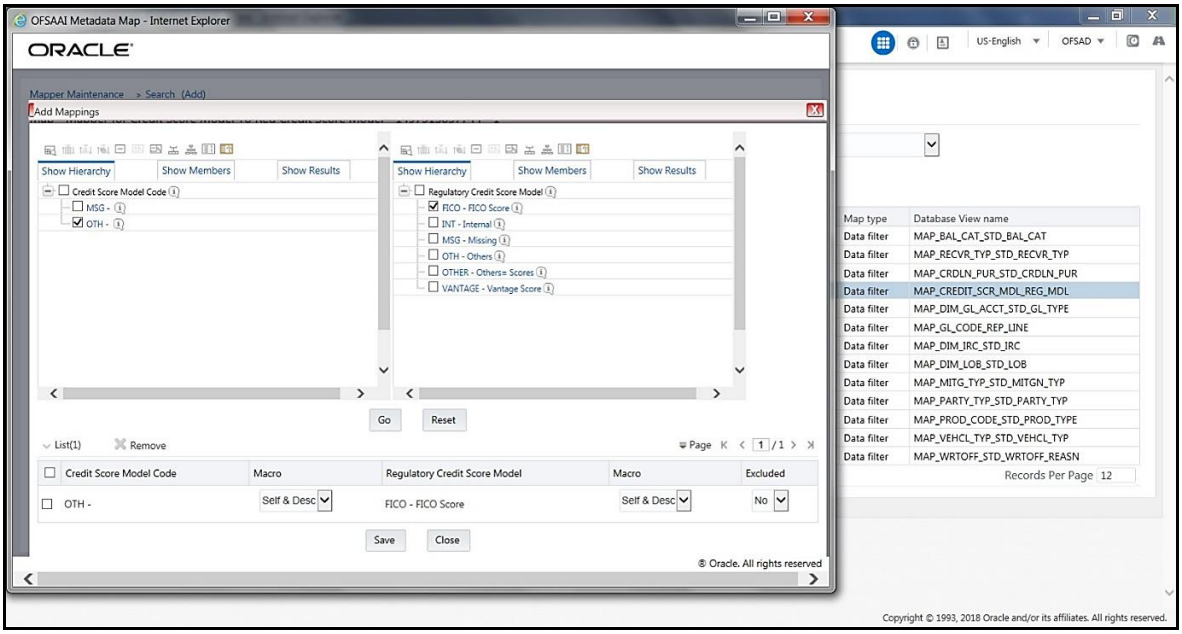
One Credit Score Model can be mapped to one Regulatory Credit Score Model using the Mapper Maintenance screen. Here, you need to select one Credit Score Model and one Regulatory Credit Score Model. This means the corresponding data in Stage Account Credit Score Details Credit Score Model column for the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

1. In the Mapper Maintenance window, click Add.



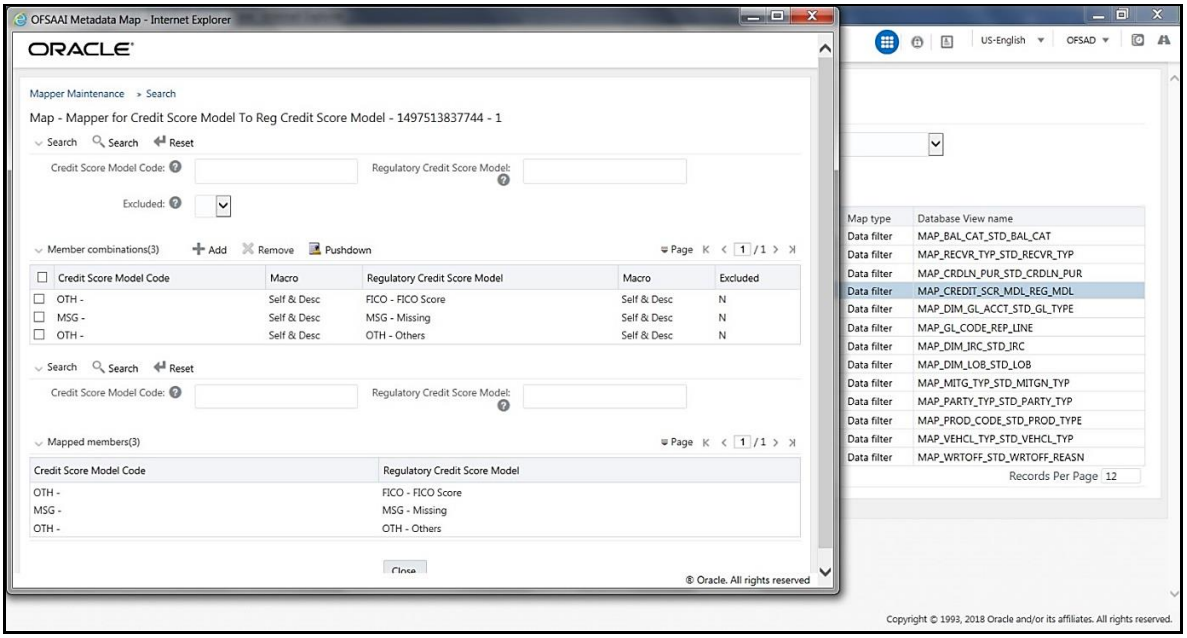
2. The Add Mappings pop-up window is displayed.

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



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

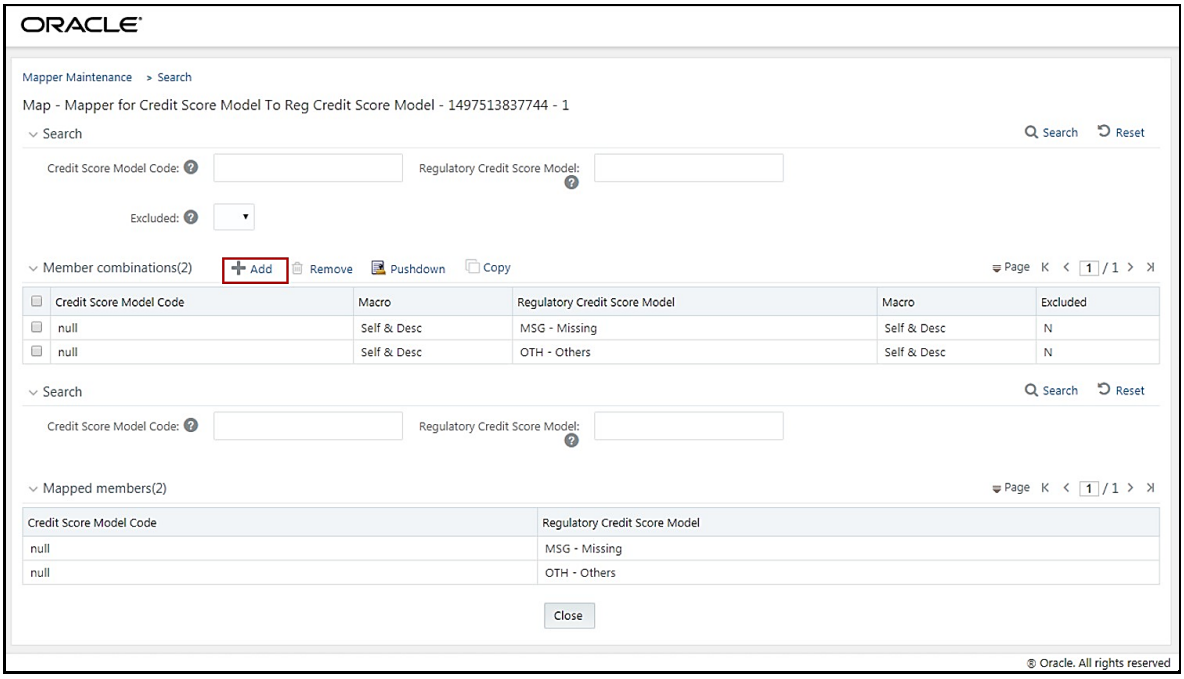
4. The Mapped Members are displayed in the Mapper Maintenance Screen.



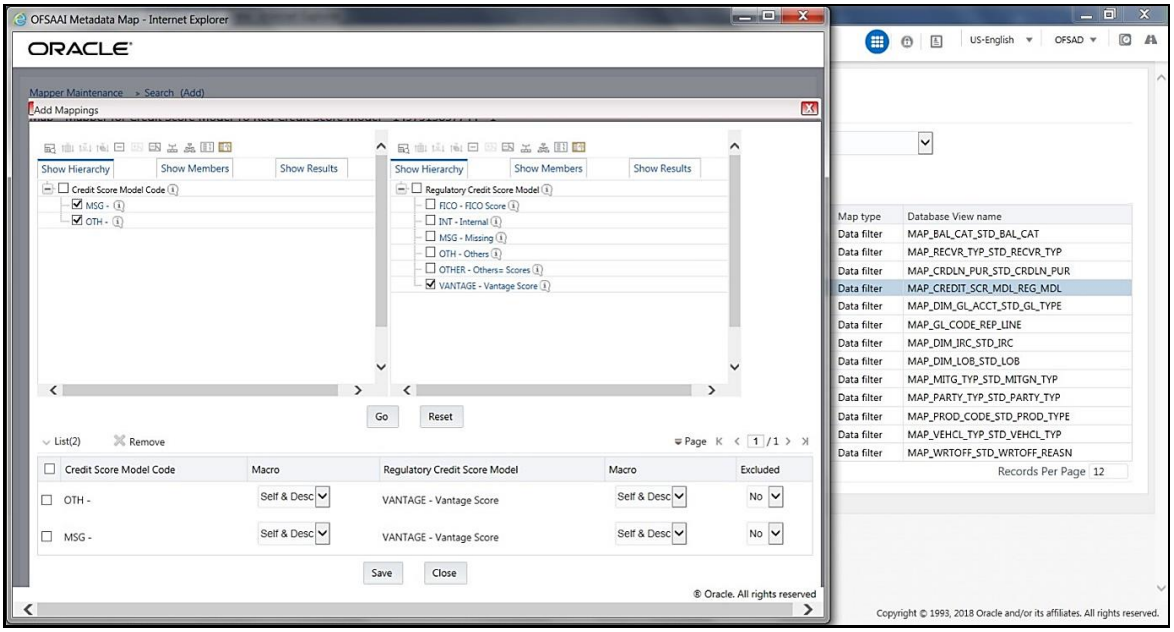
28.3.2.2 Many-to-One Mapping

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

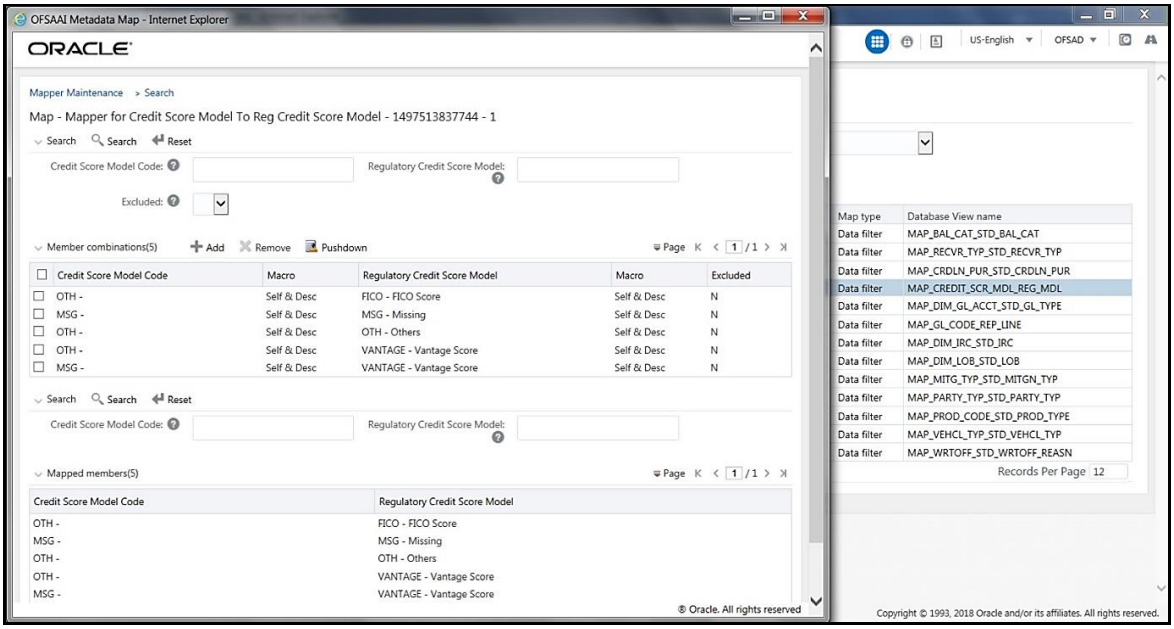
1. In the Mapper Maintenance window, click Add.



2. The Add Mappings pop-up window is displayed. 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.



3. An acknowledgement is displayed. Click Yes to confirm.
4. The Mapped Members are displayed in the Mapper Maintenance Screen.



28.4 Loading Mapper Maintenance through Backend

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

Credit Score Model Code (values from DIM_REG_CREDIT_SCORE_MODEL.V_REG_CREDIT_SCORE_MODEL_CODE).

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

```
SELECT * FROM MAP_CREDIT_SCR_MDL_REG_MDL;
```

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

28.5 Deploying Credit Score Model and Probability of Default Model Tables on Hive

All RDBMS related Credit Score Model and Probability of Default Model Result tables can also be deployed on Hive (Stage and Results on Hive).

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.

28.6 Populating Credit Score Model and Probability of Default Model T2T Result Tables

Follow this T2T process to populate data into any Credit Score Model and Probability of Default Model T2T Result table:

NOTE You can also follow this T2T process to populate data into any Hive related Credit Score Model and Probability of Default Model T2T Result table.

1. To populate data into any Credit Score Model and Probability of Default Model T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Credit Score Model and Probability of Default Model Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages

29 Other Miscellaneous Tables

This chapter provides information about populating various other Results tables in the Oracle Insurance Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topics:

Other Results tables store Party details related to fixed assets, spend obligations and assets sold. These tables are used for the purpose of financial report of the party or organization.

29.1 About Other Result T2Ts (Result T2Ts)

Other Result T2Ts and their description are given here.

T2T Name	T2T Description
T2T_FCT_ASSETS_SOLD	This T2T stores the data of assets sold over a period.
T2T_FCT_FIXED_ASSETS	This T2T stores the details pertaining to fixed assets such as real estate.
T2T_FCT_SPEND_OBLIGATIONS	This T2T stores the contract codes for purchase obligations such as long duration IT contract signed by Insurance Company or other spend obligations such as lease contracts.

The mapping details for the Other Result T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_ASSETS_SOLD	Stage Assets Sold	FCT_ASSETS_SOLD	Fact Assets Sold	T2T_FCT_ASSETS_SOLD
STG_FIXED_ASSETS_DETAILS	Stage Fixed Assets Details	FCT_FIXED_ASSETS	Fact Fixed Assets	T2T_FCT_FIXED_ASSETS
STG_SPEND_OBLIGATIONS	Stage Spend Obligations	FCT_SPEND_OBLIGATIONS	Fact Spend Obligations	T2T_FCT_SPEND_OBLIGATIONS

29.2 Deploying Other Result Tables on Hive

All RDBMS related Other Result tables can also be deployed on Hive (Stage and Results on Hive).

> **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.

29.3 Populating Other T2T Result Tables

Follow this T2T process to populate data into Other T2T Result tables:

> **NOTE** You can also follow this T2T process to populate data into any Hive related Other T2T Result table.

1. To populate data into any Other T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Other Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

30 Transaction Summary Tables

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

This chapter includes the following 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.

30.1 About Transaction Summary T2T (Result Table)

Transaction Summary T2Ts and their description are given here.

T2T Name	T2T Description
T2T_STG_ANNUITY_TXNS_FTS	This T2T stores the details of annuity transactions.
T2T_STG_BORROWINGS_TXNS_FTS	This T2T stores the details of borrowing transactions.
T2T_STG_CREDIT_DERIVATIVES_TXNS_FTS	This T2T stores the details of credit derivatives transactions.
T2T_STG_FOREX_TXNS_FTS	This T2T stores the details of Forex account transactions.
T2T_STG_FUTURES_TXNS_FTS	This T2T stores the details of futures contract transactions.
T2T_STG_GUARANTEES_TXNS_FTS	This T2T stores the details of guarantees transactions.
T2T_STG_INVESTMENT_TXNS_FTS	This T2T stores the details of investment transactions.
T2T_STG_LOAN_CONTRACT_TXNS_FTS	This T2T stores transactions that occurred on loan contracts.
T2T_STG_MM_TXNS_FTS	This T2T stores details of money market transactions.
T2T_STG_MUTUAL_FUNDS_TXNS_FTS	This T2T stores details of mutual fund transactions.
T2T_STG_OPTION_CONTRACTS_TXNS_FTS	This T2T stores details of option contracts transactions.
T2T_STG_RETIREMENT_ACCOUNTS_TXNS_FTS	This T2T stores the details of transactions that occurred on Retirement accounts.
T2T_STG_SWAP_ACCOUNT_TXNS_FTS	This T2T stores the details of swap account transactions.
T2T_STG_TRUSTS_TXNS_FTS	This T2T stores the details of transactions that occurred on trust accounts.
T2T_STG_COMMODITIES_TXNS_FTS	This T2T stores the details of commodity transactions.
T2T_STG_CUSTODIAN_ACCOUNT_TXNS_FTS	This T2T stores the details of transactions for a custodian account.
T2T_STG_REPO_TRANSACTIONS_FTS	This T2T stores the details of repurchase and reverse repurchase transactions.

The mapping details for the Transaction Summary T2Ts are given here.

Source Table Name	Logical Stage Table Name	Fact Table Name	Logical Fact Table Name	T2T Name
STG_ANNUITY_TXNS	Stage Annuity Transactions	FCT_TRANSACTION_SUMMARY	Fact Annuity Transactions	T2T_STG_ANNUITY_TXNS_FTS
STG_BORROWINGS_TXNS	Stage Borrowings Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_BORROWINGS_TXNS_FTS
STG_CREDIT_DERIVATIVES_TXNS	Stage Credit Derivatives Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_CREDIT_DERIVATIVES_TXNS_FTS
STG_FOREX_TXNS	Stage Forex Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_FOREX_TXNS_FTS
STG_FUTURES_TXNS	Stage Futures Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_FUTURES_TXNS_FTS
STG_GUARANTEES_TXNS	Stage Guarantees Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_GUARANTEES_TXNS_FTS
STG_INVESTMENT_TXNS	Stage Investment Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_INVESTMENT_TXNS_FTS
STG_LOAN_CONTRACT_TXNS	Stage Loan Contract Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_LOAN_CONTRACT_TXNS_FTS
STG_MM_TXNS	Stage Money Market Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_MM_TXNS_FTS
STG_MUTUAL_FUNDS_TXNS	Stage Mutual Funds Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_MUTUAL_FUNDS_TXNS_FTS
STG_OPTION_CONTRACTS_TXNS	Stage Option Contracts Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_OPTION_CONTRACTS_TXNS_FTS
STG_RETIREMENT_ACCOUNTS_TXNS	Stage Retirement Accounts Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_RETIREMENT_ACCOUNTS_TXNS_FTS
STG_SWAP_ACCOUNT_TXNS	Stage Swap Account Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_SWAP_ACCOUNT_TXNS_FTS
STG_TRUSTS_TXNS	Stage Trusts Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_TRUSTS_TXNS_FTS
STG_COMMODITIES_TXNS	Stage Commodities Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_COMMODITIES_TXNS_FTS
STG_CUSTODIAN_ACCOUNT_TXNS	Stage Custodian Account Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_CUSTODIAN_ACCOUNT_TXNS_FTS
STG_REPO_TRANSACTIONS	Stage Reporting Transactions	FCT_TRANSACTION_SUMMARY	Fact Transaction Summary	T2T_STG_REPO_TRANSACTIONS_FTS

30.2 Deploying Transaction Summary Result Tables on Hive

All RDBMS related Transaction Summary Result tables can also be deployed on Hive (Stage and Results on Hive).

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.

30.3 Populating Transaction Summary T2T Result Tables

Follow this T2T process to populate data into Transaction Summary T2T Result tables:

NOTE You can also follow this T2T process to populate data into any Hive related Transaction Summary T2T Result table.

1. To populate data into any Transaction Summary T2T Result table, execute the batch for T2T. For a detailed procedure, see:
   a. Prerequisites for loading T2T.
   b. Set the Run Parameters.
   c. Save the Run Parameters.
   d. Execute the Batch for T2T.
2. To check the T2T batch execution status of the Transaction Summary Result tables, follow the procedure Check the Batch Execution Status of T2T.
3. To verify log files, and check the error messages (if any), follow the procedure Verify T2T Log Files and Check Error Messages.

31 Big Data in OIDF

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

The following topics are included:

31.1 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.

31.2 OIDF Big Data Architecture

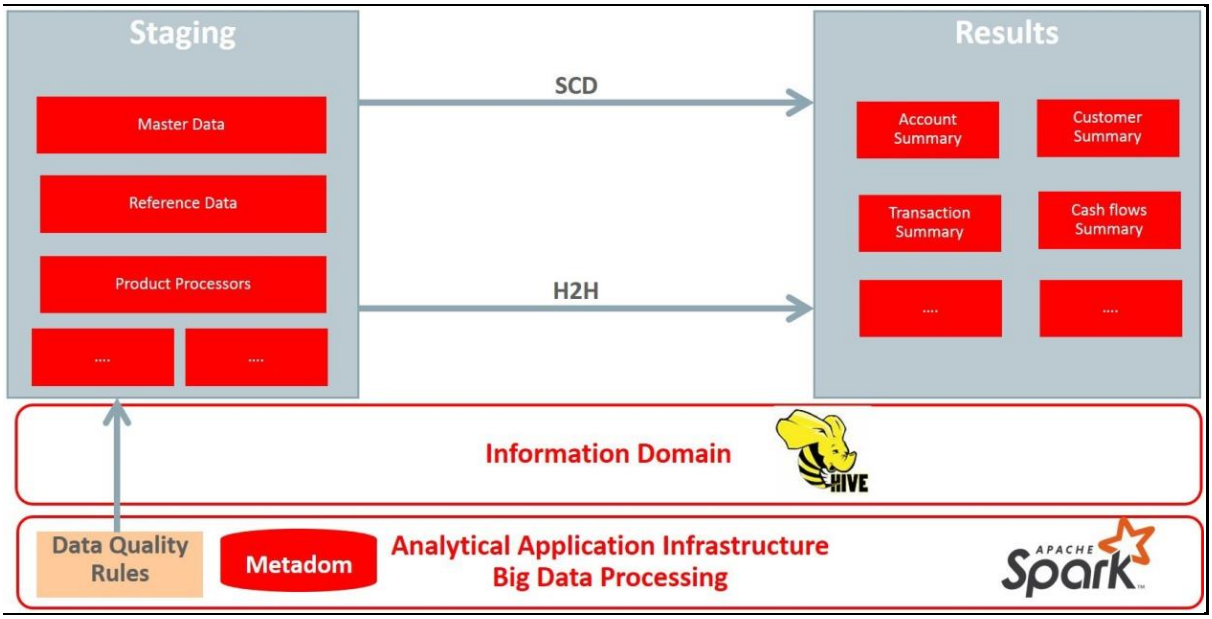
In Big Data processing, OIDF provides two types of deployment processes for Hive support. They are:

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

31.2.1 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 resides in a RDBMS schema known as Metadom. Refer to the below architecture diagram for the representation of this deployment process.

OIDF Big Data Architecture - Staging and Results on Hive



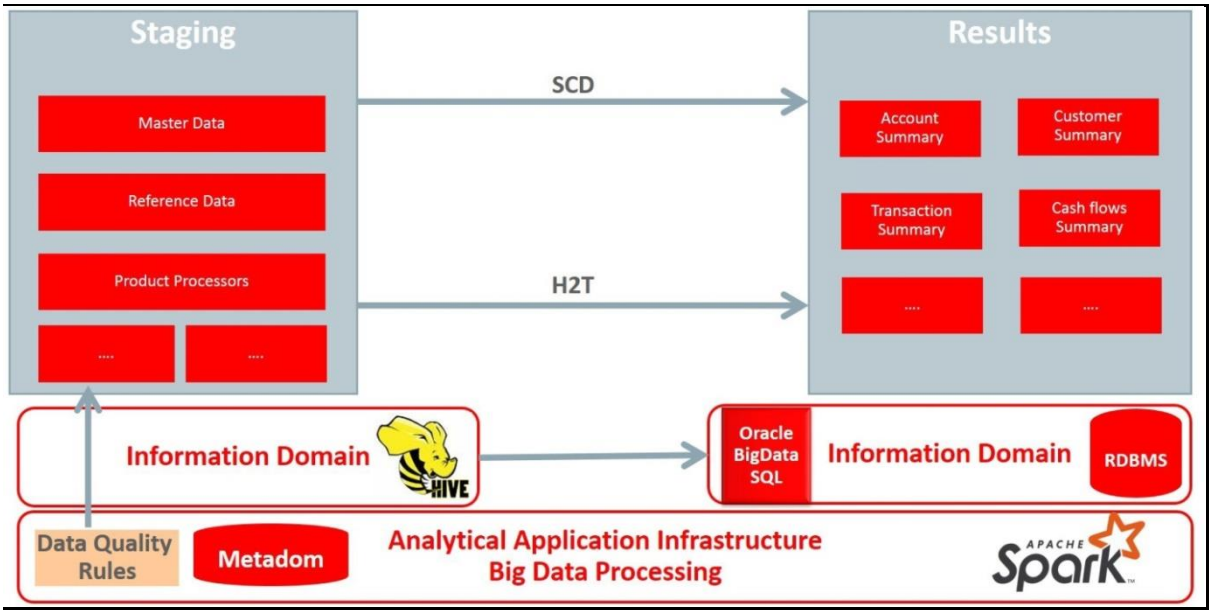
For the list of supported OIDF Hive Metadata definitions, refer to Run Chart, SCD Metadata, and Technical Metadata (Staging Source) documents at My Oracle Support.

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

31.2.2 About Staging on Hive and Results on RDBMS

In the Staging on Hive and Results on RDBMS deployment process, the Staging occurs on Hive, known as Hive Datadom, and the Results model elements are a part of RDBMS, known as RDBMS Datadom. In this deployment process, an additional software is required, which is Oracle BigData SQL (For more information, see Oracle Insurance Data Foundation Installation Guide Release 8.0.7.0.0). Oracle BigData SQL software resides on Hive and RDBMS Datadoms, and enables the user to create a link to Hive Stage tables as external tables in RDBMS. This software manages the data representation of the Hive tables in RDBMS Datadom and can be accessed as external table objects. As a result, enables the use of RDBMS infrastructure components of SCD, T2T, and DQs. For more information on External Tables, see the section Verifying Oracle's External Tables Utility in the Oracle Insurance Data Foundation Installation Guide Release 8.0.7.0.0. Refer to the below architecture diagram for the representation of this deployment process.

OIDF Big Data Architecture - Staging on Hive and Results on RDBMS



31.3 Staging and Results on Hive

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

31.3.1 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
```

31.3.2 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 AM/HM screens in OIDF (RDBMS). However, development of the underlying functionality for AM/HM feature in 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 AM/HM screens in OIDF (RDBMS). However, development of the underlying functionality for AM/HM feature in 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 a 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.

31.3.3 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).

31.3.3.1 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 8.0.7.0.0.

31.3.3.2 Run Enabled T2Ts

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

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

31.3.3.3 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 8.0.7.0.0.

31.3.4 List of Unsupported T2Ts

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

- T2T_FCT_PARTY_FAMILY_MEDICAL_DA
- T2T_FCT_PARTY_MDCAL_CONDITN_DTL
- T2T_FCT_POLICY_TRANSACTIONS_HLD
- T2T_FCT_POLICY_TRANSACTIONS_ISS
- T2T_FPT_STG_RETIREMENT_ACCOUNTS_TXNS
- T2T_FPT_STG_PROP_CASU_POLICY_TXNS
- T2T_FPT_STG_LIFE_INS_POLICY
- T2T_FPT_STG_HEALTH_INS_TXNS

- T2T_FPT_STG_ANNUITY_TXNS

31.3.5 Executing Run through Run Management for Hive

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

32 Metadata Browser

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

This chapter includes the following topics:

32.1 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.

32.2 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

32.3 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.

32.4 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.

33 Recommendation for Backdated Run

This chapter 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.

This chapter includes the following topics:

33.1 Overview of Backdated Run Execution

Backdated Run Execution is similar to any regular Run Execution in Data Foundation. You must reload or correct the data, which must be loaded for the given prior date. Refer to Oracle Insurance Data Foundation Runchart and execute the relevant Batches/Runs for the required prior date.

33.2 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.

33.3 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 Data Foundation 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.

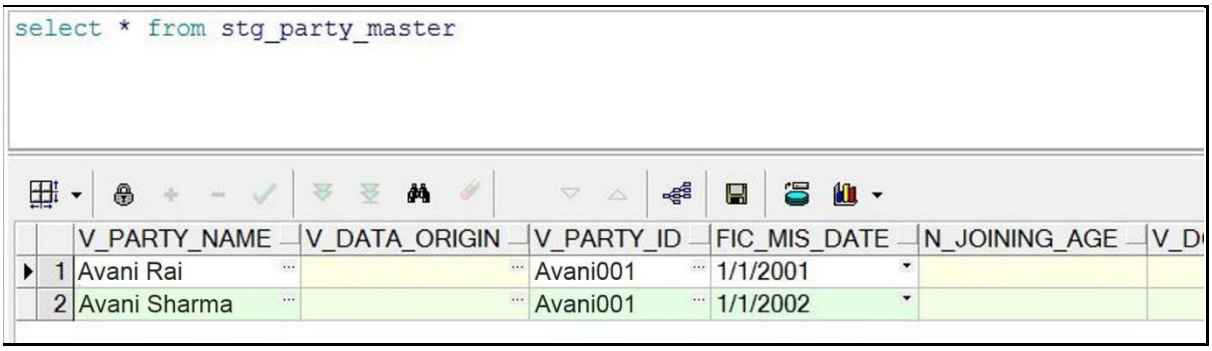
> **NOTE** It is recommended to use this only when there is Backdated Run requirement. Any normal sequential days execution can be performed using out-of-the-box T2Ts and Runs as it results in better performance.

33.4 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).

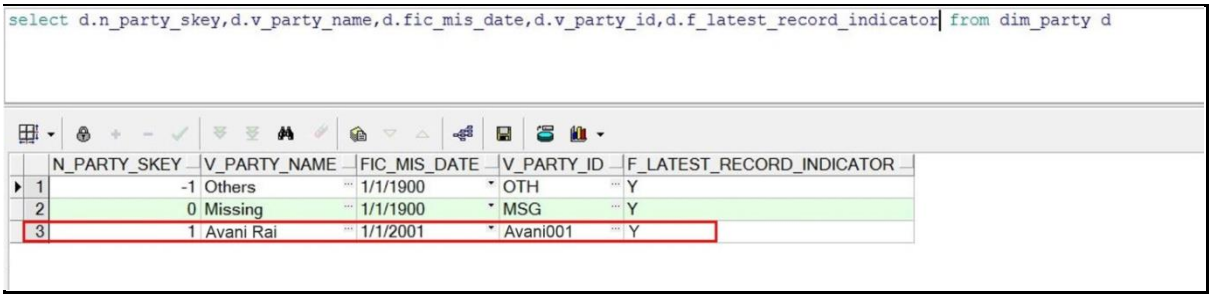
To execute Backdated Run using LRI batch:

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.



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

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



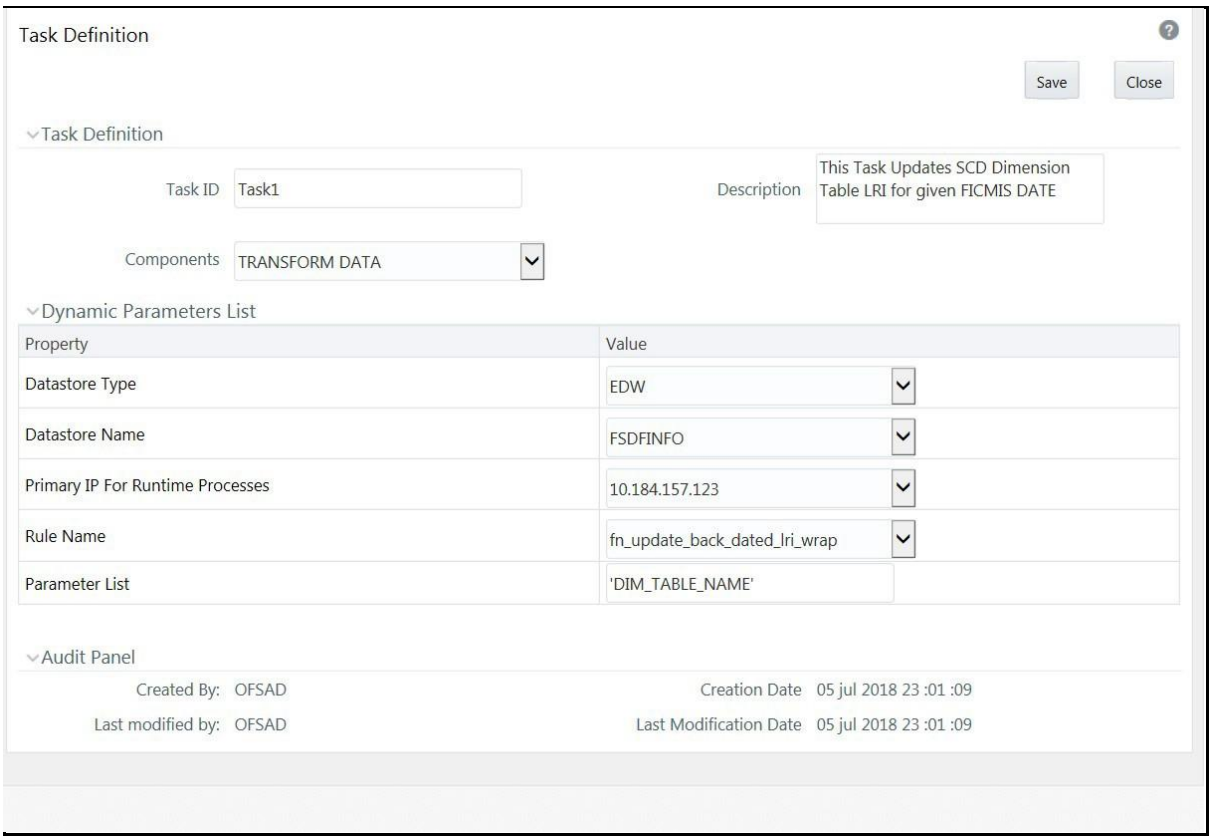
- After adding the first record to DIM_PARTY for the FIC_MIS_DATE 1/1/2001, the flag F_LATEST_RECORD_INDICATOR associated with this data record is set to Y.
- After adding the second record to DIM_PARTY for the FIC_MIS_DATE 1/1/2002, the flag F_LATEST_RECORD_INDICATOR associated with this data record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

```
select d.n_party_skey,d.v_party_name,d.fic_mis_date,d.v_party_id,d.f_latest_record_indicator from dim_party d
```

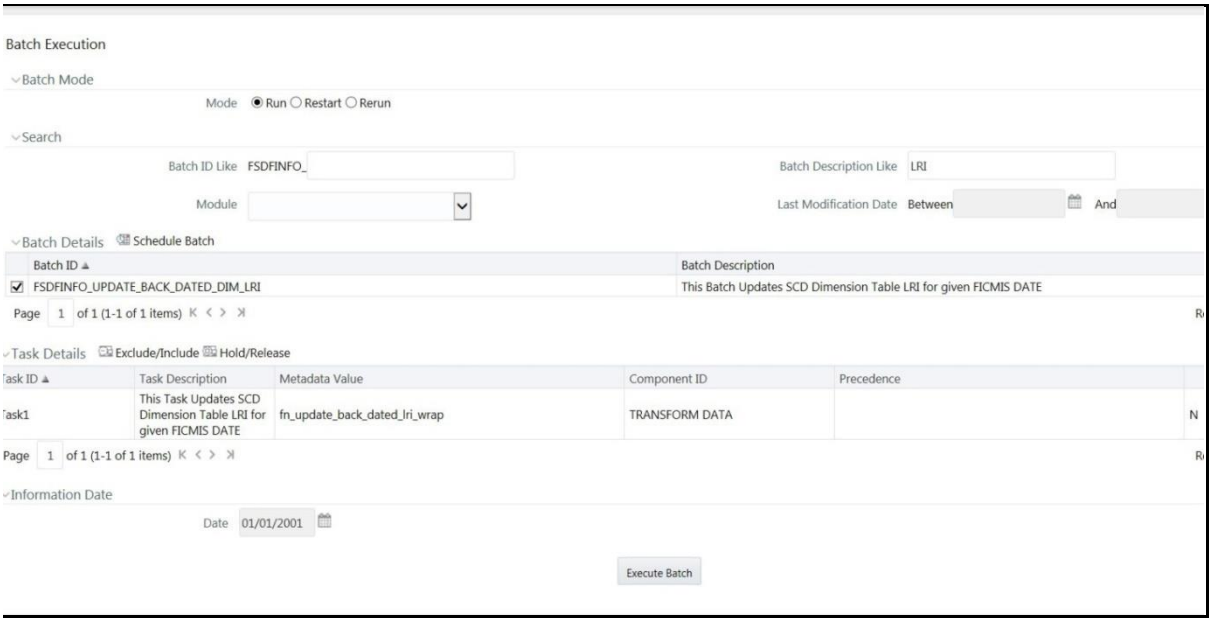
	N_PARTY_SKEY	V_PARTY_NAME	FIC_MIS_DATE	V_PARTY_ID	F_LATEST_RECORD_INDICATOR
1	-1	Others	1/1/1900	OTH	Y
2	0	Missing	1/1/1900	MSG	Y
3	1	Avani Rai	1/1/2001	Avani001	N
4	2	Avani Sharma	1/1/2002	Avani001	Y

- 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’.

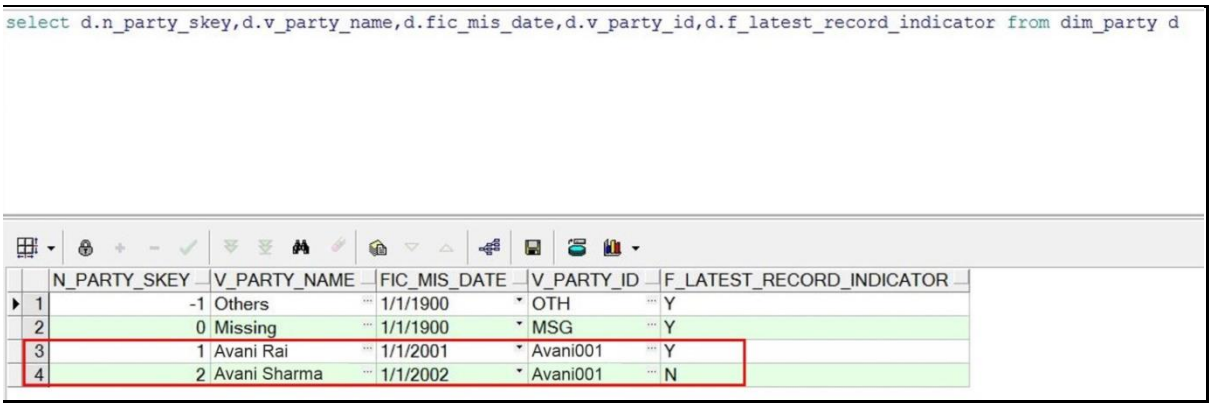


4. In the Batch Execution window, execute the UPDATE_BACK_DATED_DIM_LRI batch for different FIC_MIS_DATEs.



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

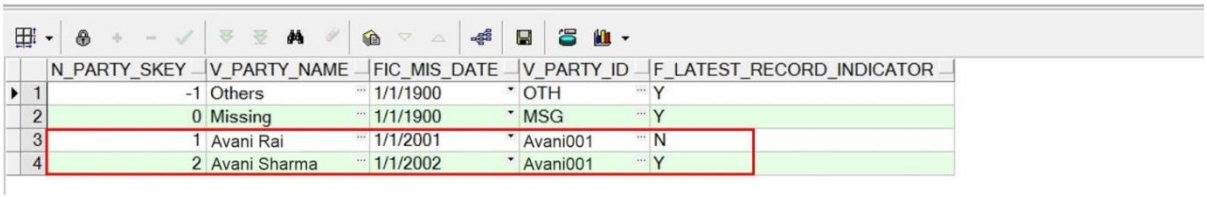
- The image depicts execution of the UPDATE_BACK_DATED_DIM_LRI batch for a backdated record for its FIC_MIS_DATE. Therefore, this record is the Latest Record. In the following image, the LRI batch is executed for the data record with historical FIC_MIS_DATE 1/1/2001. Therefore, the flag F_LATEST_RECORD_INDICATOR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



	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

```
select d.n_party_skey,d.v_party_name,d.fic_mis_date,d.v_party_id,d.f_latest_record_indicator from dim_party d
```



	N_PARTY_SKEY	V_PARTY_NAME	FIC_MIS_DATE	V_PARTY_ID	F_LATEST_RECORD_INDICATOR
1	-1	Others	1/1/1900	OTH	Y
2	0	Missing	1/1/1900	MSG	Y
3	1	Avani Rai	1/1/2001	Avani001	N
4	2	Avani Sharma	1/1/2002	Avani001	Y

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.

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

34 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 Modeling tool. The Comparing Data Model Reports section details the steps to use the OFSAA application and download the Difference Report between two Data Model release versions.

This chapter includes the following topics:

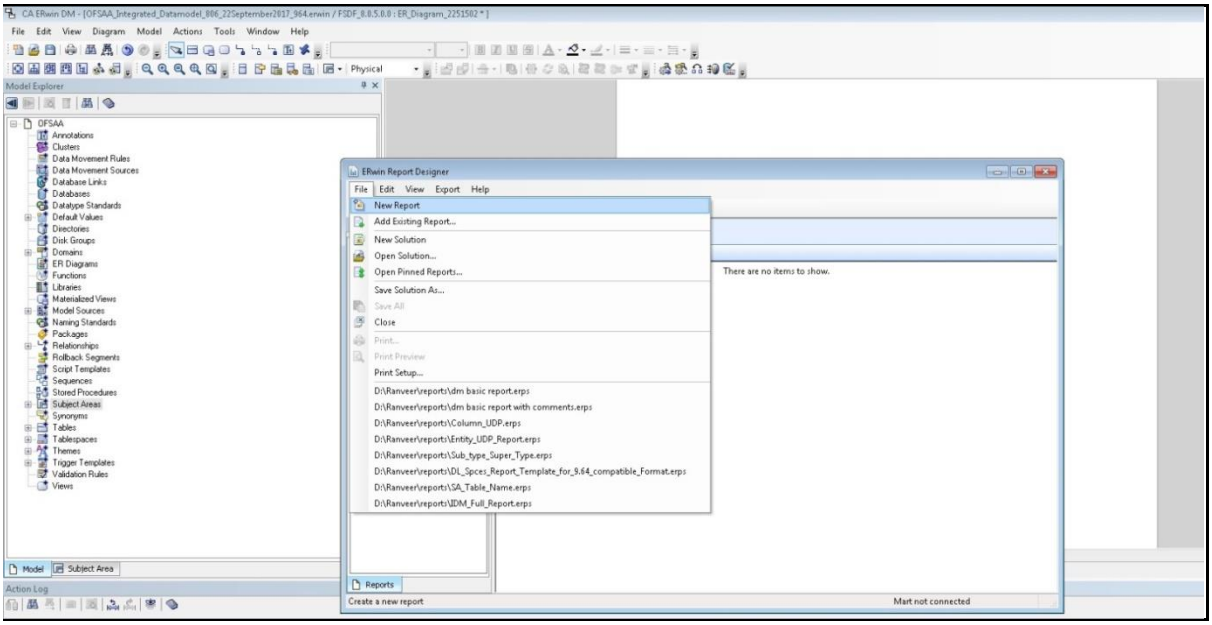
34.1 Creating Data Model Report from erwin

You can create new Data Model Reports from the erwin Data Modeling tool if there are no .eprs files to extract the information.

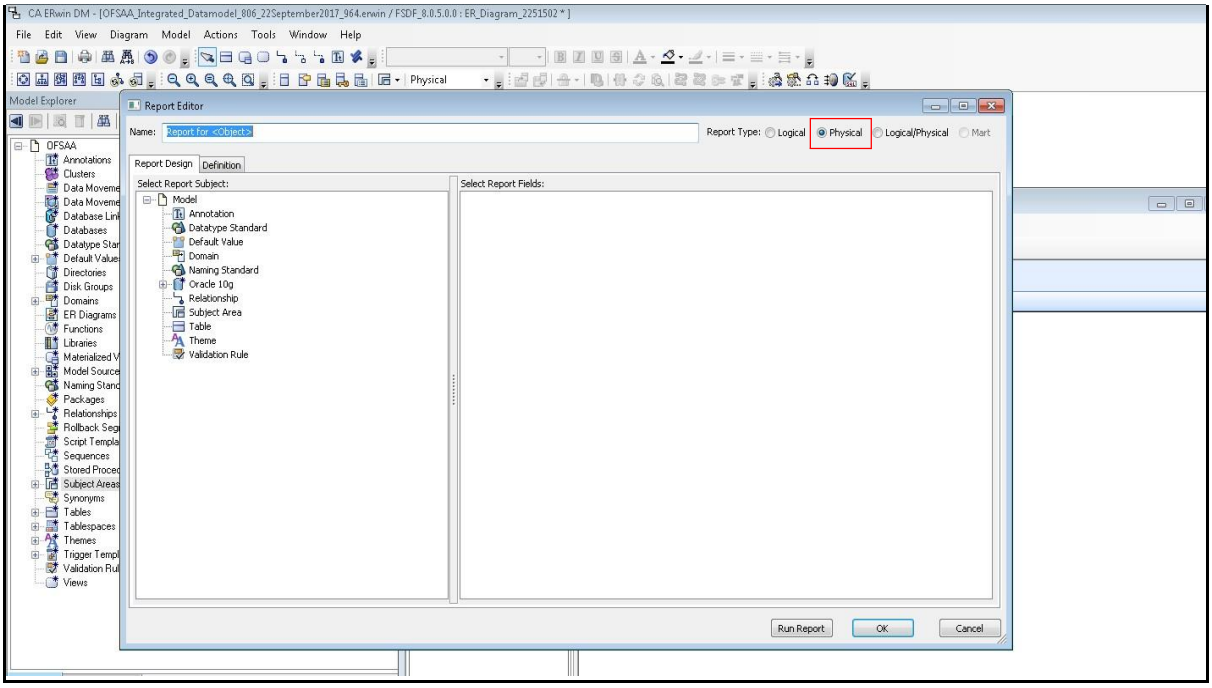
> **NOTE** If the Data Model Reports are existing, see Extracting Data Model Report from ERwin for more information.

Perform the following steps to extract the Data Model Report form ERwin Data Modeling tool:

1. In the ERwin Report Designer window, select File --> New Report (to create the .erps report file).



2. In the Report Editor window Select Report Type as Physical --> Select Report Subject as Table.



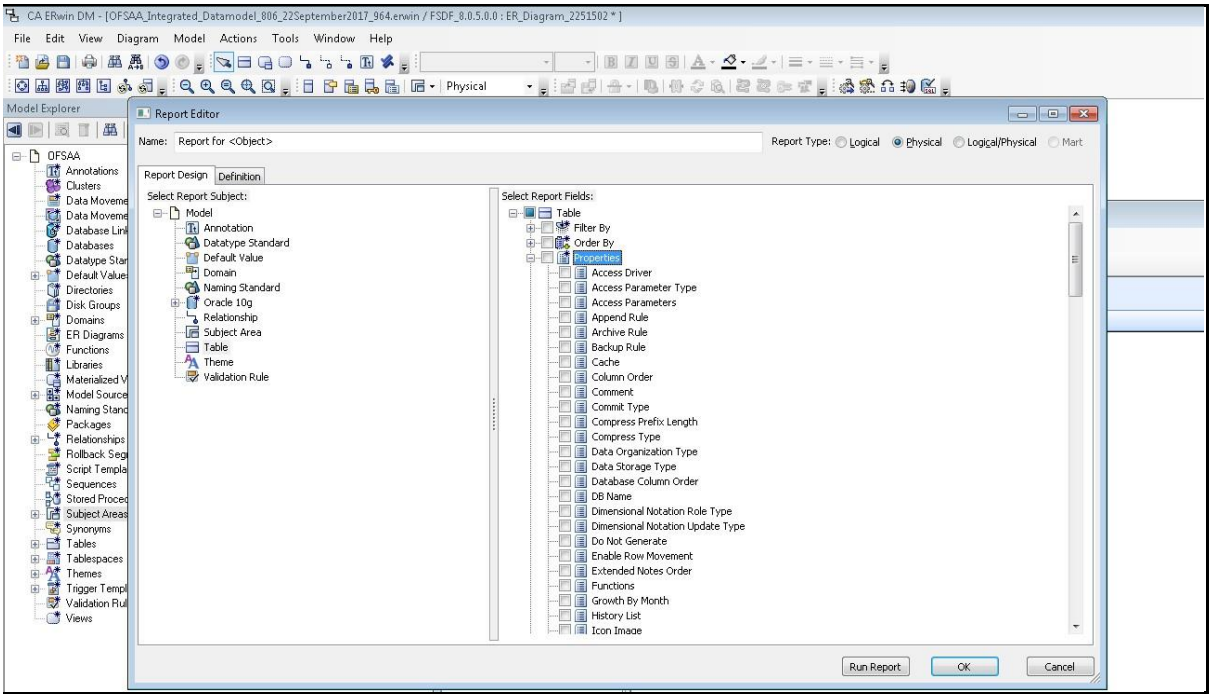
3. Select the Report Fields to match the sample Data Model Report shown as follows:

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW Acrobat

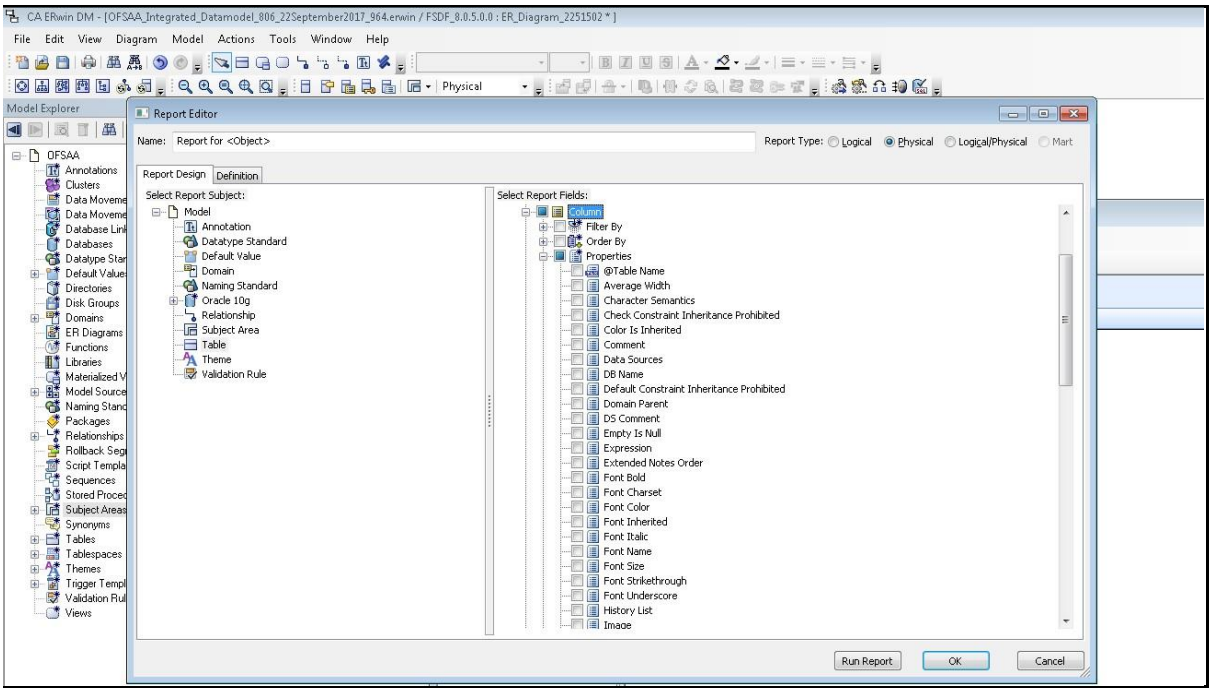
K17

	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	

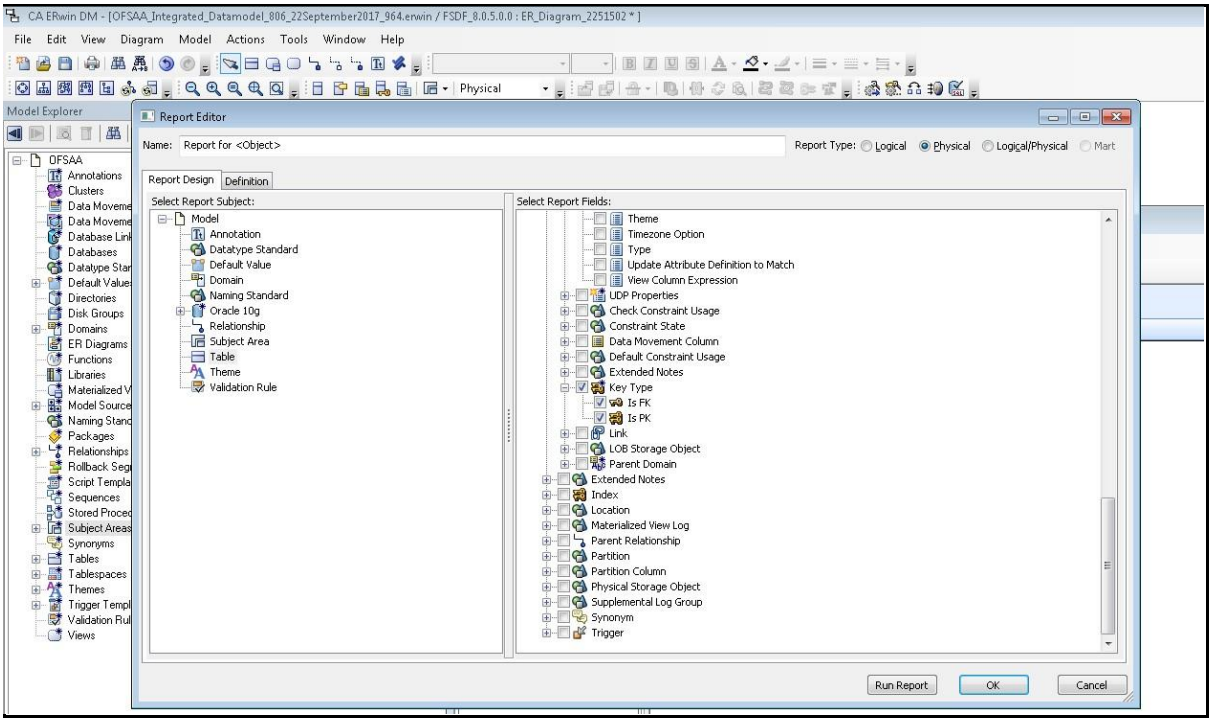
4. Go to Table > Properties, select Physical Name (Entity/Table Physical_Name).



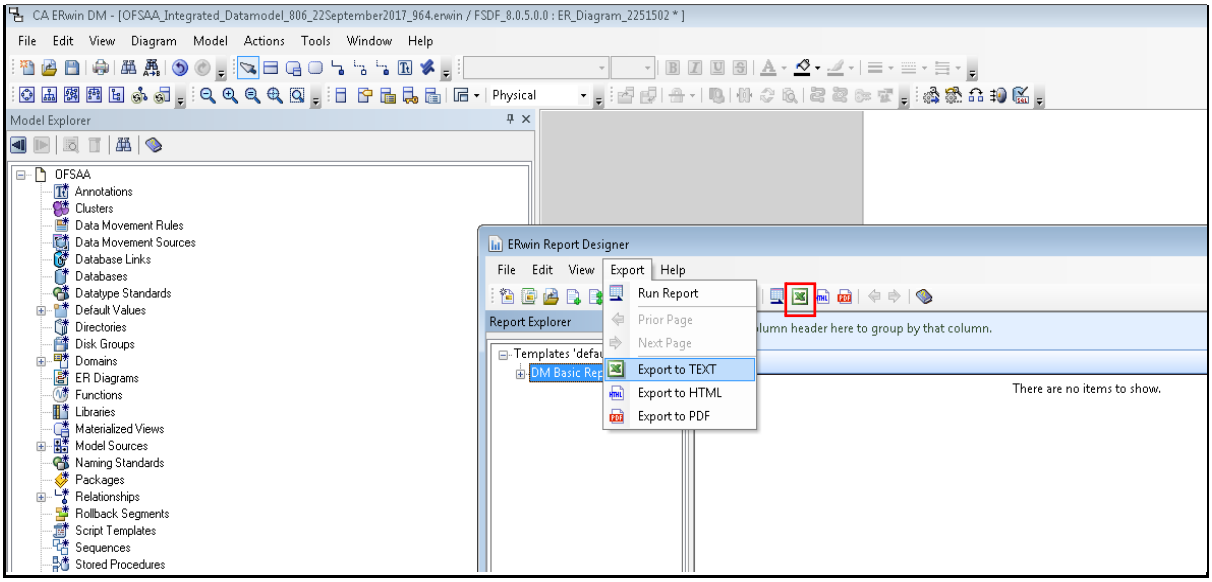
5. Go to Table > Column, select Physical Name (Attribute/Column Physical_Name), Physical Data Type, Null Option, and Domain Parent.



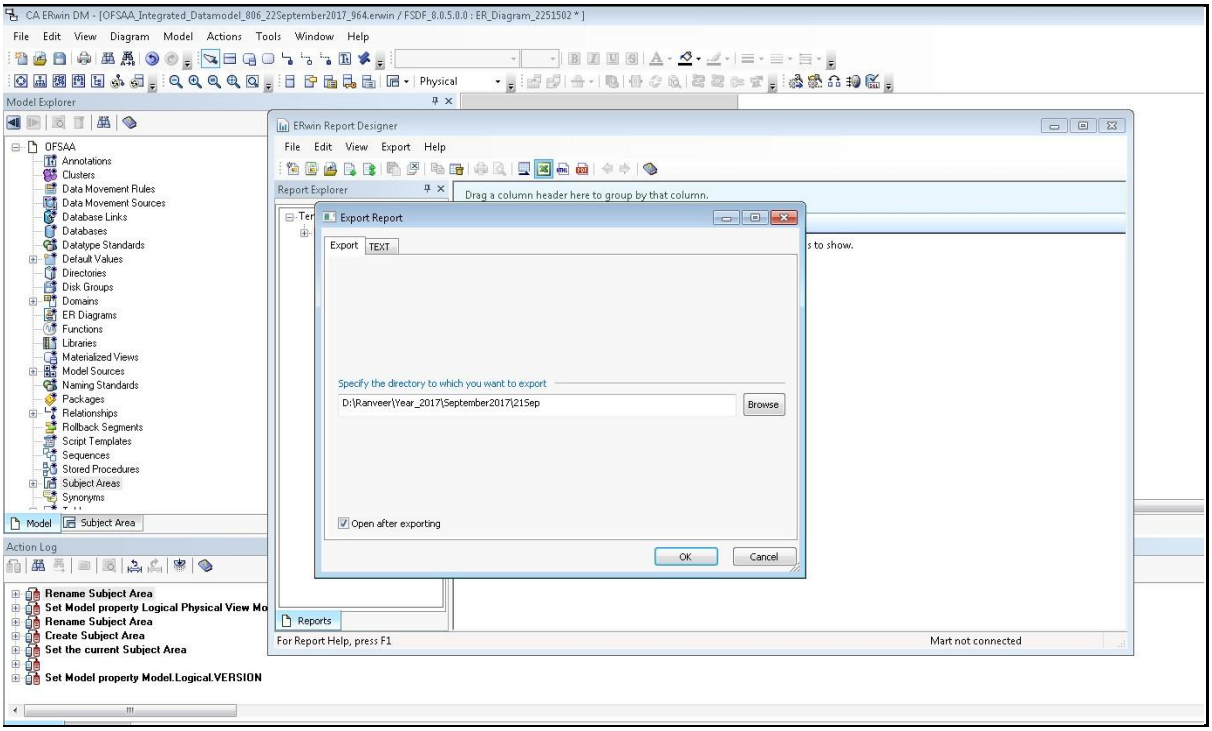
6. Go to Table > Key Type, select Is PK and Is FK. Click OK.



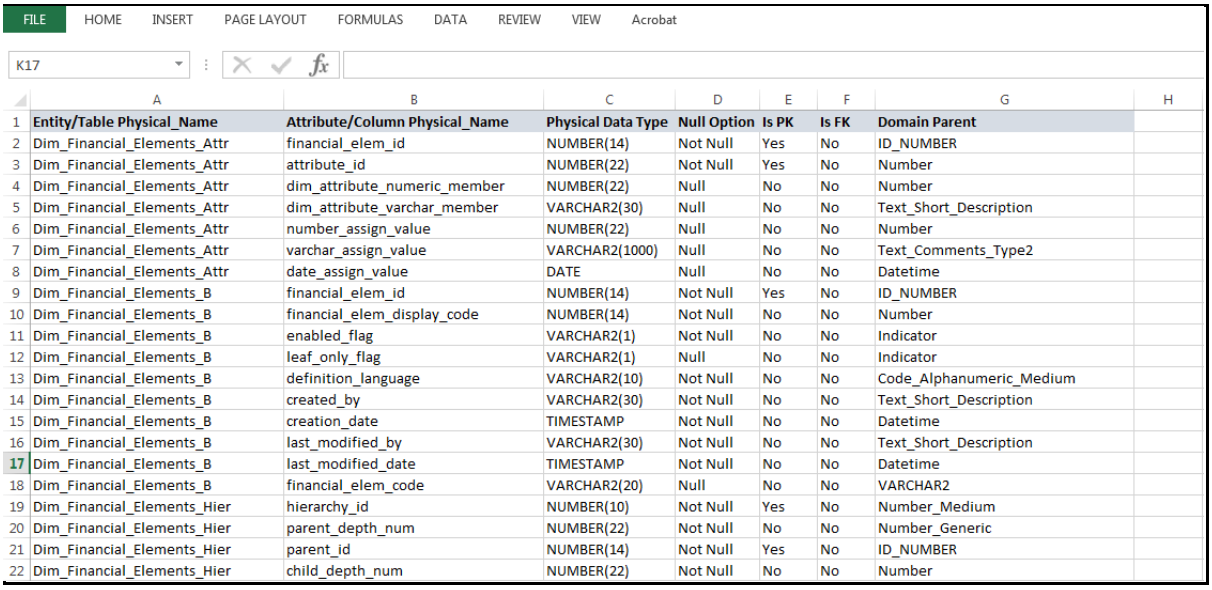
7. In the ERwin Report Designer window, select Export > Export to Text (Excel Format) or the XL icon.



8. Enter or Browse the Export file path where you want to save the Data Model Report file.



9. The generated Data Model Report file will open and must be in the following format.



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	

NOTE Ensure that the file is saved in `.xlsx` format.

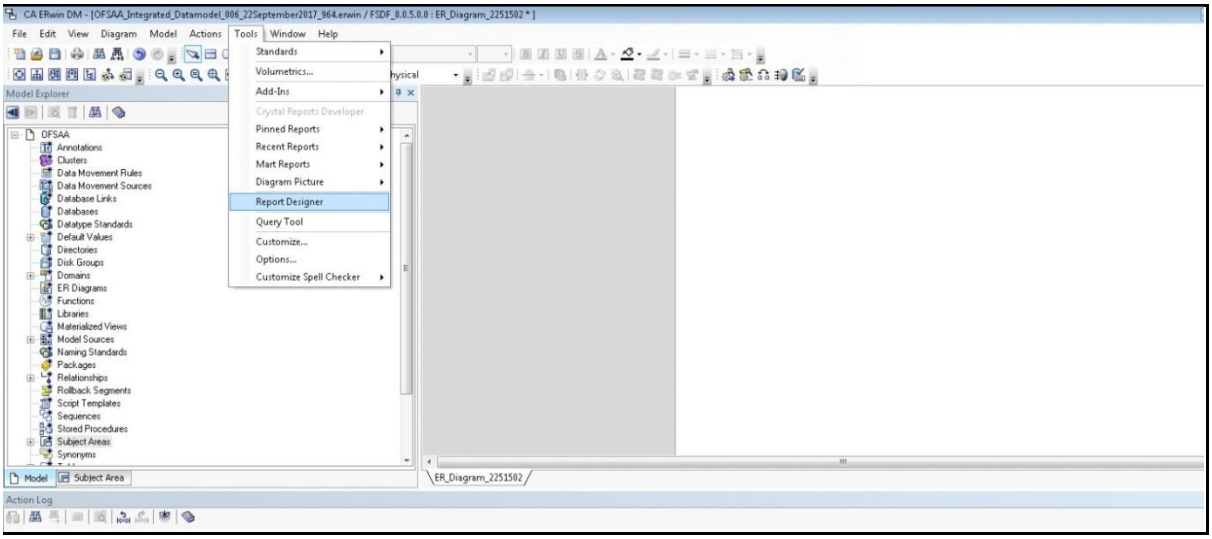
34.2 Extracting Data Model Report from ERwin

OIDF is a collection of data model artifacts delivered as ERwin files or can be extracted as .XLS file from ERwin Data Modeling tool. OIDF hence requires a license of the ERwin Data Modeling tool.

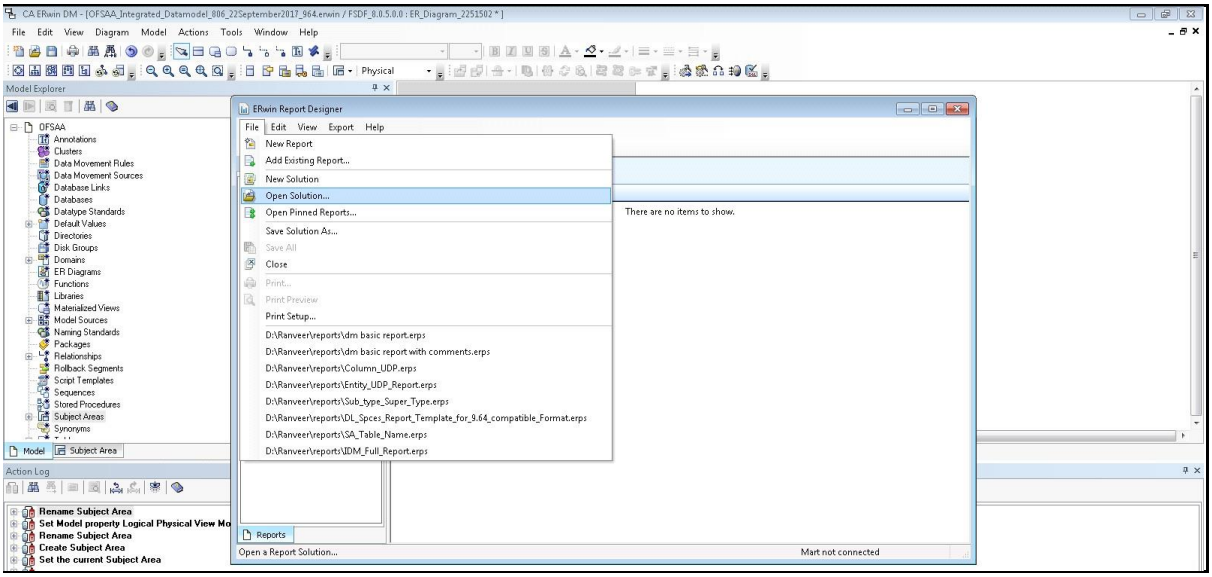
ERwin is the current and only supported modeling tool to view and edit the model. Currently, the minimum versions of ERwin supported are 9.5 and 9.64.

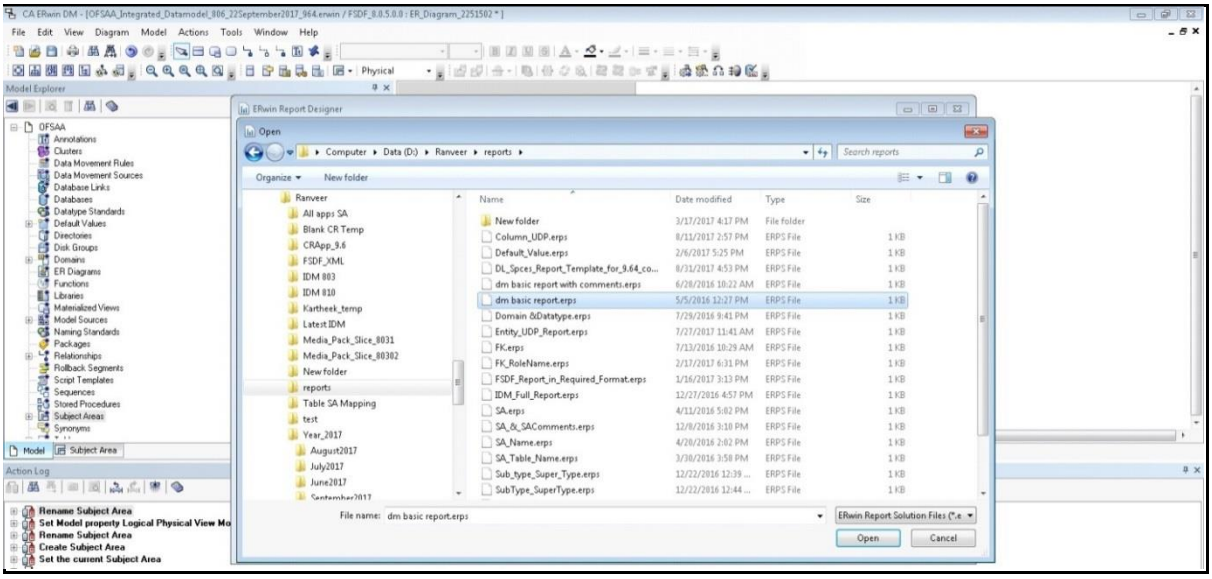
Perform the following steps to extract the Data Model Report form ERwin Data Modeling tool:

1. Open the Erwin Data Modeling tool.

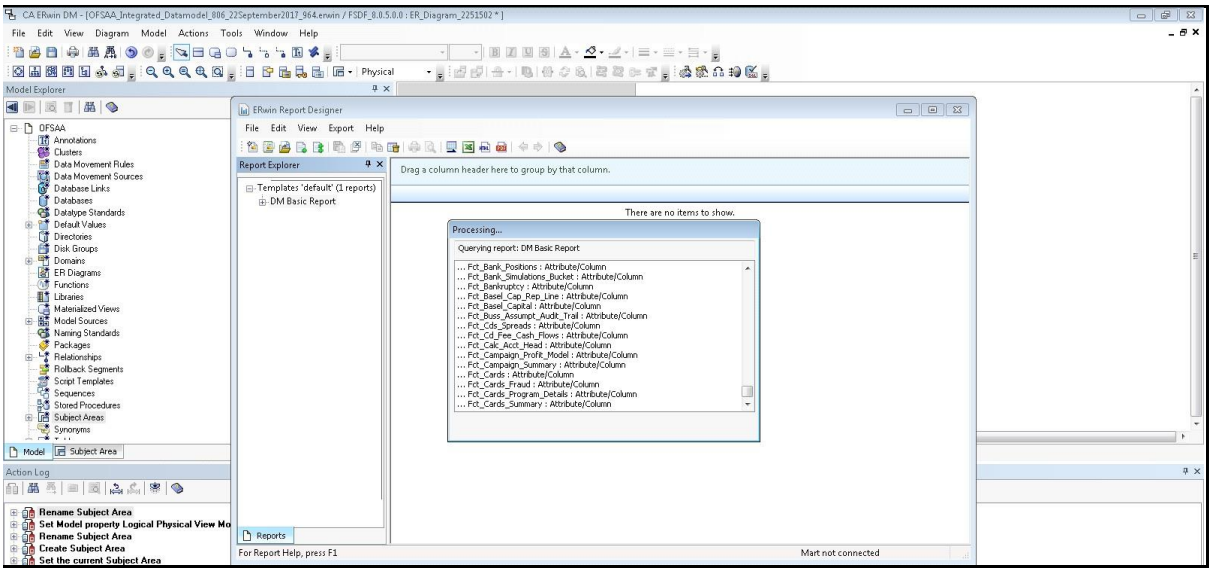


2. Select Tools > Report Designer from the Menu bar.
3. In the ERwin Report Designer window, select File > Open Solution (to extract the .erps report file).





4. Browse the path of the `.erps` file, select the file and click Open.
5. The existing Data Model Report file is extracted and processed to export the file in XL format.



6. The generated Data Model Report file in XL format is opened. Save the file in .xlsx format.

34.3 Comparing Data Model Reports

Perform the following steps to extract the Data Model Report form ERwin Data Modeling tool:

1. Open the OFSA Application with your login credentials.
2. In the Object Administration tab, select Object Administration > Utilities > Compare Models.

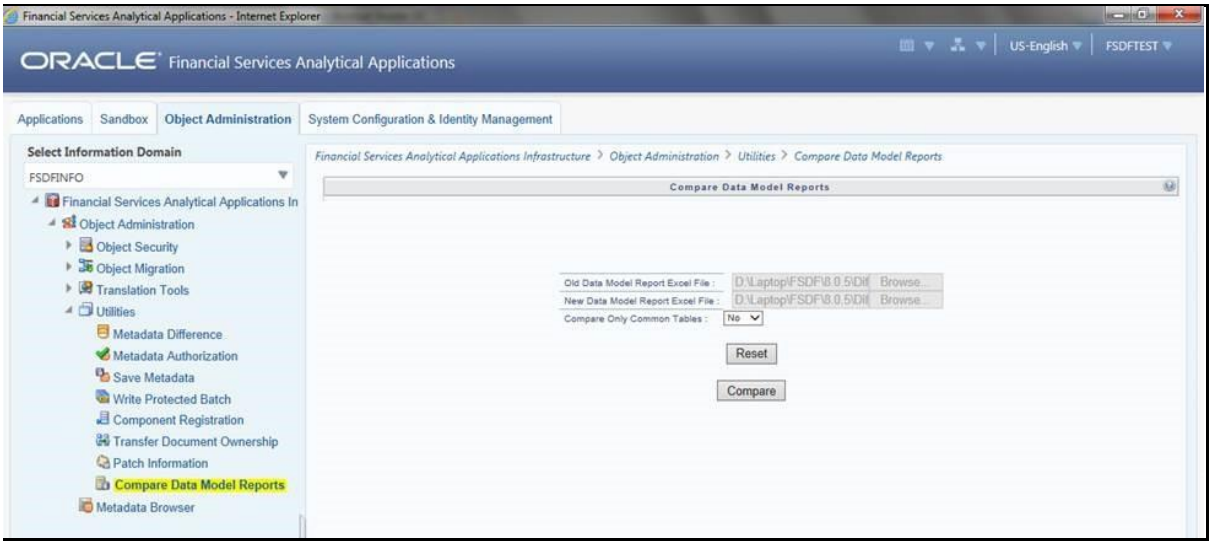
3. Browse the Old Data Model Report Excel File and New Data Model Report Excel File respectively. The Compare button is displayed.

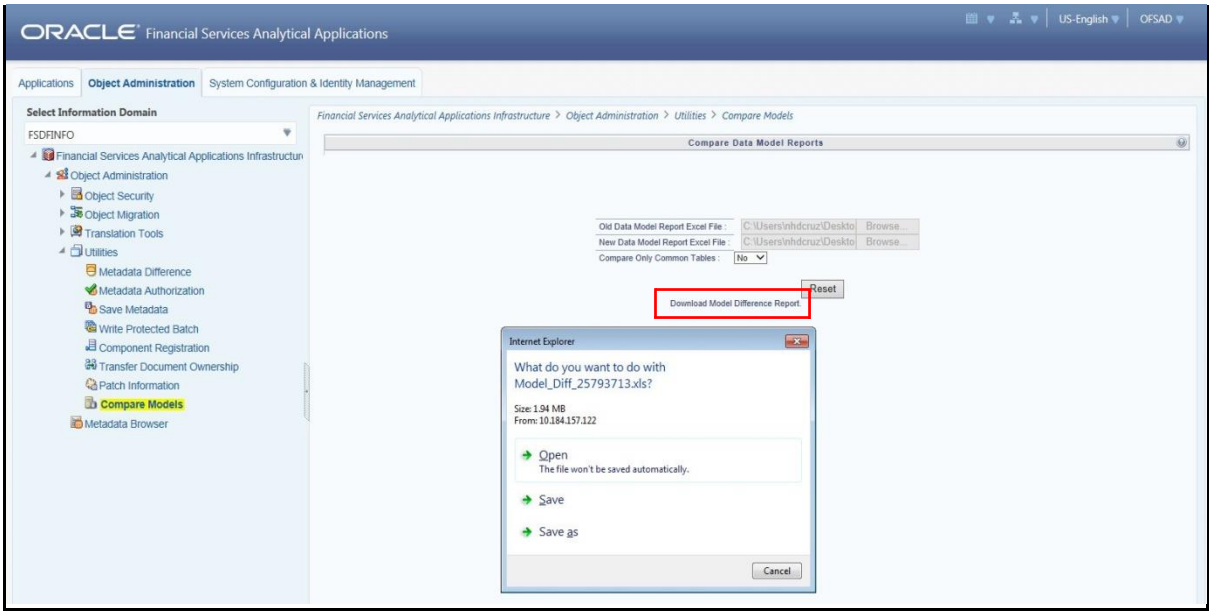
> **NOTE** The Old Data Model Report and New Data Model Report should be in same format and file extension should be .xlsx
>
> Click Reset if you want to select a different Data Model Report.

4. The Compare Only Common Tables option is “No” by default, select “Yes” only if required.

> **NOTE** The Compare Only Common Tables option “Yes” is used when comparing Data Model Report of different products. The Report fields will be different for each product and only common fields are required to compare for investigation.

5. Click Compare. The Data Model Difference Report is generated.





6. Click Download Model Difference Report link and Save the report.

35 OIDF Interface with QMR 2.0.0.0.0

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

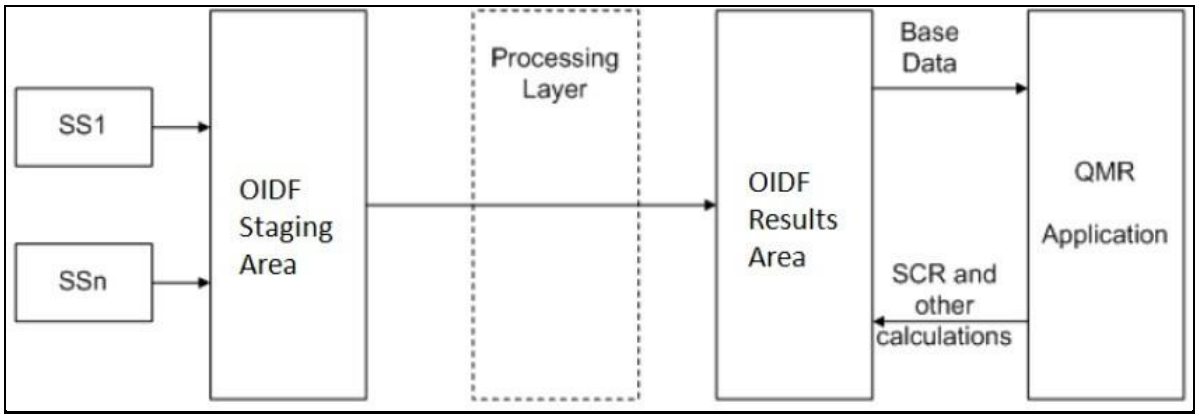
This chapter includes the following topics:

35.1.1 Overview

Oracle Insurance Data Foundation (OIDF) stores insurance specific data of an Insurance Company covering wide range of functions such as underwriting, policy, premium, claim, risk numbers, valuations, and risk reporting.

QMR is a Quantitative Management Reporting application from Oracle. QMR is a Hyperion Financial Management (HFM) based application and is used for Solvency II regulatory reporting. QMR produces highly formatted Quantitative Reporting Template (QRT) based reports as specified by European Insurance and Occupational Pensions Authority (EIOPA).

QMR integration refers to data flow between System Source (SS) of OIDF and QMR application. OIDF consists of staging and reporting area tables. This integration is between result data model and QMR.



The base data from source systems resides in OIDF staging area. From there, it moves to the OIDF processing area. The data is then processed and moved into the OIDF results area.

Data flow between QMR and OIDF is a two way process:

1. Data is extracted from OIDF using this interface and uploaded into the QRM 2.0.0.0.0 application. The application QMR 2.0.0.0.0 is then used to generate QRT based reports.
2. SCR numbers calculated in QMR will then be ported back into OIDF.

35.1.2 Prerequisites

For the installation of QMR - OIDF Integration, the following prerequisites need to be completed:

- Installation of QMR 2.0.0.0.0 application.
- Installation of OIDF 8.0.7.0.0

35.1.3 Description of Extracts from OIDF

OIDF extracts provide data pertaining to the various QRT reports in the granularity and format required by QMR

Any changes in QMR data requirement may require changes to be made to the OIDF extracts as well. Files required by QMR application depend upon the entity type and their location. Details of the kind of files generated for each entity type is as follows:

If the Legal Entity is the Group Holding Company:

- Global Data file containing exchange rates and inflation rates is generated.
- Group Data file containing ownership information, group reporting percentages, and group SCR capital add-ons is generated.
- Five files containing details of Intra Group Transactions - 4 Files, Risk Concentration - 1 File for processing through FDM.

If the Legal Entity is an Insurance Company within the group and located in European Economic Area (EEA):

- A single file containing data for all schedules processed in HFM (Balance Sheet, Own Funds, Re-Insurance, SCR and MCR inputs, Technical Provisions, Country, and Cover) is generated.
- Eleven files containing data for each schedule processed in FDM (Assets - 8 Files, Product Details – 3 Files) is generated.

If the Legal Entity is an Insurance Company within the group located outside European Economic Area (EEA):

- A single file with Insurance entity MCR, SCR, and OF equivalent data only is generated.

If the Legal Entity is Non-Insurance Company within the group:

- A single file with MCR, SCR, and OF equivalent data only is generated.

Further details regarding the QRT reports linked to each entity type is given as follows:

Sl. No.	Extract Name	Description
1	Legal Entity – Solo	Provides Data for following reports
		BS-C1 BS-C1D Cover-A1A Group-G01 MCR-B4 OF (Annual) Re-J1-Non Life Re-J1-Life Re-J2

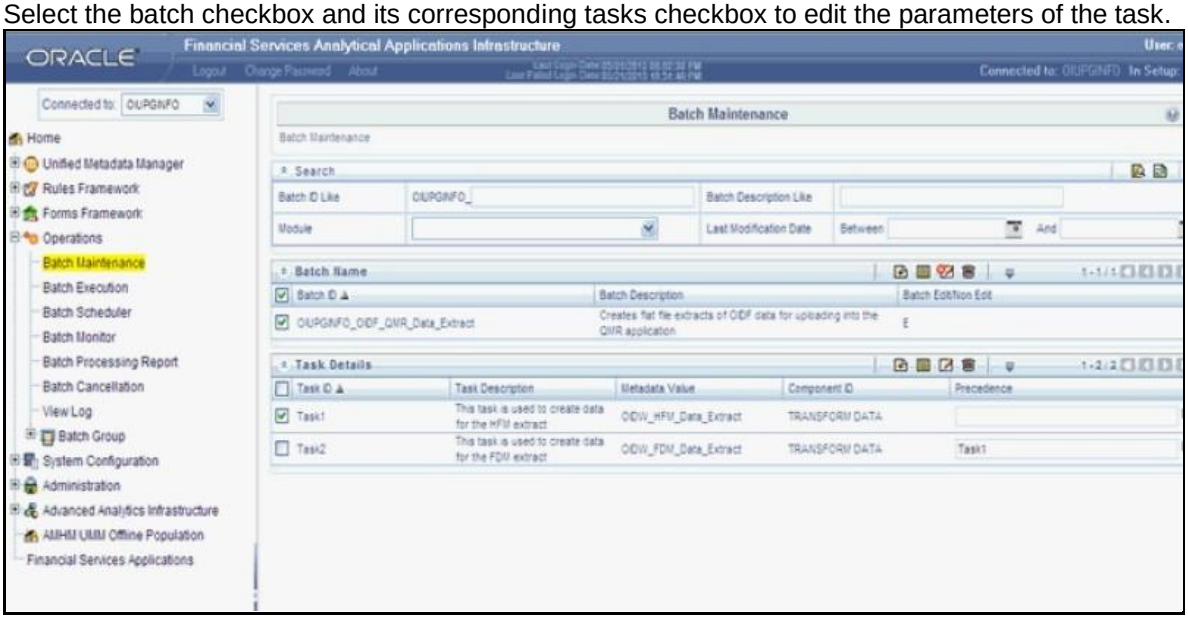
Sl. No.	Extract Name	Description
		Re-J3 Re-SPV SCR-B2A SCR-B2B SCR-B2C SCR-B3A SCR-B3B SCR-B3C SCR-B3D SCRB-3E SCRB-3F SCR-B3G TP-E1 TP-E2 TP-E3 TP-E4 TP-E6 TP-E7A TP-E7B TP-F1 TP-F2 TP-F4
2	Legal Entity – Group	P-Own P-Con OF Group SCR-B2A
3	Legal Entity-RFR (Ring Fenced Fund)	BS-C1 OF (Annual) SCR-B2A SCR-B2B SCR-B2C SCR-B3A SCR-B3B SCR-B3C SCR-B3D SCRB-3E
4	Legal Entity – Non Insurance Entity	1. Group-G04
5	Legal Entity – Non EEA Entity	1. Group-G03
6	Global Data	Exchange Rates Inflation Rates Discount Rates Correlation Data
7	Other Individual Files	IGT1 IGT2

Sl. No.	Extract Name	Description
		IGT3 IGT4 ASSETS-D1 ASSETS-D1S ASSETS-D20 ASSETS-D2T ASSETS-D3 ASSETS-D4 ASSETS-D5 ASSETS-D6

35.1.4 Procedure to Generate Flat Data Files from OIDF

The following steps enlist the process to generate the flat data files from OIDF, for QMR – HFM and Financial Data Quality Management (FDM) Applications:

1. Login to the OFSAAI Home page.
2. Select the appropriate infodom.
3. Navigate to Batch Maintenance in the Operations Menu in the left hand side panel.
4. The tasks for OIDF to QMR App Data Extract are bundled under the batch [Infodom_Name]_OIDF_QMR_Data_Extract.
5. Select the batch checkbox and its corresponding tasks checkbox to edit the parameters of the task.
6. Select the batch checkbox and its corresponding tasks checkbox to edit the parameters of the task.



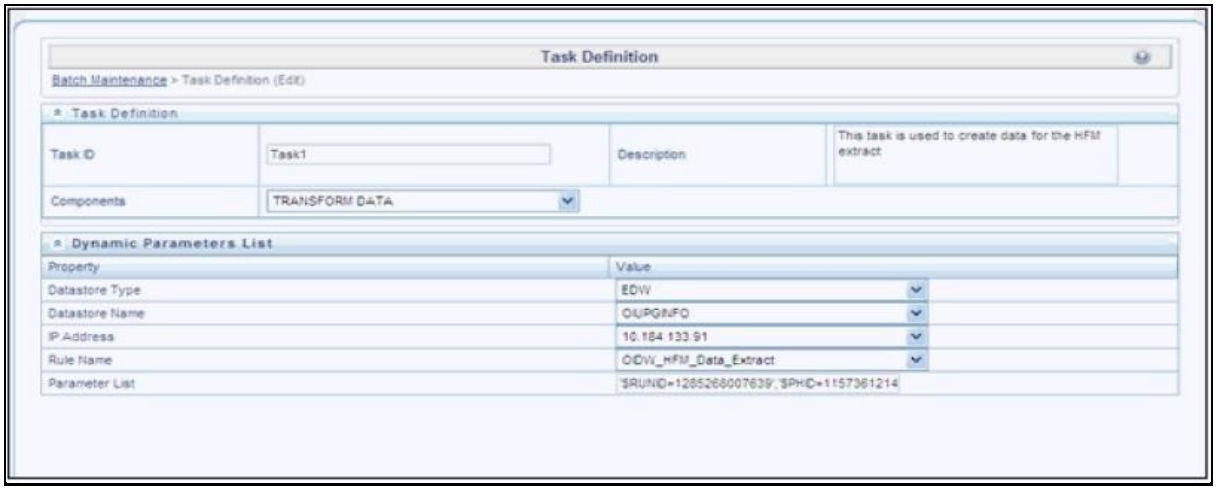
7. Click the Edit button for Task 1 (HFM Extract Task) and edit its parameter list.

   Leaving the primary parameters ($RUNID, $PHID, $EXEID, $RUNSK) unchanged, set values for the following parameters.

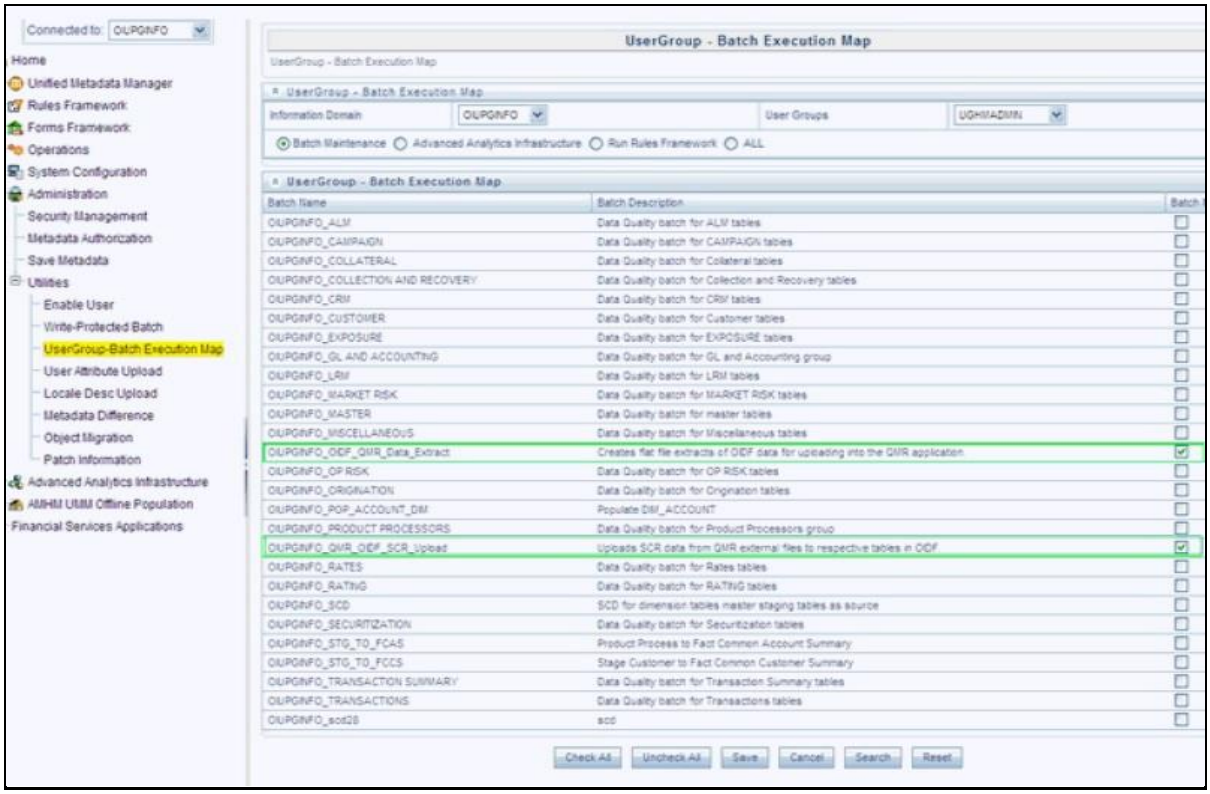
$QMRRUNSK=1	Set Run SKey for which the QMR data needs to be extracted.
$RPSTARTDATE=20120101	Set the Reporting Period Start Date in YYYYMMDD Format.
$FINANCIAL_PRD=QA	Set the Financial Period for the required Data Extract (Q1/Q2/Q3/Q4/QA).

8. Replace the parameter list (preceded with the parameter names) in Task Definition and re-save the task.
9. Click edit for Task 2 (FDM Extract Task) and edit its parameter list, similar to Task 1. Leaving the primary parameters ($RUNID, $PHID, $EXEID, $RUNSK) unchanged, set values for the following parameters.

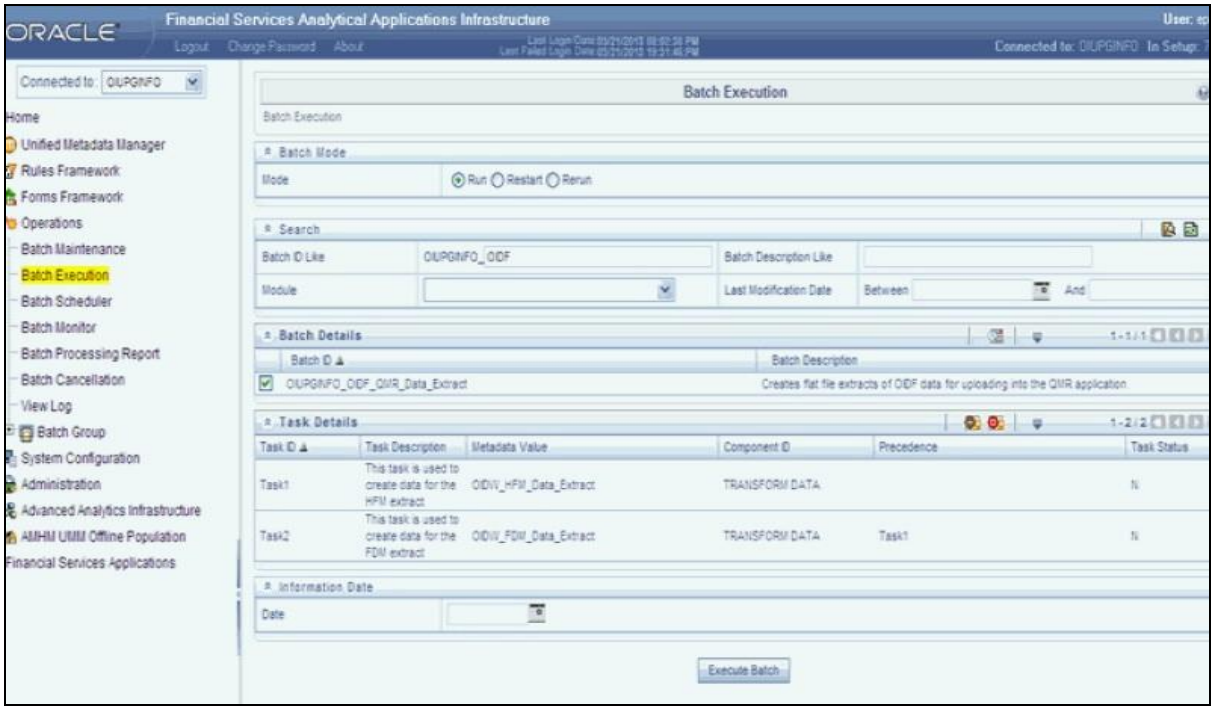
$QMRRUNSK=1	Set Run SKey for which the QMR data needs to be extracted.
$RPSTARTDATE=20120101	Set the Reporting Period Start Date in YYYYMMDD Format.
$FINANCIAL_PRD=QA	Set the Financial Period for the required Data Extract (Q1/Q2/Q3/Q4/QA).



10. Replace the parameter list (preceded with the parameter names) in Task Definition and re-save the task.
11. Map the batch or batches to be executed to the user group in the UserGroup-Batch Execution Map screen in the Administration menu.

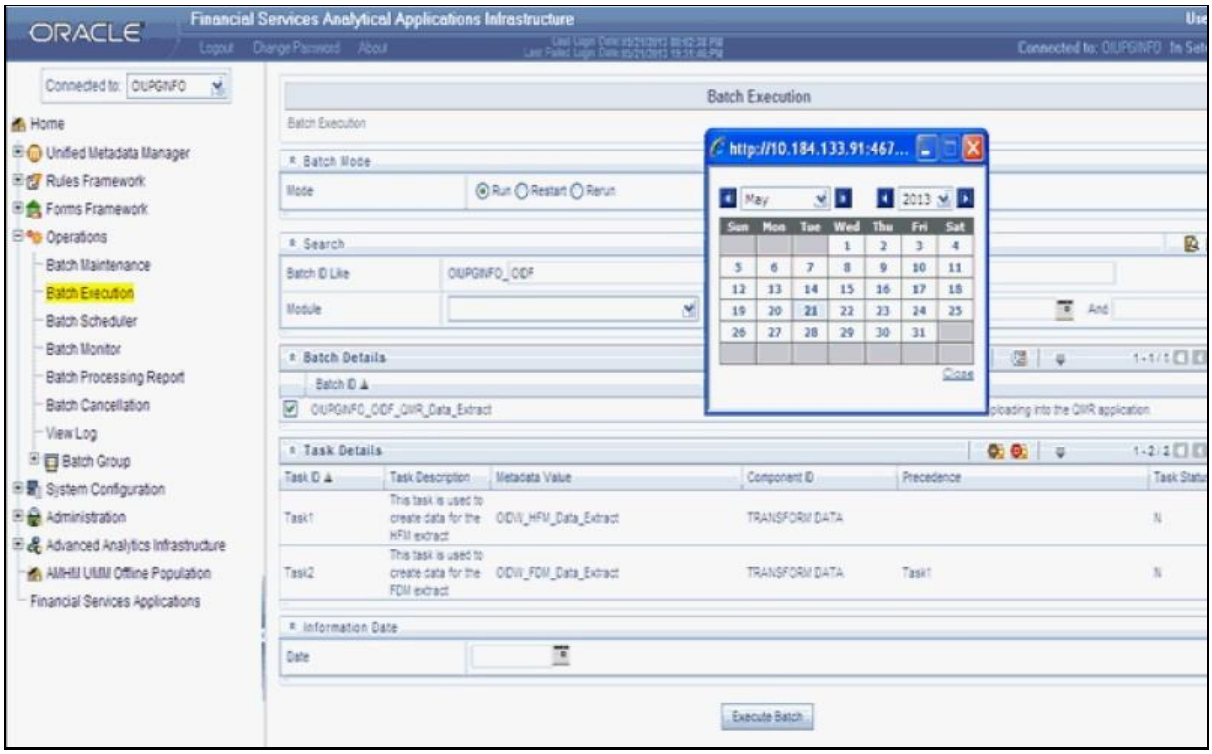


12. Navigate to Batch Execution in the Operations menu in the left hand side panel and select the batch [Infodom_Name]_OIDF_QMR_Data_Extract for execution and provide the Information Date from the Calendar dialog box.



13. Execute the Batch.

14. To check the status of the batch click Batch Monitor in the Operations menu.



15. Verify whether the batch is executed successfully, by confirming presence of all expected data file extracts under the defined Oracle Directory path for - QMR_EXTRACTS (refer Oracle Directory Script Creation). Data extract files generated would be in the QMR_Extract Directory as:

- HFM Global Data Files will be named as:

  HFM_Global_<<MIS Date>>_<<Run Skey>>.dat
- HFM Group Data Files will be named as:

  HFM_Group_<<MIS Date>>_<<Run Skey>>.dat
- HFM Insurance Non EEA Data Files will be named as:

  HFM_InsuranceNonEEA_<<Entity Code>><<MIS Date>>_<<Run_Skey>>.dat
- HFM Ring Fenced Fund Data Files will be named as:

  HFM_RFF_<<EntityCode>><<MIS Date>>_<<Run_Skey>>.dat
- HFM Insurance Data Files will be named as:

  HFM_InsuranceSolo_<<Entity Code>><<MIS Date>>_<<Run_Skey>>.dat
- HFM Non Insurance Data Files will be named as:

  HFM_NonInsurance_<<Entity Code>><<MIS Date>>_<<Run_Skey>>.dat
- FDM Group Data Files will be named as:

  FDM_Group_<<ReportName>>_<<Entity Code>><<MIS Date>>_<<Run_Skey>>.dat
- EEA data Data Files will be named as:

  FDM_EEA_<<ReportName>>_<<Entity Code>><<MIS Date>>_<<Run_Skey>>.dat
- Ring Fenced Fund Data Files will be named as:

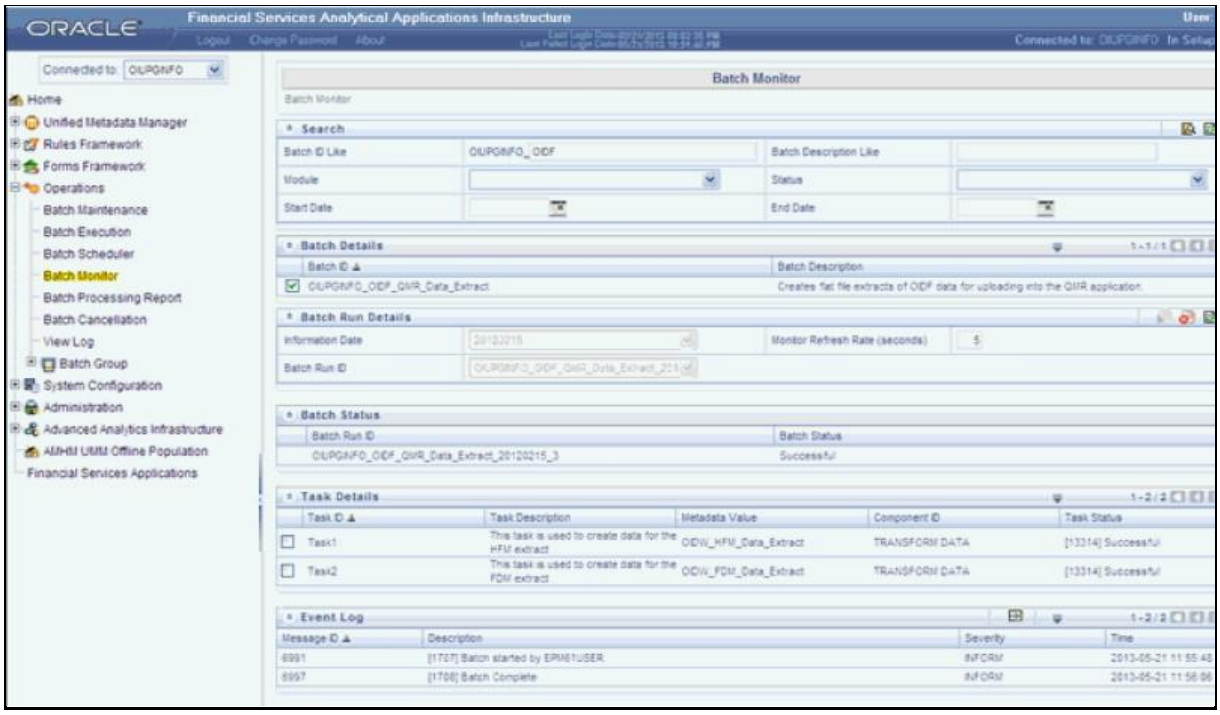
  FDM_Insurance_<<ReportName>>_<<Entity Code>><<MIS Date>>_<<Run_Skey>>.dat

35.1.5 Procedure to Upload SCR Data

The following steps enlist the process to upload Solvency Capital Requirements (SCR) data from QMR external flat (HFM Extracted File with calculated SCR and Minimum Capital Requirements (MCR) values) files to respective OIDF tables:

> **NOTE** The data extracted from HFM must include calculated data.

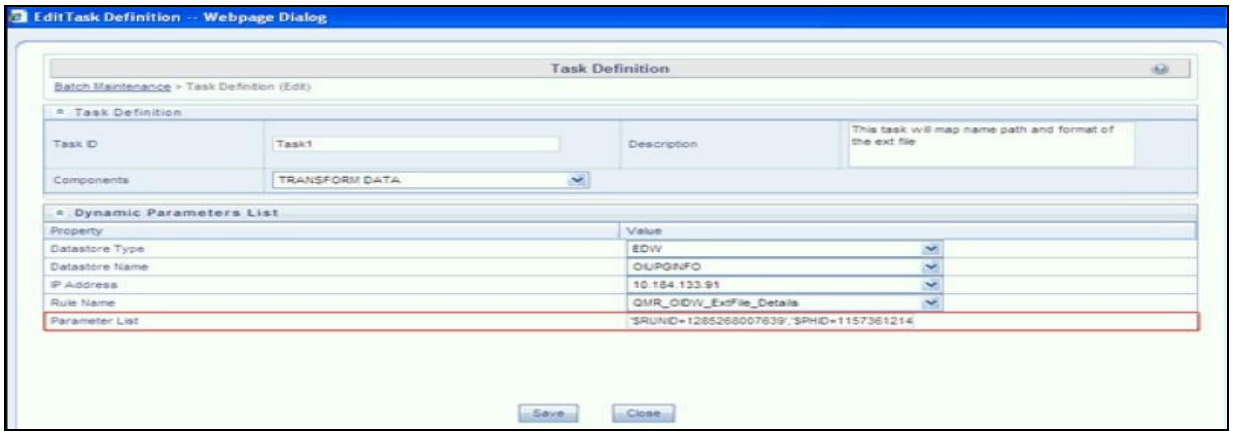
1. Verify the SCR (calculated values) Extract Data File from HFM under the defined Oracle Directory DIR_QMR_UPLOAD_FILES
2. Login to the OFSAAI Home page.
3. Select the appropriate Infodom.
4. Navigate to Batch Maintenance in the Operations menu in the left hand side panel.
5. The tasks for OIDF to QMR App Data Extract are bundled under the batch [Infodom_Name]_QMR_OIDF_SCR_Upload.
6. Select the batch checkbox and its corresponding tasks checkbox to edit the parameters for the task.



7. Click the Edit button for Task 1 (QMR_OIDW_ExtFile_Details) and edit its parameter list. Leaving the primary parameters ($RUNID, $PHID, $EXEID, $RUNSK) unchanged, set values for the following parameters.

Parameter	Set Value
$EXT_DIR_NAME=DIR_QMR_UPLOAD_FILES	Set Oracle Directory where the HFM SCR Extract Data is placed.

Parameter	Set Value
$EXT_FILE_NAME=QMR_SCR_Data_2011-QA_LE 01_Extract.dat	Set the 'HFM SCR Extract Filename' to be processed present under the Oracle Directory.

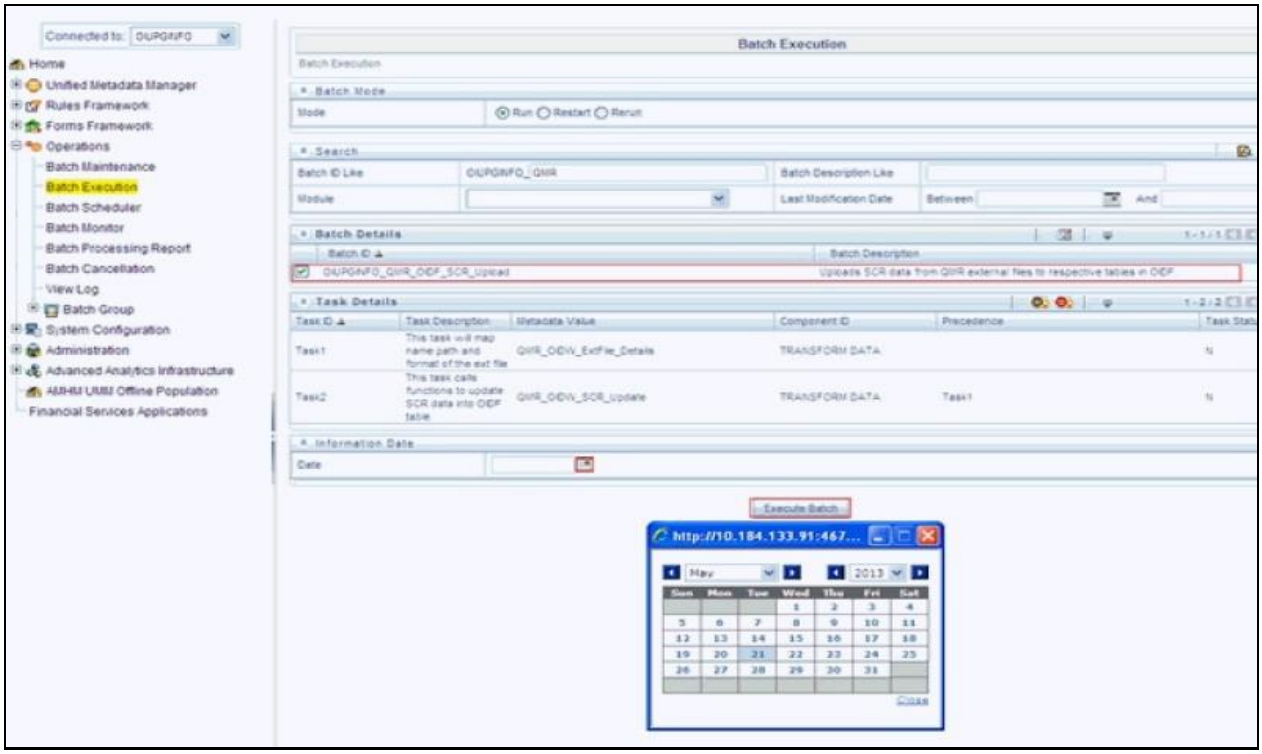


8. Replace the parameter list (preceded with the parameter names) in the Task Definition and re-save the Task.
9. Click Edit for Task 2 (QMR_OIDW_SCR_Update) and edit its parameter list.

   Leaving the primary parameters ($RUNID, $PHID, $EXEID, $RUNSK) unchanged, set values for the following parameters.

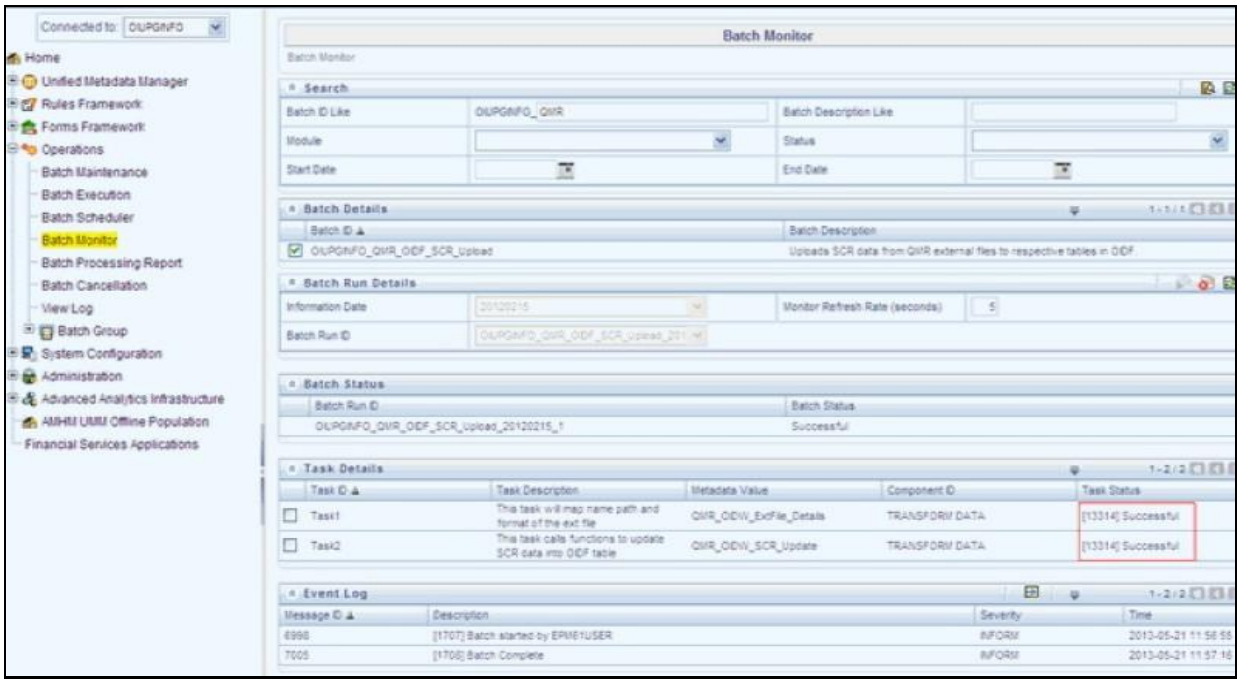
Parameter	Set Value
$QMRDATE=20120101	Set the MIS Date for which the QMR data needs to be updated in OIDF in YYYYMMDD Format.
$QMRRUNSK=1	Set Run SKey for which the QMR data needs to be updated in OIDF.
$QMRGAAP=IFRS	Set the GAAP ID for which the QMR data needs to be updated in OIDF.

10. Replace the parameter list (preceded with the parameter names) in Task Definition and re-save the task.
11. Map the batch or batches to be executed to the user group in UserGroup-Batch Execution Map screen in the Administration menu.
12. Navigate to Batch Execution in the Operations menu in the left hand side panel and select the batch ([Infodom_name]_QMR_OIDF_SCR_Upload) for execution and provide the Information Date from the Calendar dialog box.



13. Execute the Batch.

14. Verify whether the batch is executed successfully by confirming presence of calculated SCR and MCR data in all the relevant tables in OIDF.



35.1.6 Data Mapping

All reporting is based on standard codes, which are often used as codes and filters. For more information on the relevant scripts to be executed for the seeded data and download data tables, refer to Oracle Insurance Solvency II QMR Interface Download Specifications .

The following seeded data are to be populated in the Dimension/Setup Tables:

Entity Name	Table Name
Business Class Dimension	Dim_Business_Class
Business Segment Dimension	Dim_Business_Segment
Capital Computation Group Dimension	Dim_Capital_Comp_Group
Capital Computation Method Dimension	Dim_Capital_Computation_Method
Dimention Cash Flow Type	Dim_Cash_Flow_Type
Dimension Claim Status	Dim_Claim_Status
Country Dimension	Dim_Country
Coverage Type Dimension	Dim_Coverage_Type
Currency Dimension	Dim_Currency
Economic Zone Dimension	Dim_Economic_Zone
Dimension Entity Type	Dim_Entity_Type
Fund Type Dimension	Dim_Fund_Type
Insurance Risk Measures Dimension	Dim_Insurance_Risk_Measure
Insurance Risk Type Dimension	Dim_Insurance_Risk_Type
Dimension Intra Group Transaction Type	Dim_Intra_Group_Txn_Type
Dimension Market Risk Position	Dim_Market_Risk_Position
Product Benefits Type Dimension	Dim_Product_Rider_Type
Region Dimension	Dim_Region
Reporting Line Dimension	Dim_Rep_Line
Run-Off Measure Dimension	Dim_Run_Off_Measures
Run Types	Dim_Run_Type
Scenario Dimension	Dim_Scenario
Standard Lob Dimension	Dim_Standard_Lob
Standard Major Lob Dimension	Dim_Standard_Major_Lob
Standard Product Type Dimension	Dim_Standard_Product_Type
Standard Mitigant Type Dimension	Dim_Std_Mitigant_Type
Generally Accepted Accounting Principles Dimension	Dim_Gaap

35.1.7 Data Population

35.1.7.1 DIM Table Population

The sequence of Data Population for Dimension tables is a follows:

Entity Name	Table Name
Account Dimension	Dim_Account
Account Identifier Type Dimension	Dim_Account_Identifier_Type
Bands Dimension	Dim_Bands
Cedent Asset Classification Dimension	Dim_Cedent_Asst_Classification
Claim Dimension	Dim_Claim
Collateral Purpose Dimension	Dim_Collateral_Purpose
Dimension Consolidation Approach	Dim_Consolidation_Approach
Dimension Legal Nature Of Company	Dim_Constituted_Form
Contract Unwind Trigger Dimension	Dim_Contract_Unwind_Trigger
Credit Rating Dimension	Dim_Credit_Rating
Date Dimension	Dim_Dates
Account Dimension	Dim_Exposure
Exposure Type Dimension	Dim_Exposure_Type
Financial Year Dimension	Dim_Financial_Year
Fund Dimension	Dim_Fund
Hedge Status Dimension	Dim_Hedge_Status
Hedge Type Dimension	Dim_Hedge_Type
Hedging Strategy Dimension	Dim_Hedging_Strategy
Homogenous Risk Group Dimension	Dim_Homogenous_Risk_Group
Insurance Loss Modeling Component Dimension	Dim_Ins_Loss_Modeling_Comp
Instruments Contracts Dimension	Dim_Instrument_Contract
Insurance Broker Role Dimension	Dim_Insurance_Broker_Role
Insurance Scalar Factor Dimension	Dim_Insurance_Scalar_Factor
Dimension Legal Entity Group	Dim_Legal_Entity_Group
Line Of Business Dimension	Dim_Lob
Methodologies Master	Dim_Methodologies
Mitigant Dimension	Dim_Mitigant
Organization Structure Dimension	Dim_Org_Structure
Participation Type Dimension	Dim_Participation_Type
Party Dimension	Dim_Party
Dimension Policy	Dim_Policy

Entity Name	Table Name
Dimension Policy Party	Dim_Policy_Party
Portfolio Type Dimension	Dim_Portfolio_Type
Dimension Premium Payment Type	Dim_Premium_Payment_Type
Product Dimension	Dim_Product
Product Type Dimension	Dim_Product_Type
Reinsurance Progessive Layer Dimension	Dim_Reins_Progressive_Layer
Reinsurance Progessive Section Dimension	Dim_Reins_Progressive_Section
Reinsurance Contract Dimension	Dim_Reinsurance_Contract
Reinsurance Program Dimension	Dim_Reinsurance_Program
Dimension Reinsurance Risk Type	Dim_Reinsurance_Risk_Type
Reinsurance Treaty Type Dimension	Dim_Reinsurance_Treaty_Type
Dimension Reinsurer Type	Dim_Reinsurer_Type
Run Dimension	Dim_Run
Spv Authorization Options Dimension	Dim_Spv_Authorization_Options
Dimension Spv Trigger Type	Dim_Spv_Trigger_Type
Sub Account Dimension	Dim_Sub_Account
Dimension Underwriting Model Type	Dim_Underwriting_Model_Type
Valuation Method Dimension	Dim_Valuation_Method

35.1.8 FCT Table Population

The sequence of data population for FCT tables is as follows

Entity Name	Table Name
Fact Business Performance Details	Fct_Business_Performance_Detl
Fact Capital Addon	Fct_Capital_Addon
Fact Claim Details	Fct_Claim_Details
Fact Claims Inflation Rate	Fct_Claims_Inflation_Rate
Fact Counterparty Credit Risk Exposure	Fct_Counterparty_Cr_Exposure
Fact Entity Parent Information	Fct_Entity_Parent_Info
Fact Financial Elements	Fct_Financial_Element
Fact Fund Capital Details	Fct_Fund_Capital_Details
Fact Homogeneous Risk Group	Fct_Homogeneous_Rsk_Grp_Detail
Fact Homogeneous Risk Group Cash Flow	Fct_Hrg_Cashflow
Fact Insurance Internal Models Risk Details	Fct_Ins_Internal_Mdl_Risk_Dtls
Fact Insurance Scalar Factors	Fct_Ins_Risk_Scalar_Factors

Entity Name	Table Name
Fact Insurance Risk Valuation Factors	Fct_Ins_Risk_Valuation_Factors
Fact Insurance Correlation Matrix	Fct_Insurance_Correlation_Mtrx
Fact Insurance Recoverables	Fct_Insurance_Recoverables
Fact Insurance Risk Details	Fct_Insurance_Risk_Details
Fact Insurance Risk Summary	Fct_Insurance_Risk_Summary
Fact Intragroup Transaction Details	Fct_Intra_Group_Txn_Details
Fact Legal Entity Details	Fct_Legal_Entity_Details
Fact Legal Entity Group Details	Fct_Legal_Entity_Group_Details
Fact Mitigants	Fct_Mitigants
Fact Own Fund Balances	Fct_Own_Fund_Balances
Fact Own Fund Details	Fct_Own_Fund_Details
Fact Policy Details	Fct_Policy_Details
Fact Portfolio Asset Data	Fct_Portfolio_Asset_Data
Fact Product Riders	Fct_Product_Rider_Details
Fact Product Statistics	Fct_Product_Statistics
Fact Reinsurance Policy Details	Fct_Reinsurance_Policy_Details
Fact Reinsurance Spv Details	Fct_Reinsurance_Spv_Details
Fact Run-Off Detail	Fct_Run_Off
Fact Development Year-Wise Run-Off Detail	Fct_Run_Off_Development_Year
Fact Standard Insurance Lob	Fct_Std_Insurance_Lob
Fact Sub Account Details	Fct_Sub_Account_Details
Fact Sub Account Mitigant Mapping	Fct_Sub_Acct_Mitigant_Mapping
Exchange Rate History	Fsi_Exchange_Rate_Hist
Run Parameters	Run_Parameters

35.1.9 Input Data Preparation

35.1.9.1 Assumptions regarding data input

- The currency conversion is expected to be handled by the ETL layer. The data, specifically for measure columns is assumed to be converted into local currency and then loaded in the system.
- User needs to make changes in case NCY / RCY columns are required.
- The post offset amount in FCT_MARKET_RISK_EXPOSURES has to be a download after taking into account any offsetting logic that might be applicable, or the number of underlying in the contract. This value should be obtained after position conversion (if applicable).
- Scope of data extracts excludes the consolidation process. If an execution is done for the group entity, the data (for example: capital, GL based reports for composites) for group entity should be uploaded separately. Data for group entity shall not be derived from solo entity's data.

35.1.9.2 General Data Preparation Guidelines

- Granular Risk numbers are primarily stored against HRG, Policy, Sub account, credit exposures and so on. Depending on the requirement, you need to provide data as a download.
- Intra group transaction types and respective tables - The following Intra group transaction types are sourced from FCT INTRA GROUP TXN DETAILS:
  - Bonds / Debt – collateralized
  - Bonds – uncollateralized
  - Equity type – dividends
  - Equity type – shares / participations
  - Equity type – others
  - Debt - uncollateralized

However, internal reinsurance between (re)insurance undertakings of a group will be sourced from FCT REINSURANCE POLICY DETAILS.

- On role of entity in each data table, all the data elements (For example, table FCT_POLICY_DETAILS) are for an entity which has done the actual transaction (in this case sold policy) or to whom transaction belongs to and not the group entity. Hence if requirement is for group entity report, all the tables should be populated with Group Entity as respective entity and not the individual entity.
- Fct_Reinsurance_Policy_Details stores the data at insurance contract level. Progressive layer, progressive section number, standard line of business and reinsurer, apart from contract sequence number, identify the reinsurance contract. Few measure values like Sum Insured, Underwriting model amount will remain the same for above combination of parameters, whereas measures like reinsurance recoverable, premium provisions and claim provisions will differ as per requirement. For example, reinsurance recoverable would strongly depend on Line of Business and/or reinsurer.
- The user needs to specify the calendar in DIM FINANCIAL CALENDER. This will be used to calculate Year till date (YTD) numbers. Calendars are attached to entity. Calendar of Legal entity for which RUN is executed will be used for all the reports.
- In case of Fct_Business_Performance_Detl, though F_POLICY_FPS_IND, N_BUSINESS_CLASS_SKEY, N_ECONOMIC_ZONE_SKEY are part of primary key. Populate the skeys as -1 and f_policy_fps_ind as "N" in case of dashboard or wherever regulatory reporting is not required.
- In DIM_SUB_ACCOUNT, A more granular level than the account/instrument (asset) is arrived constituting Account Number or Instrument Code, Portfolio Type Code, Collateral Pledged Purpose Code. (v_account_number/v_instrument_code, v_portfolio_type_code, v_coll_pledge_purpose_code). A sub account code is assigned for the said combination. The details of this sub account have to be given in FCT SUB ACCOUNT DETAILS.

  For example, in FACT INSURANCE RISK SUMMARY the dimension dim INSURANCE RISK TYPE contains risk types(mortality risk, lapse risk), DIM INSURANCE RISK MEASURE contains definition of the measure(Gross SCR,NET SCR,ASSETS) and DIM CAPITAL COMPUTATION METHOD contains method used for computing risk numbers, which defines the way business is carried out.
- In case of fct_insurance_risk_details, though n_std_major_lob_skey,n_region_skey,n_country_skey are part of primary keys, Populate the skeys as '-1' wherever regulatory reporting is not required.

  For example in report SCRB3E, region and major lob are used for reporting geographical diversification, but country is not required. Hence this column should be populated as '-1'.

- Solvency utilizes few tables from OIDF (Oracle Insurance Data Foundation) which are common to many applications and hence their data storage is arrived according to the requirement of these applications. Hence all the columns of following table are not mandatory for Solvency II application. However, since these are functionally rich table structures, you can make use of these placeholders suitably if required. Also relevance of columns from Solvency II perspective should be referred in Download Specifications.
  - Dim_Account
  - Dim_Credit_Rating
  - Dim_Exposure
  - Dim_Instrument_Contract
  - Dim_Party
  - Dim_Product

35.1.10 Run Overview

RUN defines the purpose of data execution. Few examples are Solvency Run, Regulatory Reporting Run and so on.

In DIM RUN table, Run Surrogate Key (Run Skey) is the primary attribute in DIM RUN Table. This is to identify a particular set of data, which is defined by combination of RUN ID and Execution Date for every execution or RUN EXECUTION IDENTIFIER.

- In each of the FCT table, data is characterized by combination of RUN SKEY, Business Type, Legal Entity and scenario code as appropriate. Details of the same are given below.
  - Run Skey: is a Surrogate key for a particular processed data set. This is a numeric field and is referenced to the table DIM_RUN that contains the list of all such run surrogate keys. This table is required to be updated with the N_Run_Skey each time a report for any new instance is to be generated assuming other paramrters constant. Multiple executions on same FIC MIS DATE will have different run skey.
  - FIC_MIS_DATE/Extraction Date/Reporting Date: is a common date identifier for the records across all the tables for which you want to generate the report.
  - Business Type: can be Life, Non Life or Composite. This is not part of DIM RUN table, but all FCT tables have these as an important identifier.
  - Entity: is the list of legal entities for the institute is provided in the table DIM_ORG_STRUCTURE.
  - Scenario: the Various Scenarios used by entity for testing uncertain future developments. List of Scenarios needs to be stored in DIM SCENARIO table.
- In DIM RUN, F REPORTING FLAG should be marked 'Y' for all the run executions which will be used for reporting. If there are multiple executions for the same RUN ID and FIC MIS DATE, the reporting flag should be used to highlight one specific execution for reporting. It is assumed that single RUN will contain all the information required. Multiple scenarios and multiple FIC MIS DATE for a single RUN ID is assumed to provide the required flexibility.
- Scenario and Baseline data should be stored in the same run and are differentiated using scenario code in the fact table.
- For each Run execution, FCT_LEGAL_ENTITY_GROUP_DETAILS will have details of Legal Entity. If the run executed is SOLO execution then there will be one row entry in the stated table and lead entity and legal entity will have same entries. If the run executed is COMPOSITE execution then there will be 'n' rows entry in the stated table (where n = number of entities involved in the run), lead entity attribute will have entry of the Group entity and legal entity attribute will have entries of all the child

entities to the group entity. Parent child relationship of the entities will be available in DIM_ORG_STRUCTURE.

- Executions can be done at two different levels, composite as well as for one business segment entity. For each type of the execution composite and single business segment data should be separately calculated, solo executions will not be consolidated for composite execution.

For the list of staging area tables and attributes that can be used for QMR, refer the Oracle_Insurance_Solvency_II_QMR_Interface_-_DL_Specs_4 spreadsheet.

36 Data Quality Rules Execution

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.

This chapter includes the following topics:

36.1.1 Data Quality Framework

Data Quality Framework consists of a scalable rule-based engine which uses a single-pass integration process to standardize, match, and duplicate information across global data. Data Quality Framework within the Infrastructure system facilitates you to define rules and execute them to query, validate, and correct the transformed data existing in an Information Domain.

You can access Data Quality Framework by expanding the Data Integrator Framework within the Unified Metadata Manager section in tree structure of LHS menu.

Data Quality Framework consists of the following sections:

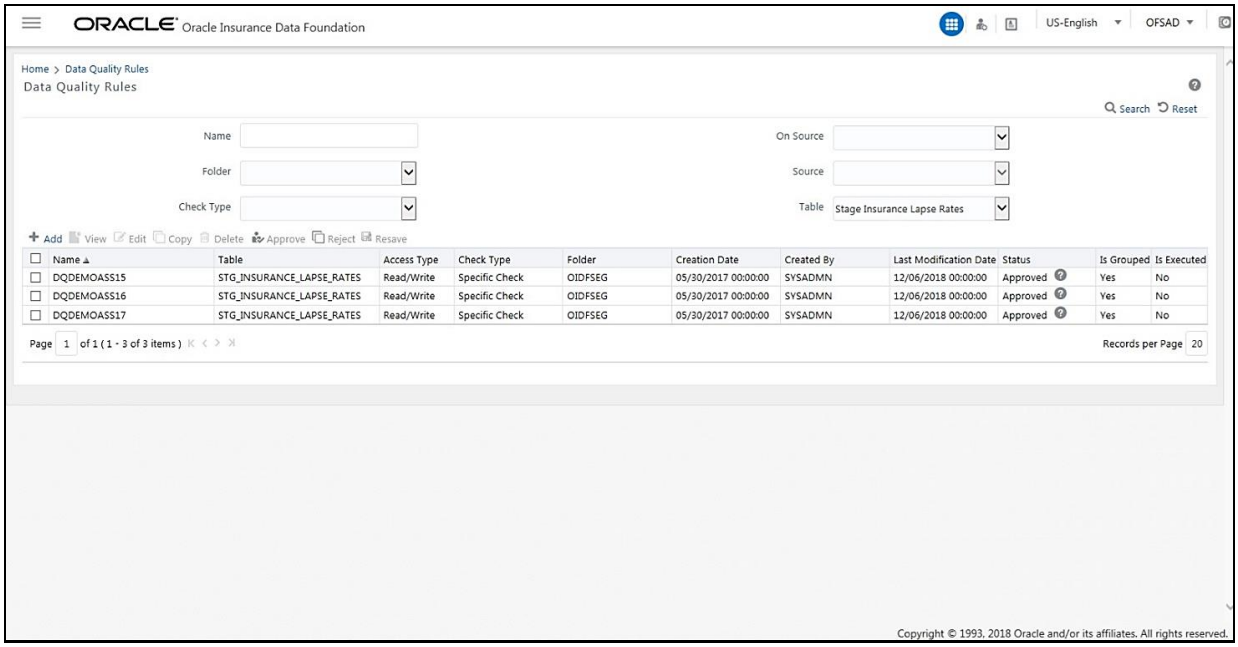
- Data Quality Rules
- Data Quality Groups Summary

36.1.1.1 Data Quality Rules

Data Quality Rules within the Data Integrator framework of Infrastructure system facilitates you to create a DQ (Data Quality) definition and define nine specific validation checks based on Range, Data Length, Column Reference/Specific Value, List of Value/Code, Null Value, Blank Value, Referential Integrity, Duplicity, and Custom Check/Business. You can also correct data for range, column reference, list of values, null value, and blank value parameters.

The defined Data Quality Rule checks can be logically grouped and executed together. You (Business Analysts) need to have ETL Analyst function role mapped to access the Data Quality Summary framework within the Infrastructure system.

Navigation to Data Quality Rules is Data Management Framework > Data Quality Framework > Data Quality Rules.



The Data Quality Rules screen displays the list of pre-defined Data Quality Rules with the other details such as DQ Name, Table Name, Access Type, Check Type, Folder, Creation Date, Created By, Last Modification Date, and Status of the Rule. A defined rule is displayed in Saved status, until it is Approved/Rejected by the approver.An Approved rule can be grouped in order for execution and a Rejected rule is sent back to the user with the Approver comments.

You can add, view, modify, copy, approve/reject, or delete Data Quality Rules within the Data Quality Rules screen. You can also make use of Search and Pagination options to search for a Data Quality Rule based on DQ Name, Table Name, Folder, or Check Type and view the existing Data Quality Rules within the system.

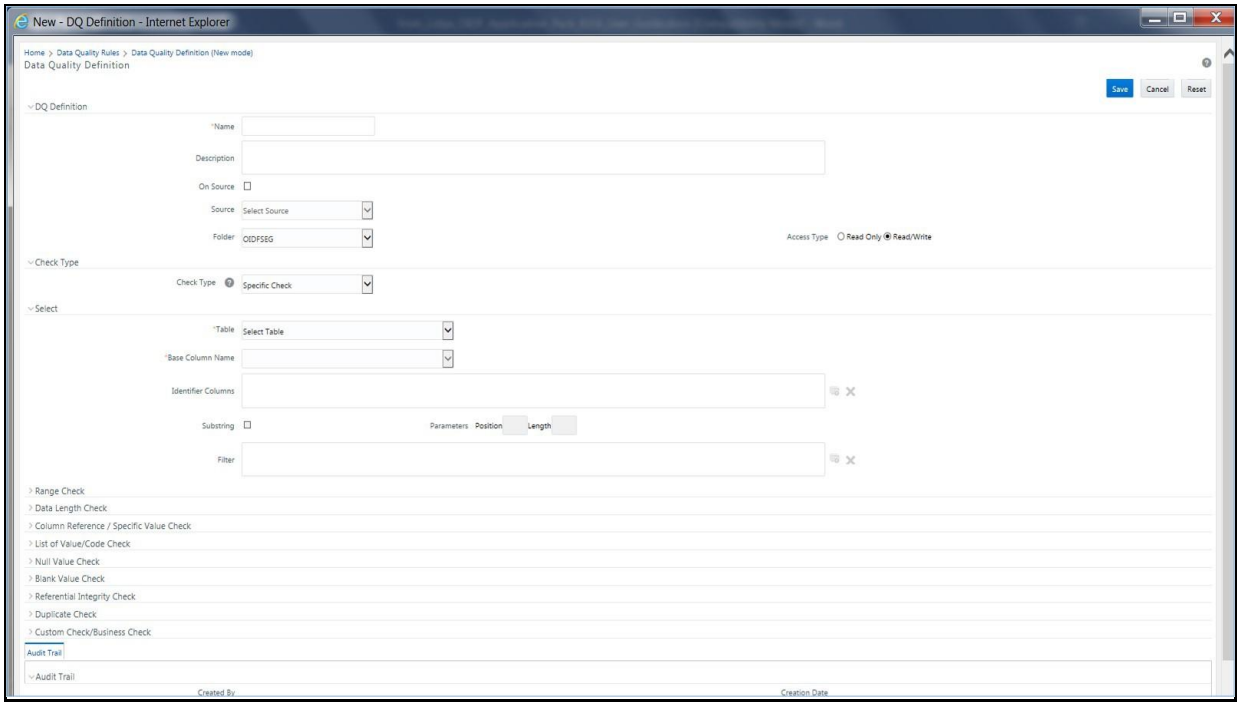
For more information, refer to Oracle Financial Services Analytical Applications Infrastructure User Manual.

36.1.1.2 Create Data Quality Rule

You can create a Data Quality Rule definition by specifying the DQ Definition details along with the type of validation check on the required table and defining the required validation conditions to query and correct the transformed data.

To create Data Quality Rule in the Data Quality Rules screen:

1. Click Add button in the Data Quality Rules tool bar. Add button is disabled if you have selected any check box in the grid. The Data Quality Definition screen is displayed.



2. In the DQ definition section, perform the following:
   - Enter the Name by which you can identify the DQ definition.
   - Enter a description or related information about the definition.
   - Select the Folder (available for selected Information Domain) from the drop down list.
   - Select the Access Type as either Read Only or Read/Write.
3. Select the Check Type from the drop down list. You can mouse-over i icon for information.
   - Select Specific Check, if the defined conditions are based on individual checks on a single column.
   - Select Generic Check, if the defined conditions are based on multiple columns of a single base table. These checks are not pre-defined and can be specified (user-defined) as required.

   If Specific Check is selected, perform the following:
   - Select Table Name and Base Column Name from the drop down list. The list displays all the tables which are marked for Data Quality Rule in a data model, which has the table classification property code set to 340.
   - (Optional) If you have selected Base Column of type Varchar/Char, select the Substring check box, enter numeric values in Parameters Position and Length fields.
   - Click the below button and define the Filter condition using the Specify Expression screen.
   - Define the required Validation Checks by selecting the appropriate grid and specify the details. You can define nine specific validation checks based on Range, Data Length, Column Reference/Specific Value, List of Value/Code, Null Value, Blank Value, Referential Integrity, Duplicity, and Custom Check/Business.

For more information, refer to Oracle Financial Services Analytical Applications Infrastructure User Manual.

NOTE A minimum of one Validation check must be defined to generate a query.

4. Click Generate Query. The details are validated and the validated query along with the status is displayed in the Generated Query section.
   - If Generic Check is selected, perform the following:
     - Select Table Name from the drop down list. The list displays all the tables which are marked for Data Quality Rule in a data model, which has the table classification property code set to 340.
     - Click the below button and define the Filter condition using the Specify Expression screen.
     - Click Add button in the Condition grid. The Specify Expression screen is displayed. Define the Condition expression.

       The Expression is displayed with the "IF" and "Else" conditions along with the Severity status as either Error or Warning. You can change the Severity by selecting from the drop down list.

> **NOTE** You can add an Assignment only when the Severity is selected as Warning. Assignments are added when you want to correct or update record(s) in base column data / selected column data. There can be one or more assignments tagged to a single condition. However, selecting severity as Error indicates there are no corrections and only facilitates in reporting the quantity of bad records.

5. Select the check box adjacent to the required Condition expression and click Add button in the Assignment grid. The assignment details are populated.

> **NOTE** You can add an Assignment only if the Severity is Warning. There can be one or more assignments tagged to a single condition.

6. Specify the Assignment details as tabulated.

Field	Description
Column Name	Select the Column Name from the drop down list.
Assignment Type	Select the Assignment Type as one of the following: No Assignment: is the default selected assignment which does not have any target column update, but the message details are pushed. Direct Value: enter the Assigned Value Another Column: select the required Column as Assigned Value from the drop down list. Code: select the required Code as Assigned Value from the drop down list if any code / leaf values exist for the selected base column. If not, you are alerted with a message indicating that No Code values exist for the selected base column.
Assignment Value	Select the Assignment Value from the drop-down list according to the Assignment Type selected.
Message Severity	Select the Message Severity as either 1 or 2 from the drop down list.

Field	Description
Message	Select the required Message for the Severity from the drop down list.

7. You can also add multiple assignments by clicking Add button in Assignment grid.

> **NOTE** Minimum of one condition needs to be defined to save the Rule.

8. Click Save. The defined Data Quality Rule definition is displayed in the Data Quality Rule Summary screen with the status as "Saved".

36.1.1.3 View Data Quality Rule

You can view individual Data Quality Rule definition details at any given point.

To view the existing Data Quality Rule definition in the Data Quality Rule Summary screen:

1. Select the check box adjacent to the required DQ Name.
2. Click View button from the Data Quality Rules tool bar.
3. The DQ Definition screen displays the details of the selected Data Quality definition. The Audit Trail section at the bottom of DQ Definition screen displays metadata information about the Data Quality Rule defined.

36.1.1.4 Modify Data Quality Rule

You can update the existing Data Quality Rule definition details except for the Definition Name, Table, and Base Column selected.

To update the required Data Quality Rule definition details in the Data Quality Rule Summary screen:

1. Select the check box adjacent to the required DQ Name.

> **NOTE** You can only edit those rules which have status either as Saved or as Rejected.

2. Click Edit button from the Data Quality Rules tool bar. The Edit button is disabled if you have selected multiple DQ Names. The DQ Definition screen is displayed.
3. Update the details as required.

   For more information, see Create Data Quality Rule section.
4. Click Save to update the changes.

36.1.1.5 Copy Data Quality Rule

You can copy the existing Data Quality Rule to quickly create a new DQ definition based on the existing rule details or by updating the required parameters.

To copy an existing Data Quality Rule definition in the Data Quality Rule Summary screen:

1. Select the check box adjacent to the required DQ Name in the list whose details are to be duplicated.

2. Click Copy button from the Data Quality Rules tool bar. Copy button is disabled if you have selected multiple check boxes. The DQ Definition screen is displayed.
3. Edit the DQ definition Name and other details as required. For more information, see Create Data Quality Rule section.
4. Click Save. The defined Data Quality Rule definition is displayed in the Data Quality Rule Summary screen with the status as "Saved".

36.1.1.6 Approve/Reject Data Quality Rule

You (Authorizer) can Approve a pre-defined Data Quality Rule definition for further execution or Reject an inappropriate DQ definition listed within the Data Quality Rule Summary screen. You should be mapped to DQ Authorizer function role to Approve or Reject a DQ definition.

To Approve/Reject Data Quality Rule in the Data Quality Rule Summary screen:

1. Select the checkbox adjacent to the required DQ Name. Ensure that you select the " Saved" DQ definition based on the Status indicated in the Data Quality Rules grid.
2. Perform one of the following:
   - To Approve the DQ definition, click Approve button. The User Comments screen is displayed. Enter the notes or additional information to the user and click OK. The selected DQ definition is approved and a confirmation dialog is displayed.
   - To Reject the DQ definition, click Reject button. The User Comments screen is displayed. Enter the notes or additional information to the user and click OK.

     The selected DQ definition is rejected and a confirmation dialog is displayed.

> **NOTE** The authorizer can approve/reject only one definition at a time.

The Approved/Rejected status of the DQ definition is indicated in the Status column of the Data Quality Rule Summary screen. You can mouse-over the i button to view the Approver comments in a pop-up.

36.1.1.7 Delete Data Quality Rule

You can remove Data Quality Rule definition(s) which are created by you and which are no longer required in the system by deleting from Data Quality Rule Summary screen.

1. Select the check box adjacent to the required DQ Name whose details are to be removed.
2. Click Delete button from the Data Quality Rules tool bar.
3. Click OK to confirm deletion.

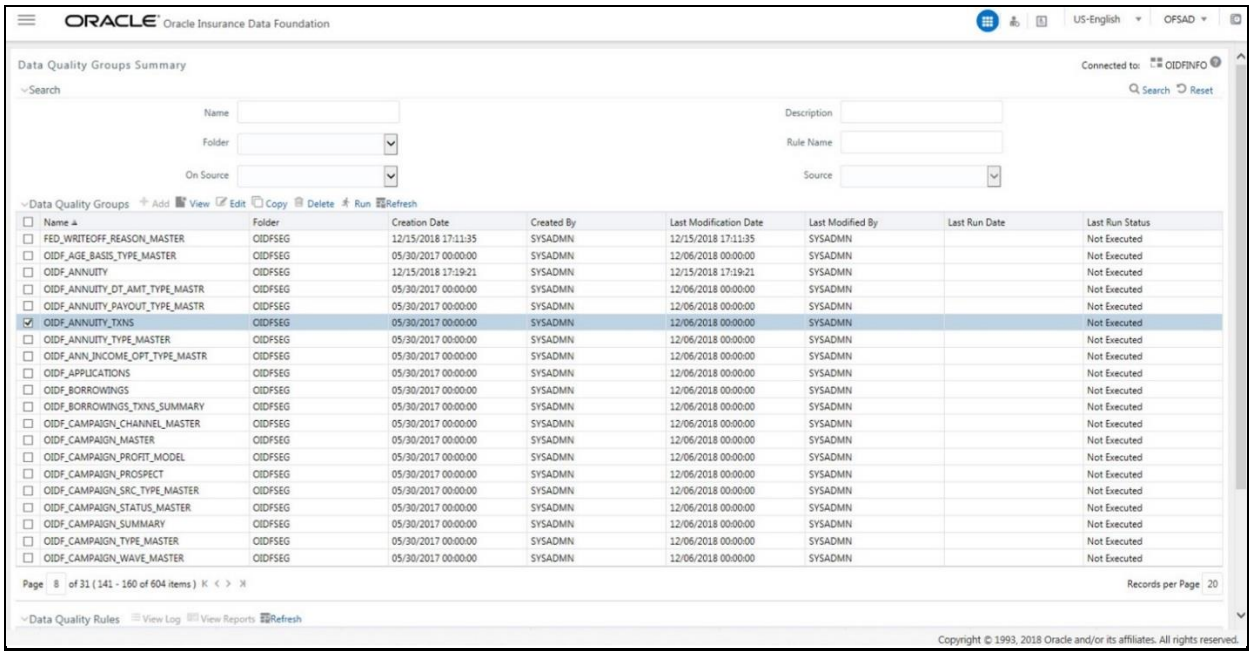
36.1.2 Data Quality Groups Summary

Data Quality Groups Summary within the Data Integrator framework of Infrastructure system facilitates you to logically group the define DQ definitions and schedule for execution. DQ definitions can be executed either through Data Quality Groups Summary screen of Data Integrator framework or in Batch Execution screen of Operations module.

When a Data Quality Group is executed for processing, the details of the execution are captured in a log file. You (Business Analysts) need to have ETL Analyst function role mapped to access the Data Quality Summary framework within the Infrastructure system. You can access Data Quality Groups Summary by

expanding the Data Quality framework within the Unified Metadata Manager section in tree structure of LHS menu.

Navigation to Data Quality Groups Summary is Data Management Framework > Data Quality Framework > Data Quality Groups Summary.



The Data Quality Groups Summary screen displays the list of pre-defined Data Quality Groups with the other details such as Group Name, Description, Creation Date, Created By, Last Modification Date, and Last Modified By. You can Create and Execute DQ Group definitions and view, modify, copy, or delete DQ Group definitions within the Data Quality Groups Summary screen.

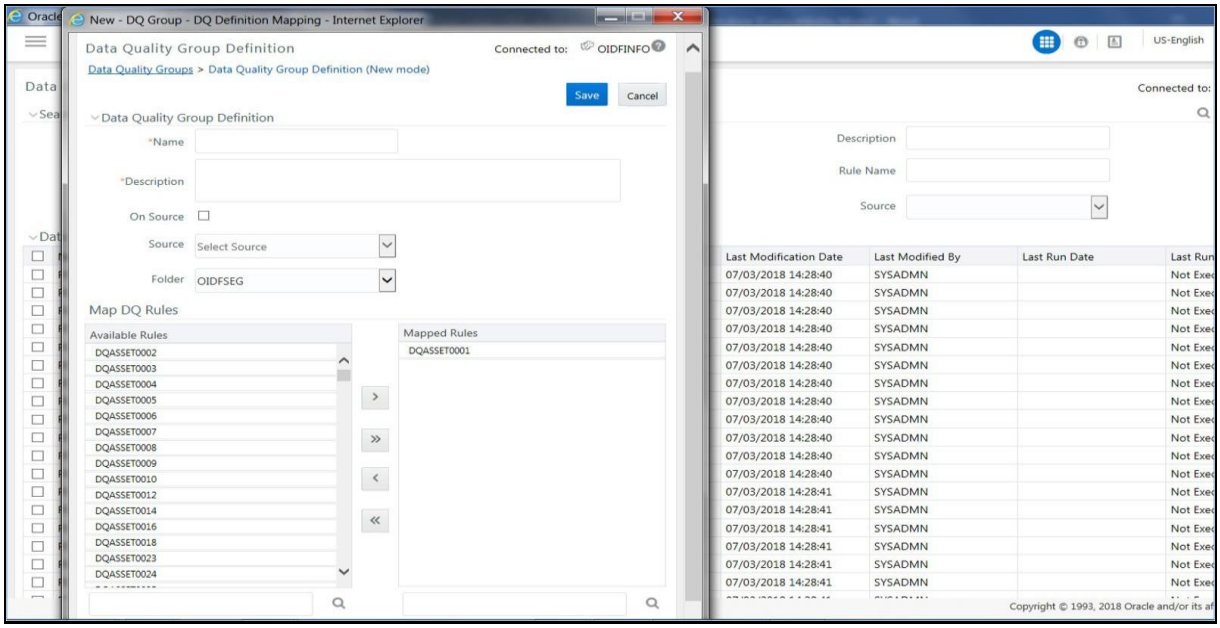
You can also make use of Search and Pagination options to search for a DQ Group definition based on Group Name, Description, or Rule Name and view the existing DQ Group definitions within the system.

36.1.2.1 Create Data Quality Group

You can create a DQ Group definition by defining the DQ Definition details and mapping the required DQ Rules which are authorized and approved within the system.

To create DQ Group in the Data Quality Groups Summary screen:

1. Click Add button in the Data Quality Groups tool bar. Add button is disabled if you have selected any check box in the grid. The Data Quality Group Definition screen is displayed.



2. In the Data Quality Group Definition section, perform the following:
   - Enter the Group Name by which you can identify the DQ Group.
   - Enter a description or related information about the DQ Group.
   - Select the Folder (available for selected Information Domain) from the drop down list.
3. In the Map DQ Rules section, perform the following:
   - Select the required DQ Rule from the Available Rules list and click Select. You can also search to select a specific DQ Rule by entering the required keyword and clicking the Find button.
   - To select all the listed DQ Rules, click Select All.

     You can also deselect a DQ Rule by selecting from the Mapped Rules list andclicking Deselect or clicking Deselect All to deselect all the mapped rules. You can search to deselect a specific DQ Rule by entering the keyword and clicking the Find button. S
4. Click Save. The defined DQ group is listed in the Data Quality Rule Summary screen and can be executed for processing.

36.1.2.2 Execute Data Quality Group

You can execute a defined DQ Group Definitions along with the mapped Rules and validation checks in the Data Quality Rule Summary screen. You can also execute a DQ Group in the Batch Execution screen of Operations module.

To Execute a DQ Group in the Data Quality Rule Summary screen:

1. Select the checkbox adjacent to the required Group Name.
2. Click Execute button from the Data Quality Groups tool bar. Execute button is disabled if you have selected multiple check boxes. The Group Execution screen is displayed.
3. In the Batch Details section, perform the following:

   Select the MIS Date using the Calendar. MIS Date refers to the date with which the data for the execution would be filtered. In case MIS date is not present in the target table, execution happens ignoring the date parameter.

NOTE The DQ Batch ID is auto populated and is not editable.

Specify the percentage of Threshold (%) limit in numeric value. This refers to the maximum percentage of records that can be rejected in a job. If the percentage of failed records exceeds the Rejection Threshold, the job will fail. If the field is left blank, the default the value is set to 100%.

4. Click Execute. A confirmation message is displayed and the DQ Group is scheduled for execution.
5. Once the DQ Group is executed, you can view the details of the execution along with the log information in the

36.1.2.3 View Logs screen.

For more information, see View Data Quality Group Summary Log section.

36.1.2.4 View Data Quality Group

You can view individual Data Quality Group definition details at any given point.

To view the existing DQ Group definition in the Data Quality Group Summary screen:

1. Select the check box adjacent to the required Group Name. The mapped DQ Rules are displayed in the Data Quality Rules section.
2. Click View button from the Data Quality Groups tool bar. The Data Quality Group Definition screen displays the DQ definition details.

36.1.2.5 Modify Data Quality Group

You can update the existing DQ Group definition details except for the Group Name.

To updated the required DQ Group definition details in the Data Quality Groups Summary screen:

1. Select the check box adjacent to the required Group Name.
2. Click Edit button from the Data Quality Groups tool bar. The Data Quality Group Definition screen is displayed.
3. Update the details and click Save to update the changes.

36.1.2.5.1 Copy Data Quality Group

You can copy the existing DQ Group details to quickly create a new DQ definition based on the existing details or by updating the required parameters.

To copy an existing DQ Group definition in the Data Quality Groups Summary screen:

1. Select the check box adjacent to the required Group Name in the list whose details are to be duplicated.
2. Click Copy button from the Data Quality Groups tool bar. Copy button is disabled if you have selected multiple check boxes. The Data Quality Group Definition screen is displayed.
3. Edit the DQ Group Name and other details as required.
4. Click Save. The new DQ Group definition is displayed in the Data Quality Groups Summary screen.

36.1.2.5.2 View Data Quality Group Summary Log

You can view the execution log details of Data Quality Rules in the View Logs screen. The View Logs screen displays the details such as Check Name, Log Message, Message Date, Message Time, Total Rows, Rows Impacted, Assignment Type, Assignment Severity, and Severity Message of the executed Data Quality Rules.

To view the Data Quality Rule execution log details in the Data Quality Groups Summary screen:

1. Select the check box adjacent to the Group Name in the Data Quality Groups grid.

   The Data Quality Rules associated with the selected Group are displayed in the Data Quality Rules grid.
2. Select the check box adjacent to the DQ Name in the Data Quality Rules grid.
3. Click View Logs button. The View Logs screen is displayed.
4. In the View Logs screen, select the Information Date from the drop down list.

   Based on the selection, you can select the Group Run ID and Iteration ID from the corresponding drop-down list.
5. Click the below button from the Group Execution details tool bar.

   The Data Quality Rule Logs grid displays the execution log details of the selected Data Quality Rule. You can also click Reset button in the Group Execution details tool bar to reset the selection.

36.1.2.6 Delete Data Quality Group

You can remove the DQ Group definition(s) which are created by you and which are no longer required in the system by deleting from Data Quality Groups Summary screen.

To delete:

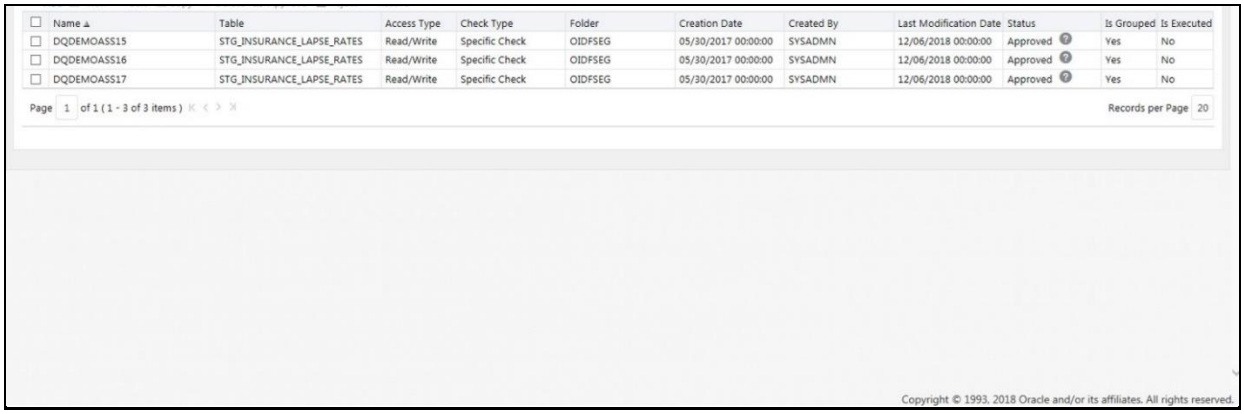
1. Select the check box adjacent to the required Group Name whose details are to be removed.
2. Click Delete button from the Data Quality Groups tool bar.
3. Click OK in the information dialog to confirm deletion.

36.1.3 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, where as in Generic Check any columns are not specified. Generic Check is useful if you have a check which is not Specific or you use IF-ELSE conditions or CASE statements.

The following screen displays the Specified DQ Rules:



36.1.4 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.

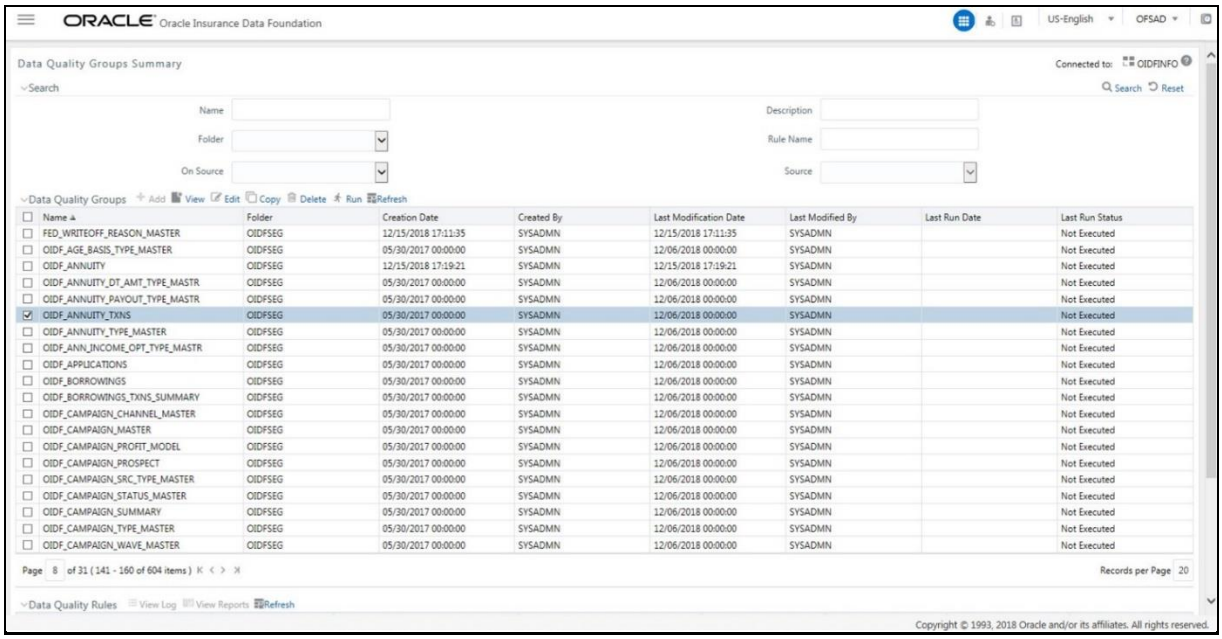
For more information regarding Group Execution, refer to the section Execute Data Quality Group as follows. The DQ_Group_Mapping spreadsheet displays the total groups and corresponding Rules mapped to that Group.

36.1.4.1 DQ Group Execution

You can execute the DQ Rules either from Data Quality Group Summary window or via a Batch execution. To execute the Data Quality Groups from the Data Quality Groups Summary window:

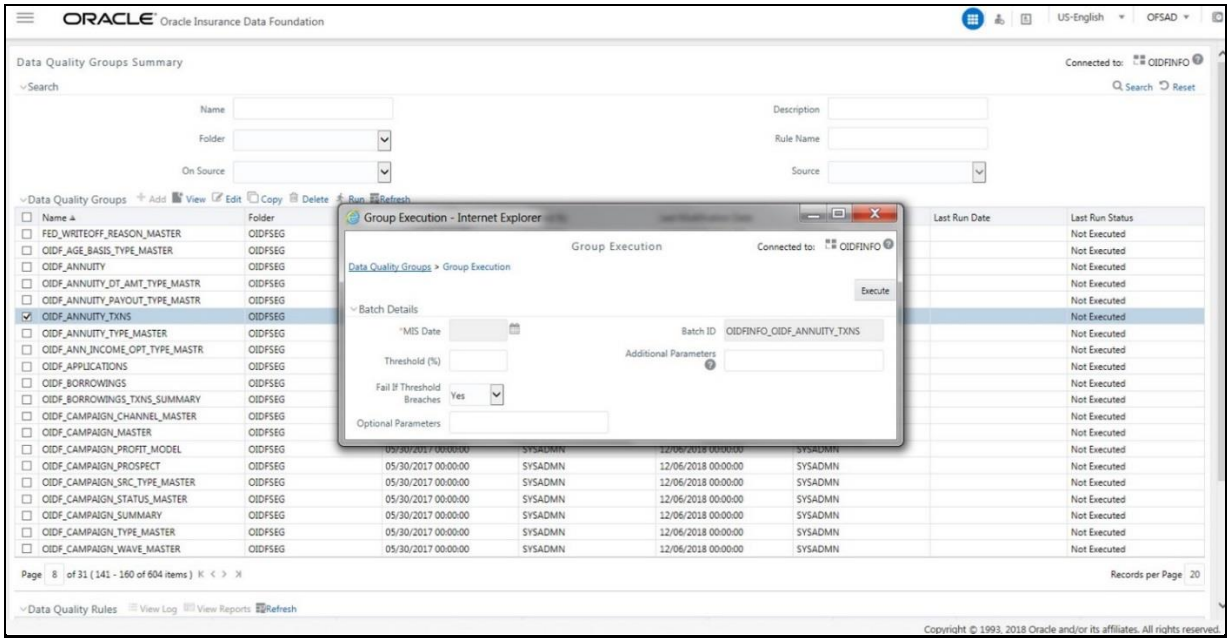
1. Select the check box adjacent to the Group, you want to execute.

   The corresponding DQ Rules are displayed in Data Quality Rules grid.



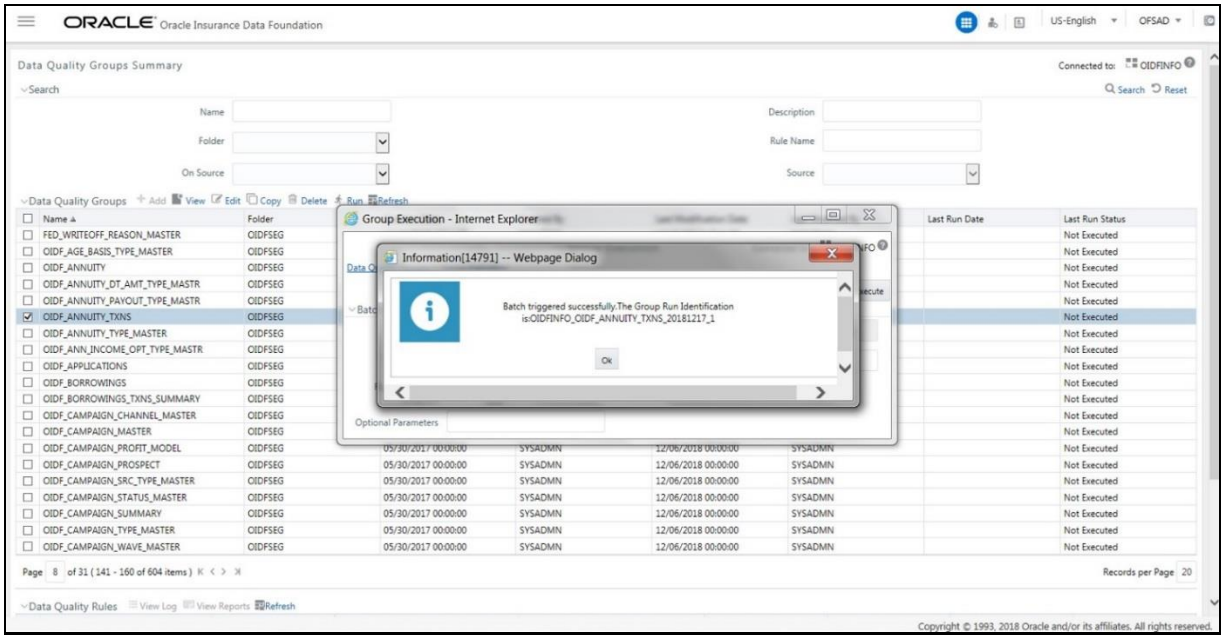
2. Click Run.

   The Group Execution window is displayed.



3. Select MIS Date.
4. Click Execute button from the Group Execution window.

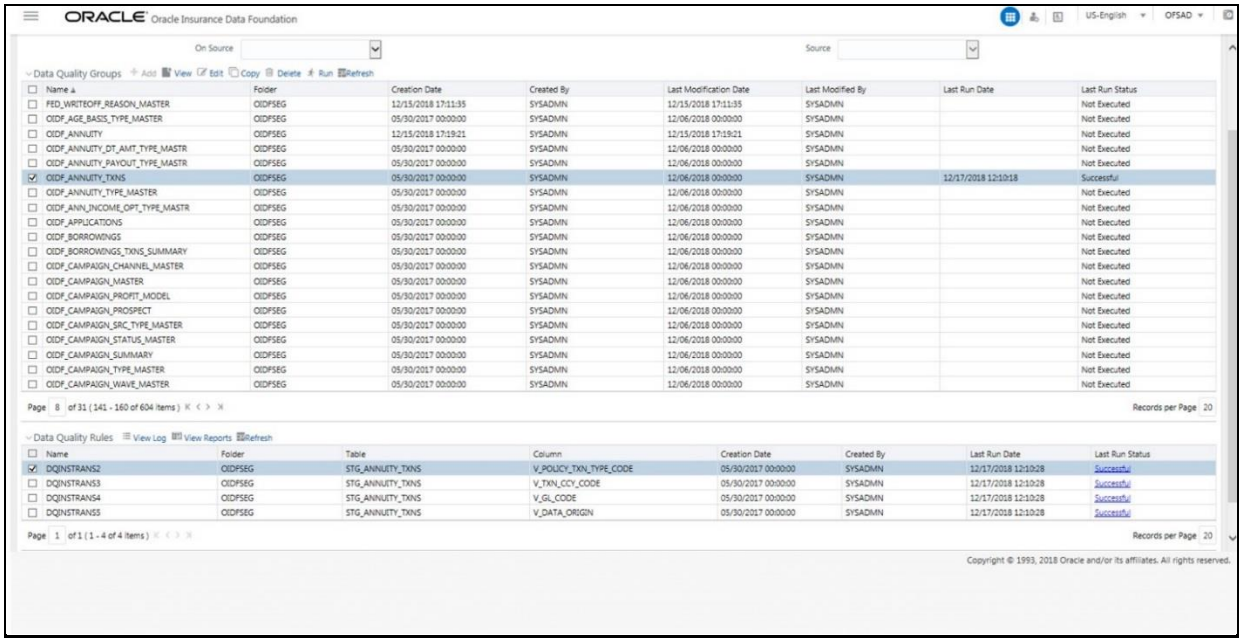
   The execution is triggered and the following message is displayed.



Once the execution is complete, you can view the details from the Data Quality Rules Summary window.

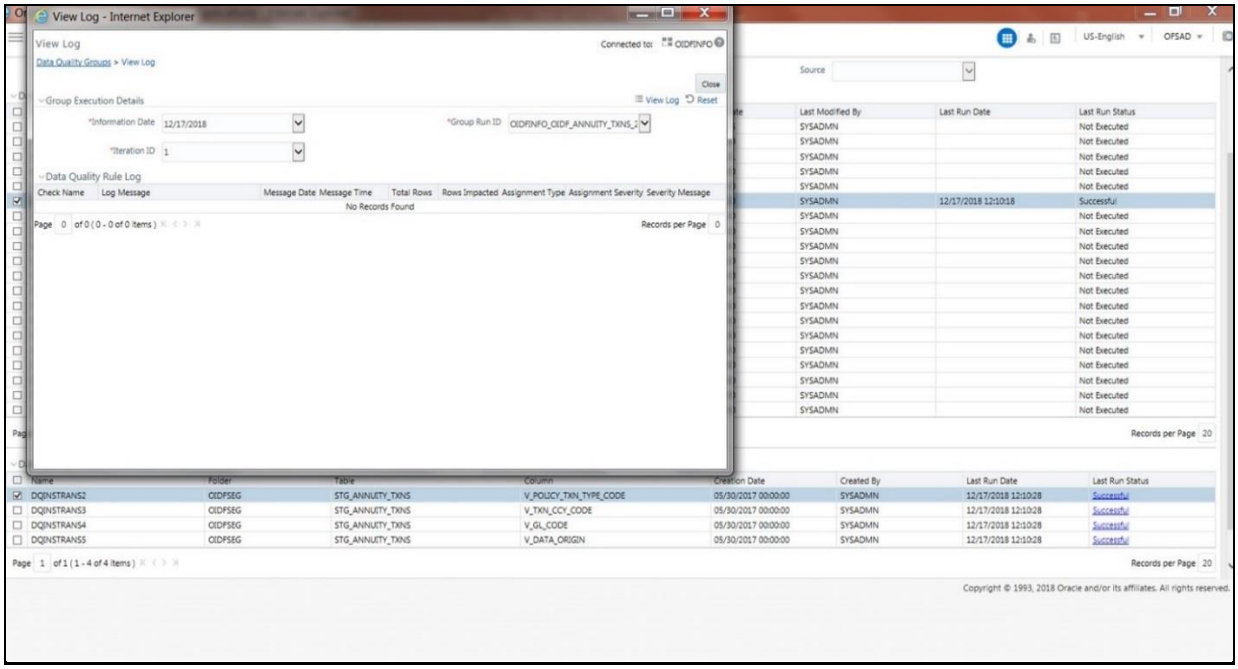
To view the execution logs:

1. Select the check box adjacent to the rule, of which you want to see the execution log.



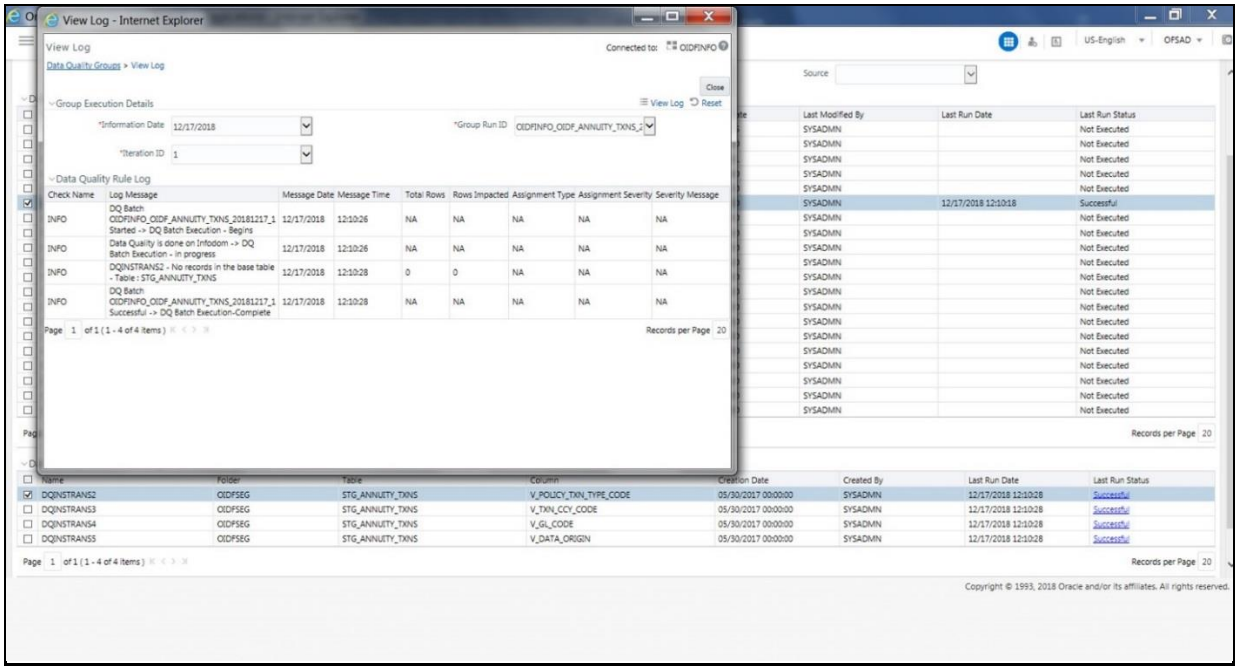
2. Click View Logs button.

   The View Logs window is displayed.



3. Select the Information Date from the drop down list.
4. Select the Group Run ID from the drop down list.
5. Select the Iteration ID from the drop down list.
6. Click the View Log button.

   The details of the selected Group Execution are displayed.



36.1.4.2 Batch Execution of DQ Rules

Create a Batch for executing DQ Rules and add a Task to the selected Batch. Add component as RUN DQ RULE and in Dynamic Parameter List, add a DQ Group.

A single Batch can have multiple number of Tasks and each Task is executing the DQ Group. Batch execution facilitates the execution of multiple Groups simultaneously.

You can also use the Include or Exclude functionality to determine which all groups have to be executed. The following batches need to be executed in OIDF.

V_BATCH_ID	V_BATCH_DESCRIPTION
<Infodom>_ALM	Data Quality batch for ALM tables
<Infodom>_CAMPAIGN	Data Quality batch for CAMPAIGN tables
<Infodom>_COLLATERAL	Data Quality batch for Collateral tables
<Infodom>_COLLECTION AND RECOVERY	Data Quality batch for Collection and Recovery tables
<Infodom>_CRM	Data Quality batch for CRM tables
<Infodom>_CUSTOMER	Data Quality batch for Customer tables
<Infodom>_EXPOSURE	Data Quality batch for EXPOSURE tables
<Infodom>_GL AND ACCOUNTING	Data Quality batch for GL and Accounting group
<Infodom>_LRM	Data Quality batch for LRM tables
<Infodom>_MARKET RISK	Data Quality batch for MARKET RISK tables
<Infodom>_MASTER	Data Quality batch for master tables
<Infodom>_MISCELLANEOUS	Data Quality batch for Miscellaneous tables
<Infodom>_OP RISK	Data Quality batch for OP RISK tables
<Infodom>_ORIGINATION	Data Quality batch for Origination tables

V_BATCH_ID	V_BATCH_DESCRIPTION
<Infodom>_PRODUCT PROCESSORS	Data Quality batch for Product Processors group
<Infodom>_RATES	Data Quality batch for Rates tables
<Infodom>_RATING	Data Quality batch for RATING tables
<Infodom>_SECURITIZATION	Data Quality batch for Securitization tables
<Infodom>_TRANSACTIONS	Data Quality batch for Transactions tables
<Infodom>_TRANSACTION SUMMARY	Data Quality batchfor Transaction Summary tables

The DQ Batches and corresponding groups are detailed in the DQ_Batch_Group_Mapping spreadsheet.

For more information regarding Batch Execution, refer to Operations module of Oracle Financial Services Analytical Applications Infrastructure User Manual.

37 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 and LDM are also listed with their descriptions.

37.1 Naming Conventions Used in OIDF LDM and PDM

ERwin Data Model consists of Logical and Physical data structures for each model file.

The following section explains the various naming conventions used in Oracle Insurance Data Foundation (OIDF) Logical Data Model (LDM) and Physical Data Model (PDM).

- OIDF Logical Data Model Naming Conventions
- OIDF Physical Data Model Naming Conventions
- Domains (PDM and LDM)

37.1.1 OIDF Logical Data Model Naming Conventions

ERwin modeler tool allows two views namely Logical view and Physical view for each model file. Accordingly, the OIDF LDM file can be viewed in logical view mode and physical view mode. Different naming conventions and standards are applied to the two views of the OIDF LDM.

> **NOTE** The physical view of the OIDF LDM is not the same as the OIDF PDM. The PDM is a distinct model and is shipped as a separate ERwin File.

- LDM Logical View Mode Entity Naming Conventions
- LDM Physical View Mode Table Naming Conventions

37.1.2 LDM Logical View Mode Entity Naming Conventions

Entities in the logical view are named in such a way that the names themselves convey the functional meaning of the entity. The first letter of each word in an entity name is capitalized (e.g. "Application Stage"). Entity names also contain common suffixes for the classes of entities listed in the table below:

Attribute names also contain common set of suffixes for the classes of attributes as listed in table below:

Class of Entity	Suffix	Example
Type	Type	Asset-Liability Type
Group	Group	Product Group
Code	Code	Collection Status Code
History	History	Account Balance History
Associative entities	Map	Customer Account Map
Unique Identifier	ID	
Reference Identifiers	Reference ID	

Class of Entity	Suffix	Example
Code	Code	
Number	Number	
Name	Name	
Description	Description	
Boolean Values	Flag	Y or N
Indicator	Indicator	More than one value ('A', 'B', 'C', and so on)
Percentage	Percent	
Rate	Rate	
Amount	Amount	
Balance	Balance	
Term	Term	
Frequency	Frequency	
Unit	Unit	
Record created by		Created By
Record modified by		Modified By
Record created on		Created On
Record modified on		Modified On
Record validity start date		From Date
Record validity end date		End Date

37.1.3 LDM Physical View Mode Table Naming Conventions

In the physical view of the OIDF LDM follows a different naming convention for entities and attributes in the model, and these reflect naming standards that are more readily acceptable to table and column naming constraints of common database systems.

The entity names in the physical view mode of the LDM are capitalized and each entity follows the following naming pattern:

- FSW_<Subject Area>_<Table Name>
- 'FSW' is a constant prefix.
- <Subject Area> is abbreviated form of the entity's primary subject area.
- <Table Name> is a meaningful name for the entity with no embedded spaces between words. Table Name may be abbreviated.

Logical View Name	Physical View Name	Description
Account Addresst	FSW_ACCT_ACCOUNT_ADDRESS	Account Address entity belonging primarily to Account (ACCT) subject area.

37.1.4 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 modeler tool 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.

- PDM Logical View Mode Naming Conventions
- PDM Physical View Mode Naming Conventions

37.1.5 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 table below:

PDM Area		Prefix	Suffix	Example
Staging (Data Sourcing)	All Tables	Stage		Stage Reinsurance Contracts Held
	Transaction Tables		Transactions	Stage Reinsurance Issued Transactions
	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

37.1.5.1 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.

PDM Area		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.

Column name prefix to indicate column datatype:

Column Data Type	Prefix
Varchar	v_
Number	n_
Date	d_
Flag	f_

Column name suffix for common classes of columns:

Column Class	Suffix
Method	_method
Percentage	_pct
Rate	_rate
Balance	_bal/_balance
Amount	_amt/_amount
Term	_term
Type	_type
Frequency	_freq

In addition, frequently occurring keywords in column names may be abbreviated as shown in the following table:

Name	Abbreviated Form	Name	Abbreviated Form
Accrual	accr	Local Currency	lcy
Account	acct	Line Of Business	lob
Accounting Currency	acy	Maximum	Max
Address	addr	Minimum	min
Adjustment	adj	Mortgage	mort
Advance	adv	Message	msg
Amount	amt	Multiplier	mult
Application	app	Number	num
Average	avg	Over Draft	od
Balance	bal	Option	opt
Business	bus	Origination	org

Name	Abbreviated Form	Name	Abbreviated Form
Currency	ccy	Percent	pct
Consolidation	cons	Payment	pmt
Customer	cust	Prepayment	ppmt
Description	desc	Product	prod
Dimension	dim	Source	src
Detail	dtl	Status	stat
Earnings at Risk	ear	Statistics	stats
End Of Period	eop	Temporary	temp
Error	err	Total	tot
Flag	flg	Transaction	txn
Frequency	freq	Value at Risk	var
Future	fut	Value	val
Forex	fx		
Generation	gen		
General Ledger	gl		
Hierarchy	hier		
History	hist		

37.1.5.2 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.

Serial No	Domain Name	Domain Description
1	Date	DATE
2	Timestamp	TIMESTAMP
3	Number	NUMBER(10)
4	Amount	NUMBER(22,3)
5	Code	NUMBER(5)
6	Flag	CHAR(1)
7	Frequency	NUMBER(5)
8	ID	VARCHAR2(25)
9	Percent	NUMBER(10,6)
10	Percent_Long	NUMBER(15,11)
11	Phone_Fax_Number	NUMBER(15)
12	Rate	NUMBER(10,6)

Serial No	Domain Name	Domain Description
13	Term	NUMBER(5)
14	Alphanumeric_Code	VARCHAR2(10)
15	Name	VARCHAR2(60)
16	Currency_Code	VARCHAR2(3)
17	Short_Description	VARCHAR2(60)
18	Description	VARCHAR2(255)
19	Account_Number	VARCHAR2(25)
20	System_Identifier	NUMBER(20)
21	Long_Description	VARCHAR2(4000)

38 APPENDIX B: Standard Data Expectations

38.1 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.

39 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.

This chapter includes the following topic:

- Batch Definition

39.1 Batch Definition

Create a batch from the OFSAAI Batch Maintenance screen as follows:

1. From the OFSAAI Home menu, navigate to Operations > Batch Maintenance.
2. In the Batch Maintenance window, Select '+' button from the Batch Name tool bar. The New Batch Definition window is displayed.
3. Enter the Batch details as tabulated.

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 should be unique across the Information Domain. The Batch Name must be alphanumeric and should not start with a number. The Batch Name should not exceed 41 characters in length. The Batch Name should not contain special characters "." and "-".
Batch Description	Enter a description for the Batch based on the Batch Name.
Duplicate Batch	(Optional) Select the check box to create a new Batch by duplicating the existing Batch details. On selection, the Batch ID field is enabled.
Batch ID (If duplicate Batch is selected)	It is mandatory to specify the Batch ID if Duplicate Batch option is selected. Select the required Batch ID from the list.
Sequential Batch	Select the check box if the Batch has to be created sequentially based on the task specified. For example, if there are 3 tasks defined in a Batch, task 3 should have precedence as task 2, and task 2 should have precedence as task 1.

4. Click Save to save the Batch definition details.

The new Batch definition details are displayed in the Batch Name section of Batch Maintenance window with the specified Batch ID.

40 APPENDIX D Reporting Code Description

40.1 Reporting Code 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 own debt and other items
SCR - B2A	Solvency capital requirement (for 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 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)
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

Report Code	Report Name
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
IGT2	IGT - Derivatives
IGT3	IGT - Internal Reinsurance
IGT4	IGT - Cost sharing, contingent liabilities, off BS items and other IGT
RC	Risk Concentration

41 APPENDIX E Template to Generate Data Dictionary and Download Specification for ERwin 9.64.x

OFSAA data models have been designed and released on 7.x version of ERwin. However, if there is a requirement to upgrade to the latest version of ERwin (9.64.x series), the existing procedure to generate DL specification using the published report templates fails in the 9.64.x version. This is because of architectural changes between the two ERwin versions. This chapter provides information about the procedure to generate data dictionary and download specification for ERwin 9.64.x, for all OFSAA data models qualified on OIDF 8.0.7.

41.1 Prerequisite

Upgrade model to 9.x series before initiating generation of the DL specification.

41.2 Procedure

Download the attached 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 and ERwin version 9.64.x.

1. Ensure that the Erwin Data Modeler version is ERwin 9.64 or later.
2. Ensure that macros are enabled in this excel before execution.
3. Open the OFSAA data model in ERwin 9.64 or a later version before executing this template.
4. Click the "UDP selection Menu" button to launch UDP selection form.
5. Select the appropriate application UDP's (optional).
6. Click the "Generate Report" button.
7. On successful execution, the following message is displayed "Report generation is Complete".
8. The download specification and the data dictionary report is located in "OFSAA_Download_Spec" worksheet.

NOTE To generate report for twoapplication UDP's 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 will throw an error. The UDP names will be fixed in the upcoming release of OIDF.

OFSAAI Support Contact Details

- Contact Infrastructure support at https://flexsupp.oracle.com if you have installed ERM and FCCM applications.
- Raise an SR in https://support.oracle.com if you have any queries related to EPM 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 which has all the revised/recently released documents.