

Oracle Financial Services  
Data Foundation  
**User Guide**

Part No. E60528\_01

*January 2015*





Oracle Financial Services Data Foundation User Guide, Release 8.0.0.0.0

Part No. E60528-01

Copyright © 2014, 2015, Oracle and/or its affiliates. All rights reserved.

Primary Author: Asma Anjum/ Vineeta Mishra

Contributing Author: Surag Ramachandran

Contributor: Aditit Saikia, Guruprasad Thiruvachi, Kiran Kumar, Sapan Kanungo

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

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

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.





	<i>Preface</i> .....	i
	<i>Intended Audience</i> .....	i
	<i>Documentation Accessibility</i> .....	i
	<i>Access to Oracle Support</i> .....	i
	<i>Related Information Sources</i> .....	i
<b>CHAPTER 1</b>	<b><i>Introduction to OFSDF</i></b> .....	<b>1</b>
	<i>Overview</i> .....	1
	<i>Components of OFSDF</i> .....	1
	<i>Relationship to Oracle Financial Services Analytical Applications</i> .....	2
	<i>OFSDF Prerequisite Components and Tools</i> .....	3
<b>CHAPTER 2</b>	<b><i>Understanding OFSDF</i></b> .....	<b>5</b>
	<i>Background</i> .....	5
	<i>OFSDF Architecture</i> .....	5
	<i>Differences from Traditional Warehouse Architecture</i> .....	7
	<i>Staging Data Model</i> .....	8
	<i>OFSDF Logical Data Model</i> .....	17
	<i>Relationship to the OFSDF Physical Model</i> .....	17
	<i>OFSDF LDM Content Details</i> .....	19
	<i>Mapping from Logical Data Model to OFSDF Staging Area</i> .....	21
<b>CHAPTER 3</b>	<b><i>Dimension Loading Process</i></b> .....	<b>23</b>
	<i>Dimension Table Population</i> .....	23
	<i>Overview of SCD Process</i> .....	23
	<i>Prerequisites</i> .....	25
	<i>Tables Used by the SCD Component</i> .....	25
	<i>Executing the SCD Component</i> .....	28
	<i>Checking the Execution Status</i> .....	29
	<i>Load DIM_ACCOUNT through SCD</i> .....	30
<b>CHAPTER 4</b>	<b><i>Customer Summary Population</i></b> .....	<b>35</b>
	<i>Overview of Common Customer Summary Tables</i> .....	35
	<i>Prerequisites</i> .....	35
	<i>Executing the Customer Summary Population T2T</i> .....	35
<b>CHAPTER 5</b>	<b><i>Loading Multiple Load Runs in OFSAA</i></b> .....	<b>37</b>
	<i>Overview</i> .....	37
	<i>Objective</i> .....	37
	<i>Design Details</i> .....	38
	<i>Implementation</i> .....	38
	<i>Loading OFSAA Staging Tables</i> .....	40
	<i>Post Stage Load Process</i> .....	43
	<i>Loading data into OFSAA Results Tables from Staging Tables</i> .....	44
<b>CHAPTER 6</b>	<b><i>Account Summary Population</i></b> .....	<b>49</b>

	<i>Overview of Account Summary Tables .....</i>	49
	<i>Overview of Account Summary Population .....</i>	49
	<i>Executing the Account Summary Population T2T .....</i>	52
	<i>Fact Common Account Summary - Run Execution .....</i>	53
	<i>Checking the Execution Status .....</i>	53
	<i>Account Summary T2Ts .....</i>	54
	<i>Handling Alternate Currency .....</i>	54
	<i>Execution of Currency Exchange Rates Population T2T .....</i>	54
<b>CHAPTER 7</b>	<b><i>Fact Transaction Summary .....</i></b>	<b>57</b>
	<i>Table to Table .....</i>	57
	<i>Executing the Fact Transaction Summary .....</i>	59
<b>CHAPTER 8</b>	<b><i>Metadata Browser .....</i></b>	<b>61</b>
	<i>Overview .....</i>	61
	<i>Object View .....</i>	61
	<i>Metadata Publish .....</i>	62
	<i>Metadata Object to Application Map .....</i>	62
<b>CHAPTER 9</b>	<b><i>Time Dimension Population .....</i></b>	<b>63</b>
	<i>Overview of Time Dimension Population .....</i>	63
	<i>Prerequisites .....</i>	63
	<i>Tables Used by the Time Dimension Population Transformation .....</i>	64
	<i>Executing the Time Dimension Population Transformation .....</i>	64
	<i>Checking the Execution Status .....</i>	65
<b>CHAPTER 10</b>	<b><i>Using OFSDF .....</i></b>	<b>67</b>
	<i>Delivery Mechanism .....</i>	67
	<i>Installing OFSDF .....</i>	67
	<i>OFSDF Supporting Documentation .....</i>	67
	<i>Data Dictionary .....</i>	68
	<i>Download Specifications .....</i>	68
	<i>Extending OFSDF Physical Data Model .....</i>	68
<b>CHAPTER 11</b>	<b><i>Data Quality Rules Execution .....</i></b>	<b>73</b>
	<i>Data Quality Framework .....</i>	73
	<i>Data Quality Rules For Staging Tables .....</i>	83
	<i>Data Quality Groups for Staging Tables .....</i>	83
	<b><i>Appendix Naming Conventions Used in OFSDF Data Model .....</i></b>	<b>89</b>
	<i>Naming Conventions Used in OFSDF LDM and PDM .....</i>	89
	<b><i>Appendix How to Define a Batch .....</i></b>	<b>97</b>





---

# Preface

This chapter discusses the following topics:

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

## **Intended Audience**

Welcome to Release 8.0.0.0.0 of the Oracle Financial Services Data Foundation User Guide.

This document is the user guide and reference guide for the Oracle Financial Services Data Foundation (OFSDF) release 8.0.0.0.0, and is intended for System Administrator and all users who are instrumental in configuring and administering OFSDF with Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) platform.

See Related Information Sources 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

## **Related Information Sources**

- Technical Whitepaper on Data Model Document Generation
- Installation Guide for Oracle Financial Services Data Foundation 8.0.0.0.0
- Installation and Configuration Guide for OFSAAI release 8.0.0.0.0



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

This chapter includes the following topics:

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

## **Overview**

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

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

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

## **Components of OFSDF**

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

**Table 1. Components**

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

## Relationship to Oracle Financial Services Analytical Applications

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

### 1. Data Model

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

- **Synchronized Releases:** The Staging and Reporting models that are part of an OFSDF release are updated to reflect prior application-specific releases. This means that the latest release of OFSDF (8.0.0.0.0) reflects all prior application releases across OFSAA from a data model perspective, with respect to the Staging and Reporting models.

## 2. Infrastructure

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

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

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

## OFSDF Prerequisite Components and Tools

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

**Table 2. Pre-requisites**

Component	Provider	Purpose
Oracle Financial Services Analytical Applications Infrastructure version 8.0.0.0.0	Oracle	OFSAAI is the platform on which the OFSDF is deployed and operated. It represents the OFSDF 'runtime' environment, and consists of a number of tools used to manage the data lifecycle within OFSDF, from sourcing to reporting*.
Oracle Database Enterprise Edition 11gR2	Oracle	OFSDF is certified on Oracle Database releases 11gR2 and higher.
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.

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



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

This chapter covers the following topics:

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

## **Background**

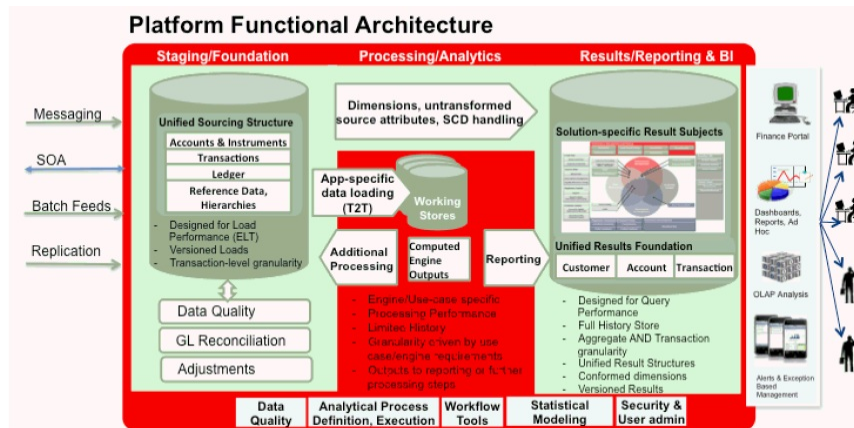
It is important to note that OFSDF architecture differs from 'traditional' data warehouse architecture in key ways.

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

## **OFSDF Architecture**

The following figure depicts the OFSDF Functional Architecture.

**Figure 1. Architecture**



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

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

The architecture illustrates the following key concepts:

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

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

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

- Metadata Management
- Data Quality
- Data Movement



- Scheduling and runtime operations
- Security/User management
- Analytical Process Definition and Execution

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

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

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

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

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

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

## Differences from Traditional Warehouse Architecture

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

**Table 3. Differences of the OFSDF**

Component	FSDf	Traditional BI Architecture
Staging Layer	Common staging Area (CSA) where data from source systems are staged and is implemented as database schema.	Usually a filesystem-based area where file-based extracts (operational images) from source systems are staged prior to loading into a target schema.

Component	FSDF	Traditional BI Architecture
3rd Normal Form Operational Data Store/Enterprise Data Warehouse	Does not provide a physicalized model for a 3NF store. The FS LDM can be physicalized as a 3NF store if desired. Operational/fine-grained reporting will be fulfilled from the reporting area.	Notification sent to all users captured as owners and user who submitted the definition.
Data Marts/Reporting Model	Set of star schemas with conformed dimensions (Ralph Kimball approach)	Set of star schemas

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

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

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

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

## Staging Data Model

### Overview/Design

The Common Staging Area Model (CSA) represents the point of entry of data into the OFSDF. The CSA provides a simplified, unified data sourcing area for inputs required by analytical applications and engines. It consists of over 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 OFSDF is to be a data repository supporting analytics, each database object in the OFSDF physical data model is necessarily mapped to a corresponding analytical use case.

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

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

**Figure 2. Download Specification**

err	Category	Description	LRM	FTI	PFT	Swapt	CRM	ROA	COA	Market Risk	Channel Analytic	Column	Exp - Credit R	Exp - Market Risk	NAAF	Market Risk	OL
ms_date	DATE	This is the date on which the codes are extracted from the source systems.	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
rejection_reason_desc	VARCHAR2(50)	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
rejection_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
application_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
ms_date_application_status_desc	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
application_status_desc	VARCHAR2(50)	This column stores the description of application status codes	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
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
ms_date_application_type_desc	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
application_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
trn_rea_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
ms_date_trn_rea_desc	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
trn_rea_desc	VARCHAR2(50)	This column holds the description of reasons	NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
trn_type	CHAR(1)		NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
ms_date_trn_type_desc	DATE		NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
trn_type_desc	VARCHAR2(50)		NO	NO	NO	NO	Yes	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
authorization_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
ms_date_authorization_decision_reason_desc	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
authorization_decision_reason_desc	VARCHAR2(50)	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
sec_base_of_exp	VARCHAR2(20)	Stores the bank role for the Securitization Exposure. 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
ms_date_sec_base_of_exp_desc	DATE		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	Yes	NO	NO	NO	NO

The Mappings can be generated from the OFSDF ERwin file using ERwin'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 OFSDF directly, or alternatively into a pre-defined file-based operational image area from which it is subsequently loaded into the OFSDF schema. In a large bank, it is not unusual to have 10s to 100s of millions of accounts and transactions.

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

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

3. **Application-managed Referential Integrity (RI):** In conjunction with the database design of the staging schema, a key feature is the management of Referential Integrity primarily in the application tier, rather than within the database. Rather than imposing foreign key relationships in the database, which could cause complex loading order dependencies, relationships between staging tables are managed by the Data Quality (DQ) framework, a toolkit within the Analytical Application Infrastructure that captures the relevant data quality checks for a specific table in the staging model. These checks include:

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

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

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

## Details of the Staging Data Model

The CSA model consists of over 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:

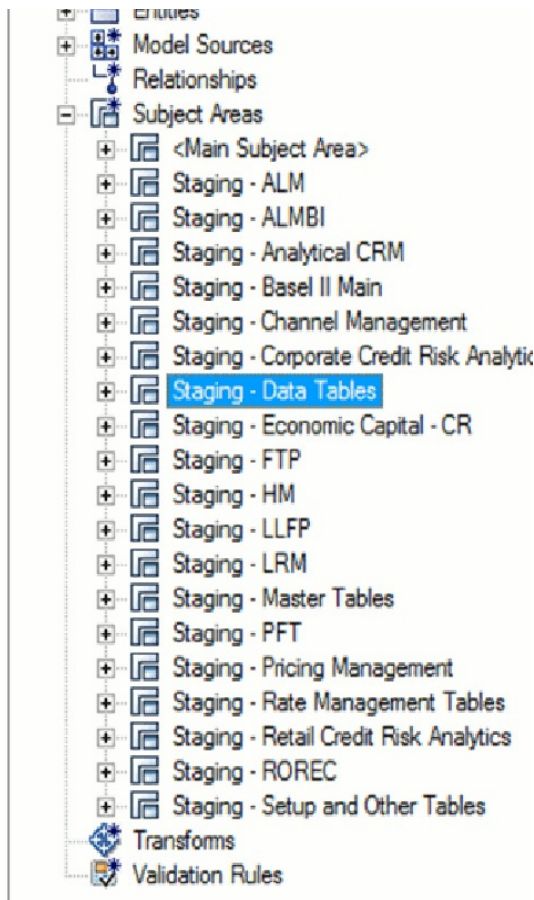
1. **Business Data:** This set of tables captures the actual business events and the resulting state of a bank from those business events. The OLTP systems (or Transactional Systems) capture this information resulting from the execution of the bank's different business processes. Broadly, this information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

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

**Business Data:**

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

**Figure 3. 'Staging-Data Tables' Subject Area**



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

**Figure 4. Business Data Tables**



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

**Table 4. Key Groups**

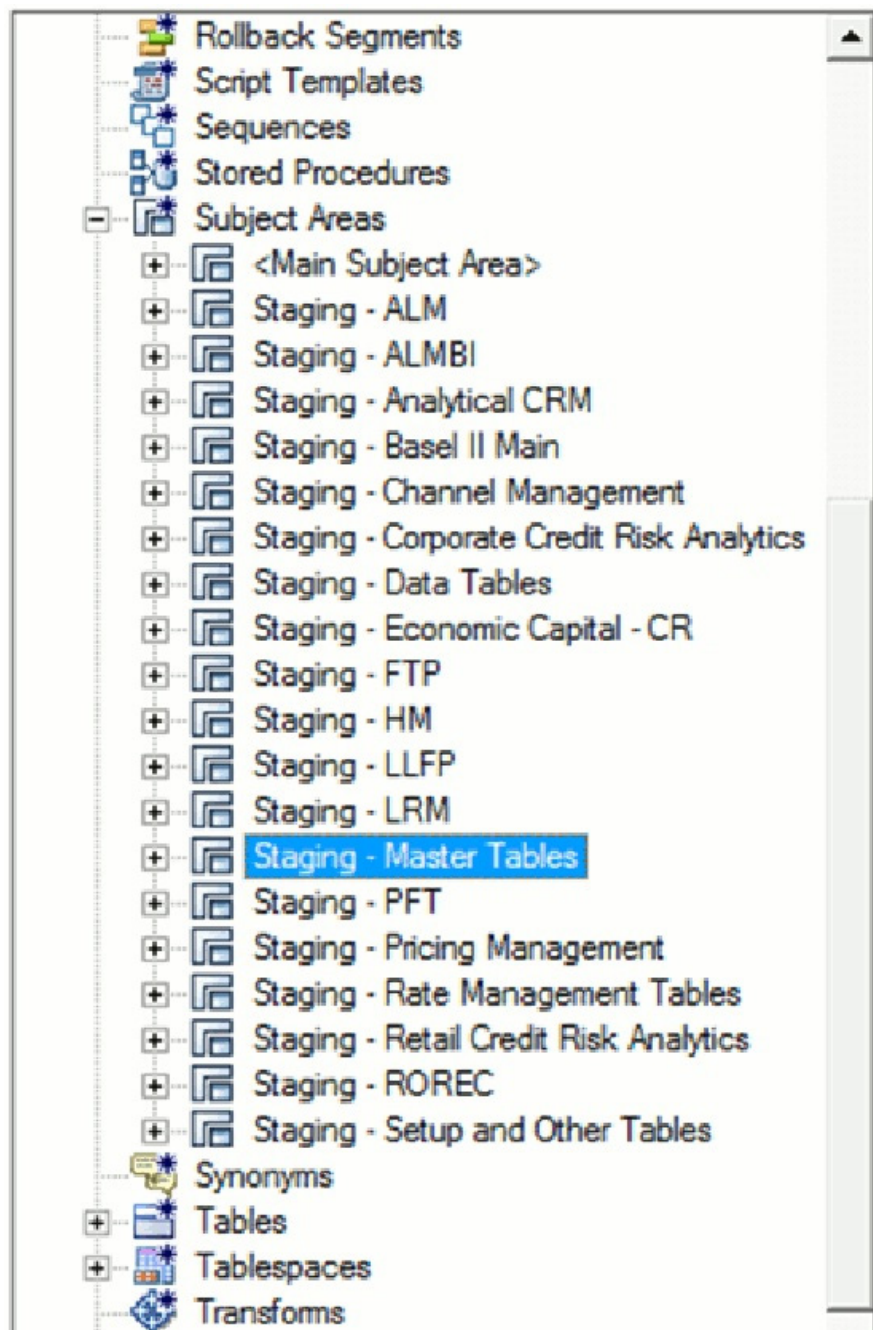
Group Name	Purpose
Product Processors	Tables for Financial Instruments and Contracts. Note that these tables can accommodate nearly 80+ types of instruments and derivatives across Banking and Trading books.
Transactions	Tables to hold Transaction/Event level data from the bank's systems. These tables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

**Reference/Master Data:**

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

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

**Table 5. Staging Master Tables**



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

## Reporting Data Model

### Overview

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

---

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

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

**1. Design:**

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

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

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

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

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

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

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

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

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

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

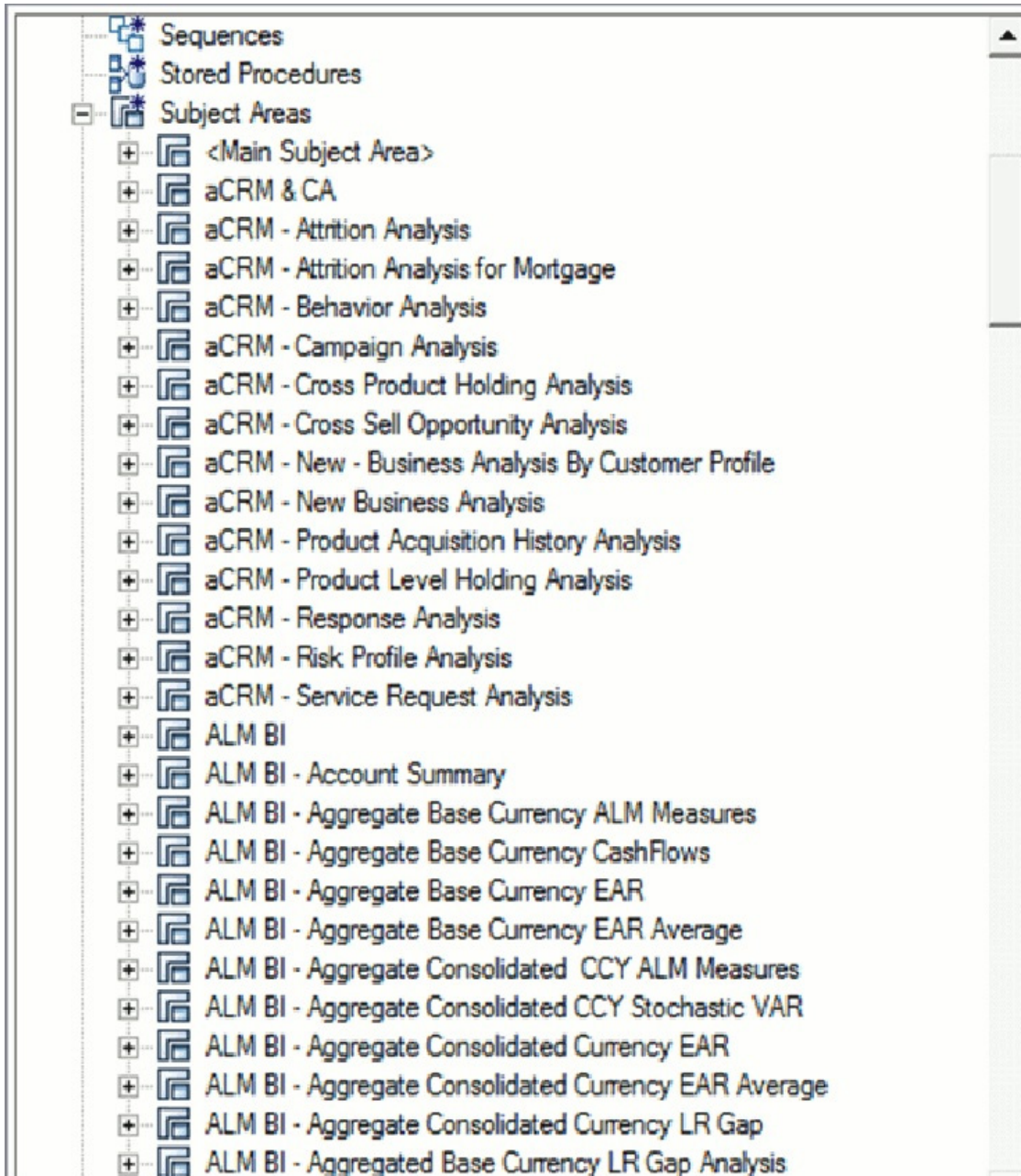
## Reporting Area Details

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

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

**Figure 5. Subject Areas**





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

**Table 6. List of Solution Supported by OFSDF**

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

**Table 6. List of Solution Supported by OFSDF**

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

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

## OFSDF Logical Data Model

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

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

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

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

## Relationship to the OFSDF Physical Model

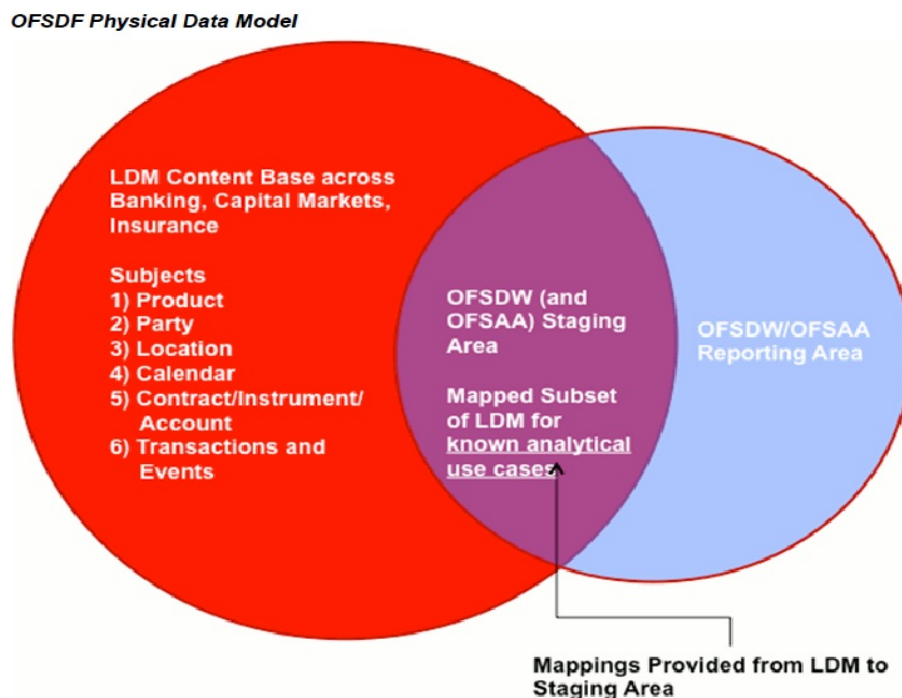
The coverage of the OFS LDM in terms of core business processes is as follows.

**Table 8. Coverage of OFS LDM**

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

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

**Figure 6. OFSDF Ohysical Data Model**



## OFSDF LDM Content Details

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

**Table 9. Subject Areas and their definitions**

Subject Area Name	Subject Area Definition
Accounting	This subject area deals with the structure of the internal accounting of a financial institution, which includes budget , GL & GL mapping to budgets.
Application	This subject area covers various aspects of application across different products. The section helps to analyze potentially risky customers and performance of different sourcing channels.
Calendar	This subject area deals with entities that contain the details for Calendar & Fiscal Date, Month, Quarter, Half year & Year.
Campaign	This subject area provides the information on the campaigns, which are organized by the financial institutions. It covers the details of the campaign treatment, offers given on a campaign, response & delivery status and so on, which helps in analyzing the profitability of a campaign.
Card	This subject area deals with entities that contains the details of the different types of media (including credit card), which are offered by the financial institution. It covers the details of the credit card reward points, its billing schedule & the payment transaction flow. It includes debit cards, credit cards, prepaid cards, and so on.
Channel	This subject area caters to all the variety of channels, which are involved in the functioning of financial institution in terms of advertisement or collecting response or sending message, or so on.
Claim	This subject area covers the details of claims originated from insurance contracts. It also captures the various claim events, the settlement history & the claim recovery process.
Collateral	This subject area records the details of various physical & financial collateral and also helps in analyzing the quality of the each collateral.
Collections and Recovery	This section captures performance of collection & recovery activities. The section helps ascertain the amount and quality of the collections and the recovery that was made using the secondary means of collecting money (by selling collaterals, and so on).
Common	This subject area covers the entities that are common across subject areas like currency, data origin, and so on.
Contract	This subject area captures contract & account details of various banking & insurance products, which includes Loans, Cards, CASA, TD, Collection, Insurance, Merchant Card, Overdraft, Commitment, Prepaid Cards, Reinsurance, and so on.

**Table 9. Subject Areas and their definitions**

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

Within the above subject areas, the FS LDM has 800+ entities. The various naming conventions used in OFSDF Logical Data Model are explained in Appendix A. For column-level details, see the Technical

---

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

## Mapping from Logical Data Model to OFSDF Staging Area

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

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

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





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

This chapter includes the following topics:

- Dimension Table Population
- Overview of SCD Process
- Prerequisites
- Tables Used by the SCD Component
- Executing the SCD Component
- Checking the Execution Status
- Load DIM\_ACCOUNT through SCD

## Dimension Table 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

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

Example:

In this example, N\_PRODUCT\_SKEY is the surrogate key column which is a unique key for each

**Table 10. Type 1 SCD Methodology Example**

N_PRODUCT_SKEY	V_PRODUCT_NAME	D_START_DATE	D_END_DATE	F_LATEST_RECORD_INDICATOR
1	PL	5/31/2010	12/31/9999	Y

record in the dimension table. V\_PRODUCT\_NAME is the product name. D\_START\_DATE indicates the date from which this product record is valid.

D\_END\_DATE indicates the date till which this product record is valid.

F\_LATEST\_RECORD\_INDICATOR with value 'Y', which indicates this is the latest record in the dimension table for this product and 'N' indicates it is not. If the V\_PRODUCT\_NAME column is set as a Type 1 SCD

column and if there is a change in the product name to 'Personal Loan' from 'PL' in the above example, in the next processing period, then when SCD is executed for the new processing period the record in the above example changes to:

**Table 11. Record Change Example**

N_PRODUC_T_ SK EY	V_PRODUCT _ NAME	D_START_D A TE	D_END_DAT E	F_LATEST_RE COR D_INDICATOR
1	Personal Loan	6/30/2010	12/31/9999	Y

### Type 2 SCDs - Creating another dimension record

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

**Table 12. Type 2 SCD Methodology Example**

N_PRODUC T_ SK EY	V_PRODUCT _ NAME	D_START_D A TE	D_END_DAT E	F_LATEST_RE COR D_INDICATOR
1	PL	5/31/2010	12/31/9999	N
1	Personal Loan	6/30/2010	12/31/9999	Y

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

### Type 3 SCDs - Creating a current value field

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

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

For more information on SCDs, see:

- *Oracle Data Integrator Best Practices for a Data Warehouse* at <http://www.oracle.com/technetwork/middleware/data-integrator/overview/odi-best-practices-datawarehouse-whi-129686.pdf>
- *Oracle® Warehouse Builder Data Modeling, ETL, and Data Quality Guide* at [http://docs.oracle.com/cd/E11882\\_01/owb.112/e10935.pdf](http://docs.oracle.com/cd/E11882_01/owb.112/e10935.pdf)  
[[http://docs.oracle.com/cd/E14072\\_01/owb.112/e10935.pdf](http://docs.oracle.com/cd/E14072_01/owb.112/e10935.pdf)]

---

Additional online sources include:

- [http://en.wikipedia.org/wiki/Slowly\\_changing\\_dimension](http://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.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 file [SCD](#).

## Prerequisites

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

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

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

## Tables Used by the SCD Component

The database tables used by the SCD component are:

- SYS\_TBL\_MASTER

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

**Table 13. SYS\_Table\_Master**

Column Name	Data Type	Column Description
MAP_REF_NUM	NUMBER(3) NOT NULL	The Mapping Reference Number for this unique mapping of a Source to a Dimension Table.
TBL_NM	VARCHAR2(30) NOT NULL	Dimension Table Name
STG_TBL_NM	VARCHAR2(30) NOT NULL	Staging Table Name

**Table 13. SYS\_Table\_Master**

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

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

**Table 14. Sample Data**

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

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

- SYS\_STG\_JOIN\_MASTER

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

**Table 15. Table for Seeded Dimension**

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

**Table 15. Table for Seeded Dimension**

Column Name	Data Type	Column Description
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_R EQD_FLG	CHAR(1) NULL	Column to determine whether Lookup is required for Priority of Source against the Source Key Column or not.
COL_DATATYPE	VARCHAR2(15) NULL	The list of possible values are VARCHAR, DATE, NUMBER based on the underlying column datatype.
COL_FORMAT	VARCHAR2(15) NULL	

The possible values for column type (the COL\_TYPE column) in SYS\_STG\_JOIN\_MASTER are:

1. PK – Primary Dimension Value (may be multiple for a given "Mapping Reference Number")
2. SK – Surrogate Key
3. DA – Dimensional Attribute (may be multiple for a given "Mapping Reference Number")
4. SD – Start Date
5. ED – End Date
6. LRI – Latest Record Indicator (Current Flag)
7. CSK – Current Surrogate Key
8. PSK – Previous Surrogate Key
9. SS – Source Key
10. LUD – Last Updated Date / Time
11. LUB – Last Updated By
12. Sample Data: This is the row put in by the solution installer for the Line of Business dimension.

**Table 16. Sample Data**

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

**Table 16. Sample Data**

MAP_REF_NUM	6
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 99999999999999999999999999999999  
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/<infodomain 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 key. Below are the columns that will be loaded during SCD population:

- V\_ACCOUNT\_NUMBER
- N\_ACCT\_SKEY
- N\_RCV\_LEG\_ACCT\_SKEY
- FIC\_MIS\_DATE

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 33 tasks under it.

These 33 tasks represent the 33 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.

## LOAD DIM TABLES THROUGH SCD

Batch <INFODOM>\_SCD has been introduced with 129 tasks under it. These 129 tasks represent the 129 SCD processes where different staging tables would be the source and Dimension Tables would be the targets. The required SCDs have been introduced into SYS\_TBL\_MASTER table, and subsequently into SYS\_STG\_JOIN\_MASTER.

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

## 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 higher. If these have to be used in OFSAAI versions 7.3.2.2.0 or 7.3.2.3.0 and higher, execute the following SQL queries:

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

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

- For improving performance, hints for the MERGE query which is generated internally by the SCD can be provided under MERGE\_HINT. Session alters could be mentioned under SESSION\_ENABLE\_STATEMENT and SESSION\_DISABLE\_STATEMENT columns.
- SESSION\_ENABLE\_STATEMENTs will be executed before the MERGE in the SCD and SESSION\_DISABLE\_STATEMENTs will be executed after the SCD MERGE.
- Since all the tasks under the SCD batch for DIM\_ACCOUNT works on the same target, the SESSION\_DISABLE\_STATEMENTs in SYS\_TBL\_MASTER cannot be provided when tasks are executed. In this case, there can be a separate SQL file to contain all the SESSION\_DISABLE\_STATEMENTs to be executed once after all the tasks in the SCD are done. The SESSION\_DISABLE\_STATEMENT will hold a null in SYS\_TBL\_MASTER table.
- SESSION\_ENABLE\_STATEMENTs are required to be mentioned only for the first task in the batch. Here the target is the same for all the tasks under a batch. In case any of the tasks are to be executed separately, then the SESSION\_ENABLE\_STATEMENTs should be mentioned for any one of the tasks which is included in the batch for the execution.

### Example

MERGE\_HINT and SESSION\_ENABLE\_STATEMENT in SYS\_TBL\_MASTER

**Table 17. Merge Hint and Session Enable Statement**

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

- 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:

**Table 18. Configuration**

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'.



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

This chapter includes the following topics:

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

## **Overview of Common Customer Summary Tables**

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

## **Prerequisites**

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

- DIM\_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\_SUMMARY 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>\_STG\_TO\_FCCS.

Following steps will help you to execute the batch:

1. Go to the *Batch Execution* screen.

- 
2. Select the seeded batch <INFODOM>\_STG\_TO\_FCCS 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.

# *Loading Multiple Load Runs in OFSAA*

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

This chapter includes the following topics:

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

## **Overview**

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

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

## **Objective**

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

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

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

---

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

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

To execute this run:

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

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

## Design Details

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

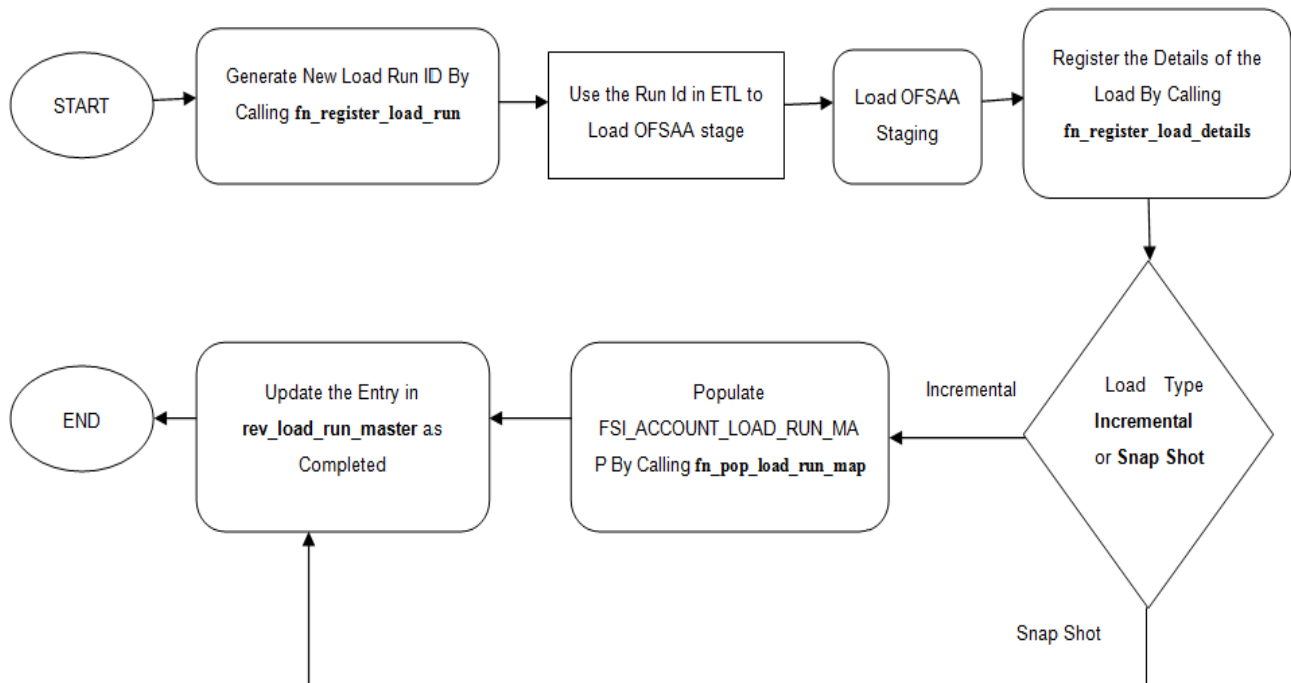
- ETL Tool
- OFSAA F2T
- OFSAA T2T
- OFSAA Excel upload
- OFSAA DIH

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

## Implementation

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





**Note:** Column n\_load\_run\_id should always be populated only by the value returned by fn\_register\_load\_run.

**Function** - Register Load Run (fn\_register\_load\_run)

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

Parameters	Source of Values	Example Values
Batch ID	Auto generated if you are using OFSAA Framework	OFSBFNDINFO_20150101_1
MIS-Date	Input from Customer	01/01/2015
Load Run Name	Input from Customer	Daily EOD Load
Load Run Purpose	Input from Customer	BA/BS (Basel Advanced Approach, Basel Standard)
Load Run Type	Input from Customer	B - Base, A - Adjustments, P- Backdated Adjustments

**Example:**

Declare

```
Result number;
```

---

Begin

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

The function registers the request in the table name rev\_load\_run\_master and marks load as "In progress". You can use columns LOAD\_RUN\_NAME and LOAD\_PURPOSE as per the requirement.

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

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

LOAD_RUN_ID	MIS_DATE	LOAD_TYPE	LOAD_PURPOSE	START_DT_TIME	LOAD_RUN_NAME	BATCH_ID	LOAD_RUN_STATUS
1	01-JAN-15	A	BA	01-JAN-15	FSDF_Load	OFSBFNDINFO_20150101_1	In Progress

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

## Loading OFSAA Staging Tables

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

Load strategy at a customer site falls across two categories:

- Complete Snapshot
- Incremental Snapshot

### Complete Snapshot Load Example

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

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

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

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

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

STG\_LOAN\_CONTRACTS after load will appear as below:

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

REV\_LOAD\_RUN\_MASTER after second function call will appear as below:

LOAD_RUN_ID	MIS_DATE	LOAD_TYPE	LOAD_PURPOSE	START_DT TIME	LOAD_RUN_NAME	BATCH_ID	LOAD_RUN_STATUS
1	01-JAN-15	B	BA	01-JAN-15 13:00 PM	FSDF_Load	OFSBFNDI NFO_201501 01_1	Complete
2	01-JAN-15	B	BA	01-JAN-15 23:00 PM	Loan Corrections	OFSBFNDI NFO_201501 01_2	In Progress

## Incremental Snapshot Load Example

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

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

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

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

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

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

**Function** - Populate Load Run Map

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

Parameters	Source of Values	Example Values
Batch ID	Auto generated if you are using OFSAA Framework	OFSBFNDINFO_20150101_1
MIS-Date	Input from Customer	01/01/2015
Stage Table Name	Input from Customer	STG_CASA
Load Run ID	Input from Customer	1
Load Run Name,	Input from Customer	FSDF_Load

### Example

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

END ;

**Note :** For troubleshooting any errors while making the function calls , refer to fsi\_message\_log table for more details.

For the example mentioned above, records in FSI\_ACCOUNT\_LOAD\_RUN\_MAP table will appear as below:

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

**Note:** If the Load Type is Snap shot, the FSI\_ACCOUNT\_LOAD\_RUN\_MAP need not be populated.

## Post Stage Load Process

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

### Register Load Run Details

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

**Function** - Register Load Run Details

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

Parameters	Source of Values	Example Values
Batch ID	Auto generated if you are using OFSAA Framework	OFSBFNDINFO_20150101_1
MIS-Date	Input from Customer	01/01/2015
Stage Table Name	Input from Customer	STG_CASA
Load_Run_Id	Input from Customer	1
Load Run Name,	Input from Customer	FSDF_Load
Load Type	Input from Customer	S - Full Snap Shot I - Incremental

---

### Example:

Declare

```
Result number;
```

Begin

```
Result: =
```

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

```
END;
```

This function populates a table named rev\_load\_run\_details. The columns load type can have only 2 values, such as:

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

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

### Updating Load as Completed

Once you complete these steps, update the status of the record inside rev\_load\_run\_master as Completed.

```
UPDATE rev_load_run_master
```

```
SET LOAD_RUN_STATUS = 'Completed'
```

```
WHERE pMis_Date = '01-Jan-15'
```

```
AND pLoad_Run_Name = 'FSDF_LOAD' and LOAD_RUN_ID = 1 ;
```

### Loading data into OFSAA Results Tables from Staging Tables

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

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

#### Complete Snapshot Load Scenario

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

For example: FCAS T2T for Stage Loan Contracts

- "Modify Filter Condition inside T2T:

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

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

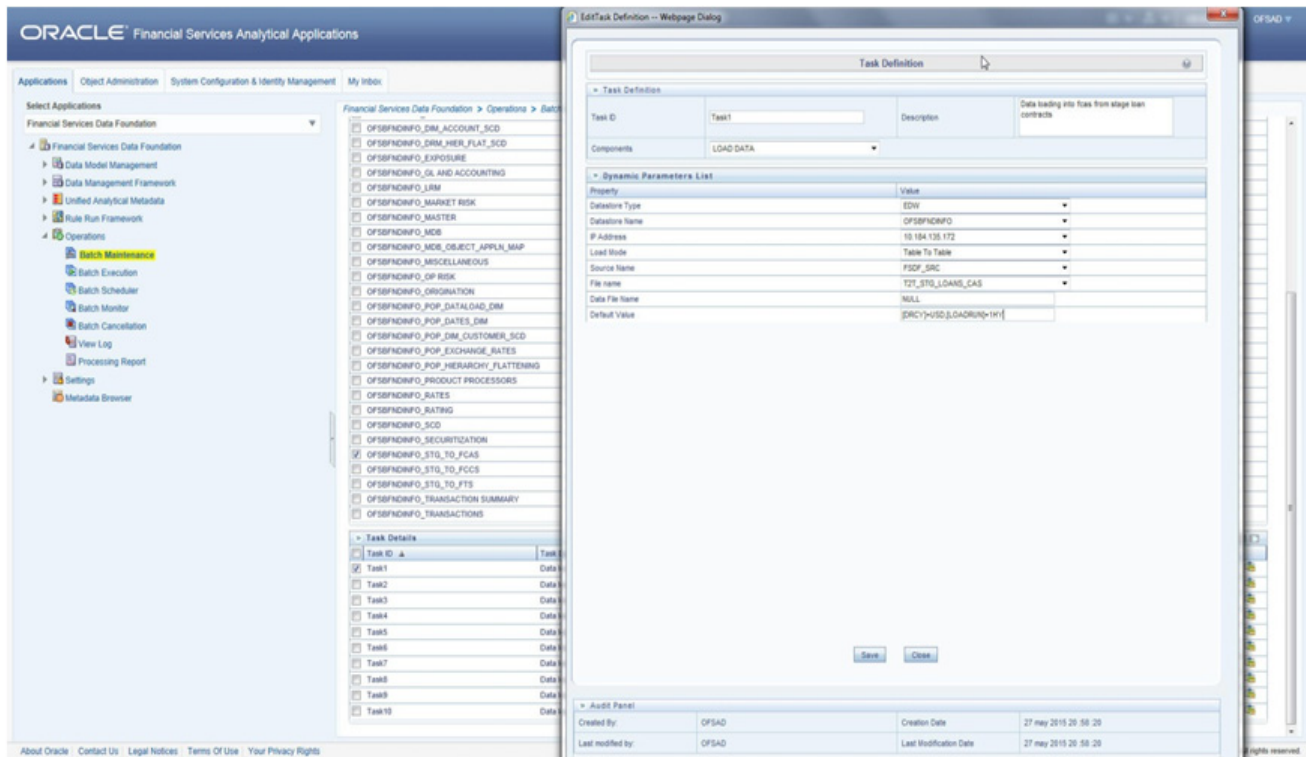
- "Modify the Corresponding Batch Task Each Time with Load Run Idxxx

1. Select Batch, Task (T2T\_STG\_LOANS\_CAS)

2. Click Edit

3. Add Highlighted Condition in Default Value and Save (Each Time we need to provide the Load Run ID which are supposed to use. Here in the below example we are using 1)

[DRCY] =USD, [LOADRUN] =1



Execute the batch after this change; it will load the Result table with the given Load Run Id Records

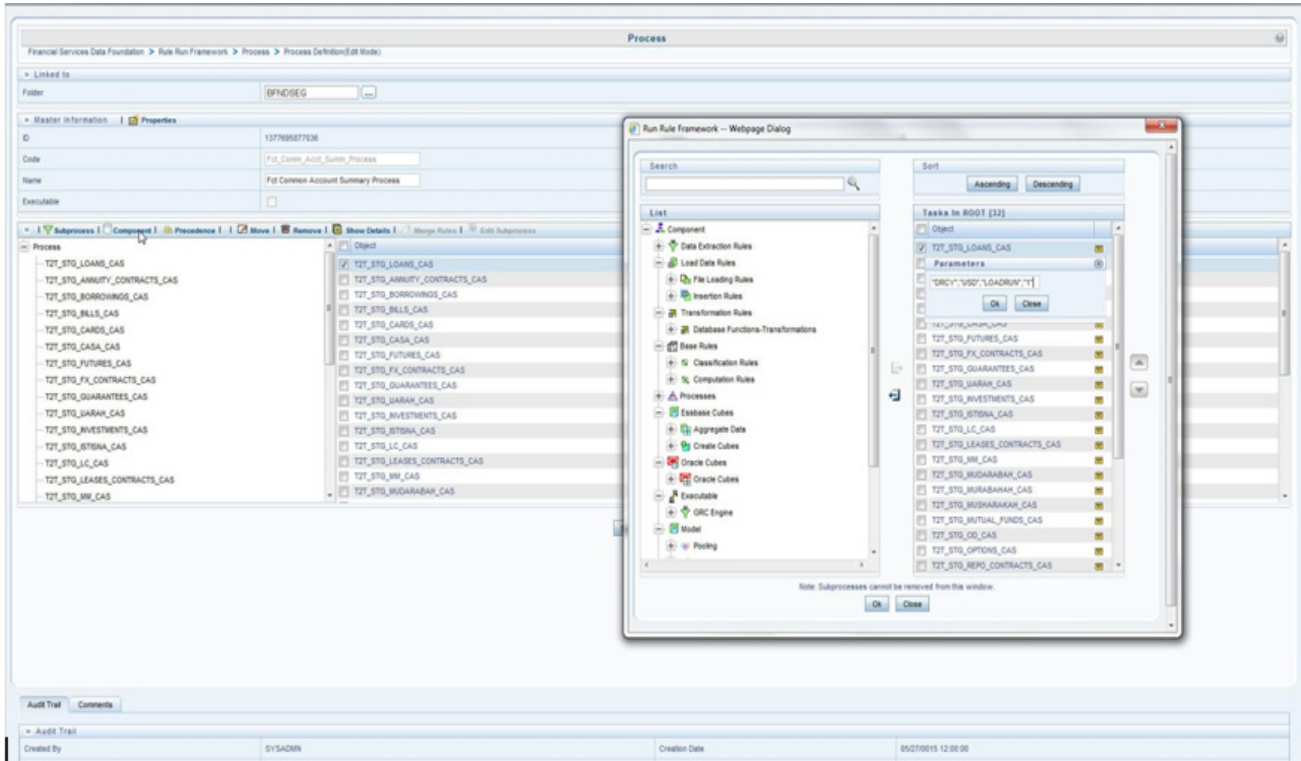
- "Modify the Corresponding Process Task of a Run Each Time with Load Run Id

1. Select Process, Click Edit

2. Choose the Components, Select the Object (T2T\_STG\_LOANS\_CAS) and Click Components

3. Click the Yellow Drop Button near the T2T

4. Add the Highlighted Condition with the text, and Save : "DRCY","USD","LOADRUN","1"



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

### Incremental Load Scenario

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

For example, FCAS T2T for Stage CASA

- "Modify Join Condition inside T2T:

Join Condition to be added in all relevant T2T

For example, STG\_CASA T2T join

```
INNER JOIN FSI_ACCOUNT_LOAD_RUN_MAP
```

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

```
AND FSI_ACCOUNT_LOAD_RUN_MAP.N_LOAD_RUN_ID = STG_CASA.N_LOAD_RUN_ID  
AND FSI_ACCOUNT_LOAD_RUN_MAP.FIC_MIS_DATE = STG_CASA.FIC_MIS_DATE  
AND FSI_ACCOUNT_LOAD_RUN_MAP.V_GAAP_CODE = STG_CASA.V_GAAP_CODE
```



---

AND FSI\_ACCOUNT\_LOAD\_RUN\_MAP.F\_LATEST\_LOAD\_RUN\_FLAG = 'Y'

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



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

This chapter includes the following topics:

- Overview of Account Summary Tables
- Overview of Account Summary Population
- Executing the Account Summary Population T2T
- Checking the Execution Status
- Account Summary T2Ts
- Handling Alternate Currency
- Execution of Currency Exchange Rates Population T2T

## **Overview of Account Summary Tables**

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

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

- FCT\_CRM\_ACCOUNT\_SUMMARY
- FCT\_PFT\_ACCOUNT\_SUMMARY
- FCT\_FTP\_ACCOUNT\_SUMMARY
- FCT\_REG\_CAP\_ACCOUNT\_SUMMARY
- FCT\_ECO\_CAPITAL\_ACCOUNT\_SUMMARY

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

## **Overview of Account Summary Population**

Table to Table seeded definitions are provided for loading data into Common Account Summar

**Table 19. Table to Table Seeded Definitions**

Sl. No.	Source Table Name	T2T Definition Name	Target Table Name
	STG_ANNUIITY_CONTRACTS	T2T_STG_ANNUIITY_CONTRACTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_BILLS_CONTRACTS	T2T_STG_BILLS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_BORROWINGS	T2T_STG_BORROWINGS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_CARDS	T2T_STG_CARDS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_CASA	T2T_STG_CASA_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_CREDIT_DERIVATIVES	T2T_STG_CREDIT_DERIVATIVES_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_FUTURES	T2T_STG_FUTURES_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_FX_CONTRACTS	T2T_STG_FX_CONTRACTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_FX_CONTRACTS	T2T_STG_FX_CONTRACTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_GUARANTEES	T2T_STG_GUARANTEES_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_IJARAH	T2T_STG_IJARAH_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_INVESTMENTS	T2T_STG_INVESTMENTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY

**Table 19. Table to Table Seeded Definitions**

Sl. No.	Source Table Name	T2T Definition Name	Target Table Name
	STG_ISTISNA	T2T_STG_ISTISNA_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_LC_CONTRACTS	T2T_STG_LC_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_LEASES_CONTRACTS	T2T_STG_LEASES_CONTRACTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_LOAN_CONTRACTS	T2T_STG_LOANS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_MM_CONTRACTS	T2T_STG_MM_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_MUDARABAH	T2T_STG_MUDARABAH_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_MURABAHAH	T2T_STG_MURABAHAH_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_MUSHARAKAH	T2T_STG_MUSHARAKAH_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_MUTUAL_FUNDS	T2T_STG_MUTUAL_FUNDS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_OD_ACCOUNTS	T2T_STG_OD_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_OPTION_CONTRACTS	T2T_STG_OPTIONS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_REPO_CONTRACTS	T2T_STG_REPO_CONTRACTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_RETIREMENT_ACCOUNTS	T2T_STG_RETIREMENT_ACCOUNTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_SALAM	T2T_STG_SALAM_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_SUKUK	T2T_STG_SUKUK_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_SWAPS_CONTRACTS	T2T_STG_TD_CONTRACTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY

**Table 19. Table to Table Seeded Definitions**

Sl. No.	Source Table Name	T2T Definition Name	Target Table Name
	STG_TRUSTS	T2T_STG_TRUSTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_COMMITMENT_CONTRACTS	T2T_STG_COMMITMENT_CONTRACTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_BORROWING_COMMITMENTS	T2T_STG_BORROWING_COMMITMENTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
	STG_CORRESPONDENT_ACCOUNT	T2T_STG_CORRESPONDENT_ACCOUNT_CAS	FCT_COMMON_ACCOUNT_SUMMARY

## Executing the Account Summary Population T2T

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

### Fact Common Account Summary - Batch Execution

A seeded batch, <Infodom>\_STG\_TO\_FCAS has to be executed for the required MIS Date.

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

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

---

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

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

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

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

## **Fact Common Account Summary - Run Execution**

A seeded batch, <Infodom>\_STG\_TO\_FCAS has to be executed for the required MIS Date.

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

Following steps will help you create a new batch:

Procedure:

1. From the Home menu, click Rules Framework and select Process.
2. Select Fct\_Comm\_Acct\_Summ\_Process Process and select Edit.
3. Click on Component if you wish to change the Parameter.
4. Select the required T2T and click on the Yellow Drop Down Button. Repeat as many as T2Ts need to be changed.
5. Update the Parameter Value and click OK. For example, default value is "DRCY","USD". Here 'USD' acts as reporting currency parameters to T2T.
6. Save the Process, Click Save
7. Click NO in the Dialogue Box as we need to save it as the same version.
8. Close the Dialogue Box.
9. Open the RUN menu from Rules Frame Work.
10. Select Comm Acc Summ Load Run and click on Fire Run button.
11. Choose the Create & Execute Option from Batch Menu and Choose the required MIS Date.
12. Click on OK to execute the RUN.
13. It will show the Batch execution is in Progress Dialogue. Close the Dialogue Box.
14. Click on Batch Progressing Report from Operations Menu. And note down the Batch Run id.
15. Click on Batch Monitor from Operations Menu.
16. Type the noted Run Id in Batch Id Like Column and Click on Search Button.
17. Select the respective batch, Select Information Date and Batch Run ID from Batch Details tab and click on Start Monitoring Button.
18. Check the Status of Each Task in the Run.

## **Checking the Execution Status**

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

---

**Note:** For a more comprehensive coverage of configuration and execution of a batch, refer to *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 execution log can be accessed on the application server in the following directory \$FIC\_DB\_HOME/log/t2t. The file name will have the batch execution id.

The file name will have the batch execution id.

• FCT\_COMMON\_ACCOUNT\_SUMMARY\$

## Account Summary T2Ts

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

## Handling Alternate Currency

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

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

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

## Execution of Currency Exchange Rates Population T2T

The batch <INFODOM>\_POP\_EXCHANGE\_RATES needs to be executed to populate fsi\_exchange\_rates as the entries in setup\_master are seeded during installation.

**Note:** FSI\_EXCHANGE\_RATES table has to be loaded prior loading any of the other Account Summary tables.

- **Metadata Browser**
- **Common Account Summary**

## Currency Execution Rates - Batch Execution

A seeded batch, <Infodom>\_POP\_EXCHANGE\_RATES has to be executed for the required MIS Date.

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

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



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

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

9. Execute the batch created in the preceding steps.



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

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

This chapter includes the following topics:

- Table to Table
- Executing the Fact Transaction Summary

### Table to Table

Table to Table seeded definitions are provided for loading data into Common Account Summary.

**Table 20.**

SI No.	Source Table Name	T2T Definition Name	Target Table Name
1	STG_TRUSTS_TXNS	STG_TRUSTS_TXNS_FTS	FCT_TRANSACTION_SUMMARY
2	STG_ANNUITY_TXNS	STG_ANNUITY_TXN_FTS	FCT_TRANSACTION_SUMMARY
3	STG_BILL_CONTRACTS_TXNS	STG_BILL_CONTRACTS_TXNS_FTS	FCT_TRANSACTION_SUMMARY
4	STG_BORROWING_COMMITMENT_TXNS	STG_BORROWING_COMMITMENT_TXNS_FTS	FCT_TRANSACTION_SUMMARY
5	STG_BORROWINGS_TXNS	STG_BORROWINGS_TXNS_FTS	FCT_TRANSACTION_SUMMARY
6	STG_CARDS_PAYMENT_TXNS	STG_CARDS_PAYMENT_TXNS_FTS	FCT_TRANSACTION_SUMMARY
7	STG_CARDS_SETTLEMENT_TXNS	STG_CARDS_SETTLEMENT_TXNS_FTS	FCT_TRANSACTION_SUMMARY
8	STG_CASA_TXNS	STG_CASA_TXNS_FTS	FCT_TRANSACTION_SUMMARY
9	STG_COMMITMENT_CONTRACT_TXNS	STG_COMMITMENT_CONTRACT_TXNS_FTS	FCT_TRANSACTION_SUMMARY
10	STG_COMMODITIES_TXNS	STG_COMMODITIES_TXNS_FTS	FCT_TRANSACTION_SUMMARY

**Table 20.**

SI No.	Source Table Name	T2T Definition Name	Target Table Name
11	STG_CORRESPONDENT_ACCT_TXNS	STG_CORRESPONDENT_ACCT_TXNS_FTS	FCT_TRANSACTION_SUMMARY
12	STG_CREDIT_DERIVATIVES_TXNS	STG_CREDIT_DERIVATIVES_TXNS_FTS	FCT_TRANSACTION_SUMMARY
13	STG_FOREX_TXNS_FTS	STG_FOREX_TXNS_FTS	FCT_TRANSACTION_SUMMARY
14	STG_GUARANTEES_TXNS	STG_GUARANTEES_TXNS_FTS	FCT_TRANSACTION_SUMMARY
15	STG_IJARAH_TXNS	STG_IJARAH_TXNS_FTS	FCT_TRANSACTION_SUMMARY
16	STG_INTERBANK_TXNS	STG_INTERBANK_TXNS_FTS	FCT_TRANSACTION_SUMMARY
17	STG_INVESTMENT_TXNS	STG_INVESTMENT_TXNS_FTS	FCT_TRANSACTION_SUMMARY
18	STG_ISTISNA_TXNS	STG_ISTISNA_TXNS_FTS	FCT_TRANSACTION_SUMMARY
19	STG_LC_TXNS	STG_LC_TXNS_FTS	FCT_TRANSACTION_SUMMARY
20	STG_LEASES_TXNS	STG_LEASES_TXNS_FTS	FCT_TRANSACTION_SUMMARY
21	STG_LOAN_CONTRACT_TXNS	STG_LOAN_CONTRACT_TXNS_FTS	FCT_TRANSACTION_SUMMARY
22	STG_MERCHANT_CARDS_TXNS	STG_MERCHANT_CARDS_TXNS_FTS	FCT_TRANSACTION_SUMMARY
23	STG_MM_TXNS	STG_MM_TXNS_FTS	FCT_TRANSACTION_SUMMARY
24	STG_MURABAHAH_TXNS	STG_MURABAHAH_TXNS_FTS	FCT_TRANSACTION_SUMMARY
25	STG_MUSHARAKAH_TXNS	STG_MUSHARAKAH_TXNS_FTS	FCT_TRANSACTION_SUMMARY
26	STG_MUTUAL_FUNDS_TXNS	STG_MUTUAL_FUNDS_TXNS_FTS	FCT_TRANSACTION_SUMMARY
27	STG_OD_ACCOUNTS_TXNS	STG_OD_ACCOUNTS_TXNS_FTS	FCT_TRANSACTION_SUMMARY
28	STG_OPTION_CONTRACTS_TXNS	STG_OPTION_CONTRACTS_TXNS_FTS	FCT_TRANSACTION_SUMMARY

**Table 20.**

SI No.	Source Table Name	T2T Definition Name	Target Table Name
29	STG_RETIREMENT_A CCOUNTS_TXNS	STG_RETIREMENT_AC COUNTS_TXNS_FTS	FCT_TRANSACTION N_SUMMARY
30	STG_SALAM_TXNS	STG_SALAM_TXNS_FTS	FCT_TRANSACTION N_SUMMARY
31	STG_SUKUK_TXNS	STG_SUKUK_TXNS_FTS	FCT_TRANSACTION N_SUMMARY
32	STG_SWAP_ACCOUN T_TXNS	STG_SWAP_ACCOUNT _TXNS_FTS	FCT_TRANSACTION N_SUMMARY
33	STG_TERMDEPOSITS _TXNS	STG_TERMDEPOSITS_T XNS_FTS	FCT_TRANSACTION N_SUMMARY
34	STG_TRADING_ACC OUNT_TXNS	STG_TRADING_ACCO UNT_TXNS_FTS	FCT_TRANSACTION N_SUMMARY
35	STG_FUTURES_TXNS	STG_FUTURES_TXNS_F TS	FCT_TRANSACTION N_SUMMARY
36	STG_MUDARABAH_ TXNS	STG_MUDARABAH_TX NS_FTS	FCT_TRANSACTION N_SUMMARY

## Executing the Fact Transaction Summary

A seeded batch, Infodom\_\_STG\_TO\_FTS has to be executed for the required MIS Date. Alternatively, following steps will help you create a new batch:

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

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

---

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

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

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

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

This chapter includes the following topics:

- Overview
- Object View
- Metadata Publish
- Metadata Object to Application Map

### **Overview**

Metadata Browser (MDB) with an Object and an Application view, provides a common repository of metadata objects created in Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) and Oracle Financial Services Analytical Applications (OFSAA) hosted in OFSAAI. Using this view, you can identify the usage of base objects in higher level objects and the mapping of Objects to Application. It enables traceability and impact analysis to the user. It also allows users to view the data flow and the work flow of the application and understand the usage of objects within the application.

The visualization of MDB supports Application view and Object view. In Application view, you can browse through the metadata created using the Applications hosted in OFSAAI. In object view, you can view the metadata created in OFSAAI.

### **Object View**

Object view provides the detailed view of the Object. It comprises of basic details, detailed properties, dependencies, usage in higher level Object and Applications consuming the Metadata objects. Object view provides the option to navigate to its dependant or higher level usage objects. Object view enables the user to identify the usage and its dependencies across other objects.

Following are the steps to be followed to migrate the Objects to the metamodel structure:

The Object view will provide the following areas in Financial Service Data Foundation:

- Data Foundation Metadata
- Target Data Model
- Data Mapping
- Data File Mapping
- Data Transformation
- Data Quality Rules
- Data Quality Groups
- Process Metadata

- 
- Process

## **Metadata Publish**

In order to publish the relevant metadata object to the metamodel structure, a seeded batch name <INFODOM\_NAME>MDB has to be executed.

Post successful publish, all the objects can be viewed in the Metadata browser. This step will need to be done on a regular basis to ensure that the metamodel is in sync with the underlying metadata.

## **Metadata Object to Application Map**

To map all the objects to the Financial Service data foundation application, a batch name <INFODOM\_NAME>\_MDB\_OBJECT\_APPLN\_MAP has to be executed.



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

Business data commonly represents information as of a point in time (for example, a balance as of a point in time) or as of a particular span of time (for example, income for the month of March). The rollup of a particular balance depending on their nature could be a simple additive rollup wherein the child member balances are added up to arrive at the parent node balance (for example, Ending Balance) or non additive rollups wherein a node formula is used to specify how to rollup the child member balances (for example, 3 month rolling average).

This chapter includes the following topics:

- Overview of Time Dimension Population
- Prerequisites
- Tables Used by the Time Dimension Population Transformation
- Executing the Time Dimension Population Transformation
- Checking the Execution Status

## Overview of Time Dimension Population

The Time dimension population transformation is used to populate the DIM\_DATES table with values between two dates specified by the user.

The database components, used by the transformations are:

1. Database function FN\_DIM\_DATES
2. Database procedure PROC\_DIM\_DATES\_POPULATION that is called by the function FN\_DIM\_DATES mentioned earlier.

## Prerequisites

The following are the prerequisites for Time dimension population.

1. All the post install steps mentioned in the *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration guide* and the solution installation manual of *Oracle Financial Services Data Foundation* have to be completed successfully.
2. Application User must be mapped to a role that has seeded batch execution function (BATPRO).
3. Before executing a Batch, check if the following services are running on the application server:
  - Iccserver
  - Router
  - AM Server
  - Message Server
  - OLAP Server

---

For more information on how to check if the services are up and on and how to start the services if you find them not running, refer to *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

4. Batches will have to be created for executing the function. For more details, refer to section How to Define a Batch.

### Tables Used by the Time Dimension Population Transformation

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

### Executing the Time Dimension Population Transformation

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

This component for OFSDf 8.0.0.0.0 has been seeded with the Batch ID **<Infodom>\_POP\_DATES\_DIM**, which can be executed from Batch Execution section of OFSAAI. In the Parameter List, enter the Start Date and End Date. For example, '19940101','19941231'.

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

To define a new task for a Batch definition:

1. Select the check box adjacent to the newly created Batch Name in the *Batch Maintenance* window.
2. Click **Add (+)** button from the Task Details grid.

The *Task Definition* window is displayed.

3. Enter the **Task ID** and **Description**.
4. Select **Transform Data** component from the drop down list.
5. Select the following from the **Dynamic Parameters** list:
  - **Datastore Type** - Select the appropriate datastore type from the list.
  - **Datastore Name** - Select the appropriate datastore name from the list.
  - **IP address** - Select the IP address from the list.
  - **Rule Name** - Select **fn\_DimDates** from the drop down list of available transformations. (This is a seeded Data Transformation which is installed as part of the OFSDf solution installer. If you do not see this in the list, contact Oracle support)
  - **Parameter List** – Enter the Start Date and End Date.
    - **Start Date** – This is the starting date, from which the Transformation will populate DIM\_DATES table. This date should be specified in 'YYYYMMDD' format.  
For example, '20081131'.
    - **End Date** - This is the end date, to which the Transformation will populate DIM\_DATES table. This date should also be specified in 'YYYYMMDD' format.  
For example, '20091231'.

---

6. Click **Save**.

The Task definition is saved for the selected Batch.

7. Execute the batch.

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

**Function Name:** FN\_DIM\_DATES

**Parameters:** P\_BATCH\_RUN\_ID, P\_AS\_OF\_DATE, P\_ST\_DT, and P\_ED\_DT

**Sample Parameter Values:** 'Batch1', '20091231', '20081131', and '20091231'

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

### Checking the Execution Status

The Batch execution status can be monitored through *Batch Monitor* section of *OFSAAI Operations* module.

The status messages in batch monitor are:

N - Not Started

O - On Going

F - Failure

S – Success

The execution log can also be accessed on the application server in the directory *\$FIC\_DB\_HOME/log/date*, where file name will have the Batch Execution ID.

You can access the database level operations log by querying the FSI\_MESSAGE\_LOG table. Filter the Batch Run ID column for identifying the relevant log.

**Note:** Check the **.profile** file in the installation home if you are unable to find the above mentioned path.



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

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

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

This chapter includes the following topics:

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

## **Delivery Mechanism**

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

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

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

## **Installing OFSDF**

As detailed earlier, OFSDF requires the Oracle Financial Services Analytical Application Infrastructure release 8.0.0.0.0 to deploy and operate.

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

## **OFSDF Supporting Documentation**

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

The OFSDF is a detailed model, with nearly 850 entities across both the Staging and Results Area in the physical data model, with another 800+ entities in the Logical Data Model.

---

Since it is delivered as an ERwin file, all the detailed metadata for the model (Table, Column, Entity, Attribute, Relationship) definitions are embedded in the file itself. The advantage of this approach is that any site-specific customizations to OFSDF can be performed within ERwin, and the updated documentation is retained in the file in the form of additional metadata.

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

1. Data Dictionary
2. Download Specifications

For more information on Dimension Management and AMHM, refer to the *Dimension Management* chapter in Oracle Financial Services Analytical Applications Infrastructure 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 OFSDF can be extracted from the ERwin file using ERwin's reporting capability, using a pre-built set of templates for data extraction.

Instructions for how to do so are provided in a separate accompanying document that provides step-by-step instructions. See the Technical Whitepaper on *Data Model Document Generation*, which details how to extract the data dictionary from ERwin section.

## Download Specifications

As detailed in the staging area section, the mapping from the Staging Data Model to use cases, called a **download specification** provides an efficient way to manage the sourcing of data into the OFSDF staging area. This is done by mapping the staging model at a column level to use cases. This mapping information is embedded in ERwin at a column level using metadata called **User Defined Properties (UDPs)**.

The Download specifications can be extracted using pre-built templates, in a manner similar to the Data Dictionary. Instructions for how to do so are also provided in the Technical Whitepaper on *Data Model Document Generation*, which details how to extract the data dictionary from ERwin section.

## Extending OFSDF Physical Data Model

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

This section consists of the following sub-sections:

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

---

## Customization Process Guidelines

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

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

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

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

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

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

## Staging Area Design Guidelines

Following are the Staging Area Design Guidelines:

1. Ensure that the naming conventions as detailed in Appendix A section are followed.
2. Entity relationships and constraints are enforced through OFSAAI data management toolkit and are not enforced via database referential integrity checks.

The model should not be changed to enforce referential integrity checks and other data quality checks via database definitions.

3. All Staging Area tables must have a column that identifies the system from where data is sourced (source system ID).
4. The code columns in master data tables and tables that contain dimension data should be designed to hold alphanumeric values.
5. The Domain dictionary maintains the list of attribute domains. New columns must be identified with an existing domain instead of explicitly defining column data type and valid values. See guidelines in Appendix A section on the use of defined Domains.
6. Tables (e.g. reference or look up tables with static data) needed for only a specific application or use case should be a part of the application specific processing area and should not be part of the common Staging Area in OFSDF.
7. OFSDF download specifications identify the tables and columns for which data needs to be sourced for a specific analytical use case. Any new tables and/or column should have its "APPLICATION USAGE" UDP set with the appropriate application value so that the generated download specification includes the customized column and table. The master list of UDP's are maintained as a central dictionary in ERwin.
8. All columns added or modified as a part of the customization should be marked as such:

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

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

## Results Area Design Guidelines

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

Following are the Results Area Design Guidelines:

1. Ensure that the naming convention for results tables and columns detailed in Appendix A section is followed.
2. Dimensional conformance should be maintained: The same dimensional information should not be represented in different forms. In addition, dimension table design should be compatible with the slowly changing dimension process design and so should have the required columns.
3. The common accounts summary fact table.

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

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

4. The common customer summary fact table.

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

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

5. **Aggregate Entities:** Depending on performance requirements for each application, information can be reported out of aggregate entities. However, a drill through to the base entity from the aggregate entity is mandatory.
6. **Reporting and local currency support:** Include additional attributes in the fact tables to store reporting and local currency equivalent of base measures. These attributes need to be computed by looking into the exchange rates.
7. **Support for full history:** Any new tables in the Results area should be designed to support maintenance of full history.



---

---

## **Upgrading Data Model**

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



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

This chapter includes the following topics:

- Data Quality Framework
- Data Quality Rules For Staging Tables
- Data Quality Groups for Staging Tables

### **Data Quality Framework**

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

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

Data Quality Framework consists of the following sections:

- Data Quality Summary
- Data Quality Group Summary

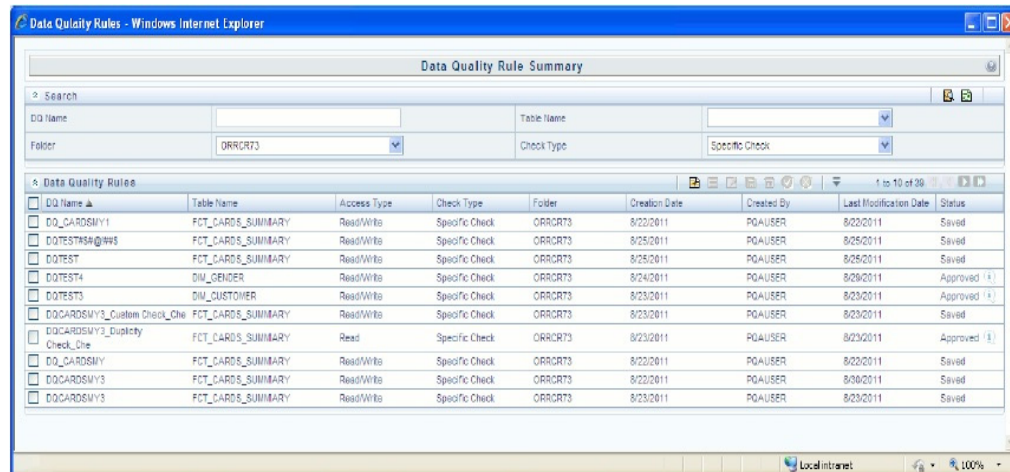
### **Data Quality Summary**

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

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

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

**Figure 7. Data Quality Rule Summary**



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

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

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

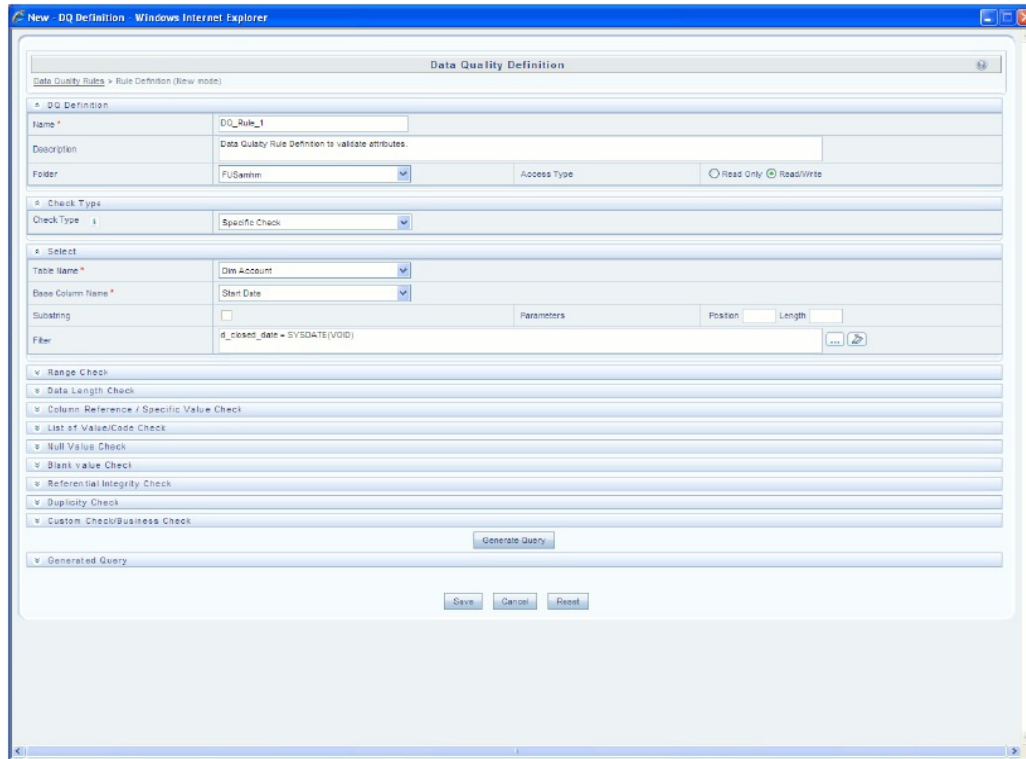
## Create Data Quality Rule

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

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

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

**Figure 8. Data Quality Definition**



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

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

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

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

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

- Select **Table Name** and **Base Column Name** from the drop down list. The list displays all the tables which are marked for Data Quality Rule in a data model, which has the table classification property code set to 340.
- (Optional) If you have selected Base Column of type Varchar/Char, select the **Substring** check box, enter numeric values in **Parameters** Position and Length fields.
- Click the below button and define the **Filter** condition using the **Specify Expression** screen.



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

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

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

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

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

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

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

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

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

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

- Specify the Assignment details as tabulated.

**Table 21. Assignment Details**

Field	Description
Column Name	Select the <b>Column Name</b> from the drop down list.
Assignment Type	Select the <b>Assignment Type</b> as one of the following: <ul style="list-style-type: none"> <li>● <b>No Assignment</b> is the default selected assignment which does not have any target column update, but the message details are pushed.</li> <li>● <b>Direct Value</b> - enter the <b>Assigned Value</b></li> <li>● <b>Another Column</b> - select the required Column as Assigned Value from the drop down list.</li> <li>● <b>Code</b> - select the required Code as Assigned Value from the drop down list if any code / leaf values exist for the selected base column.</li> </ul> <p>If not, you are alerted with a message indicating that <i>No Code values exists for the selected base column.</i></p>
Assignment Value	Select the Assignment Value from the drop-down list according to the Assignment Type selected.
Message Severity	Select the <b>Message Severity</b> as either <b>1</b> or <b>2</b> from the drop down list.
Message	Select the required <b>Message</b> for the <i>Severity</i> from the drop down list.

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

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

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

### View Data Quality Rule

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

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

1. Select the check box adjacent to the required DQ Name.
2. Click **View** button from the Data Quality Rules tool bar.

The DQ Definition screen displays the details of the selected Data Quality

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

### Modify Data Quality Rule

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

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

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

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

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

Update the details as required.

For more information, see Create Data Quality Rule section.

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

### Copy Data Quality Rule

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

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

1. Select the check box adjacent to the required DQ Name in the list whose details are to be duplicated.
2. Click **Copy** button from the Data Quality Rules tool bar. **Copy** button is disabled if you have selected multiple check boxes. The DQ Definition screen is displayed.
3. Edit the DQ definition Name and other details as required.

For more information, see Create Data Quality Rule section.

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

### Approve/Reject Data Quality Rule

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

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

1. Select the checkbox adjacent to the required DQ Name. Ensure that you select the "Saved" DQ definition based on the **Status** indicated in the **Data Quality Rules** grid.
2. Perform one of the following:
  - To **Approve** the DQ definition, click **Approve** button. The **User Comments** screen is displayed. Enter the notes or additional information to the user and click **OK**. The selected DQ definition is approved and a confirmation dialog is displayed.
  - To **Reject** the DQ definition, click **Reject** button. The **User Comments** screen is displayed. Enter the notes or additional information to the user and click **OK**.

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

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

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



## Delete Data Quality Rule

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

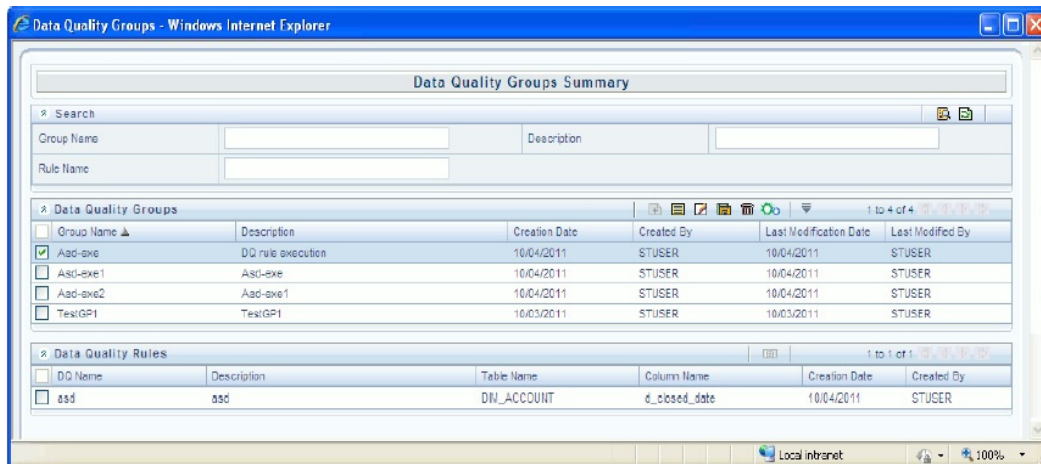
1. Select the check box adjacent to the required DQ Name whose details are to be removed.
2. Click **Delete** button from the Data Quality Rules tool bar.
3. Click **OK** to confirm deletion.

## Data Quality Groups Summary

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

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

**Figure 9. Data Quality Group Summary**



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

Group definitions and view, modify, copy, or delete DQ Group definitions within the **Data Quality Groups Summary** screen.

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

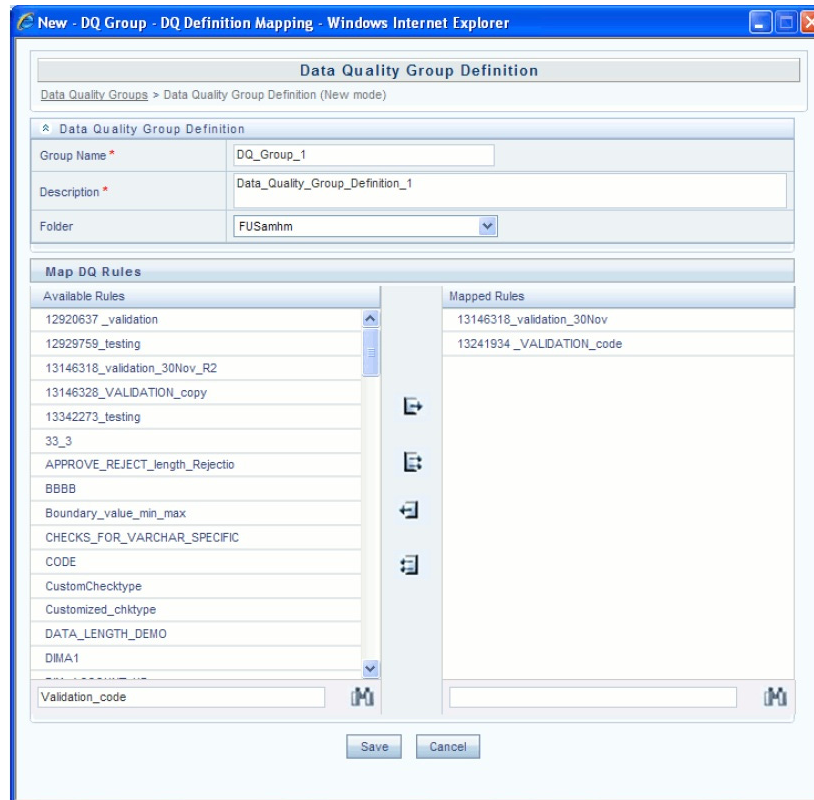
## Create Data Quality Group

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

To create DQ Group in the **Data Quality Groups Summary** screen:

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

**Figure 10. Data Quality Group Definition**



2. In the **Data Quality Group Definition** section, perform the following:
  - Enter the **Group Name** by which you can identify the DQ Group.
  - Enter a description or related information about the DQ Group.
  - Select the **Folder** (available for selected Information Domain) from the drop down list.
3. In the Map DQ Rules section, perform the following:
  - Select the required DQ Rule from the **Available Rules** list and click **Select**. You can also search to select a specific DQ Rule by entering the required keyword and clicking **Find** button.
  - To select all the listed DQ Rules, click **Select All**.

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

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

---

## Execute Data Quality Group

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

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

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

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

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

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

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

## View Data Quality Group

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

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

1. Select the check box adjacent to the required Group Name. The mapped DQ Rules are displayed in the Data Quality Rules section.
2. Click **View** button from the Data Quality Groups tool bar. The Data Quality Group Definition screen displays the DQ definition details.

## Modify Data Quality Group

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

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

1. Select the check box adjacent to the required Group Name.
2. Click **Edit** button from the Data Quality Groups tool bar. The Data Quality Group Definition screen is displayed.
3. Update the details and click **Save** to update the changes.

---

## Copy Data Quality Group

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

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

1. Select the check box adjacent to the required Group Name in the list whose details are to be duplicated.
2. Click **Copy** button from the Data Quality Groups tool bar. Copy button is disabled if you have selected multiple check boxes. The Data Quality Group Definition screen is displayed.
3. Edit the DQ Group Name and other details as required.
4. Click **Save**. The new DQ Group definition is displayed in the **Data Quality Groups Summary** screen.

## View Data Quality Group Summary Log

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

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

1. Select the check box adjacent to the Group Name in the **Data Quality Groups** grid.

The Data Quality Rules associated with the selected Group are displayed in the Data Quality Rules grid.

2. Select the check box adjacent to the DQ Name in the Data Quality Rules grid.
3. Click **View Logs** button. The View Logs screen is displayed.
4. In the View Logs screen, select the **Information Date** from the drop down list.

Based on the selection, you can select the **Group Run ID** and **Iteration ID** from the corresponding drop-down list.

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



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

## Delete Data Quality Group

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

**To delete:**

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

## Data Quality Rules For Staging Tables

Data Quality (DQ) Rules are framed and created based on Staging Tables. Each rule is based on specified staging table column Specific Check or table Generic Check. The rules created for each of the tables are detailed in the [DQ Check Rules](#) spreadsheet.

In Specific Check, a particular column is checked based on rule's predefined checks, where as in Generic Check any columns are not specified. Generic Check is useful if you have a check which is not Specific or you use IF-ELSE conditions or CASE statements.

The following screen displays the Specified DQ Rules:

Figure 11. Specified DQ Rules

ID	Name	Table	Access Type	Check Type	Folder	Creation Date	Created By	Last Modification Date	Status	Is Grouped
DQFSDW01002	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01003	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01004	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01005	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01006	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01012	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01013	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01016	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01019	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01021	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:21	FSDFT4USER	3/19/2013 22:16:21	Approved	Yes
DQFSDW01023	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:21	FSDFT4USER	3/19/2013 22:16:21	Approved	Yes
DQFSDW01014	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:21	FSDFT4USER	3/19/2013 22:16:21	Approved	Yes
DQFSDW01015	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:21	FSDFT4USER	3/19/2013 22:16:21	Approved	Yes
DQFSDW01018	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:21	FSDFT4USER	3/19/2013 22:16:21	Approved	Yes
DQFSDW01029	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01030	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01031	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01032	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01033	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01034	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01035	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01037	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01048	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01049	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01051	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01052	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:21	FSDFT4USER	3/19/2013 22:16:21	Approved	Yes
DQFSDW01053	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:21	FSDFT4USER	3/19/2013 22:16:21	Approved	Yes
DQFSDW01054	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:21	FSDFT4USER	3/19/2013 22:16:21	Approved	Yes
DQFSDW01055	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:21	FSDFT4USER	3/19/2013 22:16:21	Approved	Yes
DQFSDW01056	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:21	FSDFT4USER	3/19/2013 22:16:21	Approved	Yes
DQFSDW01057	STG_ANNUITY_CONTRACTS	ReadWrite	Specific Check	FSDFT4SEGC		3/19/2013 22:16:21	FSDFT4USER	3/19/2013 22:16:21	Approved	Yes
DQFSDW01058	STG_ANNUITY_CONTRACTS	ReadWrite	Generic Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01059	STG_ANNUITY_CONTRACTS	ReadWrite	Generic Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01064	STG_ANNUITY_CONTRACTS	ReadWrite	Generic Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01067	STG_ANNUITY_CONTRACTS	ReadWrite	Generic Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01074	STG_ANNUITY_CONTRACTS	ReadWrite	Generic Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	3/19/2013 22:16:20	Approved	Yes
DQFSDW01089	STG_ANNUITY_CONTRACTS	ReadWrite	Generic Check	FSDFT4SEGC		3/19/2013 22:16:20	FSDFT4USER	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.

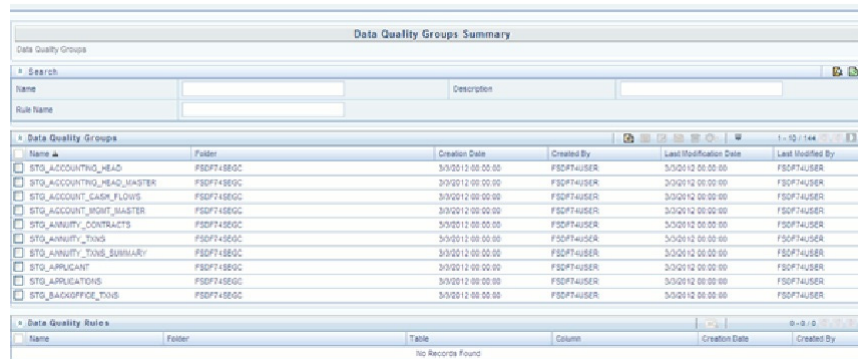
The [DQ Group Mapping](#) spreadsheet displays the total groups and corresponding Rules mapped to that Group.

### 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:

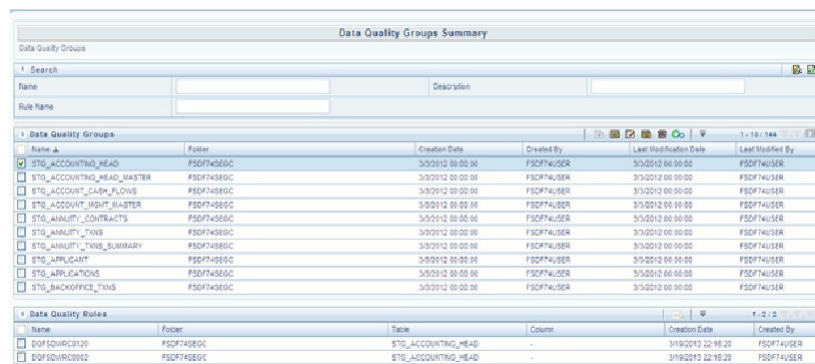
**Figure 12. DQ Group Execution**



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

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

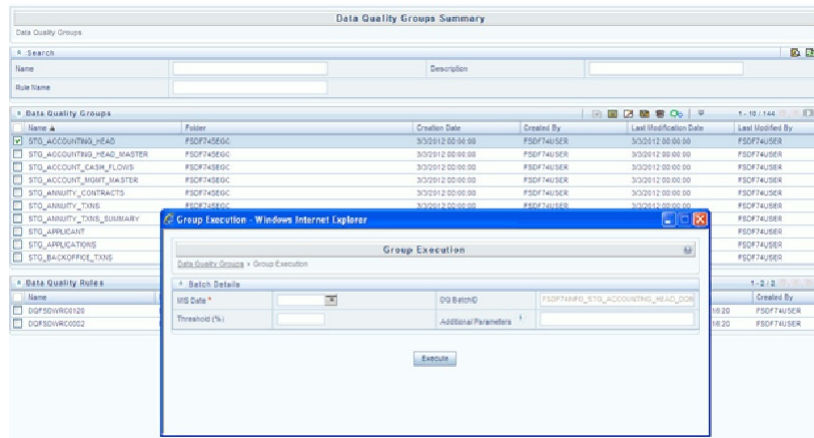
**Figure 13. DQ Rules**



2. Click **Execute** button.

The *Group Execution* window is displayed.

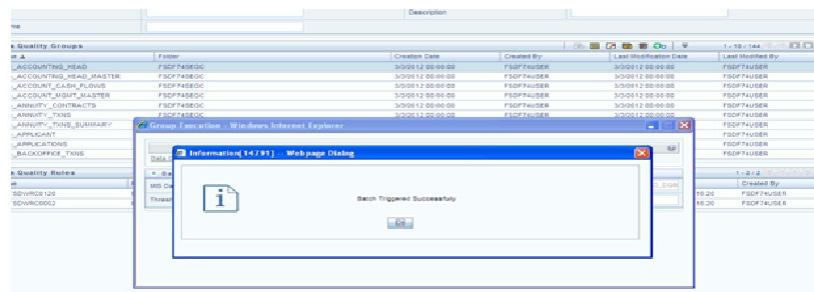
Figure 14. Group Execution



3. Enter the MIS Date.
4. Click **Execute** button from the *Group Execution* window.

The execution is triggered and the following message is displayed.

Figure 15. Execution Message



Once the execution is complete, you can view the details from the *Data Quality Rules Summary* window.

To view the execution logs:

1. Select the check box adjacent to the rule, of which you want to see the execution log.

Figure 16. Data Quality Summary Window

Data Quality Groups Summary						
Data Quality Groups						
Name	Folder	Creation Date	Created By	Last Modification Date	Last Modified By	
<input checked="" type="checkbox"/>	STG_ACCOUNTING_HEAD	FSDPT4SEGC	3/3/2012 00:00:00	FSDPT4USER	3/3/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_ACCOUNTING_HEAD_MASTER	FSDPT4SEGC	3/3/2012 00:00:00	FSDPT4USER	3/3/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_ACCOUNT_CASH_FLOWS	FSDPT4SEGC	3/3/2012 00:00:00	FSDPT4USER	3/3/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_ACCOUNT_MGMT_MASTER	FSDPT4SEGC	3/3/2012 00:00:00	FSDPT4USER	3/3/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_ANNUITY_CONTRACTS	FSDPT4SEGC	3/3/2012 00:00:00	FSDPT4USER	3/3/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_ANNUITY_TXNS	FSDPT4SEGC	3/3/2012 00:00:00	FSDPT4USER	3/3/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_ANNUITY_TXNS_SUMMARY	FSDPT4SEGC	3/3/2012 00:00:00	FSDPT4USER	3/3/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_APPLICATION	FSDPT4SEGC	3/3/2012 00:00:00	FSDPT4USER	3/3/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_APPLICATIONS	FSDPT4SEGC	3/3/2012 00:00:00	FSDPT4USER	3/3/2012 00:00:00	FSDPT4USER
<input type="checkbox"/>	STG_BACKGROUND_TXNS	FSDPT4SEGC	3/3/2012 00:00:00	FSDPT4USER	3/3/2012 00:00:00	FSDPT4USER

Data Quality Rules						
Name	Folder	Table	Column	Create Date	Created By	
<input checked="" type="checkbox"/>	DQFSDVRC0120	FSDPT4SEGC	STG_ACCOUNTING_HEAD	-	3/18/2013 22:18:21	FSDPT4USER
<input type="checkbox"/>	DQFSDVRC0202	FSDPT4SEGC	STG_ACCOUNTING_HEAD	-	3/18/2013 22:18:21	FSDPT4USER

2. Click **View Logs** button.

The *View Logs* window is displayed.

Figure 17. View Logs

View Logs								
Data Quality Groups > View Logs								
Group Execution details								
Information Date	Select Information Date	Group Run ID	Select Group Run ID					
Iteration ID								
Data Quality Rule Logs								
Check Name	Log Message	Message Date	Message Time	Total Rows	Rows Impacted	Assignment Type	Assignment Severity	Severity Message
No Records Found								

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

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

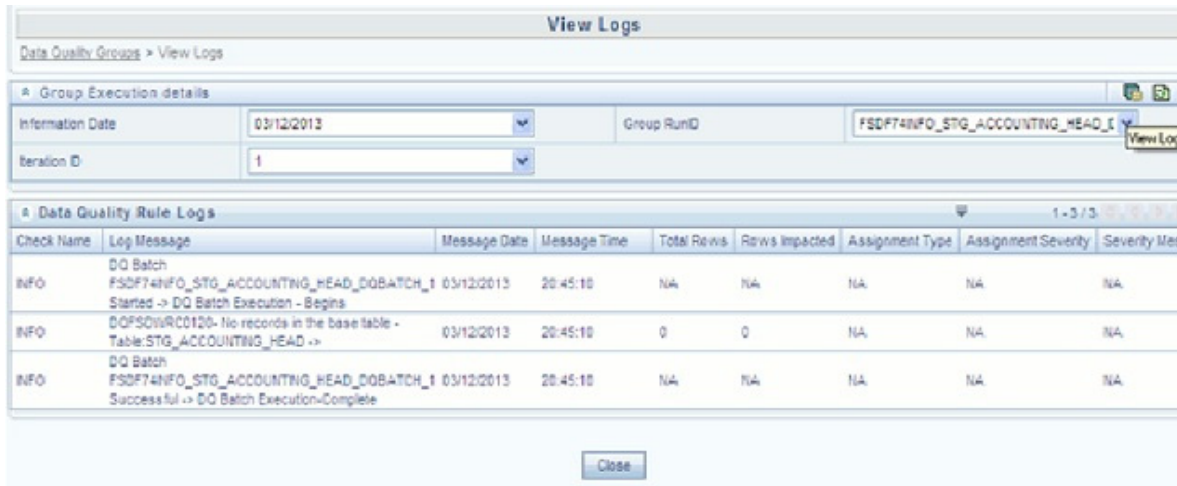
5. Select the **Iteration ID** from the drop down list.

6. Click **View Logs** button.

The details of the selected Group Execution are displayed.



**Figure 18. Group Execution Details**



### Batch Execution of DQ Rules

Create a Batch for executing DQ Rules and add a Task to the selected Batch. Add component as RUN DQ RULE and in Dynamic Parameter List, add a DQ Group.

A single Batch can have multiple number of Tasks and each Task is executing the DQ Group. Batch execution facilitates the execution of multiple Groups simultaneously.

You can also use the **Include** or **Exclude** functionality to determine which all groups have to be executed.

The following batches need to be executed in OFSDF.

**Table 22.**

V_BATCH_ID	V_BATCH_DESCRIPTION
XXXX_ALM	Data Quality batch for ALM tables
XXXX_CAMPAIGN	Data Quality batch for CAMPAIGN tables
XXXX_COLLATERAL	Data Quality batch for Collateral tables
XXXX_COLLECTIONAND RECOVERY	Data Quality batch for Collection and Recovery tables
XXXX_CRM	Data Quality batch for CRM tables
XXXX_CUSTOMER	Data Quality batch for Customer tables
XXXX_EXPOSURE	Data Quality batch for EXPOSURE tables
XXXX_GLANDACCOUNT ING	Data Quality batch for GL and Accounting group
XXXX_LRM	Data Quality batch for LRM tables
XXXX_MARKETRISK	Data Quality batch for MARKET RISK tables
XXXX_MASTER	Data Quality batch for master tables
XXXX_MISCELLANEOUS	Data Quality batch for Miscellaneous tables

**Table 22.**

V_BATCH_ID	V_BATCH_DESCRIPTION
XXXX_OPRIK	Data Quality batch for OP RISK tables
XXXX_ORIGINATION	Data Quality batch for Origination tables
XXXX_PRODUCTPROCESSORS	Data Quality batch for Product Processors group
XXXX_RATES	Data Quality batch for Rates tables
XXXX_RATING	Data Quality batch for RATING tables
XXXX_SECURITIZATION	Data Quality batch for Securitization tables
XXXX_TRANSACTIONS	Data Quality batch for Transactions tables
XXXX_TRANSACTIONSUMMARY	Data Quality batch for Transaction Summary tables

The DQ Batches and corresponding groups are detailed in the file [DQ Batch Group Mapping](#) spreadsheet.

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

# *Naming Conventions Used in OFSDF Data Model*

This Appendix chapter explains the various naming conventions used in OFSDF Logical Data Model and Physical Data Model. In addition, the domains in PDM and LDM are also listed with their descriptions.

## **Naming Conventions Used in OFSDF LDM and PDM**

ERwin Data Model consists of Logical and Physical data structures for each model file.

The following section explains the various naming conventions used in Oracle Financial Services Data Foundation (OFSDF) Logical Data Model (LDM) and Physical Data Model (PDM).

- OFSDF Logical Data Model Naming Conventions
- OFSDF Physical Data Model Naming Conventions
- Domains (PDM and LDM)

### **OFSDF Logical Data Model Naming Conventions**

ERwin modeler tool allows two views namely Logical view and Physical view for each model file. Accordingly, the OFSDF LDM file can be viewed in logical view mode and physical view mode. Different naming conventions and standards are applied to the two views of the OFSDF LDM.

**Note:** The physical view of the OFSDF LDM is not the same as the OFSDF PDM. The PDM is a distinct model and is shipped as a separate ERwin File.

- LDM Logical View Mode Entity Naming Conventions
- LDM Physical View Mode Table Naming Conventions

### **LDM Logical View Mode Entity Naming Conventions**

Entities in the logical view are named in such a way that the names themselves convey the functional meaning of the entity. The first letter of each word in an entity name is capitalized (e.g. "Application Stage"). Entity names also contain common suffixes for the classes of entities listed in the following table.

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

**Table 23. Entity Naming Conventions**

Class of Entity	Suffix	Example
Type	Type	Asset-Liability Type
Group	Group	Product Group
Code	Code	Collection Status Code
History	History	Account Balance History
Associative entities	Map	Customer Account Map

**Table 24. Attribute Name Suffixes**

Class of Attribute	Suffix	Required Values (If any)
Unique Identifier	ID	
Reference Identifiers	Reference ID	
Code	Code	
Number	Number	
Name	Name	
Description	Description	
Boolean Values	Flag	Y or N
Indicator	Indicator	More than one value ('A', 'B', 'C' etc)
Percentage	Percent	
Rate	Rate	
Amount	Amount	
Balance	Balance	
Term	Term	
Frequency	Frequency	
Unit	Unit	
Record created by		Created By
Record modified by		Modified By
Record created on		Created On
Record modified on		Modified On
Record validity start date		From Date
Record validity end date		End Date

### LDM Physical View Mode Table Naming Conventions

In the physical view of the OFSDF LDM follows a different naming convention for entities and attributes in the model, and these reflect naming standards that are more readily acceptable to table and column naming constraints of common database systems.

The entity names in the physical view mode of the LDM are capitalized and each entity follows the following naming pattern:

- FSW\_<Subject Area>\_<Table Name>
- 'FSW' is a constant prefix.
- <Subject Area> is abbreviated form of the entity's primary subject area.
- <Table Name> is a meaningful name for the entity with no embedded spaces between words. Table Name may be abbreviated.

**Table 25. Naming Conventions**

Logical View Name	Physical View Name	Description
Account Address	FSW_ACCT_ACCOUNT_AD DRESS	Account Address entity belonging primarily to Account (ACCT) subject area.
Letter Of Credit	FSW_CON_LETTER_OF_CR EDIT	Letter of Credit entity belongs primarily to the Contracts (CON) subject area.

## OFSDF Physical Data Model Naming Conventions

The OFSDF PDM consists of tables grouped into two distinct areas namely the Staging Area and Results Area. The Staging Area consists of tables for data sourcing and the Results Area consists of the star schemas / datamarts for reporting and BI.

ERwin modeler tool allows two views namely Logical view and Physical view for each model file. Accordingly, the OFSDF PDM (Physical Data Model) file can be viewed in logical view mode and physical view mode. Different naming conventions and standards are applied to the two views of the OFSDF Physical Data Model.

- PDM Logical View Mode Naming Conventions
- PDM Physical View Mode Naming Conventions

### PDM Logical View Mode Naming Conventions

In the logical view model, OFSDF PDM model tables and columns have descriptive names that readily convey the meaning and use of the element. In the logical view, names of tables and columns can have more than one word with the first letter of each word capitalized. Staging Area structures and Results Area structures have different name prefixes and suffixes as outlined in the table below:

**Table 26. Prefixes and Suffixes**

PDM Area		Prefix	Suffix	Example
Staging (Data Sourcing)	All Tables	Stage		Stage Internet Transactions
	Transaction Tables		Transactions	Stage Card Settlement Transactions
	Master Tables		Master	Stage Product Master
Results (Data- mart)	All Fact Tables	Fact		Fact Process Cashflow
	All Dimension Tables		Dimension	Account Dimension

### PDM Physical View Mode Naming Conventions

In the physical view model, OFSDF PDM tables and columns may have abbreviated words joined by underscore character to form more meaningful and descriptive names.

Table names in the physical view are capitalized

**Table 27. Physical view**

PDM Area		Prefix	Suffix	Example
Staging (Data Sourcing)	All Tables	STG_		STG_LOAN_CO NTRACTS
	Interface tables for data from MDM and core banking systems		_INTF	STG_ORG_UNI T_ATTR_INTF
	Transaction Tables		_TXNS	STG_CARDS_SE TTLEMENT_TX NS

**Table 27. Physical view**

PDM Area		Prefix	Suffix	Example
	Master Tables		_MASTER	STG_CUSTOME R_MASTER
Results (Datamart)	All Fact Tables	FCT_		FCT_ACCOUNT _SUMMARY
	All Dimension Tables	DIM_		DIM_ DIM_PRODUCT

The table below lists the prefix and/or suffix used for columns names in the physical view of the OFSDF PDM. The prefix or suffix depends on the class and data type of the column.

Column name prefix to indicate column datatype:

**Table 28. Column Name Prefix**

Column Data Type	Prefix
Varchar	v_
Number	n_
Date	d_
Flag	f_

Column name suffix for common classes of columns:

**Table 29. Column Name Suffix**

Column Class	Suffix
Method	_method
Percentage	_pct
Rate	_rate
Balance	_bal/_balance
Amount	_amt/_amount
Term	_term
Type	_type
Frequency	_freq

In addition, frequently occurring keywords in column names may be abbreviated as shown in table below:

**Table 30. Abbreviations**

Name	Abbreviated Form	Name	Abbreviated Form
Accrual	accr	Local Currency	lcy
Account	acct	Line Of Business	lob
Accounting Currency	acy	Maximum	max
Address	addr	Minimum	min
Adjustment	adj	Mortgage	mort
Advance	adv	Message	msg
Amount	amt	Multiplier	mult
Application	app	Number	num
Average	avg	Over Draft	od
Balance	bal	Option	opt
Business	bus	Origination	org
Currency	ccy	Percent	pct
Consolidation	cons	Payment	pmt
Customer	cust	Prepayment	ppmt
Description	desc	Product	prod
Dimension	dim	Source	src
Detail	dtl	Status	stat
Earnings at Risk	ear	Statistics	stats
End Of Period	eop	Temporary	temp
Error	err	Total	tot
Flag	flg	Transaction	txn
Frequency	freq	Value at Risk	var
Future	fut	Value	val
Forex	fx		
Generation	gen		
General Ledger	gl		
Hierarchy	hier		
History	hist		

**Domains (PDM and LDM)**

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



**Table 31. Domains and their Descriptions**

Serial No	Domain Name	Domain Description
1	Date	DATE
2	Timestamp	TIMESTAMP
3	Number	NUMBER(10)
4	Amount	NUMBER(22,3)
5	Code	NUMBER(5)
6	Flag	CHAR(1)
7	Frequency	NUMBER(5)
8	ID	VARCHAR2(25)
9	Percent	NUMBER(10,6)
10	Percent_Long	NUMBER(15,11)
11	Phone_Fax_Number	NUMBER(15)
12	Rate	NUMBER(10,6)
13	Term	NUMBER(5)
14	Alphanumeric_Code	VARCHAR2(10)
15	Name	VARCHAR2(60)
16	Currency_Code	VARCHAR2(3)
17	Short_Description	VARCHAR2(60)
18	Description	VARCHAR2(255)
19	Account_Number	VARCHAR2(25)
20	System_Identifier	NUMBER(20)
21	Long_Description	VARCHAR2(4000)



This Appendix provides information about How to Define a Batch in the Oracle Financial Services Data Foundation application and step-by-step instructions to use this section.

This chapter includes the following topic:

- Batch Definition

## Batch Definition

Create a batch from the OFSAAI *Batch Maintenance* screen as follows:

### Procedure

1. From the OFSAAI **Home** menu, navigate to **Operations > Batch Maintenance**.
2. In the *Batch Maintenance* window, Select '+' button from the *Batch Name* tool bar.

The *New Batch Definition* window is displayed.

3. Enter the Batch details as tabulated.

**Table 32. Batch Details**

Field	Description
Batch Name	The <b>Batch Name</b> is auto generated by the system. You can edit to specify a Batch name based on the following conditions: <ul style="list-style-type: none"> <li>• The Batch Name should be unique across the Information Domain.</li> <li>• The Batch Name must be alphanumeric and should not start with a number.</li> <li>• The Batch Name should not exceed 41 characters in length.</li> <li>• The Batch Name should not contain special characters "." and "-".</li> </ul>
Batch Description	Enter a description for the Batch based on the Batch Name.
Duplicate Batch	(Optional) Select the check box to create a new Batch by duplicating the existing Batch details. On selection, the <b>Batch ID</b> field is enabled.
Batch ID (If duplicate Batch is selected)	It is mandatory to specify the Batch ID if Duplicate Batch option is selected. Select the required <b>Batch ID</b> from the list.
Sequential Batch	Select the check box if the Batch has to be created sequentially based on the task specified. For example, if there are 3 tasks defined in a Batch, task 3 should have precedence as task 2, and task 2 should have precedence as task 1.

- 
4. Click **Save** to save the Batch definition details.

The new Batch definition details are displayed in the *Batch Name* section of *Batch Maintenance* window with the specified **Batch ID**.



