Oracle® Insurance Data Foundation

User Guide

Release 7.4

Part No. E39352-01

April 2014

Oracle Insurance Data Foundation User Guide, Release 7.4

Part No. E39352-01

Primary Author: Aneesh Kurian

Contributing Author: Surag Ramachandran, Arpana Danayak , Sindhu Koppara

Contributor: Sunil Mathew, Guruprasad Thiruvachi, Ashok Mallegowda

Contents

Send Us Your Comments

Preface

1 Introduction to OIDF

2 Understanding OIDF

3 Dimension Loading Process

4 Customer Summary Population

5 Using OIDF

6 OIDF Interface with QMR 2.0.0.0.0

7 Data Quality Rules Execution

8 Slowly Changing Dimension (SCD)

A Naming Conventions used in OIDF Data Model

B Reporting Code Description

Send Us Your Comments

Oracle Insurance Data Foundation User Guide, Release 7.4

Part No. E39352-01

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

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

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

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

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

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

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

Preface

Intended Audience

Welcome to Release 7.4 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 7.4.1.0.0, 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 on page x for more Oracle product information.

Documentation Accessibility

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

Access to Oracle Support

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

Structure

1 Introduction to OIDF

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

2 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 and Logical Data Model, and Mapping Logical Data Model to OIDF Staging Area are explained in detail.

3 Dimension Loading Process

4 Customer Summary Population

5 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. (For details on what Download Specifications are, see Staging Area, page 2-5)

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

6 OIDF Interface with QMR 2.0.0.0.0

7 Data Quality Rules Execution

8 Slowly Changing Dimension (SCD)

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.

B Reporting Code Description

Related Information Sources

1. Technical Whitepaper on Data Model Document Generation
2. Installation Guide for Oracle Insurance Data Foundation 7.4.1.0.0
3. Installation and Configuration Guide for OFSAAI release 7.3.2.1.0

1

Introduction to OIDF

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

This chapter covers the following topics:

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

Overview

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

OIDF combines an **industry data model** for Financial Services along with a **set of management and infrastructure tools** that allows Financial Services Institutions to develop, deploy, and operate analytical solutions spanning key functional areas in Financial Services, including:

1. Enterprise Risk Management

2. Enterprise Performance Management

3. Customer Insight

4. Financial Crime and Compliance Management

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

Components of OIDF

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

Component	Purpose
Logical Data Model (LDM)	A reference model of the Financial Services domain spanning Banking and Capital Markets. This model captures the business processes of a typical Financial Services institution in detail. These primarily include core banking business activities such as Retail Banking, Corporate/Merchant Banking, Wealth Management, Trading, and Securities Services. This is a **logical data model**, which means that it cannot be readily used. The structures modeled in the OIDF LDM provide an abstract, graphical model of the Financial Services domain using Entity-Relationship modeling. It can be thought of as a detailed blueprint for organizing data within a Financial Services institution. It provides a reference guide for institutions to understand the salient data related to a specific business process. To actually use the OIDF in practice, customers deploy the OIDF Analytical Warehouse Model, which is derived from the LDM (blueprint).

Component	Purpose
Financial Services Analytical Warehouse Data Model	A physical data model that supports data sourcing and reporting related to key analytical use cases in the Financial Services 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**, which facilitates data sourcing from the bank's internal operational systems such as Lending systems, Trading Systems, Collateral Management systems, and Master Data Management systems. The **Reporting model** which facilitates the storage of outputs from analytical tools, applications, and engines in a manner that is conducive to BI reporting. The warehouse model is typically deployed into production via a set of management tools called the Oracle Financial Services Analytical Application Infrastructure (OFSAAI). **The AAI is a separate product, and is a prerequisite for OIDF** (see Infrastructure).
Supporting scripts	As part of the OIDF package, there are additionally a number of scripts provided for basic operations such as internal data movement between the staging and reporting areas.

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:

1. **Data Model**

    1. **OIDF Staging model** provides the complete data sourcing foundation for OFSAA applications. All application-specific input data requirements are captured as part of the staging data model. "OIDF Staging Model" is a combination of all staging models supplied with each OFSAA application.

    2. **OIDF Reporting model** provides the complete reporting data model common to all the OFSAA Business Intelligence (BI) applications. This includes a single set of conformed dimensions as well as unified fact tables used for cross-functional reporting. For more information, see OIDF Reporting model,

page 2-13. OIDF reporting model is the superset of all the BI-application specific reporting models.

3. **Logical Data Model (LDM)**: The OIDF Logical Data Model is independent of OFSAA analytical applications, and more aligned to the underlying business processes of a financial institution. However, there is significant content overlap in the lower-level details of the LDM entities/attributes. This is captured in the form of attribute level mappings between LDM entities/attributes and their corresponding equivalents in the OIDF staging area data model. This mapping is made available as part of the OIDF release packaging.

4. **Synchronized Releases:** The Staging and Reporting models that are part of an OIDF release are updated to reflect prior application-specific releases. This means that the latest release of OIDF (7.4.1.0.0) reflects all prior application releases across OFSAA from a data model perspective, with respect to the Staging and Reporting models.

2. **Infrastructure**

   1. The Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) is the same infrastructure that is used to deploy and manage an OIDF instance.

   2. The same collection of tools (OFSAAI) is used to manage a deployed instance of the Oracle Insurance Data Foundation (OIDF).

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

      - Model Upload
      - Unified Metadata Manager (UMM)
      - Data Quality Framework
      - T2T framework
      - Mart Management framework

OIDF Prerequisite Components and Tools

The key prerequisites for running the OIDF are as listed below:

Component	Provider	Purpose
Oracle Financial Services Analytical Applications Infrastructure version 7.3.2.1.0	Oracle	OFSAAI is the platform on which the OIDF is deployed and operated. It represents the OIDF 'runtime' environment, and consists of a number of tools used to manage the data lifecycle within OIDF, from sourcing to reporting*.
Oracle Database Enterprise Edition 11gR2	Oracle	OIDF is certified on Oracle Database releases 11gR2 and later.
ERwin data modeler version 7.3+	Computer Associates (CA)	ERwin is a data modeling tool that provides a visual environment to manage the complex enterprise data environment.

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

2

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 and Logical Data Model, and Mapping Logical Data Model to OIDF Staging Area are explained in detail.

This chapter covers the following topics:

- Background
- OIDF Architecture
- OIDF Physical Data Model
- OIDF Logical Data Model
- Mapping from Logical Data Model to OIDF Staging Area

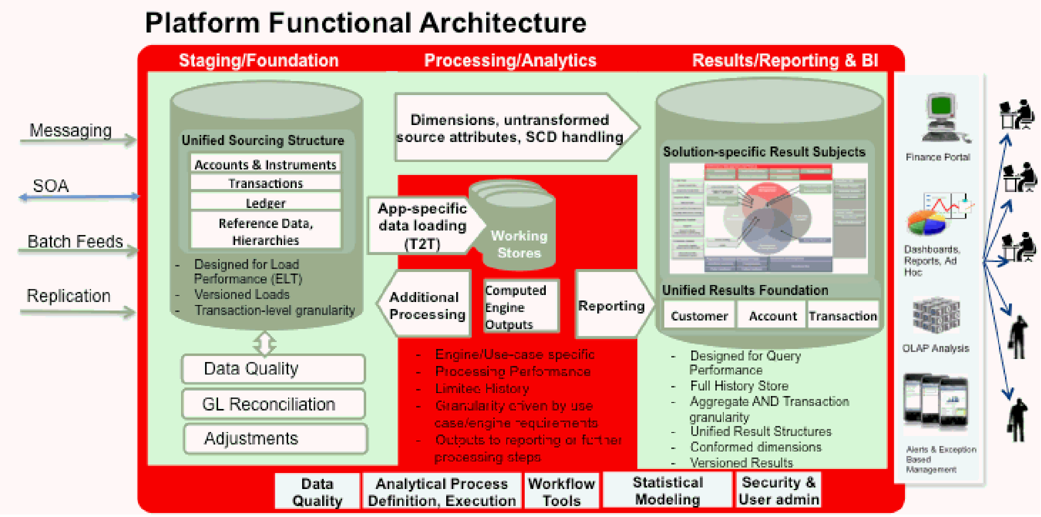
Background

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

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

OIDF Architecture

The following figure depicts the OIDF Functional Architecture.


Platform Functional Architecture
Staging/Foundation
Processing/Analytics
Results/Reporting & BI
Messaging
SOA
Batch Feeds
Replication
Unified Sourcing Structure
Accounts & Instruments
Transactions
Ledger
Reference Data, Hierarchies
- Designed for Load Performance (ELT)
- Versioned Loads
- Transaction-level granularity
Data Quality
GL Reconciliation
Adjustments
Dimensions, untransformed source attributes, SCD handling
App-specific data loading (T2T)
Working Stores
Additional Processing
Computed Engine Outputs
Reporting
- Engine/Use-case specific
- Processing Performance
- Limited History
- Granularity driven by use case/engine requirements
- Outputs to reporting or further processing steps
Solution-specific Result Subjects
Unified Results Foundation
Customer
Account
Transaction
- Designed for Query Performance
- Full History Store
- Aggregate AND Transaction granularity
- Unified Result Structures
- Conformed dimensions
- Versioned Results
Data Quality
Analytical Process Definition, Execution
Workflow Tools
Statistical Modeling
Security & User admin
Finance Portal
Dashboards, Reports, Ad Hoc
OLAP Analysis
Alerts & Exception Based Management

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.

1. **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.*

2. **A management toolkit** provided by OFSAAI that is used to manage the data repository, by providing a collection of tools and frameworks based on a common metadata foundation. *This user guide DOES NOT cover the details of the Analytical Application Infrastructure, as that is a separate product with its own documentation.*

The architecture illustrates the following key concepts:

1. A **unified data-sourcing area for analytics:** The OIDF Staging Data Model provides the basis for central, unified data-sourcing layer for a wide variety of analytical needs. The staging layer faces the operational (OLTP) and front office systems of a bank. It consists of tables to capture key business and operational data from these systems, which is then processed by one or more analytical engines.

2. A **unified reporting/consumption layer:** Analytical results can be simple to complex, quantitative and qualitative measures of a bank's Risk, Compliance, Customer and Financial Performance. The 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 as well as cross-departmental and cross-functional reporting.

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

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

    1. Metadata Management
    2. Data Quality
    3. Data Movement
    4. Scheduling and runtime operations
    5. Security/User management
    6. Analytical Process Definition and Execution

4. **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 Finance to Risk to Compliance.

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

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

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

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

The processing area tables/schemas are NOT part of the 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.

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 filesystem-based area where file-based extracts (operational images) from source systems are staged prior to loading into a target schema.
3rd Normal Form Operational Data Store/Enterprise Data Warehouse	Does not provide a physicalized model for a 3NF store. The LDM can be physicalized as a 3NF store if desired. Operational/fine-grained reporting will be fulfilled from the reporting area.	Typically a 3rd normal form repository of 'atomic data' that is loaded via ETL from the staging area (Bill Inmon approach)
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.

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:

1. **Readily Deployable:** The Physical Data Model is a readily deployable physical schema. It is provided as an ERwin data model file (for details on ERwin, see http://erwin.com/products/modeler/) and consists of tables grouped into distinct subject areas depending on function. The tables are either used to gather source data (Staging Area), or as containers of outputs/results from analytical processing and engines for reporting purposes (Reporting Area).

2. **Use-case Driven:** The OIDF Physical Data model is driven by a set of clearly identified analytical use cases spanning Risk, Performance, Customer Insight, and Compliance.

3. **Extensible:** While the OIDF Physical Data Model satisfies a very large number of analytical use cases across Risk, Finance, Marketing, and Compliance subject areas, customers may find the need to customize the model for a specific installation. These customizations may be done in accordance with guidelines published in Using OIDF section, page 5-1 of this manual.

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

Staging Data Model

Overview/Design

The Common Staging Area Model (CSA) represents the point of entry of data into the 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:

1. **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 Basel II, Market Risk Analytics, ALM and others.

   These mappings can readily be extracted into a Download Specification, page 5-3, which lists the data demands for a specific analytical use case. An example is shown below:

Download Specification - Example

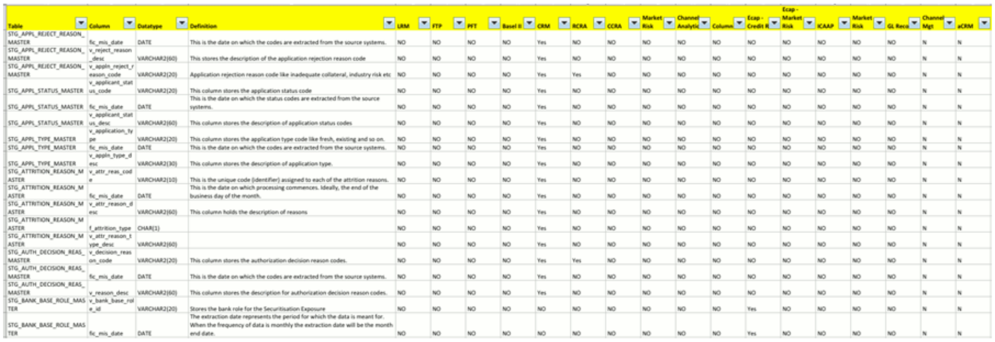


Table	Column	Datatype	Definition	LRM	FTP	PFT	Basel II	CRM	RCRA	CCRA	Market Risk	Channel Analytic	Column	Ecap - Credit R	Ecap - Market Risk	ICAAP	Market Risk	GL Reco	Channel Mgt	aCRM
STG_APPL_REJECT_REASON_MASTER	fic_mis_date	DATE	This is the date on which the codes are extracted from the source systems.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_APPL_REJECT_REASON_MASTER	v_reject_reason_desc	VARCHAR2(60)	This stores the description of the application rejection reason code	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_APPL_REJECT_REASON_MASTER	v_appln_reject_reason_code	VARCHAR2(20)	Application rejection reason code like inadequate collateral, industry risk etc	NO	NO	NO	NO	Yes	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_APPL_STATUS_MASTER	v_applicant_status_code	VARCHAR2(20)	This column stores the application status code	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_APPL_STATUS_MASTER	fic_mis_date	DATE	This is the date on which the status codes are extracted from the source systems.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_APPL_STATUS_MASTER	v_applicant_status_desc	VARCHAR2(60)	This column stores the description of application status codes	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_APPL_TYPE_MASTER	v_application_type	VARCHAR2(20)	This column stores the application type code like fresh, existing and so on.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_APPL_TYPE_MASTER	fic_mis_date	DATE	This is the date on which the codes are extracted from the source systems.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_APPL_TYPE_MASTER	v_appln_type_desc	VARCHAR2(30)	This column stores the description of application type.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_ATTRITION_REASON_MASTER	v_attr_reas_code	VARCHAR2(10)	This is the unique code (identifier) assigned to each of the attrition reasons.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_ATTRITION_REASON_MASTER	fic_mis_date	DATE	This is the date on which processing commences. Ideally, the end of the business day of the month.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_ATTRITION_REASON_MASTER	v_attr_reason_desc	VARCHAR2(60)	This column holds the description of reasons	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_ATTRITION_REASON_MASTER	f_attrition_type	CHAR(1)		NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_ATTRITION_REASON_MASTER	v_attr_reason_type_desc	VARCHAR2(60)		NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_AUTH_DECISION_REAS_MASTER	v_decision_reason_code	VARCHAR2(20)	This column stores the authorization decision reason codes.	NO	NO	NO	NO	Yes	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_AUTH_DECISION_REAS_MASTER	fic_mis_date	DATE	This is the date on which the codes are extracted from the source systems.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_AUTH_DECISION_REAS_MASTER	v_reason_desc	VARCHAR2(60)	This column stores the description for authorization decision reason codes.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	N	N
STG_BANK_BASE_ROLE_MASTER	v_bank_base_role_id	VARCHAR2(20)	Stores the bank role for the Securitisation Exposure	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	Yes	NO	NO	NO	NO	N	N
STG_BANK_BASE_ROLE_MASTER	fic_mis_date	DATE	The extraction date represents the period for which the data is meant for. When the frequency of data is monthly the extraction date will be the month end date.	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	Yes	NO	NO	NO	NO	N	N

The Mappings can be generated from the OIDF ERwin file using ERwin's reporting tools.

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

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

   Typically, this data is extracted from source systems and loaded into 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 bank, it is not unusual to have 10s to 100s of millions of accounts and transactions.

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

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

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

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

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

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

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

Details of the Staging Data Model

The CSA model consists of over 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 analytical application that requires it:

1. **Business Data:** This set of tables captures the actual business events and the resulting state of a bank from those business events. The OLTP systems (or Transactional Systems) capture this information resulting from the execution of the bank's different business processes. Broadly, this information can be categorized as:
    1. **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.
    2. **State:** The net effect of business transactions is to change the bank's overall financial/non-financial state. An example of this – when a customer opens a checking/current account and deposits money into it, the account tracks the net effect of all withdrawals and deposits using a numeric quantity called a '**balance**'. The **account** will also contain a list of all events (Withdrawals, deposits, fees, etc) that resulted in the balance. This state information is typically captured by product-specific systems in a bank or FSI. For example, there is a lending system that captures details of loans, and a current account

system that captures details of checking and savings accounts, which are distinct products. In both cases, the accounts are governed by contracts, which refer to the terms and conditions governing business on that account.

2. **Reference/Master Data:** Events and state refer to 'business activities' of a bank or FSI. To provide more detail on these, banks need to capture additional data that provides context for these activities. This data may be variously called as 'reference data' or 'master data', and covers various business dimensions of a given transaction or account. For example – a bank has a master list of products that it sells to customers (Product Master). Similarly, it has a list of customers (Customer master). A trading firm may hold a list of securities it transacts in (Securities master). These and other lists provide context for each business transaction or account. Banks typically maintain 'Master' data for this purpose.

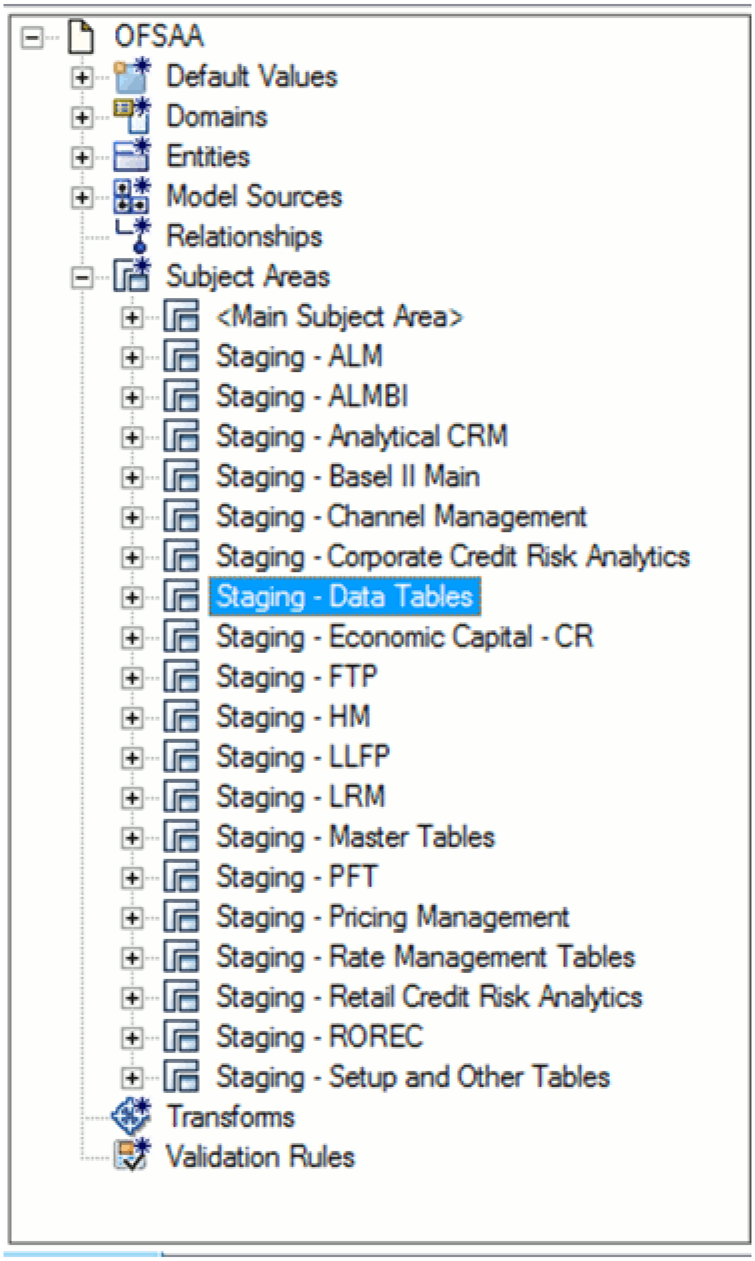
   With this background, the following lists the key categories of business data and reference data in the staging data model.

Business Data:

To view the current set of **business data** tables in the staging area, open the OIDF Staging Area model in the ERwin data modeling tool, and click on subject areas as shown below.

Right click and switch to the 'Staging-Data Tables' subject area.

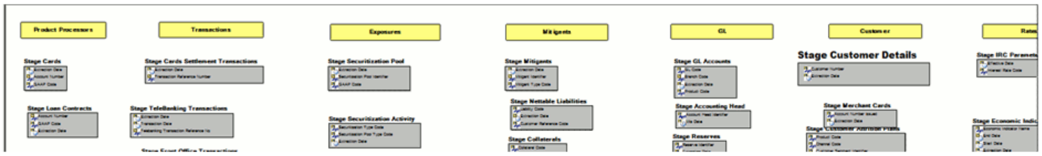
'Staging-Data Tables' Subject Area



- OFSAA
  - Default Values
  - Domains
  - Entities
  - Model Sources
  - Relationships
  - Subject Areas
    - <Main Subject Area>
    - Staging - ALM
    - Staging - ALMBI
    - Staging - Analytical CRM
    - Staging - Basel II Main
    - Staging - Channel Management
    - Staging - Corporate Credit Risk Analytics
    - Staging - Data Tables
    - Staging - Economic Capital - CR
    - Staging - FTP
    - Staging - HM
    - Staging - LLFP
    - Staging - LRM
    - Staging - Master Tables
    - Staging - PFT
    - Staging - Pricing Management
    - Staging - Rate Management Tables
    - Staging - Retail Credit Risk Analytics
    - Staging - ROREC
    - Staging - Setup and Other Tables
  - Transforms
  - Validation Rules

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

Grouped List of Business Data Tables in Current Staging Area Model



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

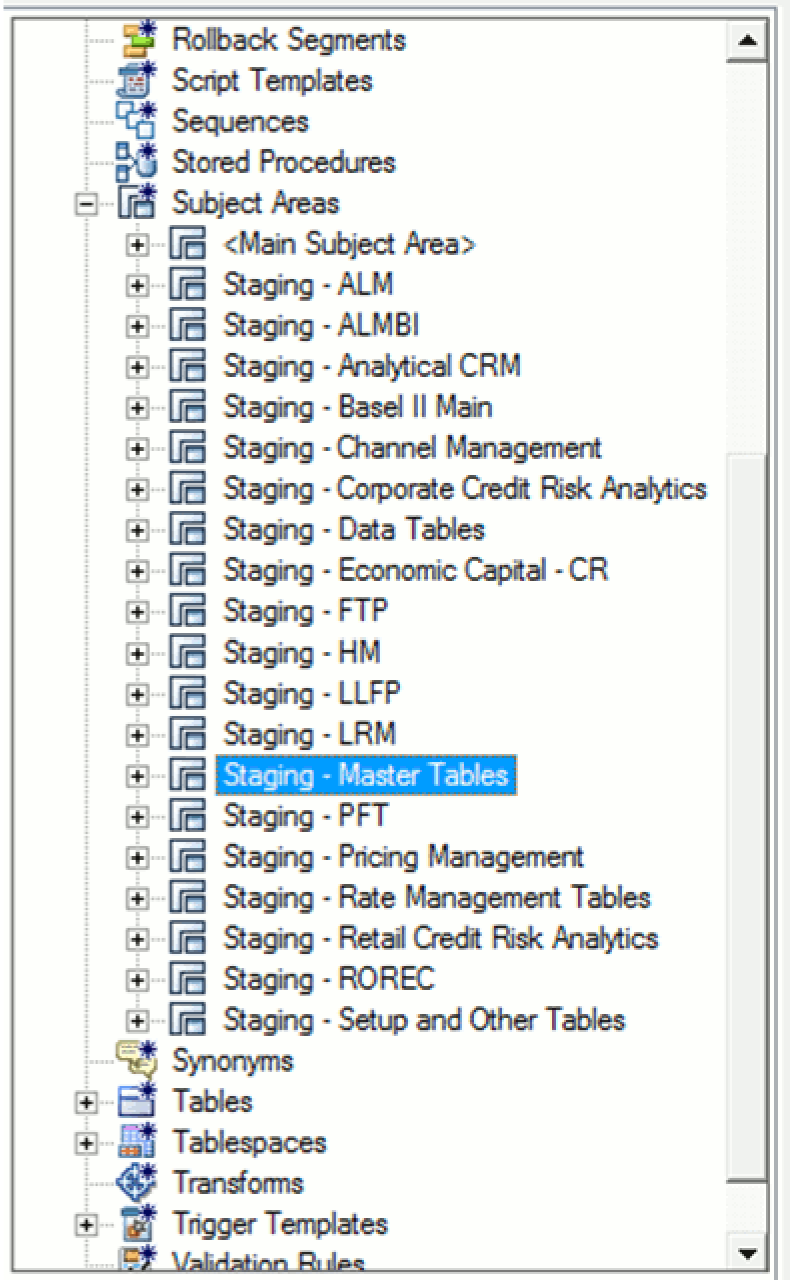
Group Name	**Purpose**
Product Processors	Tables for Financial Instruments and Contracts. Note that these tables can accommodate nearly 80+ types of instruments and derivatives across Banking and Trading books.
Transactions	Tables to hold Transaction/Event level data from the bank's systems. These tables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

Reference/Master Data:

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

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

Staging-Master Tables



Rollback Segments
Script Templates
Sequences
Stored Procedures
Subject Areas
<Main Subject Area>
Staging - ALM
Staging - ALMBI
Staging - Analytical CRM
Staging - Basel II Main
Staging - Channel Management
Staging - Corporate Credit Risk Analytics
Staging - Data Tables
Staging - Economic Capital - CR
Staging - FTP
Staging - HM
Staging - LLFP
Staging - LRM
Staging - Master Tables
Staging - PFT
Staging - Pricing Management
Staging - Rate Management Tables
Staging - Retail Credit Risk Analytics
Staging - ROREC
Staging - Setup and Other Tables
Synonyms
Tables
Tablespaces
Transforms
Trigger Templates
Validation Rules

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

Reporting Data Model

Overview

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

Similar to the Common Staging Area being the foundation for data provisioning to analytical applications, engines and processes, the Reporting Data Model is the common data store for the **outputs** of these processes. Outputs are computed quantitative measures and Key Performance Indicators that involve simple to complex, mathematical and statistical processing using the raw data, which is performed by specialized engines and computational models. In the 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:

1. **Design:**

   The Reporting Area data model is a dimensional data model. This means that it consists primarily of central fact tables (de-normalized), related to multiple dimension tables, also called a Star Schema.

   Additionally, the dimension tables are shared across the star schemas in the reporting mode, meaning they are Conformed Dimensions. This means that Drill-across reporting is naturally supported by the 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.

2. **Support for multiple scenarios of analysis:**

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

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

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

3. **Support for Cross Functional Reporting:**

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

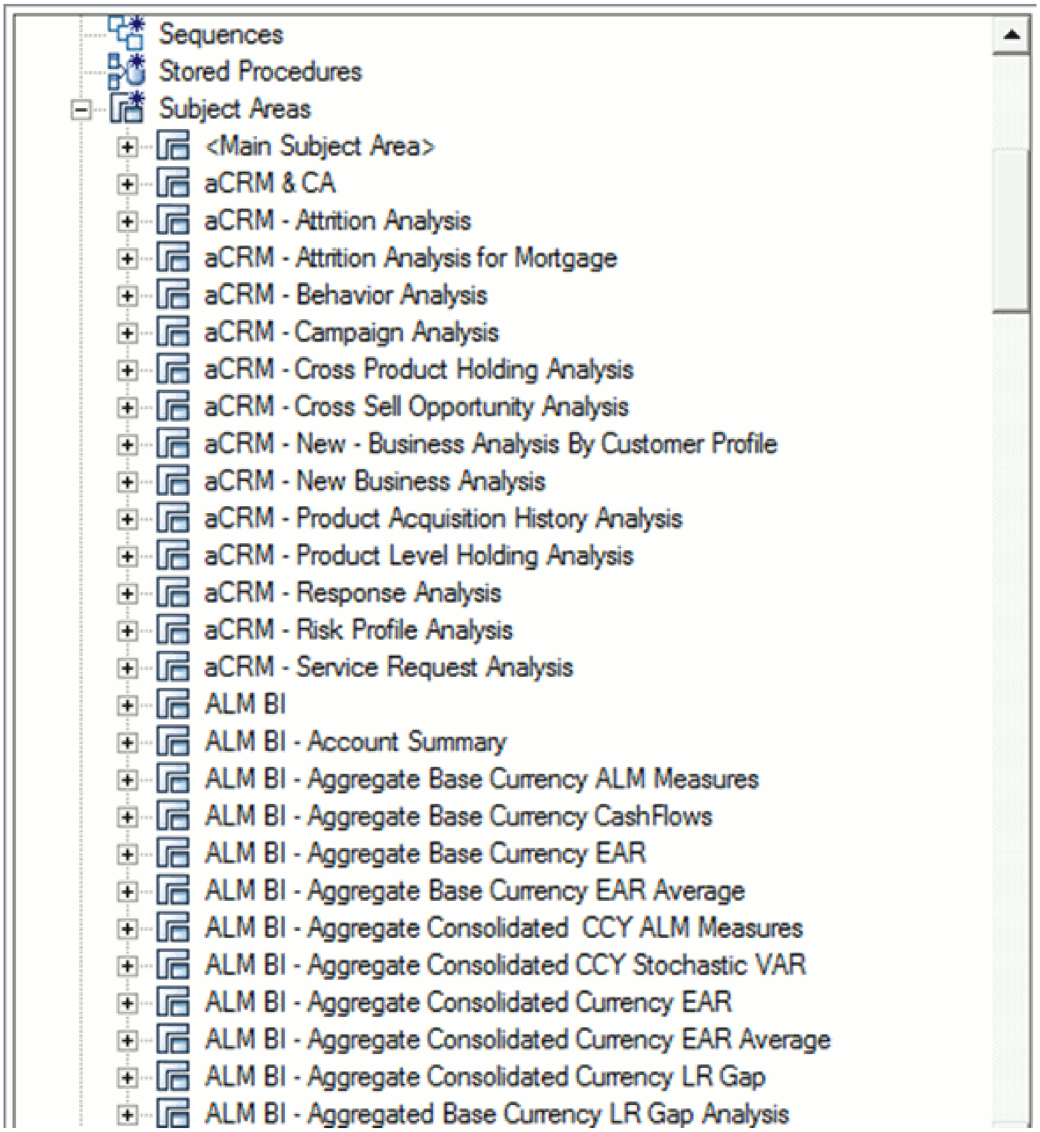
Typically, Business Intelligence and Reporting Solutions work off a dedicated, purpose-specific data store called a data mart. Data marts are function-specific data stores typically star schemas (eg. Marketing data marts, Risk Data Marts, Customer Data mart), that provide the necessary reporting and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

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

Reporting Area Details

Similar to the staging data model, the reporting model is divided into Subject Areas. These subject areas are visible by clicking on the **Subject Areas** view in the ERwin data modeler tool as show below.

Subject Areas View in ERwin Data Modeler Tool



- Sequences
- Stored Procedures
- Subject Areas
  - <Main Subject Area>
  - aCRM & CA
  - aCRM - Attrition Analysis
  - aCRM - Attrition Analysis for Mortgage
  - aCRM - Behavior Analysis
  - aCRM - Campaign Analysis
  - aCRM - Cross Product Holding Analysis
  - aCRM - Cross Sell Opportunity Analysis
  - aCRM - New - Business Analysis By Customer Profile
  - aCRM - New Business Analysis
  - aCRM - Product Acquisition History Analysis
  - aCRM - Product Level Holding Analysis
  - aCRM - Response Analysis
  - aCRM - Risk Profile Analysis
  - aCRM - Service Request Analysis
  - ALM BI
  - ALM BI - Account Summary
  - ALM BI - Aggregate Base Currency ALM Measures
  - ALM BI - Aggregate Base Currency CashFlows
  - ALM BI - Aggregate Base Currency EAR
  - ALM BI - Aggregate Base Currency EAR Average
  - ALM BI - Aggregate Consolidated CCY ALM Measures
  - ALM BI - Aggregate Consolidated CCY Stochastic VAR
  - ALM BI - Aggregate Consolidated Currency EAR
  - ALM BI - Aggregate Consolidated Currency EAR Average
  - ALM BI - Aggregate Consolidated Currency LR Gap
  - ALM BI - Aggregated Base Currency LR Gap Analysis

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

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

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

Name	Definition
PFT BI	Supports reporting related to Profitability analysis, part of the Enterprise Performance Management solution area.
Reputational Risk	Support for Reputational Risk measurement and analysis
Retail Credit Risk	The set of tables/subject areas supporting Retail Credit Risk Analysis.
Retail Pooling	Reporting support for Pooling for Retail Exposures – required by ALM and Retail Credit Risk analysis.
Solvency II	Support for the emerging Solvency II regulatory framework for Insurance providers.
Strategic Risk	Support for reporting related to Strategic Risk – measurement of these risks is a qualititative process, and part of the ICAAP framework.
aCRM	Reporting related to analytical Customer Relationship Management.

Within the above higher-level areas, there are over 165 subject areas consisting of star schemas supporting detailed analysis. Each Subject Area typically corresponds to one or possibly more star schemas, depending on the reporting need. There are currently 500+ fact and dimension tables in the reporting area. The details of the various naming conventions used in OIDF Data Model are explained in Appendix A, page A-1. For column-level details, see the Technical Whitepaper on *Data Model Document Generation*, which details how to extract the data dictionary from ERwin section.

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

Relationship to the OIDF Physical Model

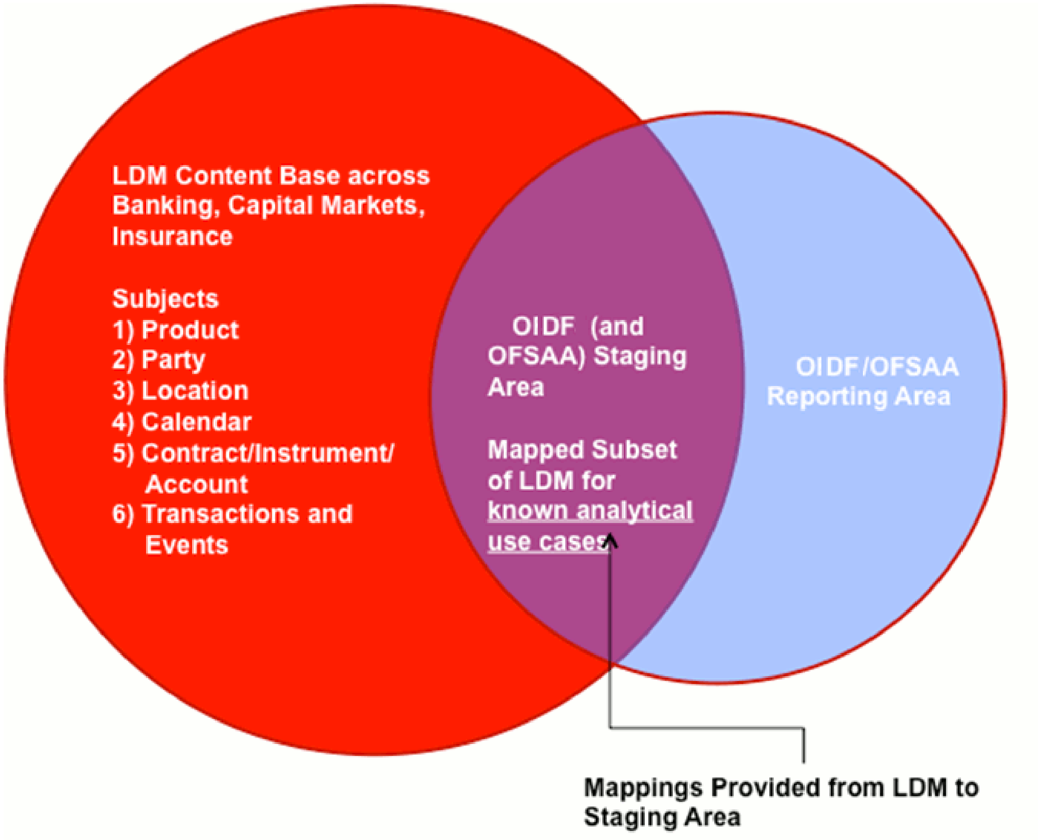
The coverage of the LDM in terms of core business processes is as follows.

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

Business Area	Comments
Insurance	Preliminary coverage of Insurance liabilities in the non P & C insurance domains.

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, Performance, Customer Insight, and Compliance, as illustrated below.

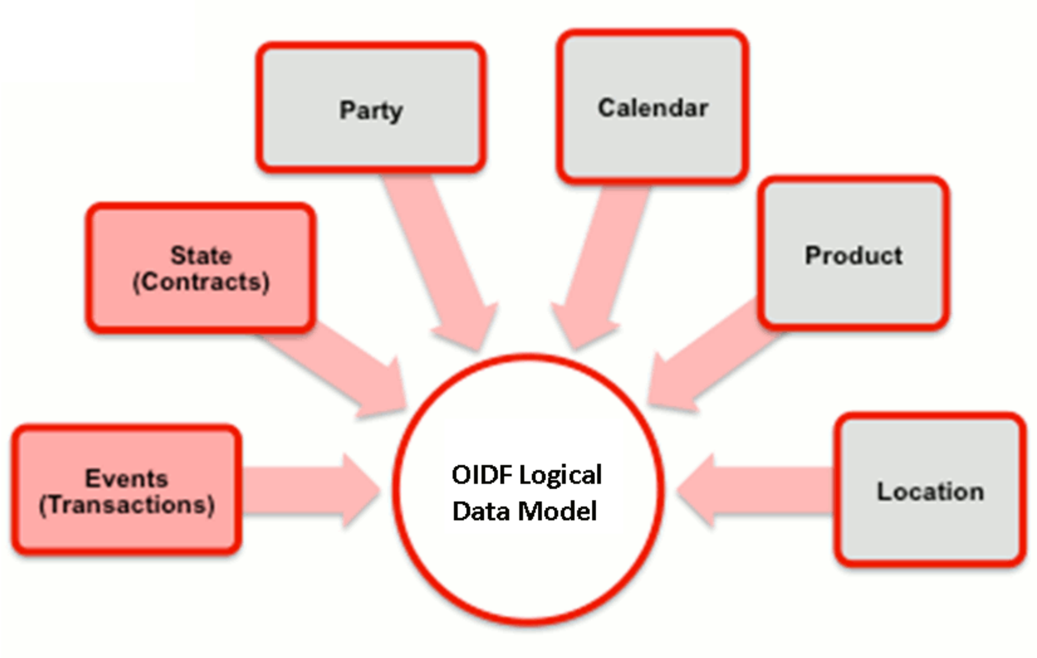
OIDF Physical Data Model


LDM Content Base across Banking, Capital Markets, Insurance
Subjects
1) Product
2) Party
3) Location
4) Calendar
5) Contract/Instrument/ Account
6) Transactions and Events
OIDF (and OFSAA) Staging Area
Mapped Subset of LDM for known analytical use cases
OIDF/OFSAA Reporting Area
Mappings Provided from LDM to Staging Area

OIDF LDM Content Details

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

Key Subjects in OIDF Logical Data Model



Within the above subject areas, the OIDF LDM has 800+ entities. The various naming conventions used in OIDF Logical Data Model are explained in Appendix A, page A-1 . For column-level details, see the Technical Whitepaper on *Data Model Document Generation*, which details how to extract the data dictionary from ERwin section.

Mapping from Logical Data Model to 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 Excel spreadsheets. Click the corresponding spreadsheet icon that you want to view.

LDM_Staging_mapping_Application.xlsx

LDM_Staging_mapping_Collateral.xlsx

LDM_Staging_mapping_Contract.xlsx

LDM_Staging_mapping_Geography.xlsx

LDM_Staging_mapping_Instrument.xlsx

LDM_Staging_mapping_Organization.xlsx

LDM_Staging_mapping_Party.xlsx

LDM_Staging_mapping_Product.xlsx

LDM_Staging_mapping_Accounting.xlsx

LDM_Staging_mapping_Campaign.xlsx

LDM_Staging_mapping_Campaign_7.3.xlsx

LDM_Staging_mapping_Claim.xlsx

LDM_Staging_mapping_Collection_Recove...

LDM_Staging_mapping_Credit_Card_7.3.xlsx

LDM_Staging_mapping_Transaction.xlsx

3

Dimension Loading Process

Dimension Tables Population

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

Overview of SCD Process

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

- **Type 1 SCDs - Overwriting**

  In a Type 1 SCD the new data overwrites the existing data. Thus the existing data is lost as it is not stored anywhere else. This is the default type of dimension you create. You do not need to specify any additional information to create a Type 1 SCD.

- **Type 2 SCDs - Creating another dimension record**

  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.

- **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/E14072_01/owb.112/e10935.pdf

Additional online sources include:

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

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

The SCDs used in Data Foundation solutions are listed in the following spreadsheet.

SCD.xls

Prerequisites

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

2. The user executing the SCD component should have execute rights on the file mentioned as prerequisite in point 2.

3. The setup tables accessed by SCD component are SYS_TBL_MASTER and SYS_STG_JOIN_MASTER.

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

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

Tables Used by the SCD Component

The database tables used by the SCD component are:

- SYS_TBL_MASTER

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

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	NUMBER(2) NULL	Priority of the Source when multiple sources are mapped to the same target.
SRC_PROC_SEQ	NUMBER(2) NOT NULL	The sequence in which the various sources for the DIMENSION will be taken up for processing.
SRC_TYP	VARCHAR2(30) NULL	The type of the Source for a Dimension, that is, Transaction Or Master Source.

Column Name	Data Type	Column Description
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.

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

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

- SYS_STG_JOIN_MASTER

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

Column Name	Data Type	Column Description
MAP_REF_NUM	NUMBER(3) NOT NULL	The Mapping Reference Number for this unique mapping of a Source to a Dimension Table.
COL_NM	VARCHAR2(30) NOT NULL	Name of the column in the Dimension Table.
COL_TYP	VARCHAR2(30) NOT NULL	Type of column. The possible values are given in the following section.
STG_COL_NM	VARCHAR2(60) NULL	Name of the column in the Staging Table.
SCD_TYP_ID	NUMBER(3) NULL	SCD type for the column.
PRTY_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	

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

1. PK – Primary Dimension Value (may be multiple for a given "Mapping Reference Number")

2. SK – Surrogate Key

3. DA – Dimensional Attribute (may be multiple for a given "Mapping Reference Number")

4. SD – Start Date

5. ED – End Date

6. LRI – Latest Record Indicator (Current Flag)

7. CSK – Current Surrogate Key

8. PSK – Previous Surrogate Key

9. SS – Source Key

10. LUD – Last Updated Date / Time

11. LUB – Last Updated By

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

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

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

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

Example

Dim_Bands_V

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

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

- DIM_<dimensionname> – Output table to which SCD writes the dimension data.

A sequence should be added for every user-defined dimension.

Example

```
create sequence SEQ_DIM_<DIM>  minvalue 1
maxvalue 9999999999999999999999999
increment by 1
```

Executing the SCD Component

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

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

- From the **Home** menu, select **Operations**, then select **Batch Maintenance**.
- Click **New Batch** ('+' symbol in Batch Name container) and enter the Batch Name and Description.
- Click **Save.**
- Select the Batch you created in the earlier step by clicking the check box in the Batch Name container.
- Click **New Task** ('+' symbol in Task Details container).
- Enter the Task ID and Description.
- Select **Run Executable**, from the Component ID list.
- Click **Parameters.** Select the following from the Dynamic Parameters List and then click **Save**:
  - Datastore Type - Select the appropriate datastore from the list
  - Datastore Name - Select the appropriate name from the list

- IP address - Select the IP address from the list
- Executable - scd,<map ref num>

  Example
  scd, 61 (Refer the following sections for details)
- Wait: When the file is being executed you have the choice to either wait till the execution is complete or proceed with the next task. Click the list box of the field provided for Wait in the Value field to select 'Yes' or 'No'. Clicking **Yes** confirms that you wish to wait for the execution to be complete. Clicking **No** indicates that you wish to proceed.
- Batch Parameter: Clicking **Yes** would mean that the batch parameters are also passed to the executable being started; else the batch parameters will not be passed to the executable.

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

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

- -1 (if you want to process all the dimensions). The *Executable* parameter mentioned earlier would be

  scd,-1
- If you want to process for a single dimension, query the database table SYS_TBL_MASTER and give the number in the map_ref_num column for the dimension you want to process. These are the ones which come seeded with the install.

- Execute the batch from Batch Execution by choosing the batch created following the steps mentioned in the preceding sections for a date.

  > **Note:** A seeded batch <Infodom>_SCD is provided which has all the required dimensions as different tasks that are part of SCD.

Checking the Execution Status

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

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

> **Note:** For a more comprehensive coverage, see *Oracle Financial Services*

Analytical Applications Infrastructure User Guide.

The status messages in Batch Monitor are :

N - Not Started

O - On Going

F - Failure

S – Success

The ICC execution log can be accessed on the application server in the following directory: $FIC_DB_HOME/log/ficgen.

The file name will have the batch execution id.

Sample

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

The detailed SCD component log can be accessed on the application server in the directory $FIC_HOME, go one folder up from there and then accessing the following path /ftpshare/<infodom name>/logs

The file name will have the batch execution id.

Sample

/dbfiles/home/oracle/ftpshare/OFSAADEMO/logs

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

Load DIM_ACCOUNT through SCD

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

- V_ACCOUNT_NUMBER
- N_ACCT_SKEY
- N_RCV_LEG_ACCT_SKEY
- FIC_MIS_DATE

This approach replaces the function load in which the table DIM_ACCOUNT is getting loaded through the function, FN_POPDIMACCOUNT. This loads the following columns into DIM_ACCOUNT table:

- V_ACCOUNT_NUMBER

- N_ACCT_SKEY
- N_RCV_LEG_ACCT_SKEY
- FIC_MIS_DATE

Where, the sources are the different product processor tables present in the solution, which are configured in FSI_DIM_ACCOUNT_SETUP_DETAILS table.

DIM_ACCOUNT SCD

Batch <INFODOM>DIM_ACCOUNT_SCD has been introduced with 30 tasks under it. These 30 tasks represent the 30 SCD processes where different product processors would be the source and DIM_ACCOUNT would be the target. MAP_REF_NUMs 188 to 217 have been introduced into SYS_TBL_MASTER table, and subsequently into SYS_STG_JOIN_MASTER.

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

Improve 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 are existing.

- For improving performance, hints for the MERGE query which is generated internally by the SCD can be provided under MERGE_HINT. Session alters could be mentioned under SESSION_ENABLE_STATEMENT and SESSION_DISABLE_STATEMENT columns.

- SESSION_ENABLE_STATEMENTs will be executed before the MERGE in the SCD and SESSION_DISABLE_STATEMENTs will be executed after the SCD MERGE.

- Since all the tasks under the SCD batch for DIM_ACCOUNT works on the same target, the SESSION_DISABLE_STATEMENTs in SYS_TBL_MASTER cannot be provided when tasks are executed. In this case, there can be a separate SQL file to contain all the SESSION_DISABLE_STATEMENTs to be executed once after all the tasks in the SCD are done. The SESSION_DISABLE_STATEMENT will hold a null in SYS_TBL_MASTER table.

- SESSION_ENABLE_STATEMENTs are required to be mentioned only for the first task in the batch. Here the target is the same for all the tasks under a batch. In case any of the tasks are to be executed separately, then the SESSION_ENABLE_STATEMENTs should be mentioned for any one of the tasks which is included in the batch for the execution.

Example

MERGE_HINT and SESSION_ENABLE_STATEMENT in SYS_TBL_MASTER

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

- All the tasks can be executed in parallel. This might cause the N_RCV_LEG_ACCT_SKEY to have an incremental value as compared to N_ACCT_SKEY.

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

  - Once the DIM_ACCOUNT table is populated using this approach, you will not be able to use the initial approach (FN_POPDIMACCOUNT) as this will lead to skey conflict.

  - 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.

- The F_LATEST_RECORD_INDICATOR for an existing DIM_ACCOUNT data already loaded by the function should be updated to 'Y' before running the SCD, failing which a new skey might get generated for the same account number.
- SCD execution occurs based on the GAAP code which is configured in SETUP_MASTER table. This has been introduced to tackle the scenario of multiple GAAP codes. Whether or not there exist multiple GAAP codes, SETUP_MASTER should be manually configured as follows:

V_COMPONENT_CODE	V_COMPONENT_DESC	V_COMPONENT_VALUE
DEFAULT_GAAP	DEFAULT_GAAP	USGAAP

Where V_COMPONENT_VALUE should be manually populated with the required GAAP code.

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

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

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

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

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

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 all Product Processors, 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.

4

Customer Summary Population

Overview of Common Customer Summary Tables

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

Prerequisites

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

- DIM_BANDS
- DIM_CHANNEL
- DIM_CUSTOMER
- DIM_CUSTOMER_TYPE
- DIM_GEOGRAPHY
- DIM_INDUSTRY
- DIM_MANAGEMENT
- FCT_COMMON_ACCOUNT_SUMMARY
- STG_CUSTOMER_DETAILS

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

Executing the Customer Summary Population T2T

Fact Common Customer Summary T2T can be executed by executing task present in the seeded batch <INFODOM>_aCRM_CommCust_Appln.

Following steps will help you to execute the batch:

1. Go to the *Batch Execution* screen.
2. Select the seeded batch <INFODOM>_aCRM_CommCust_Appln where INFODOM is the information domain where application is installed.
3. Select the AS_OF_DATE for which source customer information is required to be loaded into the table.
4. Click **Execute Batch**.
5. Monitor the status of the batch from *Batch Monitor* screen of OFSAAI.

Error Messages

Following are the most common error message which will be logged in the T2T log file present in the *$FIC_DB_HOME/logs/t2t* folder:

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

5

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. (For details on what Download Specifications are, see Staging Area, page 2-5)

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

This chapter covers the following topics:

- Delivery Mechanism
- Installing OIDF
- OIDF Supporting Documentation
- Data Dictionary
- Download Specifications
- Extending OIDF Physical Data Model

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+. ERwin 8.0 is not supported as of version 7.4.0.0.0.

Installing OIDF

As detailed earlier, OIDF requires the Oracle Financial Services Analytical Application Infrastructure release 7.3.2.1.0 to deploy and operate.

Please refer to the separate **OIDF Install Guide** for stepwise instructions how to configure and install OIDF into an AAI instance.

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 2 key detailed artifacts of OIDF documentation that can be extracted from within the ERwin data model are:

1. Data Dictionary

2. Download Specifications

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

Data Dictionary

The data dictionary for 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.

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.

Extending OIDF Physical Data Model

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, page 5-3
- Staging Area Design Guidelines, page 5-4
- Results Area Design Guidelines, page 5-5
- Upgrading Data Model, page 5-6

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.

Staging Area Design Guidelines

Following are the Staging Area Design Guidelines:

1. Ensure that the naming conventions as detailed in Appendix A, page A-1 section are followed.

2. Entity relationships and constraints are enforced through OFSAAI data management toolkit and are not enforced via database referential integrity checks. The model should not be changed to enforce referential integrity checks and other data quality checks via database definitions.

3. All Staging Area tables must have a column that identifies the system from where data is sourced (source system ID).

4. The code columns in master data tables and tables that contain dimension data should be designed to hold alphanumeric values.

5. The Domain dictionary maintains the list of attribute domains. New columns must be identified with an existing domain instead of explicitly defining column data type and valid values. See guidelines in Appendix A, page A-1 section on the use of defined Domains.

6. Tables (e.g. reference or look up tables with static data) needed for only a specific application or use case should be a part of the application specific processing area and should not be part of the common Staging Area in OIDF.

7. 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.

8. All columns added or modified as a part of the customization should be marked as such:

    1. The column level UDP named "CUSTOM" must be marked YES, identifying the column as a custom property.

    2. The "Customization Reason" UDP should be specified. Valid values are provided as a drop down list and can be "Pending Enhancement Request" or " Specific to Customer".

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

Results Area Design Guidelines

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

Following are the Results Area Design Guidelines:

1. Ensure that the naming convention for results tables and columns detailed in Appendix A, page A-1 section is followed.

2. Dimensional conformance should be maintained: The same dimensional information should not be represented in different forms. In addition, dimension table design should be compatible with the slowly changing dimension process design and so should have the required columns. Design considerations for adding new Dimensions and for adding columns to existing dimensions are detailed in a separate OIDF Model extensions guide.

3. The common accounts summary fact table (FCT_COMMON_ACCOUNTS_SUMMARY) consolidates measures at an account level granularity for all applications. Account level attributes captured from source systems in staging and those attributes that do not vary between runs should be part of the common accounts summary table. This enables integrated reporting of account information.

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

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

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

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

6. **Reporting and local currency support**: Include additional attributes in the fact tables to store reporting and local currency equivalent of base measures. These attributes need to be computed by looking into the exchange rates.

7. **Support for full history**: Any new tables in the Results area should be designed to support maintenance of full history.

Upgrading Data Model

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

6

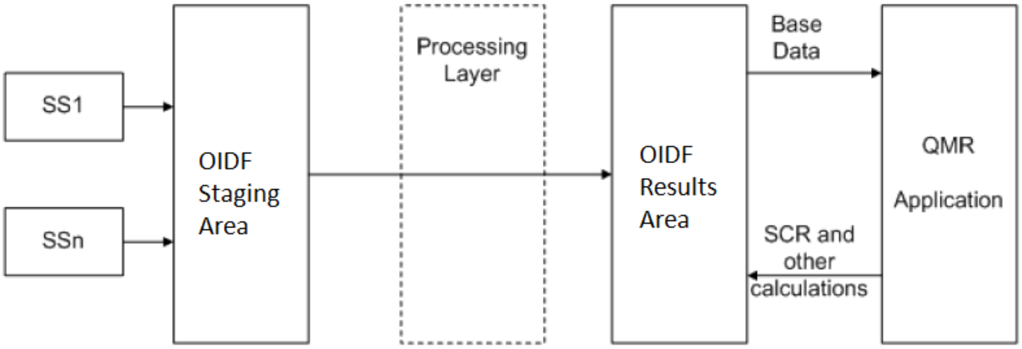
OIDF Interface with QMR 2.0.0.0.0

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.

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 7.4.1.0.0.

Description of Extracts from OIDF

OIDF extracts provide data pertaining to the various QRT reports in the granularity and format required by QMR 2.0.0.0.0. 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:

Sr. No.	Extract Name	Description
1	Legal Entity – Solo	Provides Data for following reports 1. BS-C1 2. BS-C1D 3. Cover-A1A 4. Group-G01 5. MCR-B4 6. OF (Annual) 7. Re-J1-Non Life 8. Re-J1-Life 9. Re-J2 10. Re-J3 11. Re-SPV 12. SCR-B2A 13. SCR-B2B 14. SCR-B2C 15. SCR-B3A 16. SCR-B3B

Sr. No.	Extract Name	Description
		17. SCR-B3C 18. SCR-B3D 19. SCRB-3E 20. SCRB-3F 21. SCR-B3G 22. TP-E1 23. TP-E2 24. TP-E3 25. TP-E4 26. TP-E6 27. TP-E7A 28. TP-E7B 29. TP-F1 30. TP-F2 31. TP-F4
2	Legal Entity – Group	1. P-Own 2. P-Con 3. OF Group 4. SCR-B2A

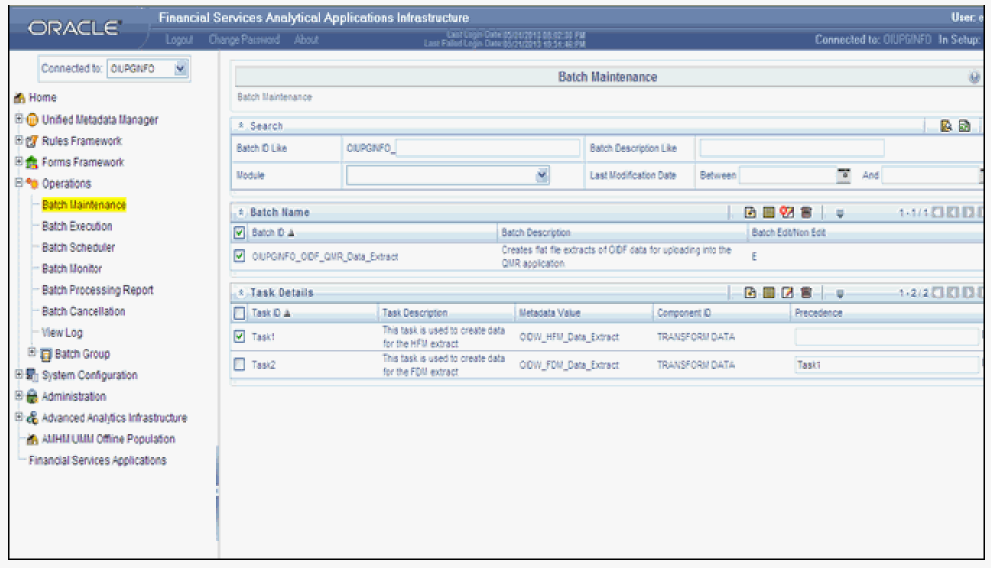
Sr. No.	Extract Name	Description
3	Legal Entity-RFR (Ring Fenced Fund)	1. BS-C1 2. OF (Annual) 3. SCR-B2A 4. SCR-B2B 5. SCR-B2C 6. SCR-B3A 7. SCR-B3B 8. SCR-B3C 9. SCR-B3D 10. SCRB-3E
4	Legal Entity – Non Insurance Entity	1. Group-G04
5	Legal Entity – Non EEA Entity	1. Group-G03
6	Global Data	1. Exchange Rates 2. Inflation Rates 3. Discount Rates 4. Correlation Data
7	Other Individual Files	1. IGT1 2. IGT2 3. IGT3

Sr. No.	Extract Name	Description
		4. IGT4 5. ASSETS-D1 6. ASSETS-D1S 7. ASSETS-D20 8. ASSETS-D2T 9. ASSETS-D3 10. ASSETS-D4 11. ASSETS-D5 12. ASSETS-D6

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.



Batch ID Like: OIUPGINFO_

Batch Description Like:

Module:

Last Modification Date: Between And

Batch Name

☑	Batch ID	Batch Description	Batch Edit/Non Edit
☑	OIUPGINFO_OIDF_QMR_Data_Extract	Creates flat file extracts of OIDF data for uploading into the QMR application	E

Task Details

□	Task ID	Task Description	Metadata Value	Component ID	Precedence
☑	Task1	This task is used to create data for the HFM extract	OIDW_HFM_Data_Extract	TRANSFORM DATA	
□	Task2	This task is used to create data for the FDM extract	OIDW_FDM_Data_Extract	TRANSFORM DATA	Task1

6. 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.

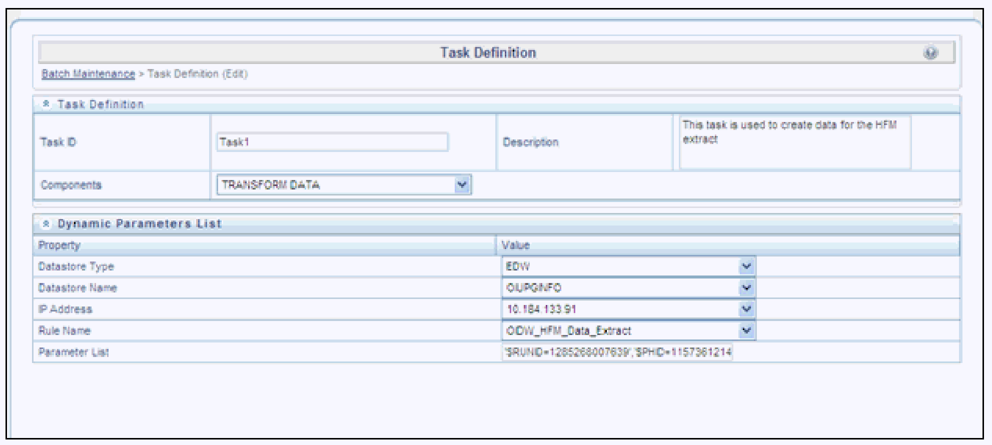
$QMRRUNSK=1	Set Run SKey for which the QMR data needs to be extracted.
$RPSTARTDATE=20120101	Set the Reporting Period Start Date in YYYYMMDD Format.
$FINANCIAL_PRD=QA	Set the Financial Period for the required Data Extract (Q1/Q2/Q3/Q4/QA).

7. Replace the parameter list (preceded with the parameter names) in **Task Definition** and re-save the task.

8. 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).



Task Definition

Batch Maintenance > Task Definition (Edit)

Task Definition

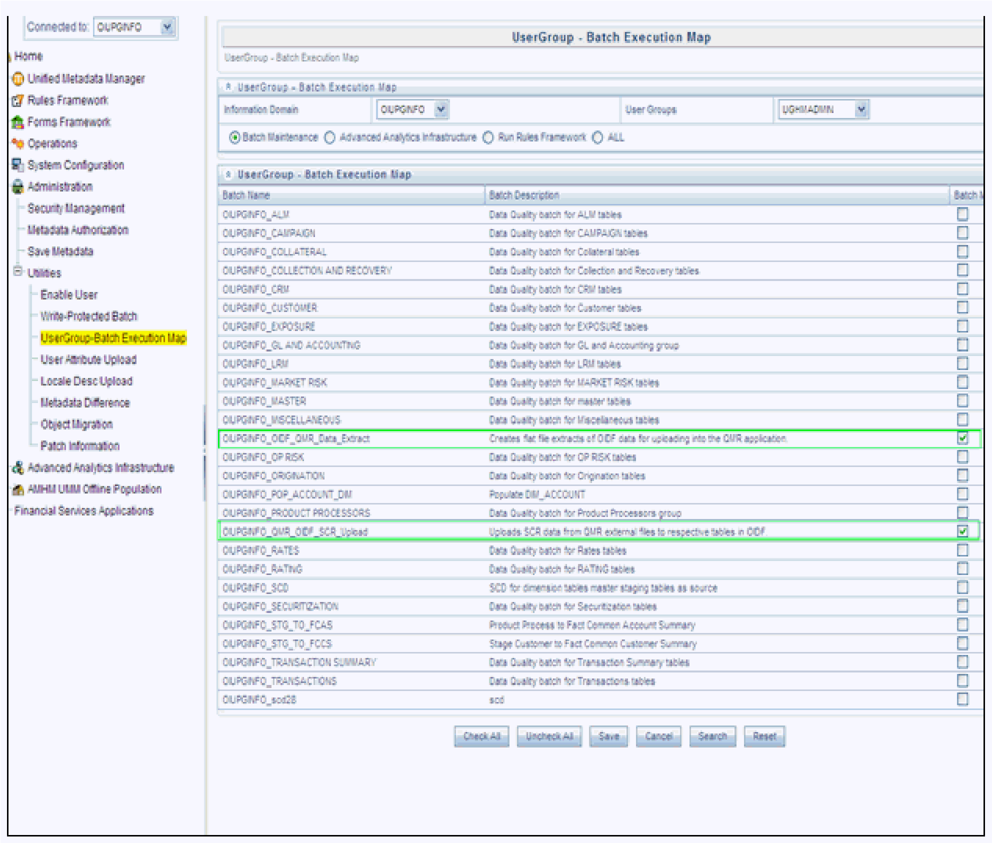
Task ID	Task1	Description	This task is used to create data for the HFM extract
Components	TRANSFORM DATA		

Dynamic Parameters List

Property	Value
Datastore Type	EDW
Datastore Name	OIUPGINFO
IP Address	10.184.133.91
Rule Name	OIDW_HFM_Data_Extract
Parameter List	'$RUNID=1285268007639','$PHID=1157361214

9. Replace the parameter list (preceded with the parameter names) in **Task Definition** and re-save the task.

10. Map the batch or batches to be executed to the user group in the **UserGroup-Batch Execution Map** screen in the **Administration** menu.



UserGroup - Batch Execution Map

UserGroup - Batch Execution Map

UserGroup - Batch Execution Map

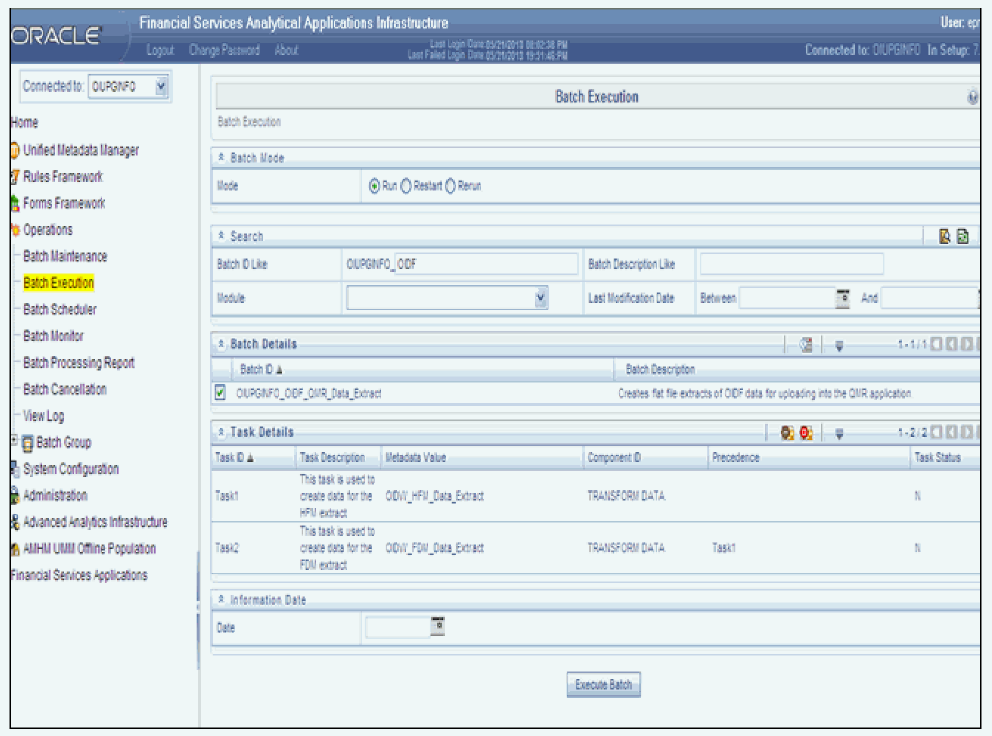
Information Domain: OIUPGINFO   User Groups: UGHMADMIN

● Batch Maintenance ○ Advanced Analytics Infrastructure ○ Run Rules Framework ○ ALL

UserGroup - Batch Execution Map

Batch Name	Batch Description	Batch
OIUPGINFO_ALM	Data Quality batch for ALM tables	□
OIUPGINFO_CAMPAIGN	Data Quality batch for CAMPAIGN tables	□
OIUPGINFO_COLLATERAL	Data Quality batch for Collateral tables	□
OIUPGINFO_COLLECTION AND RECOVERY	Data Quality batch for Collection and Recovery tables	□
OIUPGINFO_CRM	Data Quality batch for CRM tables	□
OIUPGINFO_CUSTOMER	Data Quality batch for Customer tables	□
OIUPGINFO_EXPOSURE	Data Quality batch for EXPOSURE tables	□
OIUPGINFO_GL AND ACCOUNTING	Data Quality batch for GL and Accounting group	□
OIUPGINFO_LRM	Data Quality batch for LRM tables	□
OIUPGINFO_MARKET RISK	Data Quality batch for MARKET RISK tables	□
OIUPGINFO_MASTER	Data Quality batch for master tables	□
OIUPGINFO_MISCELLANEOUS	Data Quality batch for Miscellaneous tables	□
OIUPGINFO_OIDF_QMR_Data_Extract	Creates flat file extracts of OIDF data for uploading into the QMR application	☑
OIUPGINFO_OP RISK	Data Quality batch for OP RISK tables	□
OIUPGINFO_ORIGINATION	Data Quality batch for Origination tables	□
OIUPGINFO_POP_ACCOUNT_DIM	Populate DIM_ACCOUNT	□
OIUPGINFO_PRODUCT PROCESSORS	Data Quality batch for Product Processors group	□
OIUPGINFO_QMR_OIDF_SCR_Upload	Uploads SCR data from QMR external files to respective tables in OIDF	☑
OIUPGINFO_RATES	Data Quality batch for Rates tables	□
OIUPGINFO_RATING	Data Quality batch for RATING tables	□
OIUPGINFO_SCD	SCD for dimension tables master staging tables as source	□
OIUPGINFO_SECURITIZATION	Data Quality batch for Securitization tables	□
OIUPGINFO_STG_TO_FCAS	Product Process to Fact Common Account Summary	□
OIUPGINFO_STG_TO_FCCS	Stage Customer to Fact Common Customer Summary	□
OIUPGINFO_TRANSACTION SUMMARY	Data Quality batch for Transaction Summary tables	□
OIUPGINFO_TRANSACTIONS	Data Quality batch for Transactions tables	□
OIUPGINFO_scd28	scd	□

11. 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.



Connected to: OIUPGINFO

Batch Execution

Batch Mode

Mode: ● Run ○ Restart ○ Rerun

Search

Batch ID Like	OIUPGINFO_OIDF	Batch Description Like	
Module		Last Modification Date	Between And

Batch Details 1 - 1 / 1

	Batch ID ▲	Batch Description
☑	OIUPGINFO_OIDF_QMR_Data_Extract	Creates flat file extracts of OIDF data for uploading into the QMR application

Task Details 1 - 2 / 2

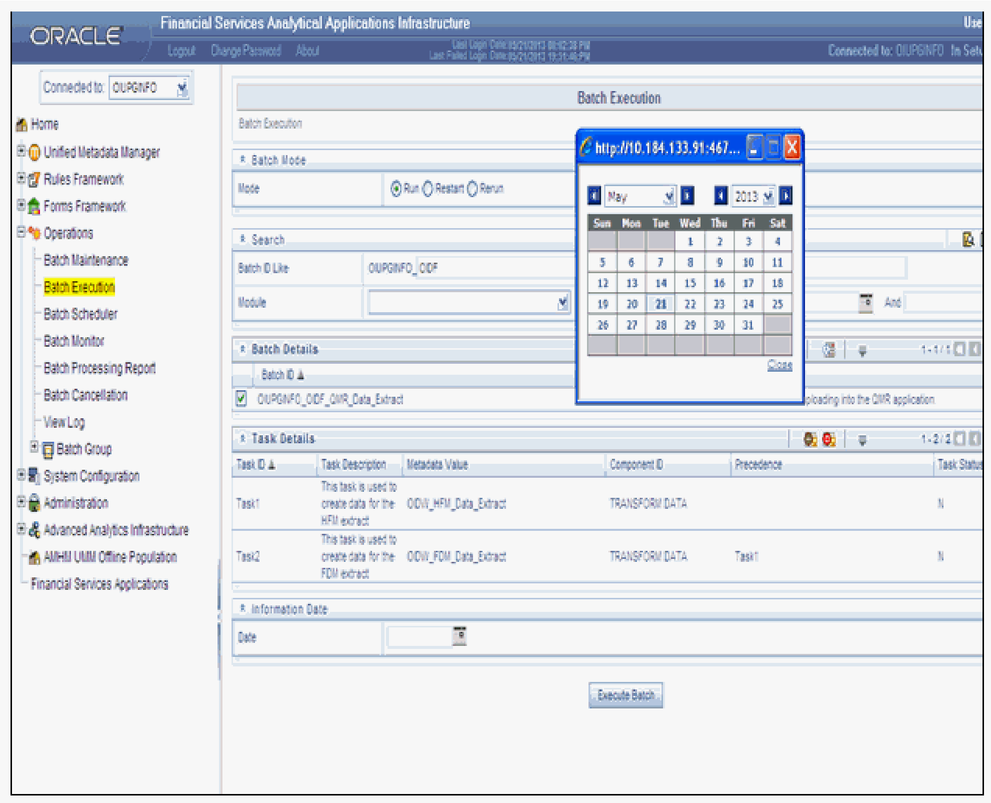
Task ID ▲	Task Description	Metadata Value	Component ID	Precedence	Task Status
Task1	This task is used to create data for the HFM extract	OIDW_HFM_Data_Extract	TRANSFORM DATA		N
Task2	This task is used to create data for the FDM extract	OIDW_FDM_Data_Extract	TRANSFORM DATA	Task1	N

Information Date

Date:

12. Execute the Batch.

13. To check the status of the batch click **Batch Monitor** in the **Operations** menu.



Batch Execution

Batch Mode

Mode: ● Run ○ Restart ○ Rerun

Search

Batch ID Like: OIUPGINFO_OIDF

Module:

Batch Details

☑ OIUPGINFO_OIDF_QMR_Data_Extract ... uploading into the QMR application

Task Details

Task ID	Task Description	Metadata Value	Component ID	Precedence	Task Status
Task1	This task is used to create data for the HFM extract	OIDW_HFM_Data_Extract	TRANSFORM DATA		N
Task2	This task is used to create data for the FDM extract	OIDW_FDM_Data_Extract	TRANSFORM DATA	Task1	N

Information Date

Date:

14. 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

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

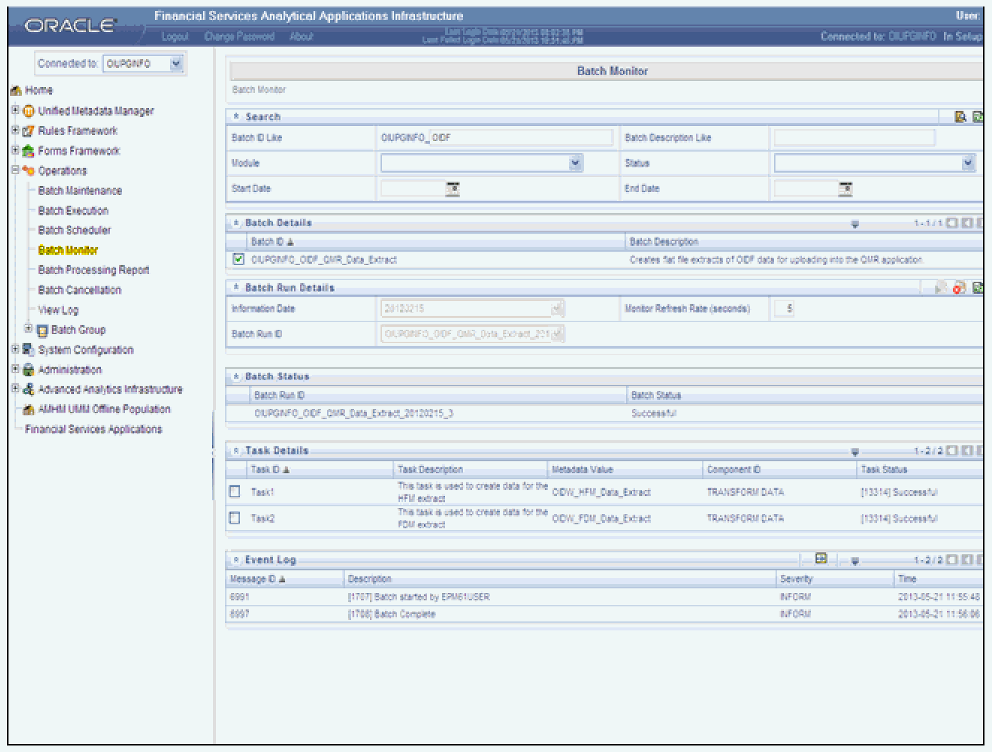
2. Login to the OFSAAI Home page.

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.



Connected to: OIUPGINFO

Batch Monitor

Search

Batch ID Like	OIUPGINFO_OIDF	Batch Description Like	
Module		Status	
Start Date		End Date	

Batch Details 1 - 1 / 1

	Batch ID	Batch Description
☑	OIUPGINFO_OIDF_QMR_Data_Extract	Creates flat file extracts of OIDF data for uploading into the QMR application.

Batch Run Details

Information Date	20120215	Monitor Refresh Rate (seconds)	5
Batch Run ID	OIUPGINFO_OIDF_QMR_Data_Extract_201		

Batch Status

Batch Run ID	Batch Status
OIUPGINFO_OIDF_QMR_Data_Extract_20120215_3	Successful

Task Details 1 - 2 / 2

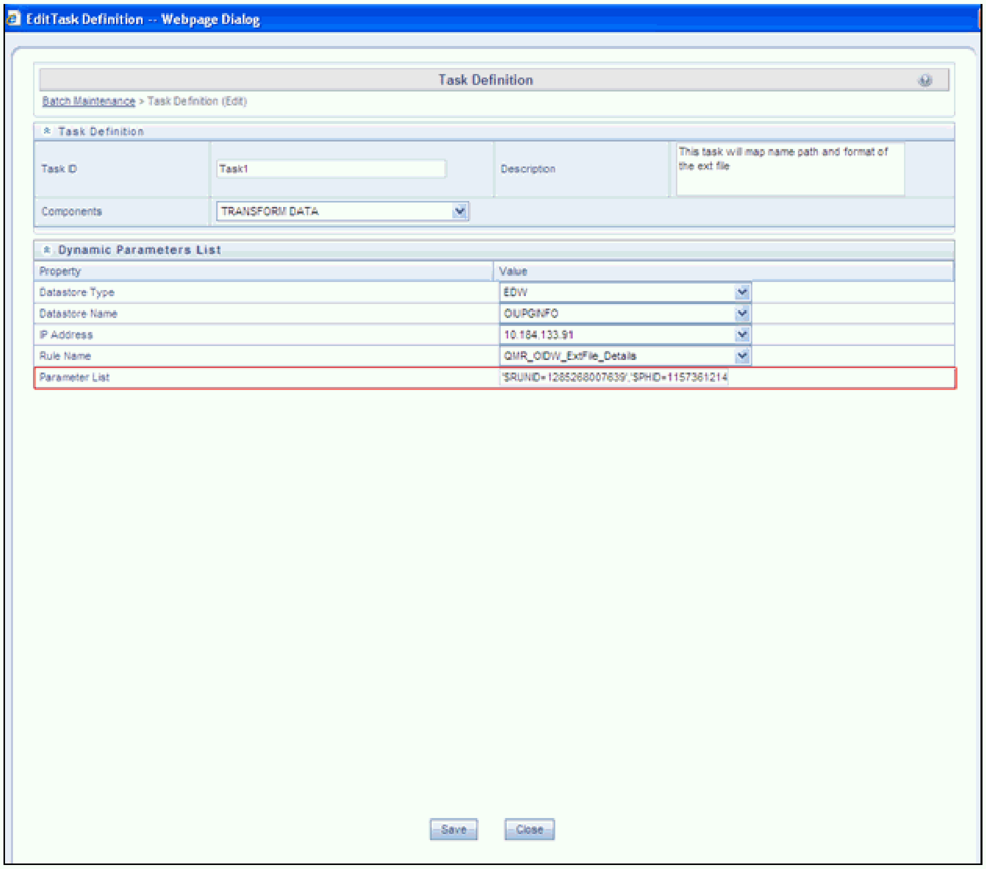
	Task ID	Task Description	Metadata Value	Component ID	Task Status
□	Task1	This task is used to create data for the HFM extract	OIDW_HFM_Data_Extract	TRANSFORM DATA	[13314] Successful
□	Task2	This task is used to create data for the FDM extract	OIDW_FDM_Data_Extract	TRANSFORM DATA	[13314] Successful

Event Log 1 - 2 / 2

Message ID	Description	Severity	Time
6991	[1707] Batch started by EPM61USER	INFORM	2013-05-21 11:55:48
6997	[1708] Batch Complete	INFORM	2013-05-21 11:56:06

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.

$EXT_DIR_NAME=DIR_QMR_UPLOAD_FILES	Set Oracle Directory where the HFM SCR Extract Data is placed.
$EXT_FILE_NAME=QMR_SCR_Data_2011-QA_LE01_Extract.dat	Set the 'HFM SCR Extract Filename' to be processed present under the Oracle Directory.


EditTask Definition -- Webpage Dialog
Task Definition
Batch Maintenance > Task Definition (Edit)
Task Definition
Dynamic Parameters List
Save
Close

Task ID	Task1	Description	This task will map name path and format of the ext file
Components	TRANSFORM DATA		

Property	Value
Datastore Type	EDW
Datastore Name	OIUPGINFO
IP Address	10.184.133.91
Rule Name	QMR_OIDW_ExtFile_Details
Parameter List	'$RUNID=1285268007639','$PHID=1157361214

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.

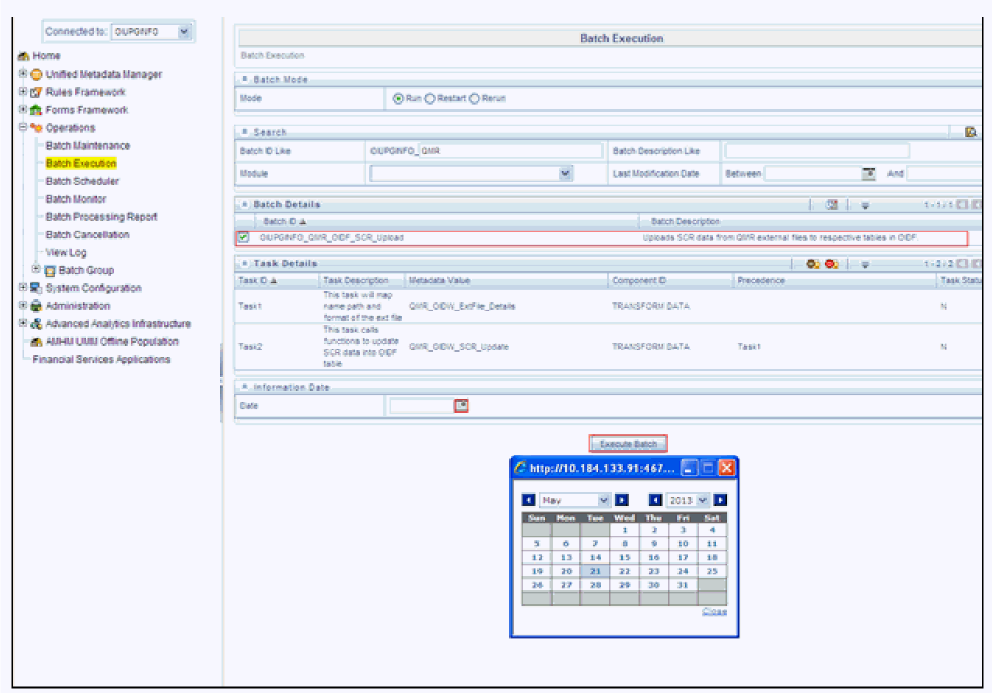
$QMRDATE=20120101	Set the MIS Date for which the QMR data needs to be updated in OIDF in YYYYMMDD Format.
$QMRRUNSK=1	Set Run SKey for which the QMR data needs to be updated in OIDF.
$QMRGAAP=IFRS	Set the GAAP ID for which the QMR data needs to be updated in OIDF.

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.



Mode ● Run ○ Restart ○ Rerun

Batch ID Like	OIUPGINFO_QMR
Batch Description Like	
Module	
Last Modification Date	Between And

	Batch ID ▲	Batch Description
☑	OIUPGINFO_QMR_OIDF_SCR_Upload	Uploads SCR data from QMR external files to respective tables in OIDF.

Task ID ▲	Task Description	Metadata Value	Component ID	Precedence	Task Statu
Task1	This task will map name path and format of the ext file	QMR_OIDW_ExtFile_Details	TRANSFORM DATA		N
Task2	This task calls functions to update SCR data into OIDF table	QMR_OIDW_SCR_Update	TRANSFORM DATA	Task1	N

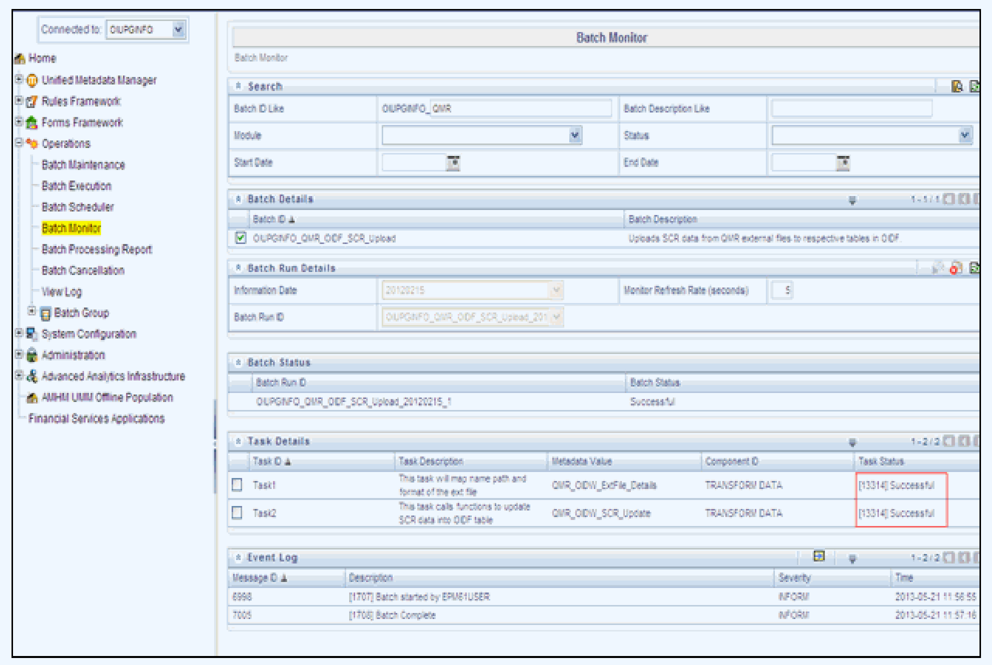
Date

May 2013

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

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.



Search

Batch ID Like	OIUPGINFO_ QMR	Batch Description Like	
Module		Status	
Start Date		End Date	

Batch Details

	Batch ID	Batch Description
☑	OIUPGINFO_QMR_OIDF_SCR_Upload	Uploads SCR data from QMR external files to respective tables in OIDF.

Batch Run Details

Information Date	20120215	Monitor Refresh Rate (seconds)	5
Batch Run ID	OIUPGINFO_QMR_OIDF_SCR_Upload_201		

Batch Status

Batch Run ID	Batch Status
OIUPGINFO_QMR_OIDF_SCR_Upload_20120215_1	Successful

Task Details

	Task ID	Task Description	Metadata Value	Component ID	Task Status
□	Task1	This task will map name path and format of the ext file	QMR_OIDW_ExtFile_Details	TRANSFORM DATA	[13314] Successful
□	Task2	This task calls functions to update SCR data into OIDF table	QMR_OIDW_SCR_Update	TRANSFORM DATA	[13314] Successful

Event Log

Message ID	Description	Severity	Time
6998	[1707] Batch started by EPM61USER	INFORM	2013-05-21 11:56:55
7005	[1708] Batch Complete	INFORM	2013-05-21 11:57:16

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

Data Population

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
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

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
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

Input Data Preparation

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.

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 FSDF 7.4 (Financial Services 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

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 – This 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 – The list of legal entities for the institute is provided in the table DIM_ORG_STRUCTURE.
  - Scenario – 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 attached spreadsheet.

Oracle_Insurance_Solvency_II_QMR_Interface...

7

Data Quality Rules Execution

Data Quality Framework

Data Quality Framework consists of a scalable rule-based engine which uses a single-pass integration process to standardize, match, and duplicate information across global data. Data Quality Framework within the Infrastructure system facilitates you to define rules and execute them to query, validate, and correct the transformed data existing in an Information Domain.

You can access Data Quality Framework by expanding the Data Integrator Framework within the Unified Metadata Manager section in tree structure of LHS menu.

Data Quality Framework consists of the following sections:

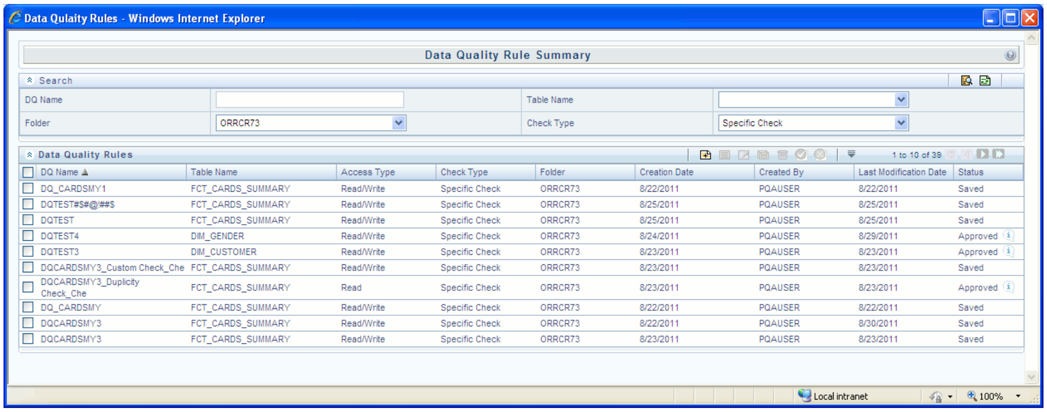
- Data Quality Summary
- Data Quality Group Summary

Data Quality Summary

Data Quality Summary within the Data Integrator framework of Infrastructure system facilitates you to create a DQ (Data Quality) definition and define nine specific validation checks based on Range, Data Length, Column Reference/Specific Value, List of Value/Code, Null Value, Blank Value, Referential Integrity, Duplicity, and Custom Check/Business. You can also correct data for range, column reference, list of values, null value, and blank value parameters.

The defined Data Quality Rule checks can be logically grouped and executed together. You (Business Analysts) need to have ETL Analyst function role mapped to access the Data Quality Summary framework within the Infrastructure system.

You can access Data Quality Summary by expanding the Data Quality framework within the Unified Metadata Manager section in tree structure of LHS menu.



Data Quality Rule Summary

Search

DQ Name		Table Name	
Folder	ORRCR73	Check Type	Specific Check

Data Quality Rules — 1 to 10 of 39

□	DQ Name	Table Name	Access Type	Check Type	Folder	Creation Date	Created By	Last Modification Date	Status
□	DQ_CARDSMY1	FCT_CARDS_SUMMARY	Read/Write	Specific Check	ORRCR73	8/22/2011	PQAUSER	8/22/2011	Saved
□	DQTEST#$#@!##$	FCT_CARDS_SUMMARY	Read/Write	Specific Check	ORRCR73	8/25/2011	PQAUSER	8/25/2011	Saved
□	DQTEST	FCT_CARDS_SUMMARY	Read/Write	Specific Check	ORRCR73	8/25/2011	PQAUSER	8/25/2011	Saved
□	DQTEST4	DIM_GENDER	Read/Write	Specific Check	ORRCR73	8/24/2011	PQAUSER	8/29/2011	Approved ℹ
□	DQTEST3	DIM_CUSTOMER	Read/Write	Specific Check	ORRCR73	8/23/2011	PQAUSER	8/23/2011	Approved ℹ
□	DQCARDSMY3_Custom Check_Che	FCT_CARDS_SUMMARY	Read/Write	Specific Check	ORRCR73	8/23/2011	PQAUSER	8/23/2011	Saved
□	DQCARDSMY3_Duplicity Check_Che	FCT_CARDS_SUMMARY	Read	Specific Check	ORRCR73	8/23/2011	PQAUSER	8/23/2011	Approved ℹ
□	DQ_CARDSMY	FCT_CARDS_SUMMARY	Read/Write	Specific Check	ORRCR73	8/22/2011	PQAUSER	8/22/2011	Saved
□	DQCARDSMY3	FCT_CARDS_SUMMARY	Read/Write	Specific Check	ORRCR73	8/22/2011	PQAUSER	8/30/2011	Saved
□	DQCARDSMY3	FCT_CARDS_SUMMARY	Read/Write	Specific Check	ORRCR73	8/23/2011	PQAUSER	8/23/2011	Saved

The **Data Quality Rule Summary** screen displays the list of pre-defined Data Quality Rules with the other details such as DQ Name, Table Name, Access Type, Check Type, Folder, Creation Date, Created By, Last Modification Date, and Status of the Rule. A defined rule is displayed in Saved status, until it is Approved/Rejected by the approver. An Approved rule can be grouped in order for execution and a Rejected rule is sent back to the user with the Approver comments.

You can add, view, modify, copy, approve/reject, or delete Data Quality Rules within the **Data Quality Rule Summary** screen. You can also make use of Search and Pagination options to search for a Data Quality Rule based on DQ Name, Table Name, Folder, or Check Type and view the existing Data Quality Rules within the system.

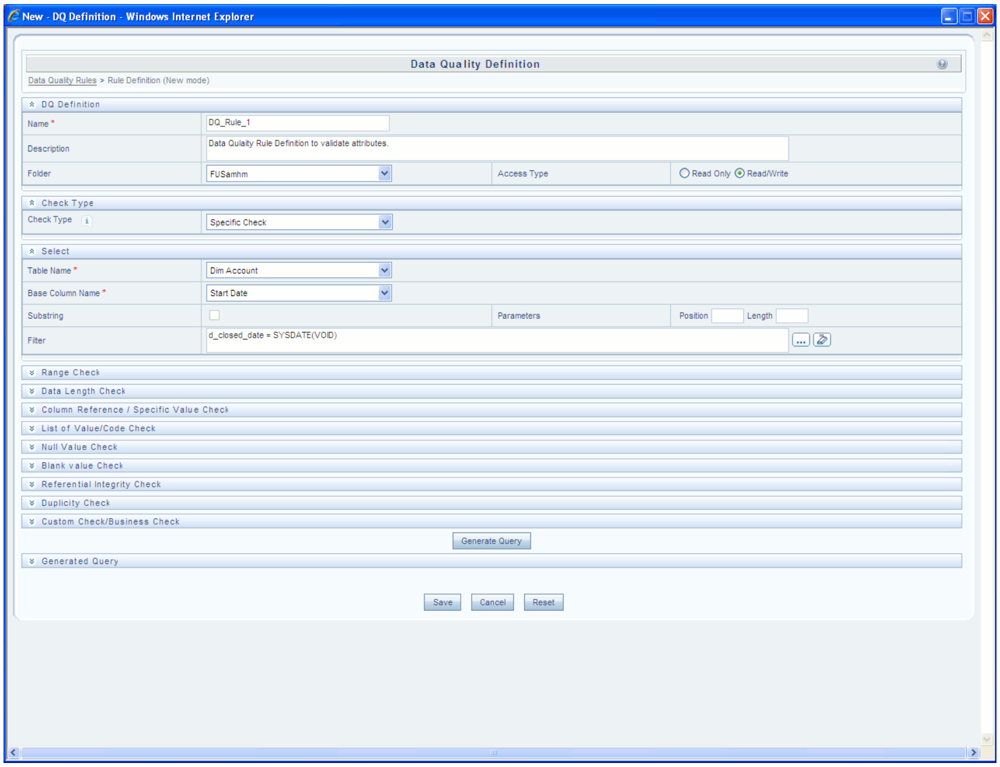
For more information, refer to *Oracle Financial Services Analytical Applications Infrastructure User Manual*.

Create Data Quality Rule

You can create a Data Quality Rule definition by specifying the DQ Definition details along with the type of validation check on the required table and defining the required validation conditions to query and correct the transformed data.

To create Data Quality Rule in the **Data Quality Rule Summary** screen:

1. Click **Add** button in the Data Quality Rules tool bar. Add button is disabled if you have selected any check box in the grid. The **Data Quality Definition** screen is displayed.



New - DQ Definition - Windows Internet Explorer

Data Quality Definition

Data Quality Rules > Rule Definition (New mode)

DQ Definition

Name *	DQ_Rule_1		
Description	Data Qulaity Rule Definition to validate attributes.		
Folder	FUSamhm	Access Type	○ Read Only ● Read/Write

Check Type

Check Type	Specific Check

Select

Table Name *	Dim Account		
Base Column Name *	Start Date		
Substring	□	Parameters	Position Length
Filter	d_closed_date = SYSDATE(VOID)		

2. In the DQ definition section, perform the following:

   - Enter the **Name** by which you can identify the DQ definition.
   - Enter a description or related information about the definition.
   - Select the **Folder** (available for selected Information Domain) from the drop down list.
   - Select the **Access Type** as either Read Only or Read/Write.

3. Select the **Check Type** from the drop down list. You can mouse-over i icon for information.

   - Select **Specific Check**, if the defined conditions are based on individual checks on a single column.
   - Select **Generic Check**, if the defined conditions are based on multiple columns of a single base table. These checks are not pre-defined and can be specified (user-defined) as required.

   If *Specific Check* is selected, perform the following:

   - Select **Table Name** and **Base Column Name** from the drop down list. The list displays all the tables which are marked for Data Quality Rule in a data model, which has the table classification property code set to 340.

- (Optional) If you have selected Base Column of type Varchar/Char, select the **Substring** check box, enter numeric values in **Parameters** Position and Length fields.
- Click the below button and define the **Filter** condition using the **Specify Expression** screen.

  ...

- Define the required **Validation Checks** by selecting the appropriate grid and specify the details. You can define nine specific validation checks based on Range, Data Length, Column Reference/Specific Value, List of Value/Code, Null Value, Blank Value, Referential Integrity, Duplicity, and Custom Check/Business.

  For more information, refer to *Oracle Financial Services Analytical Applications Infrastructure User Manual*.

  > **Note:** A minimum of one Validation check must be defined to generate a query.

- Click **Generate Query**. The details are validated and the validated query along with the status is displayed in the Generated Query section.

If *Generic Check* is selected, perform the following:

- Select **Table Name** from the drop down list. The list displays all the tables which are marked for Data Quality Rule in a data model, which has the table classification property code set to 340.
- Click the below button and define the **Filter** condition using the **Specify Expression** screen.

  ...

- Click **Add** button in the **Condition** grid. The Specify Expression screen is displayed. Define the Condition expression.

  The Expression is displayed with the "IF" and "Else" conditions along with the **Severity** status as either **Error** or **Warning**. You can change the **Severity** by selecting from the drop down list.

Note: You can add an Assignment only when the Severity is selected as **Warning**. Assignments are added when you want to correct or update record(s) in base column data / selected column data. There can be one or more assignments tagged to a single condition. However, selecting severity as **Error** indicates there are no corrections and only facilitates in reporting the quantity of bad records.

- Select the check box adjacent to the required Condition expression and click **Add** button in the **Assignment** grid. The assignment details are populated.

  Note: You can add an Assignment only if the Severity is **Warning**. There can be one or more assignments tagged to a single condition.

- Specify the Assignment details as tabulated.

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 existsfor the selected base column*.
Assignment Value	Select the Assignment Value from the drop-down list according to the Assignment Type selected.

Field	Description
Message Severity	Select the **Message Severity** as either **1** or **2** from the drop down list.
Message	Select the required **Message** for the *Severity* from the drop down list.

You can also add multiple assignments by clicking **Add** button in **Assignment** grid.

> **Note:** Minimum of one condition needs to be defined to save the Rule.

4. Click **Save**. The defined Data Quality Rule definition is displayed in the **Data Quality Rule Summary** screen with the status as "Saved".

View Data Quality Rule

You can view individual Data Quality Rule definition details at any given point.

To view the existing Data Quality Rule definition in the **Data Quality Rule Summary** screen:

1. Select the check box adjacent to the required DQ Name.

2. Click **View** button from the Data Quality Rules tool bar.

   The DQ Definition screen displays the details of the selected Data Quality definition. The **Audit Trail** section at the bottom of DQ Definition screen displays metadata information about the Data Quality Rule defined.

Modify Data Quality Rule

You can update the existing Data Quality Rule definition details except for the Definition Name, Table, and Base Column selected.

To update the required Data Quality Rule definition details in the **Data Quality Rule Summary** screen:

1. Select the check box adjacent to the required DQ Name.

   > **Note:** You can only edit those rules which have status either as **Saved** or as **Rejected**.

2. Click **Edit** button from the Data Quality Rules tool bar. The Edit button is disabled if you have selected multiple DQ Names. The DQ Definition screen is displayed. Update the details as required.

   For more information, see Create Data Quality Rule section.

3. Click **Save** to update the changes.

Copy Data Quality Rule

You can copy the existing Data Quality Rule to quickly create a new DQ definition based on the existing rule details or by updating the required parameters.

To copy an existing Data Quality Rule definition in the **Data Quality Rule Summary** screen:

1. Select the check box adjacent to the required DQ Name in the list whose details are to be duplicated.

2. Click **Copy** button from the Data Quality Rules tool bar. **Copy** button is disabled if you have selected multiple check boxes. The DQ Definition screen is displayed.

3. Edit the DQ definition Name and other details as required.

   For more information, see Create Data Quality Rule section.

4. Click **Save**. The defined Data Quality Rule definition is displayed in the **Data Quality Rule Summary** screen with the status as "Saved".

Approve/Reject Data Quality Rule

You (Authorizer) can Approve a pre-defined Data Quality Rule definition for further execution or Reject an inappropriate DQ definition listed within the **Data Quality Rule Summary** screen. You should be mapped to DQ Authorizer function role to Approve or Reject a DQ definition.

To Approve/Reject Data Quality Rule in the Data Quality Rule Summary screen:

1. Select the checkbox adjacent to the required DQ Name. Ensure that you select the " Saved" DQ definition based on the **Status** indicated in the **Data Quality Rules** grid.

2. Perform one of the following:

   - To **Approve** the DQ definition, click **Approve** button. The **User Comments** screen is displayed. Enter the notes or additional information to the user and click **OK**. The selected DQ definition is approved and a confirmation dialog is displayed.

   - To **Reject** the DQ definition, click **Reject** button. The **User Comments** screen is displayed. Enter the notes or additional information to the user and click **OK**.

The selected DQ definition is rejected and a confirmation dialog is displayed.

> **Note:** The authorizer can approve/reject only one definition at a time.

The Approved/Rejected status of the DQ definition is indicated in the **Status** column of the **Data Quality Rule Summary** screen. You can mouse-over i button to view the Approver comments in a pop-up.

Delete Data Quality Rule

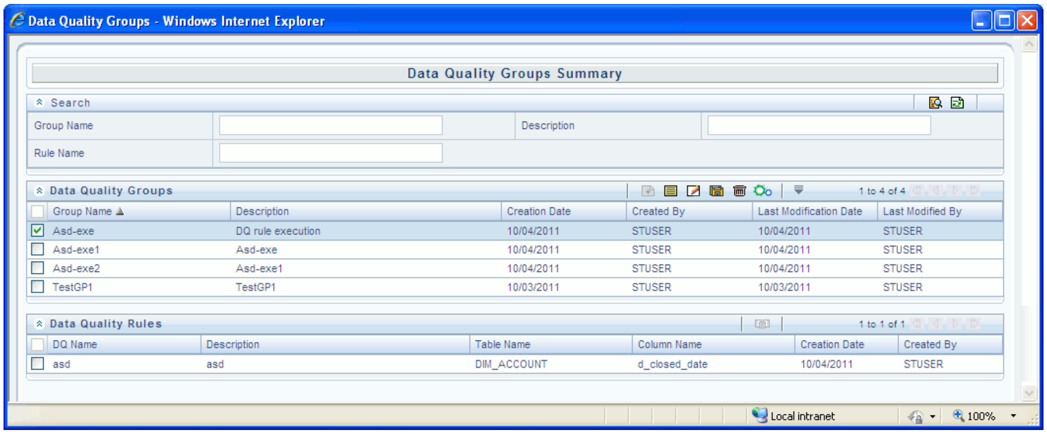
You can remove Data Quality Rule definition(s) which are created by you and which are no longer required in the system by deleting from **Data Quality Rule Summary** screen.

1. Select the check box adjacent to the required DQ Name whose details are to be removed.
2. Click **Delete** button from the Data Quality Rules tool bar.
3. Click **OK** to confirm deletion.

Data Quality Groups Summary

Data Quality Groups Summary within the Data Integrator framework of Infrastructure system facilitates you to logically group the define DQ definitions and schedule for execution. DQ definitions can be executed either through Data Quality Groups Summary screen of Data Integrator framework or in Batch Execution screen of Operations module.

When a Data Quality Group is executed for processing, the details of the execution are captured in a log file. You (Business Analysts) need to have ETL Analyst function role mapped to access the Data Quality Summary framework within the Infrastructure system. You can access Data Quality Groups Summary by expanding the Data Quality framework within the Unified Metadata Manager section in tree structure of LHS menu.



Data Quality Groups - Windows Internet Explorer
Data Quality Groups Summary
Search

Group Name		Description	
Rule Name			

Data Quality Groups
1 to 4 of 4

□	Group Name ▲	Description	Creation Date	Created By	Last Modification Date	Last Modified By
☑	Asd-exe	DQ rule execution	10/04/2011	STUSER	10/04/2011	STUSER
□	Asd-exe1	Asd-exe	10/04/2011	STUSER	10/04/2011	STUSER
□	Asd-exe2	Asd-exe1	10/04/2011	STUSER	10/04/2011	STUSER
□	TestGP1	TestGP1	10/03/2011	STUSER	10/03/2011	STUSER

Data Quality Rules
1 to 1 of 1

□	DQ Name	Description	Table Name	Column Name	Creation Date	Created By
□	asd	asd	DIM_ACCOUNT	d_closed_date	10/04/2011	STUSER

The **Data Quality Groups Summary** screen displays the list of pre-defined Data Quality Groups with the other details such as Group Name, Description, Creation Date, Created By, Last Modification Date, and Last Modified By. You can Create and Execute DQ Group definitions and view, modify, copy, or delete DQ Group definitions within the **Data Quality Groups Summary** screen.

You can also make use of Search and Pagination options to search for a DQ Group definition based on Group Name, Description, or Rule Name and view the existing DQ Group definitions within the system.

Create Data Quality Group

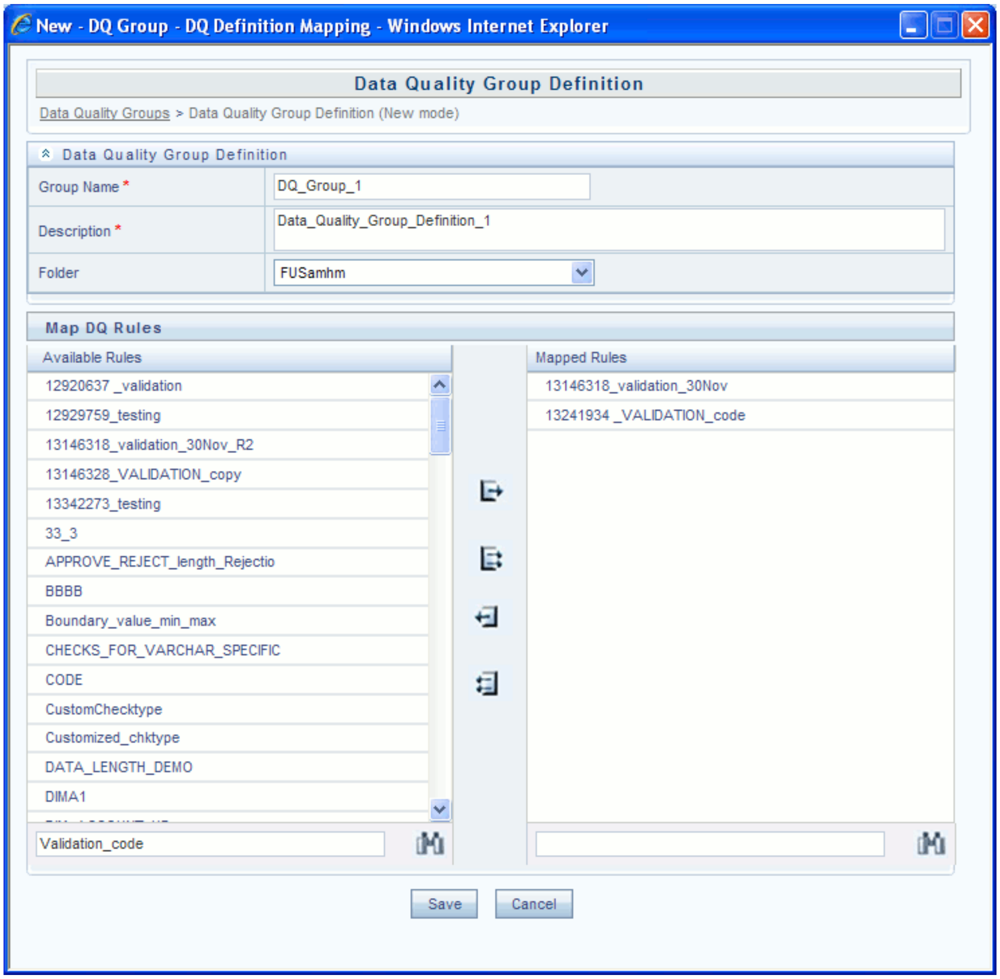
You can create a DQ Group definition by defining the DQ Definition details and mapping the required DQ Rules which are authorized and approved within the system.

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.


New - DQ Group - DQ Definition Mapping - Windows Internet Explorer
Data Quality Group Definition
Data Quality Groups > Data Quality Group Definition (New mode)
Data Quality Group Definition
Map DQ Rules
Save
Cancel

Group Name *	DQ_Group_1
Description *	Data_Quality_Group_Definition_1
Folder	FUSamhm

Available Rules	Mapped Rules
12920637 _validation	13146318_validation_30Nov
12929759_testing	13241934 _VALIDATION_code
13146318_validation_30Nov_R2	
13146328_VALIDATION_copy	
13342273_testing	
33_3	
APPROVE_REJECT_length_Rejectio	
BBBB	
Boundary_value_min_max	
CHECKS_FOR_VARCHAR_SPECIFIC	
CODE	
CustomChecktype	
Customized_chktype	
DATA_LENGTH_DEMO	
DIMA1	
Validation_code	

2. In the **Data Quality Group Definition** section, perform the following:

   - Enter the **Group Name** by which you can identify the DQ Group.

   - Enter a description or related information about the DQ Group.

   - Select the **Folder** (available for selected Information Domain) from the drop down list.

3. In the Map DQ Rules section, perform the following:

   - Select the required DQ Rule from the **Available Rules** list and click **Select**. You can also search to select a specific DQ Rule by entering the required keyword and clicking **Find** button.

   - To select all the listed DQ Rules, click **Select All.**

     You can also deselect a DQ Rule by selecting from the Mapped Rules list and clicking **Deselect** or clicking **Deselect All** to deselect all the mapped rules. You can search to deselect a specific DQ Rule by entering the keyword and clicking button. You can search to deselect a specific DQ Rule by entering the keyword and clicking **Find** button.

4. Click **Save**. The defined DQ group is listed in the **Data Quality Rule Summary** screen and can be executed for processing.

Execute Data Quality Group

You can execute a defined DQ Group Definitions along with the mapped Rules and validation checks in the **Data Quality Rule Summary** screen. You can also execute a DQ Group in the Batch Execution screen of Operations module.

To Execute a DQ Group in the **Data Quality Rule Summary** screen:

1. Select the checkbox adjacent to the required Group Name.
2. Click **Execute** button from the Data Quality Groups tool bar. Execute button is disabled if you have selected multiple check boxes. The **Group Execution** screen is displayed.
3. In the Batch Details section, perform the following:
   - Select the **MIS Date** using the Calendar. MIS Date refers to the date with which the data for the execution would be filtered. In case MIS date is not present in the target table, execution happens ignoring the date parameter.

     Note: The **DQ Batch ID** is auto populated and is not editable.

   - Specify the percentage of **Threshold (%)** limit in numeric value. This refers to the maximum percentage of records that can be rejected in a job. If the percentage of failed records exceeds the Rejection Threshold, the job will fail. If the field is left blank, the default the value is set to 100%.
4. Click **Execute**. A confirmation message is displayed and the DQ Group is scheduled for execution.

   Once the DQ Group is executed, you can view the details of the execution along with the log information in the **View Logs** screen.

   For more information, see View Data Quality Group Summary Log section.

View Data Quality Group

You can view individual Data Quality Group definition details at any given point.

To view the existing DQ Group definition in the **Data Quality Group Summary** screen:

1. Select the check box adjacent to the required Group Name. The mapped DQ Rules are displayed in the Data Quality Rules section.

2. Click **View** button from the Data Quality Groups tool bar. The Data Quality Group Definition screen displays the DQ definition details.

Modify Data Quality Group

You can update the existing DQ Group definition details except for the Group Name.

To updated the required DQ Group definition details in the **Data Quality Groups Summary** screen:

1. Select the check box adjacent to the required Group Name.

2. Click **Edit** button from the Data Quality Groups tool bar. The Data Quality Group Definition screen is displayed.

3. Update the details and click **Save** to update the changes.

Copy Data Quality Group

You can copy the existing DQ Group details to quickly create a new DQ definition based on the existing details or by updating the required parameters.

To copy an existing DQ Group definition in the **Data Quality Groups Summary** screen:

1. Select the check box adjacent to the required Group Name in the list whose details are to be duplicated.

2. Click **Copy** button from the Data Quality Groups tool bar. Copy button is disabled if you have selected multiple check boxes. The Data Quality Group Definition screen is displayed.

3. Edit the DQ Group Name and other details as required.

4. Click **Save**. The new DQ Group definition is displayed in the **Data Quality Groups Summary** screen.

View Data Quality Group Summary Log

You can view the execution log details of Data Quality Rules in the View Logs screen. The **View Logs** screen displays the details such as Check Name, Log Message, Message Date, Message Time, Total Rows, Rows Impacted, Assignment Type, Assignment Severity, and Severity Message of the executed Data Quality Rules.

To view the Data Quality Rule execution log details in the **Data Quality Groups Summary** screen:

1. Select the check box adjacent to the Group Name in the **Data Quality Groups** grid. The Data Quality Rules associated with the selected Group are displayed in the Data Quality Rules grid.

2. Select the check box adjacent to the DQ Name in the Data Quality Rules grid.

3. Click **View Logs** button. The View Logs screen is displayed.

4. In the View Logs screen, select the **Information Date** from the drop down list. Based on the selection, you can select the **Group Run ID** and **Iteration ID** from the corresponding drop-down list.

5. Click the below button from the Group Execution details tool bar.

   The Data Quality Rule Logs grid displays the execution log details of the selected Data Quality Rule. You can also click **Reset** button in the Group Execution details tool bar to reset the selection.

Delete Data Quality Group

You can remove the DQ Group definition(s) which are created by you and which are no longer required in the system by deleting from Data Quality Groups Summary screen.

To delete:

1. Select the check box adjacent to the required Group Name whose details are to be removed.

2. Click **Delete** button from the Data Quality Groups tool bar.

3. Click **OK** in the information dialog to confirm deletion.

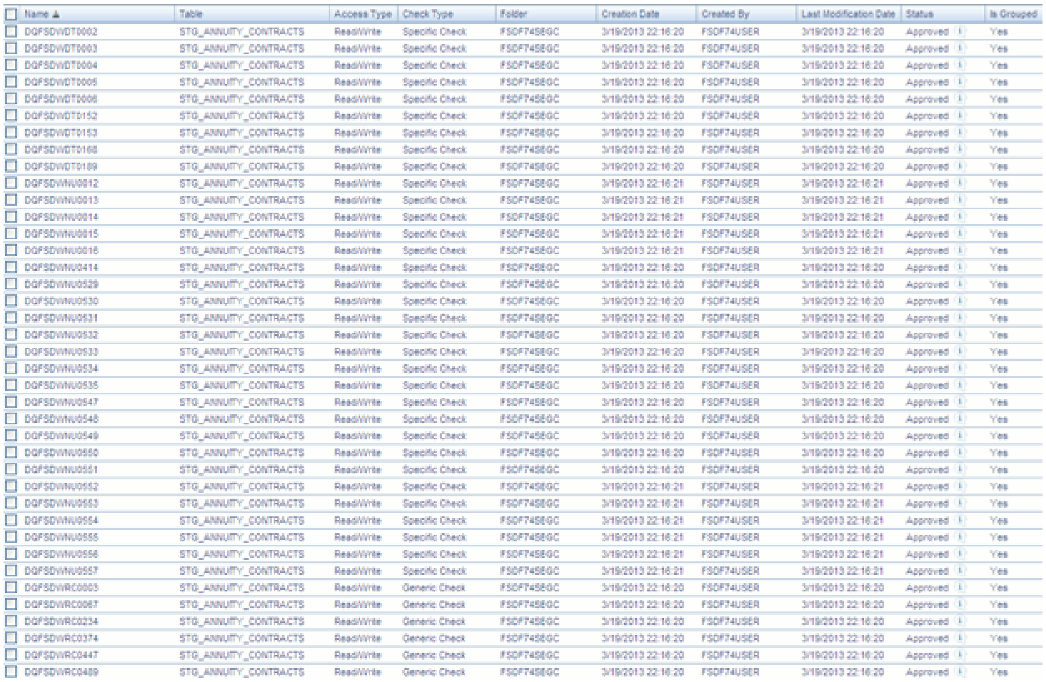
Data Quality Rules For Staging Tables

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 the following spreadsheet.

DQ_CHECK_RULES.xlsx

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



□	Name ▲	Table	Access Type	Check Type	Folder	Creation Date	Created By	Last Modification Date	Status	Is Grouped
□	DQFSDWDT0002	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWDT0003	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWDT0004	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWDT0005	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWDT0006	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWDT0152	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWDT0153	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWDT0168	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWDT0189	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWNU0012	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:21	FSDF74USER	3/19/2013 22:16:21	Approved	Yes
□	DQFSDWNU0013	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:21	FSDF74USER	3/19/2013 22:16:21	Approved	Yes
□	DQFSDWNU0014	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:21	FSDF74USER	3/19/2013 22:16:21	Approved	Yes
□	DQFSDWNU0015	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:21	FSDF74USER	3/19/2013 22:16:21	Approved	Yes
□	DQFSDWNU0016	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:21	FSDF74USER	3/19/2013 22:16:21	Approved	Yes
□	DQFSDWNU0414	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWNU0529	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWNU0530	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWNU0531	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWNU0532	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWNU0533	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWNU0534	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWNU0535	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWNU0547	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWNU0548	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWNU0549	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWNU0550	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWNU0551	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWNU0552	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:21	FSDF74USER	3/19/2013 22:16:21	Approved	Yes
□	DQFSDWNU0553	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:21	FSDF74USER	3/19/2013 22:16:21	Approved	Yes
□	DQFSDWNU0554	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:21	FSDF74USER	3/19/2013 22:16:21	Approved	Yes
□	DQFSDWNU0555	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:21	FSDF74USER	3/19/2013 22:16:21	Approved	Yes
□	DQFSDWNU0556	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:21	FSDF74USER	3/19/2013 22:16:21	Approved	Yes
□	DQFSDWNU0557	STG_ANNUITY_CONTRACTS	Read/Write	Specific Check	FSDF74SEGC	3/19/2013 22:16:21	FSDF74USER	3/19/2013 22:16:21	Approved	Yes
□	DQFSDWRC0003	STG_ANNUITY_CONTRACTS	Read/Write	Generic Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWRC0067	STG_ANNUITY_CONTRACTS	Read/Write	Generic Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWRC0234	STG_ANNUITY_CONTRACTS	Read/Write	Generic Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWRC0374	STG_ANNUITY_CONTRACTS	Read/Write	Generic Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWRC0447	STG_ANNUITY_CONTRACTS	Read/Write	Generic Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes
□	DQFSDWRC0489	STG_ANNUITY_CONTRACTS	Read/Write	Generic Check	FSDF74SEGC	3/19/2013 22:16:20	FSDF74USER	3/19/2013 22:16:20	Approved	Yes

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, page 7-11.

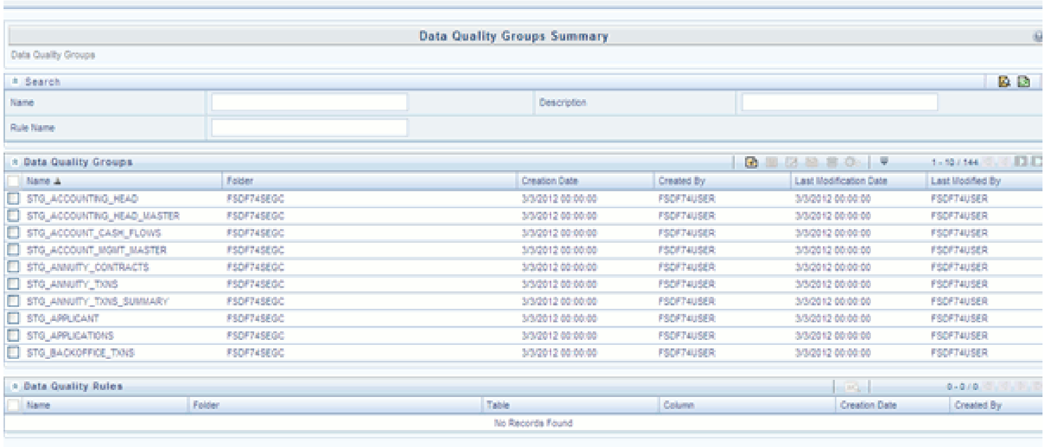
The following spreadsheet displays the total groups and corresponding Rules mapped to that Group.

DQ_GROUP_MAPPING.xlsx

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:



Data Quality Groups Summary

Data Quality Groups

Search

Name

Description

Rule Name

Data Quality Groups 1 - 10 / 144

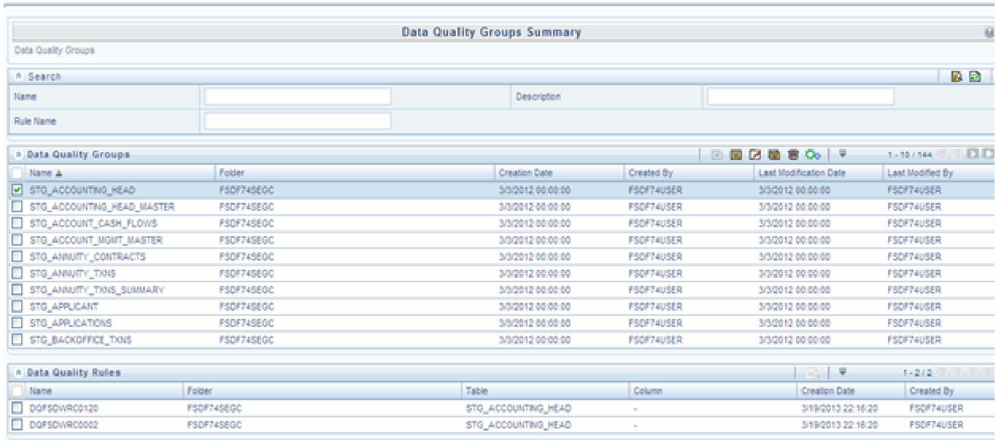
	Name	Folder	Creation Date	Created By	Last Modification Date	Last Modified By
□	STG_ACCOUNTING_HEAD	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ACCOUNTING_HEAD_MASTER	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ACCOUNT_CASH_FLOWS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ACCOUNT_MGMT_MASTER	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ANNUITY_CONTRACTS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ANNUITY_TXNS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ANNUITY_TXNS_SUMMARY	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_APPLICANT	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_APPLICATIONS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_BACKOFFICE_TXNS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER

Data Quality Rules 0 - 0 / 0

	Name	Folder	Table	Column	Creation Date	Created By
			No Records Found			

1. Select the check box adjacent to the Group, you want to execute.

   The corresponding DQ Rules are displayed in *Data Quality Rules* grid.



Data Quality Groups Summary

Data Quality Groups

Search

Name

Description

Rule Name

Data Quality Groups 1 - 10 / 144

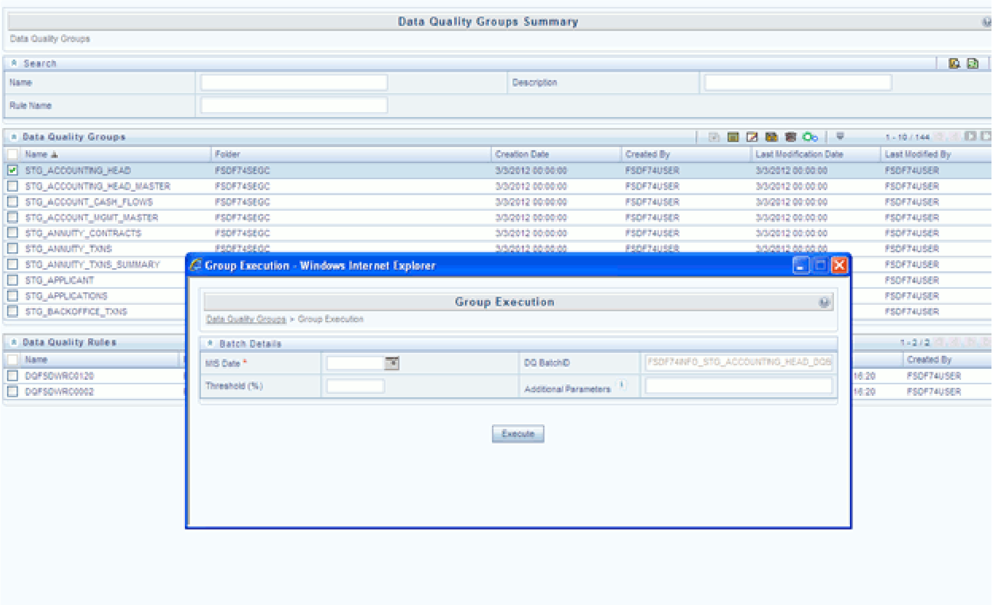
	Name	Folder	Creation Date	Created By	Last Modification Date	Last Modified By
☑	STG_ACCOUNTING_HEAD	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ACCOUNTING_HEAD_MASTER	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ACCOUNT_CASH_FLOWS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ACCOUNT_MGMT_MASTER	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ANNUITY_CONTRACTS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ANNUITY_TXNS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ANNUITY_TXNS_SUMMARY	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_APPLICANT	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_APPLICATIONS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_BACKOFFICE_TXNS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER

Data Quality Rules 1 - 2 / 2

	Name	Folder	Table	Column	Creation Date	Created By
□	DQFSDWRC0120	FSDF74SEGC	STG_ACCOUNTING_HEAD	-	3/19/2013 22:16:20	FSDF74USER
□	DQFSDWRC0002	FSDF74SEGC	STG_ACCOUNTING_HEAD	-	3/19/2013 22:16:20	FSDF74USER

2. Click **Execute** button.

   The *Group Execution* window is displayed.



Data Quality Groups Summary

Data Quality Groups

Search

Name | Description

Rule Name

Data Quality Groups — 1 - 10 / 144

	Name	Folder	Creation Date	Created By	Last Modification Date	Last Modified By
☑	STG_ACCOUNTING_HEAD	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ACCOUNTING_HEAD_MASTER	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ACCOUNT_CASH_FLOWS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ACCOUNT_MGMT_MASTER	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ANNUITY_CONTRACTS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ANNUITY_TXNS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ANNUITY_TXNS_SUMMARY					FSDF74USER
□	STG_APPLICANT					FSDF74USER
□	STG_APPLICATIONS					FSDF74USER
□	STG_BACKOFFICE_TXNS					FSDF74USER

Data Quality Rules — 1 - 2 / 2

	Name		Created By
□	DQFSDWRC0120	16:20	FSDF74USER
□	DQFSDWRC0002	16:20	FSDF74USER

Group Execution - Windows Internet Explorer

Group Execution

Data Quality Groups > Group Execution

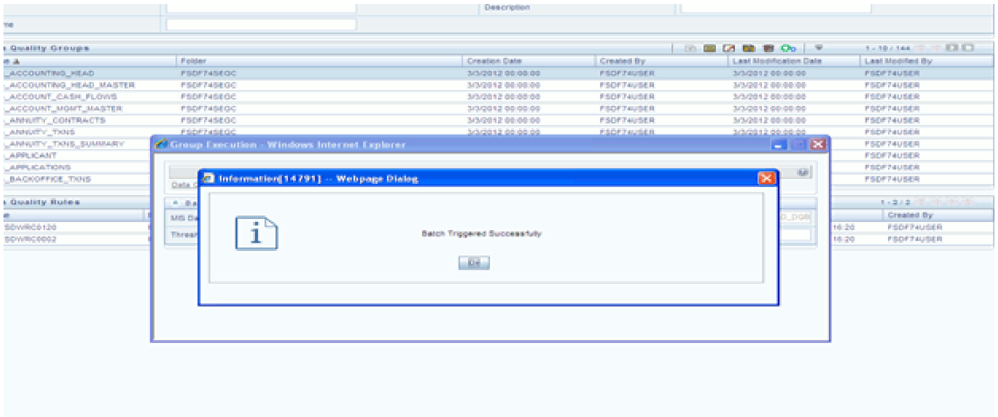
Batch Details

MIS Date *		DQ BatchID	FSDF74INFO_STG_ACCOUNTING_HEAD_DQB
Threshold (%)		Additional Parameters	

3. Enter the MIS Date.

4. Click **Execute** button from the *Group Execution* window.

   The execution is triggered and the following message is displayed.



Group Execution - Windows Internet Explorer

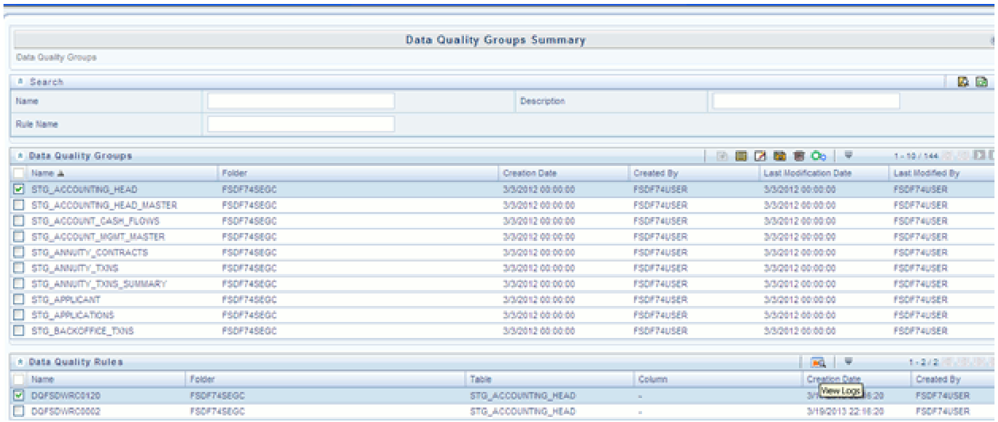
Information[14791] -- Webpage Dialog

Batch Triggered Successfully

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 tot see the execution log.



Data Quality Groups Summary

Data Quality Groups

Search

Name		Description	
Rule Name			

Data Quality Groups 1 - 10 / 144

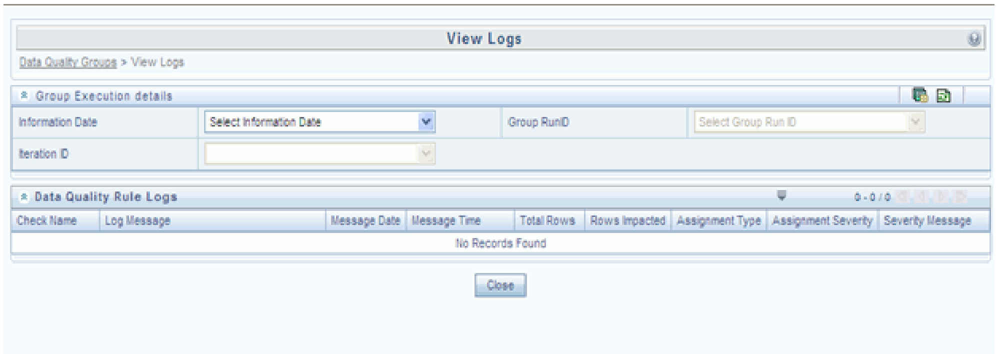
	Name	Folder	Creation Date	Created By	Last Modification Date	Last Modified By
☑	STG_ACCOUNTING_HEAD	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ACCOUNTING_HEAD_MASTER	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ACCOUNT_CASH_FLOWS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ACCOUNT_MGMT_MASTER	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ANNUITY_CONTRACTS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ANNUITY_TXNS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_ANNUITY_TXNS_SUMMARY	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_APPLICANT	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_APPLICATIONS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER
□	STG_BACKOFFICE_TXNS	FSDF74SEGC	3/3/2012 00:00:00	FSDF74USER	3/3/2012 00:00:00	FSDF74USER

Data Quality Rules 1 - 2 / 2

	Name	Folder	Table	Column	Creation Date	Created By
☑	DQFSDWRC0120	FSDF74SEGC	STG_ACCOUNTING_HEAD	-	[illegible]	FSDF74USER
□	DQFSDWRC0002	FSDF74SEGC	STG_ACCOUNTING_HEAD	-	3/19/2013 22:16:20	FSDF74USER

2. Click **View Logs** button.

   The *View Logs* window is displayed



View Logs

Data Quality Groups > View Logs

Group Execution details

Information Date	Select Information Date	Group RunID	Select Group Run ID
Iteration ID			

Data Quality Rule Logs 0 - 0 / 0

Check Name	Log Message	Message Date	Message Time	Total Rows	Rows Impacted	Assignment Type	Assignment Severity	Severity Message
No Records Found								

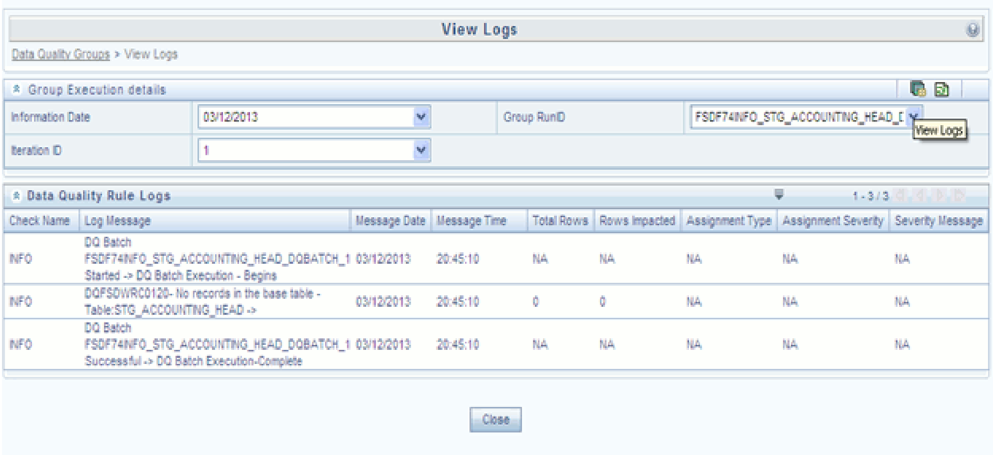
3. Select the **Information Date** from the drop down list.

4. Select the **Group Run ID** from the drop down list.

5. Select the **Iteration ID** from the drop down list.

6. Click **View Logs** button.

   The details of the selected Group Execution are displayed.



Information Date: 03/12/2013

Group RunID: FSDF74INFO_STG_ACCOUNTING_HEAD_I

Iteration ID: 1

Check Name	Log Message	Message Date	Message Time	Total Rows	Rows Impacted	Assignment Type	Assignment Severity	Severity Message
INFO	DQ Batch FSDF74INFO_STG_ACCOUNTING_HEAD_DQBATCH_1 Started -> DQ Batch Execution - Begins	03/12/2013	20:45:10	NA	NA	NA	NA	NA
INFO	DQFSDWRC0120- No records in the base table - Table:STG_ACCOUNTING_HEAD ->	03/12/2013	20:45:10	0	0	NA	NA	NA
INFO	DQ Batch FSDF74INFO_STG_ACCOUNTING_HEAD_DQBATCH_1 Successful -> DQ Batch Execution-Complete	03/12/2013	20:45:10	NA	NA	NA	NA	NA

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 DQ Batches and corresponding groups are detailed in the following spreadsheet.

DQ_BATCH_GROUP_MAPPING.xlsx

For more information regarding Batch Execution, refer to **Operations** module of *Oracle Financial Services Analytical Applications Infrastructure User Manual*.

8

Slowly Changing Dimension (SCD)

Overview of SCD Process

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

- **Type 1 SCDs - Overwriting**

  In a Type 1 SCD the new data overwrites the existing data. Thus the existing data is lost as it is not stored anywhere else. This is the default type of dimension you create. You do not need to specify any additional information to create a Type 1 SCD.

- **Type 2 SCDs - Creating another dimension record**

  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.

- **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.

The SCDs used in OFSDF are listed in the following spreadsheet.

SCD.xls

> **Note:** The preceding spreadsheet contains the list of all the SCDs. Only those are relevant to OIDF must be used.

Load DIM_ACCOUNT through SCD

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

- V_ACCOUNT_NUMBER
- N_ACCT_SKEY
- N_RCV_LEG_ACCT_SKEY
- FIC_MIS_DATE

This approach replaces the function load in which the table DIM_ACCOUNT is getting loaded through the function, FN_POPDIMACCOUNT. This loads the following columns into DIM_ACCOUNT table:

- V_ACCOUNT_NUMBER
- N_ACCT_SKEY
- N_RCV_LEG_ACCT_SKEY
- FIC_MIS_DATE

Where, the sources are the different product processor tables present in the solution, which are configured in FSI_DIM_ACCOUNT_SETUP_DETAILS table.

DIM_ACCOUNT SCD

Batch <INFODOM>DIM_ACCOUNT_SCD has been introduced with 30 tasks under it. These 30 tasks represent the 30 SCD processes where different product processors would be the source and DIM_ACCOUNT would be the target. MAP_REF_NUMs 188 to 217 have been introduced into SYS_TBL_MASTER table, and subsequently into SYS_STG_JOIN_MASTER.

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

Improve 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 are existing.

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

Example

MERGE_HINT and SESSION_ENABLE_STATEMENT in SYS_TBL_MASTER

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

- All the tasks can be executed in parallel. This might cause the N_RCV_LEG_ACCT_SKEY to have an incremental value as compared to N_ACCT_SKEY.
- Execute the SQL file with all the SESSION_DISABLE_STATEMENTs, after the successful completion of the SCD batch.
  - Once the DIM_ACCOUNT table is populated using this approach, you will not be able to use the initial approach (FN_POPDIMACCOUNT) as this will lead to skey conflict.
  - 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.
  - The F_LATEST_RECORD_INDICATOR for an existing DIM_ACCOUNT data already loaded by the function should be updated to 'Y' before running the SCD, failing which a new skey might get generated for the same account number.
  - SCD execution occurs based on the GAAP code which is configured in SETUP_MASTER table. This has been introduced to tackle the scenario of multiple GAAP codes. Whether or not there exist multiple GAAP codes, SETUP_MASTER should be manually configured as follows:

V_COMPONENT_CODE	V_COMPONENT_DESC	V_COMPONENT_VALUE
DEFAULT_GAAP	DEFAULT_GAAP	USGAAP

Where V_COMPONENT_VALUE should be manually populated with the required GAAP code.

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

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

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

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

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

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.

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.

This appendix covers the following topics:

- Naming Conventions Used in OIDF LDM and PDM

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 OIDF Logical Data Model (LDM) and Physical Data Model (PDM).

- OIDF Logical Data Model Naming Conventions, page A-1
- OIDF Physical Data Model Naming Conventions, page A-4
- Domains (PDM and LDM), page A-8

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, page A-2
- LDM Physical View Mode Table Naming Conventions, page A-3

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:

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

Attribute names also contain common set of suffixes for the classes of attributes as listed in table below:

Class of Attribute	Suffix	Required Values (If any)
Unique Identifier	ID	
Reference Identifiers	Reference ID	
Code	Code	
Number	Number	
Name	Name	
Description	Description	
Boolean Values	Flag	Y or N

Class of Attribute	Suffix	Required Values (If any)
Indicator	Indicator	More than one value ('A', 'B', 'C' etc)
Percentage	Percent	
Rate	Rate	
Amount	Amount	
Balance	Balance	
Term	Term	
Frequency	Frequency	
Unit	Unit	
Record created by		Created By
Record modified by		Modified By
Record created on		Created On
Record modified on		Modified On
Record validity start date		From Date
Record validity end date		End Date

LDM Physical View Mode Table Naming Conventions

In the physical view of the 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 Address	FSW_ACCT_ACCOUNT_ADDRESS	Account Address entity belonging primarily to Account (ACCT) subject area.
Letter Of Credit	FSW_CON_LETTER_OF_CREDIT	Letter of Credit entity belongs primarily to the Contracts (CON) subject area.

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, page A-4

- PDM Physical View Mode Naming Conventions, page A-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 Internet Transactions

PDM Area		Prefix	Suffix	Example
	Transaction Tables		Transactions	Stage Card Settlement Transactions
	Master Tables		Master	Stage Product Master
Results (Datamart)	All Fact Tables	Fact		Fact Process Cashflow
	All Dimension Tables		Dimension	Account Dimension

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_LOAN_CONTRACTS
	Interface tables for data from MDM and core banking systems		_INTF	STG_ORG_UNIT_ATTR_INTF
	Transaction Tables		_TXNS	STG_CARDS_SETTLEMENT_TXNS
	Master Tables		_MASTER	STG_CUSTOMER_MASTER
Results (Datamart)	All Fact Tables	FCT_		FCT_ACCOUNT_SUMMARY
	All Dimension Tables	DIM_		DIM_PRODUCT

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 table below:

Name	Abbreviated Form	Name	Abbreviated Form
Accrual	accr	Local Currency	lcy
Account	acct	Line Of Business	lob
Accounting Currency	acy	Maximum	max
Address	addr	Minimum	min
Adjustment	adj	Mortgage	mort
Advance	adv	Message	msg
Amount	amt	Multiplier	mult
Application	app	Number	num
Average	avg	Over Draft	od
Balance	bal	Option	opt
Business	bus	Origination	org
Currency	ccy	Percent	pct
Consolidation	cons	Payment	pmt
Customer	cust	Prepayment	ppmt
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

Name	Abbreviated Form	Name	Abbreviated Form
Flag	flg	Transaction	txn
Frequency	freq	Value at Risk	var
Future	fut	Value	val
Forex	fx		
Generation	gen		
General Ledger	gl		
Hierarchy	hier		
History	hist		

Domains (PDM and LDM)

Domains are Logical data types that are attached to each column within the model. The following table lists the domains and their descriptions.

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)

Serial No	Domain Name	Domain Description
9	Percent	NUMBER(10,6)
10	Percent_Long	NUMBER(15,11)
11	Phone_Fax_Number	NUMBER(15)
12	Rate	NUMBER(10,6)
13	Term	NUMBER(5)
14	Alphanumeric_Code	VARCHAR2(10)
15	Name	VARCHAR2(60)
16	Currency_Code	VARCHAR2(3)
17	Short_Description	VARCHAR2(60)
18	Description	VARCHAR2(255)
19	Account_Number	VARCHAR2(25)
20	System_Identifier	NUMBER(20)
21	Long_Description	VARCHAR2(4000)

B

Reporting Code Description

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
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