Oracle® Financial Services Retail Performance Analytics

User Guide

Release 8

Part No. E59607-01

February 2015

ORACLE

Contents

5 Customer Dimension Population

6 Account Dimension Population

7 Exchange Rate History Population

8 Loading Multiple Load Runs in OFSAA

9 Account Summary Population

10 Fact Transaction Summary

11 Customer Summary Population

12 Fact Data Population

13 Cube Build Process

14 Predictive Modeling

15 Overview of OFSRPA Reports

Preface

Intended Audience

Welcome to Release 8 of the *Oracle Financial Services Retail Performance Analytics User Guide.*

See Related Information Sources on page viii for more Oracle product information.

Documentation Accessibility

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

Access to Oracle Support

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

Structure

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). Time dimension makes it possible to report the balances by Year, Quarter, or Month. For example, the monthly data for January, February, and March gets rolled up to Quarter 1 and the Quarter 1, 2, 3, and 4 data get rolled up to, say Year 2013. 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, three month rolling average).

5 Customer Dimension Population
6 Account Dimension Population
7 Exchange Rate History Population
8 Loading Multiple Load Runs in OFSAA
9 Account Summary Population

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.

10 Fact Transaction Summary
11 Customer Summary Population

This chapter explains the process flow for populating Fact Common Customer Summary table.

12 Fact Data Population
13 Cube Build Process
14 Predictive Modeling
15 Overview of OFSRPA Reports
16 Segmentation
17 Service Calls to RPA
A How to Add a New Dimension
B How to Add a New Measure
C How to Develop a New Cube
D List of Members
E How to Define a Batch
F Run Rule Framework

Related Information Sources

Oracle Financial Services Channel Analytics (OFSCA) User Guide

Oracle Financial Services Institutional Performance Analytics (OFSIPA) User Guide

Oracle Financial Services Retail Customer Analytics (OFSRCA) User Guide

1

Introduction

Overview of Oracle Financial Services Retail Performance Analytics (OFSRPA)

Oracle Financial Services Retail Performance Analytics (OFSRPA) is a complete end-to-end web-based Business Intelligence solution which provides a 360 degree view of the customer relationship for key insights into the customer life-cycle.

OFSRPA provides tools for data integration and includes customizable, pre-built dashboards and reports, a reporting data model, and user friendly functional subject areas for ad-hoc reporting.

It also provide you deep insights into customer engagements across target segments and products/Line Of Business (LOB) including lending, credit cards, and so on.

It proactively manage the growth through strategic insights into the retail business performance.

OFSRPA helps you to monitor customer distribution across credit and delinquency bands and related exposures.

The OFSRPA solution is a part of Profitability Pack and is packaged along with AAI 8.0 and other applications. This OFSRPA is supported for Oracle 11g and 12c.

OFSRPA solution is built using:

- OBIEE 11.1.1.7.1 for Dashboard and Reports activities
- Essbase 11.1.2.3+ for 12c database

This manual deals with essential Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) required for OFSRPA activities, process flow for the data transformation, cube building processes, and functional details about the dash boards and reports. In addition, it includes subject areas which could be used for ad-hoc reporting using OBIEE Answers tool.

2

Overview of Process Flow

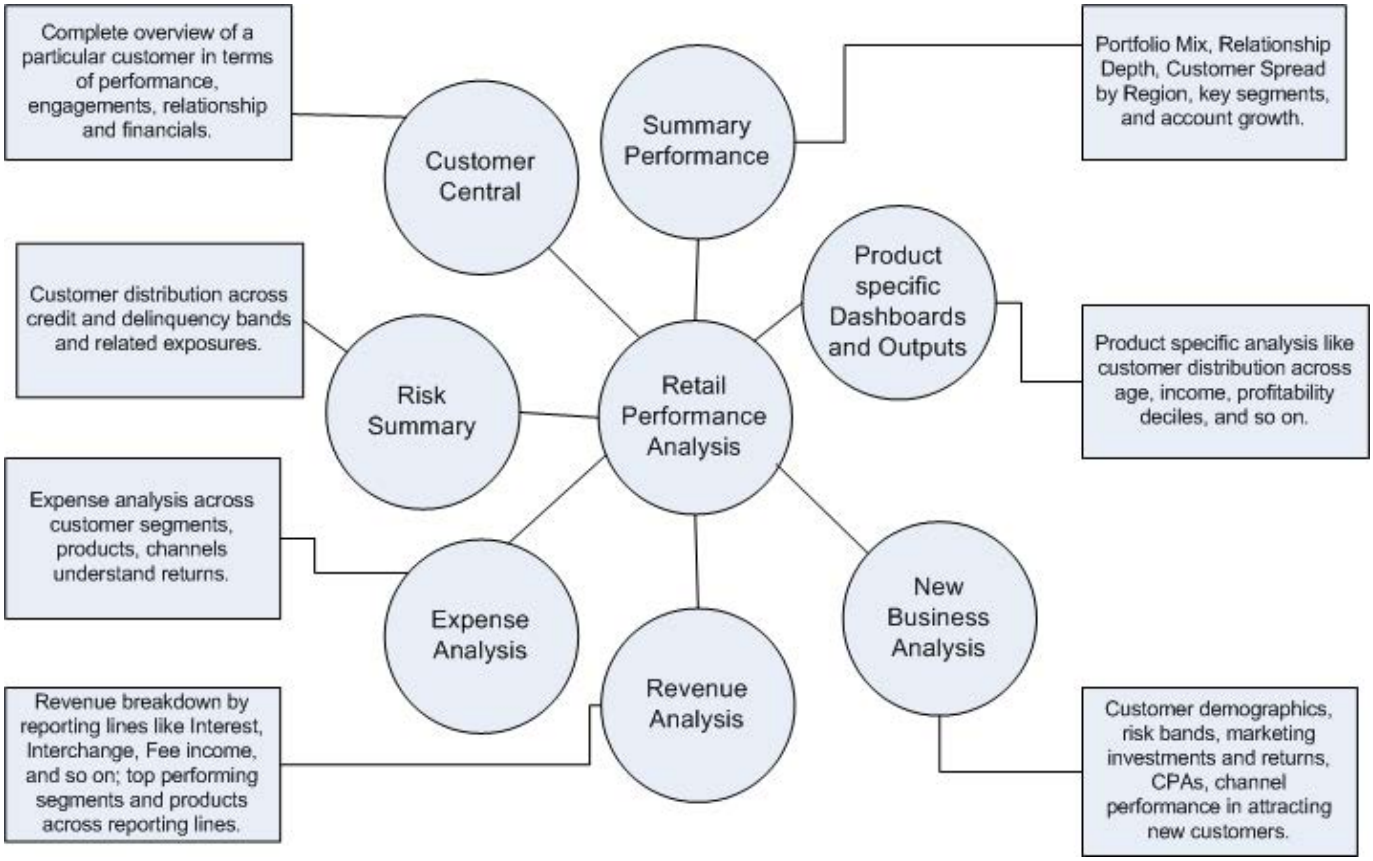
Introduction

Oracle Financial Services Retail Performance Analytics (OFSRPA) 8.0 utilizes OBIEE technology to:

- Gain deep insight into customer engagements across target segments and products/LOB including lending, credit cards, and so on.
- Perform Wallet share analysis and Customer Profitability.
- Understand the efficiency of investments (like marketing, branch, and channel and so on) over time.
- Monitor customer distribution across credit and delinquency bands and related exposures.
- Perform an enterprise-wide revenue analysis across customer segments, products, and reporting lines including fee income, interest, and interchange.
- Summary performance of the LOBs, overall Profitability, and Portfolio mix.
- Customer trends across performance drivers like Sales, Balances, Deposits, Product subscriptions (revenue services), Credit scores, Delinquency bands, Losses, and so on.
- LOB specific performance reports can be analyzed against key dimensions like customer segments, product family, region, branch, risk scores, and so on.
- Analyze expenses across customer segments, products, and channels to understand ROI.

Following explains the product objectives of OFSRPA 8.0:

Following explains the product objectives of OFSRPA 8.0:



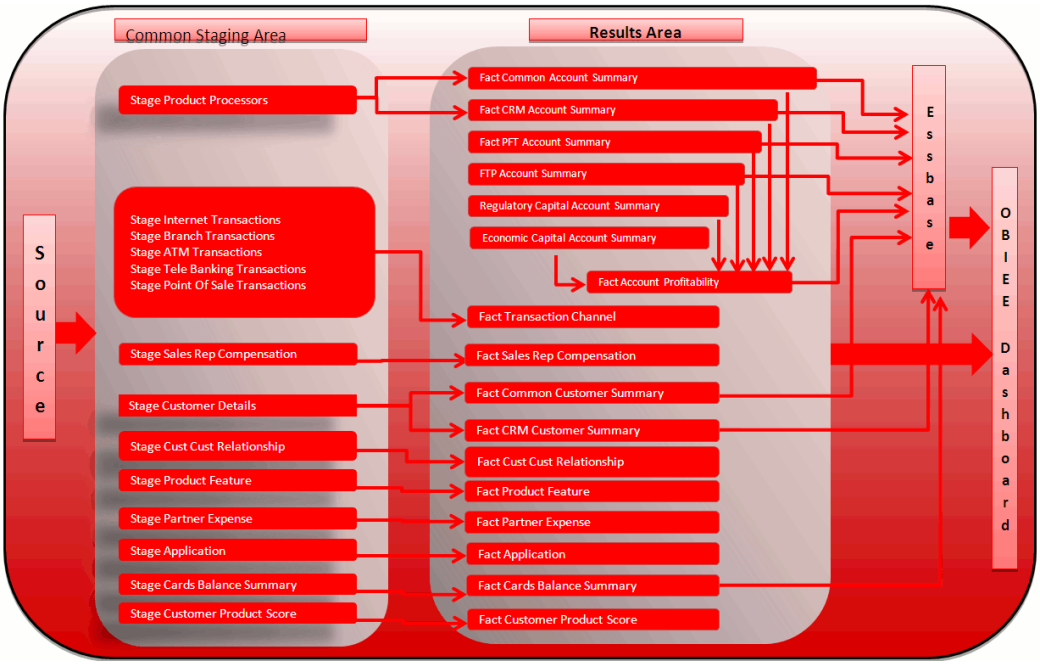
For details on OFSRPA reports and how OBIEE is being utilized, see Overview of OFSRPA reports, page 15-1.

OFSRPA 8.0 is designed for OBIEE reading data from relational database. The relational database comprises of various dimensions and facts in the BI data model.

OFSRPA 8.0 can be independently licensed and installed to work on top of the OFSAAI 8.0 Infrastructure.

Data Flow

Retail Performance Analytics data model contains the staging tables from which data is loaded in to the dimensions and fact tables. Staging tables include the master staging tables, detail staging tables, staging product processor tables, and so on. The user has to populate data into these staging tables.



Dimension Data Flow

Dimension data in OFSRPA application is loaded from staging master tables using the Slowly Changing Dimensions (SCD) process. Data from source systems can be loaded into staging through flat file or source system interfaces.

SCD process tracks the changes in the dimensional attributes and loads data into dimension tables. Few examples of dimension tables that follow the SCD process are Product, Customer Type, Customer, Campaign, and so on.

Some dimensions are static or maintained internally within the application and are not expected as a download from source system. For example, Reporting Line. These dimensions are maintained through the AMHM (Attribute Member Hierarchy

Maintenance) component of OFSAAI or through other framework components like DEFI.

Following are the list of Dimensions used in OFSRPA:

Dimension Entity Name	Staging Entity Name(s)	Loading/Maintenance Method
Account Status Dimension	Stage Account Status Master	SCD
Application Reject Reasons Dimension	Stage Application Reject Reason Master	SCD
Application Status Dimension	Stage Application Status Master	SCD
Application Type Dimension	Stage Application Type Master	SCD
Attrition Dimension	Stage Attrition Reason Master	SCD
Authorization Decision Reasons Dimension	Stage Auth Decision Reason Master	SCD
Balance Category Dimension	Stage Credit Card Balance Category Master	SCD
Card Type Dimension	Stage Card Type Master	SCD
Channel Transaction Dimension	Stage Transaction Channel Type Master	SCD
Country Dimension	Stage Country Master	SCD
Credit Center Dimension	Stage Credit Center Master	SCD
Credit Officer Dimension	Stage Credit Officer Master	SCD
Customer Dimension	Stage Customer Master	SCD
Customer Type Dimension	Stage Customer Type Master	SCD
Decision Status Dimension	Stage Decision Status Master	SCD

Dimension Entity Name	Staging Entity Name(s)	Loading/Maintenance Method
Deviation Reasons Dimension	Stage Deviation Reason Master	SCD
Education Dimension	Stage Customer Education Master	SCD
Geography Dimension	Stage Geography Master	SCD
Home Ownership Dimension	Stage Home Ownership Master	SCD
Household Dimension	Stage Household Master	SCD
Industry Dimension	Stage Industry Master	SCD
LoB Dimension	Stage LOB Master	SCD
Management Dimension	Stage Account Mgmt Master	SCD
Merchant Dimension	Stage Merchant Master	SCD
Merchant Category Dimension	Stage Merchant Category Master	SCD
Migration Reasons Dimension	Stage Migration Reason Master	SCD
Offer Dimension	Stage Offer Master	SCD
Reason Dimension	Stage Opportunity Win Loss Reason Master	SCD
Organization Structure Dimension	Stage Organization Structure Dimension	SCD
Partner Dimension	Stage Partner Master	SCD
Pool Identification Dimension	Stage Pool Identification Master	SCD

Dimension Entity Name	Staging Entity Name(s)	Loading/Maintenance Method
Prepayment Reason Dimension	Stage Prepayment Reason Master	SCD
Product Dimension	Stage Product Master	SCD
Loan Product Category Dimension	Stage Product Category Master	SCD
Product Feature Dimension	Stage Product Feature Master	SCD
Product Type Dimension	Stage Product Type Master	SCD
Prospect Dimension	Stage Prospect Master	SCD
Retention Offer Type Dimension	Stage Retention Offer Master	SCD
Sales Representative Dimension	Stage Sales Rep Master	SCD
Sales Stage Dimension	Stage Sales Stage Master	SCD
Terminal Dimension	Stage Terminal Master	SCD
Terminal Type Dimension	Stage Terminal Type Master	SCD
Transaction Dimension	Stage Transaction Master	SCD
Transaction Channel Dimension	Stage TXN Channel Master	SCD
Txn Failure Reason Dimension	Stage Transactions Failure Reason Master	SCD
Transaction Status Dimension	Stage Transactions Status Master	SCD
Vendor Dimension	Stage Vendor Master	SCD

Dimension Entity Name	Staging Entity Name(s)	Loading/Maintenance Method
Vintage Dimension	Stage Vintage Master	SCD
Reporting Line Dimension	Reporting Line Dimension Members, Reporting Line Member Translation, Reporting Line Member Attributes, Reporting Line Hierarchies	AMHM/DT
Band Dimension	Band Dimension Members, Band Member Translation, Band Member Attributes	AMHM/SCD
Account Dimension	Stage LC Contracts	SCD
Account Dimension	Stage Commitment Contracts	SCD
Party Dimension	Stage Party	SCD
Account Dimension	Stage Stage OD accounts	SCD
Account Dimension	Stage Stage TD contracts	SCD
Account Dimension	Stage Stage Trusts	SCD
Account Dimension	Stage Stage Loan Contracts	SCD
Account Dimension	Stage Stage Mutual Funds	SCD
Account Dimension	Stage Bills Contracts	SCD
Account Dimension	Stage CASA Accounts	SCD
Account Dimension	Stage Guarantees	SCD
Account Dimension	Stage Stage leases contracts	SCD
Account Dimension	Stage Stage mm contracts	SCD

Dimension Entity Name	Staging Entity Name(s)	Loading/Maintenance Method
Account Dimension	Stage Annuity Contracts	SCD
Account Dimension	Stage Borrowings	SCD
Account Dimension	Stage Card Accounts	SCD
Account Dimension	Stage Investments	SCD
Region Dimension		Direct Load
Acquisition Channel Dimension		Direct Load
Instrument Category Dimension		Seeded
Currency Dimension		Seeded
Gender Dimension		Seeded
Marital Status Dimension		Seeded
Calendar Dimension		DT
Account Dimension	Staging Product Processor Tables like	SCD
	Stage Annuity Contracts,	
	Stage Bill Contracts,	
	Stage Borrowings,	
	Stage Cards,	
	Stage CASA Accounts,	
	Stage Guarantees,	
	Stage Investments,	

Dimension Entity Name	Staging Entity Name(s)	Loading/Maintenance Method
	Stage LC Contracts,	
	Stage Leases Contracts,	
	Stage Loan Contracts,	
	Stage Money Market Contracts,	
	Stage Over Draft Accounts,	
	Stage Term Deposit Contracts,	
	Stage Trusts,	
	Stage Swaps Contracts,	
	Stage Repo Contracts,	
	Stage Option Contracts,	
	Stage Mutual Funds,	
	Stage Futures And Forwards	

For account dimension, view is created for the population of accounts from staging to DIM account using SCD.

Some of the stage data can also come from master data management interfaces.

In such a case, data from interface is loaded into staging interface tables and SCD is run on the interface tables. Mapping of dimensional attributes to staging can be obtained by querying SYS_STG_JOIN_MASTER and SYS_TBL_MASTER table in the atomic schema.

Fact Data Flow

Most of the Fact tables are mapped to staging counterparts through Table to Table (T2T) mappings. Data from source systems can be loaded into staging through flat file or source system interfaces. T2T process then loads data to fact tables. Few examples are Fact Common Account Summary, Fact Campaign, and so on.

Some of the Fact tables are loaded with processed fact information from other fact tables. Few examples are Fact CRM Customer Summary, Fact Account Profitability, and

so on.

Fact Entity Name	Source	Source Entities	Method of populating measures
Fact Common Account Summary	Stage	Stage Annuity Contracts,	T2T
		Stage Bill Contracts,	
		Stage Borrowings,	
		Stage Cards,	
		Stage CASA Accounts,	
		Stage Guarantees,	
		Stage Investments,	
		Stage LC Contracts,	
		Stage Leases Contracts,	
		Stage Loan Contracts,	
		Stage Money Market Contracts,	
		Stage Over Draft Accounts,	
		Stage Term Deposit Contracts,	
		Stage Trusts	
Fact CRM Account Summary	Stage	Stage Annuity Contracts,	T2T
		Stage Bill Contracts,	

Fact Entity Name	Source	Source Entities	Method of populating measures
		Stage Borrowings,	
		Stage Cards,	
		Stage CASA Accounts,	
		Stage Guarantees,	
		Stage Investments,	
		Stage LC Contracts,	
		Stage Leases Contracts,	
		Stage Loan Contracts,	
		Stage Money Market Contracts,	
		Stage Over Draft Accounts,	
		Stage Term Deposit Contracts,	
		Stage Trusts	
Fact Common Customer Summary	Stage	Stage Customer Details,	T2T
		Stage Party Rating Details,	
		Stage Party Financials	
Fact CRM Customer Summary	Stage and Fact	Stage Customer Master,	T2T

Fact Entity Name	Source	Source Entities	Method of populating measures
		Stage Customer Details,	
		Fact Common Account Summary	
		Fact Transaction Channel	
Fact Application	Stage	Stage Applications	T2T
Transaction Channel	Stage	Stage Internet Transactions	T2T
		Stage Branch Transactions	
		Stage ATM Transactions	
		Stage TeleBanking Transactions	
		Stage Point Of Sale Transactions	
Fact Cards Balance Summary	Stage	Stage Credit Card Balance Summary	T2T
Fact Account Feature Map	Stage	Stage Account Feature Map	T2T
Fact Customer to Customer Relationship	Stage	Stage Customer to Customer Relationships	T2T
Fact Account Profitability	Fact	Fact Common Account Summary,	DT
		Fact FTP Account Summary,	

Fact Entity Name	Source	Source Entities	Method of populating measures
		Fact PFT Account Summary, Fact Regulatory Capital Account Summary, Fact Economic Capital Account Summary	
Exchange Rate History	Stage	Stage Exchange Rates	T2T
Fact Account Segment Score	Fact	Fact Common Account Summary	T2T
Fact Account Segment MOB Summary	Fact	Fact Account Profitability, Fact Common Account Summary, Fact Account Segment Score	DT
Fact Party Account Role Map	Stage	Stage Party Account Role Map	T2T
Fact Party Financials	Stage	Stage Party Financials	T2T

This table will be populated in presence of RCA application with the t2t. In absence of RCA, this table has to be directly loaded if the data is available.

The OFSRPA uses some materialized views registered as "Derived Entity", that has to be refreshed as and when the dependent table has fresh data. The MVs can be refreshed

by running the batches crated for the purpose. The list of Derived Entity and the dependent objects can be found in the following table.

MATERIALIZED_VIEW	REFERENCED_NAME	REFERENCED_OBJECT_NAME
ACNTSMRM	FCT_COMMON_ACCOUNT_SUMMARY	Table
ACNTSMRM	FCT_CRM_ACCOUNT_SUMMARY	Table
CUSTDETM	DIM_CUSTOMER	Table
CUSTDETM	DIM_CUSTOMER_TYPE	Table
CUSTDETM	DIM_GENDER	Table
CUSTDETM	FCT_COMMON_CUSTOMER_SUMMARY	Table
FCSTCUSR	VW_ACCT_VAL_FCST_CUSTAGG_RPA	Table
FCSTLTMR	VW_FORECAST_LTV_RPA	Table
FCSTREPR	VW_ACCT_VAL_FCST_REPAGG_RPA	Table
FSIUSRD	FSI_USER_DATA_ACCESS	Table
MGMTPFTM	ACNTSMRM	Table
MGMTPFTM	CUSTDETM	Table
MGMTPFTM	FCT_ACCOUNT_MGR_REL	Table
MGMTPFTM	FCT_ACCOUNT_PROFITABILITY	Table
MVRACPRO	A_DIM_REP_CURRENCY	Table
MVRACPRO	DIM_ACCOUNT	Table

MATERIALIZED_VIEW	REFERENCED_NAME	REFERENCED_OBJECT_NAME
MVRACPRO	DIM_CONSOLIDATION	Table
MVRACPRO	DIM_CURRENCY	Table
MVRACPRO	DIM_CUSTOMER	Table
MVRACPRO	DIM_CUSTOMER_TYPE	Table
MVRACPRO	DIM_DATES	Table
MVRACPRO	DIM_LOB	Table
MVRACPRO	DIM_ORG_UNIT	Table
MVRACPRO	DIM_PRODUCT	Table
MVRACPRO	DIM_REP_LINE	Table
MVRACPRO	DIM_VINTAGE	Table
MVRACPRO	FCT_ACCOUNT_PROFITABILITY	Table
MVRACPRO	FCT_COMMON_CUSTOMER_SUMMARY	Table
MVRACPRO	FCT_CRM_ACCOUNT_SUMMARY	Table
MVRACPRO	MVUSRACC	Table
MVRCUSAG	A_DIM_REP_CURRENCY	Table
MVRCUSAG	DIM_ACCOUNT	Table
MVRCUSAG	DIM_CONSOLIDATION	Table
MVRCUSAG	DIM_CURRENCY	Table
MVRCUSAG	DIM_CUSTOMER	Table

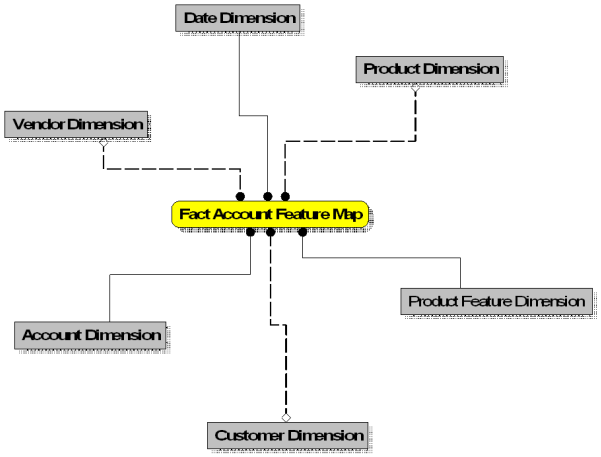
MATERIALIZED_VIEW	REFERENCED_NAME	REFERENCED_OBJECT_NAME
MVRCUSAG	DIM_CUSTOMER_TYPE	Table
MVRCUSAG	DIM_DATES	Table
MVRCUSAG	DIM_LOB	Table
MVRCUSAG	DIM_ORG_UNIT	Table
MVRCUSAG	DIM_PRODUCT	Table
MVRCUSAG	DIM_REP_LINE	Table
MVRCUSAG	DIM_VINTAGE	Table
MVRCUSAG	FCT_ACCOUNT_PROFITABILITY	Table
MVRCUSAG	FCT_COMMON_CUSTOMER_SUMMARY	Table
MVRCUSAG	FCT_CRM_ACCOUNT_SUMMARY	Table
MVRCUSAG	MVUSRACC	Table
MVRPROAG	A_DIM_REP_CURRENCY	Table
MVRPROAG	DIM_ACCOUNT	Table
MVRPROAG	DIM_CONSOLIDATION	Table
MVRPROAG	DIM_CURRENCY	Table
MVRPROAG	DIM_CUSTOMER	Table
MVRPROAG	DIM_CUSTOMER_TYPE	Table
MVRPROAG	DIM_DATES	Table
MVRPROAG	DIM_LOB	Table

MATERIALIZED_VIEW	REFERENCED_NAME	REFERENCED_OBJECT_NAME
MVRPROAG	DIM_ORG_UNIT	Table
MVRPROAG	DIM_PRODUCT	Table
MVRPROAG	DIM_REP_LINE	Table
MVRPROAG	DIM_VINTAGE	Table
MVRPROAG	FCT_ACCOUNT_PROFITABILITY	Table
MVRPROAG	FCT_COMMON_CUSTOMER_SUMMARY	Table
MVRPROAG	MVUSRACC	Table
MVUSRACC	DIM_ACCOUNT	Table
MVUSRACC	FCT_COMMON_ACCOUNT_SUMMARY	Table
MVUSRACC	FSIUSRD	Table
RTHREPMV	WITH_REP_LINE_DIRECT_IND_RPA	Table
USRMGRMV	FSI_M_USER_MANAGER_MAP	Table

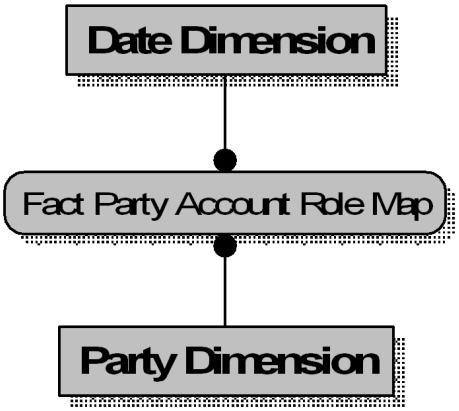
BI Data Model

Following are the subject areas in ERwin data model:

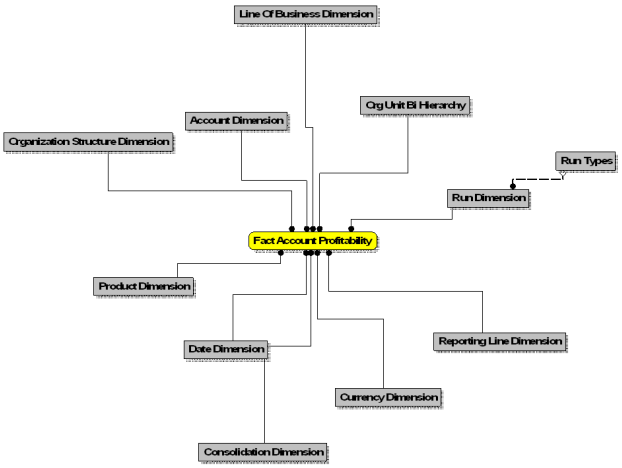
- Fact Account Feature Map



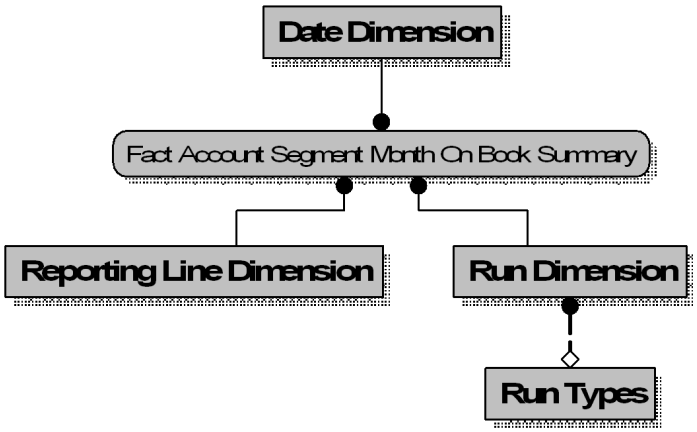
- Fact Account Party Role



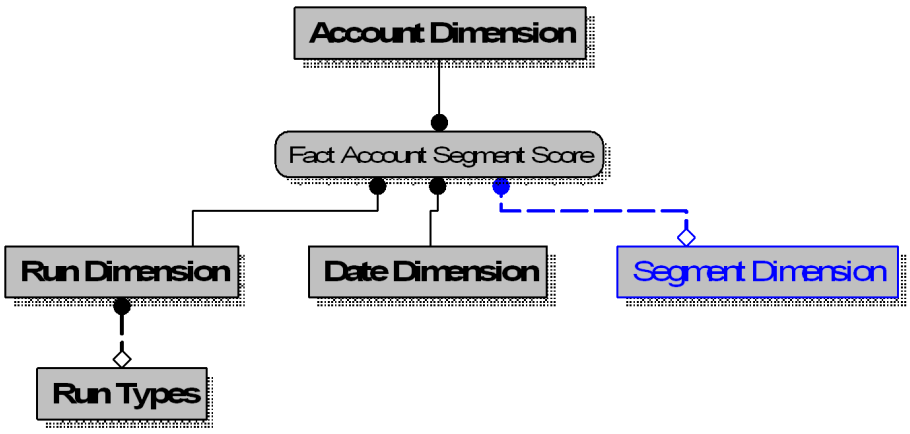
- Fact Account Profitability



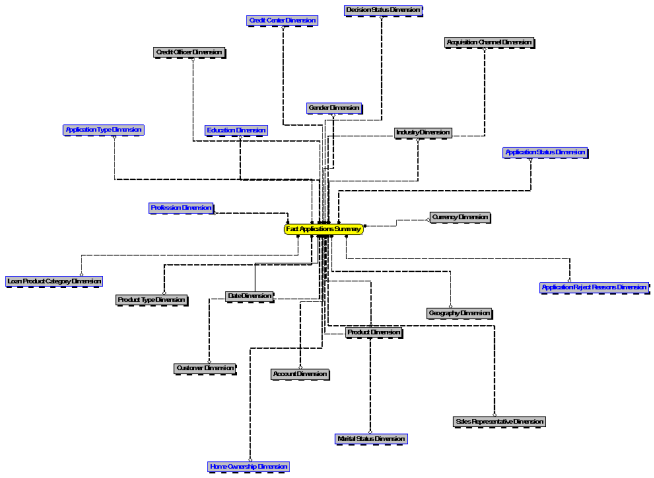
- Fact Account Segment MOB Summary



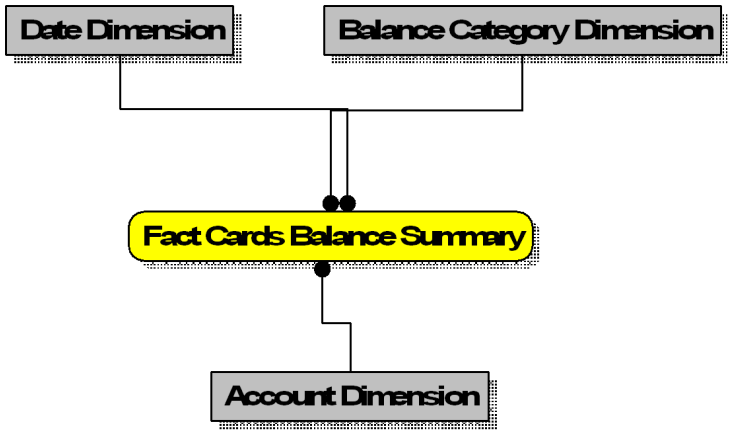
- Fact Account Segment Score



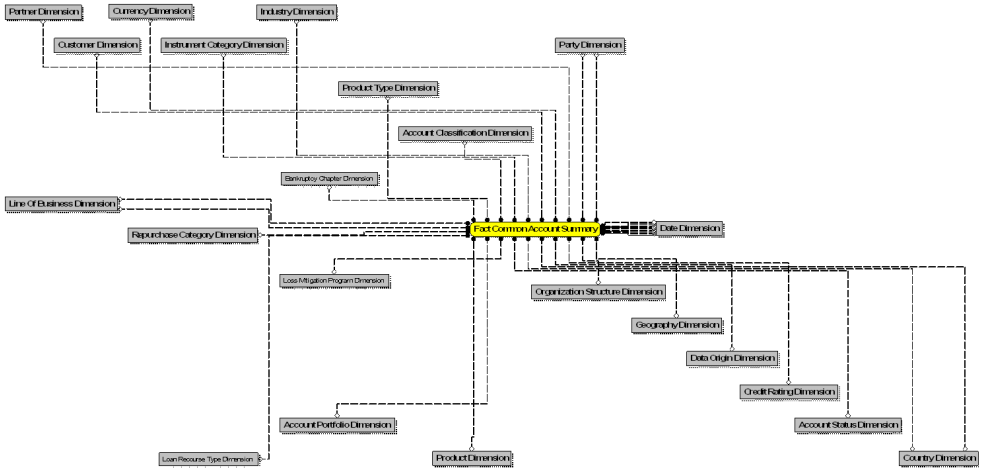
- Fact Applications Summary



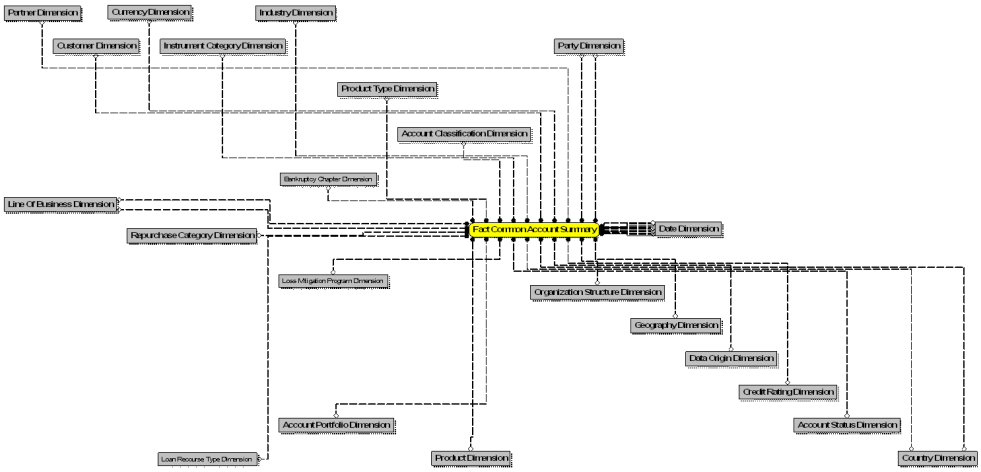
- Fact Cards Balance Summary



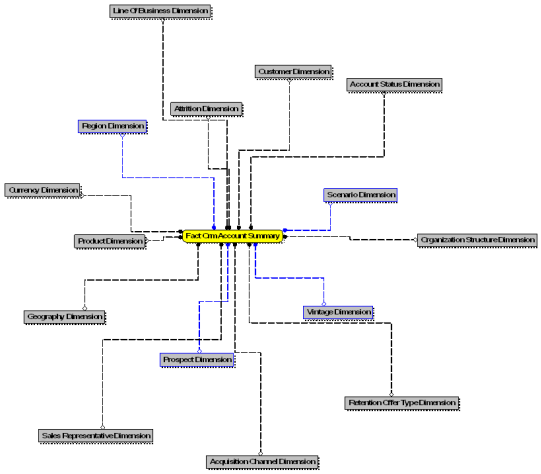
- Fact Common Account Summary



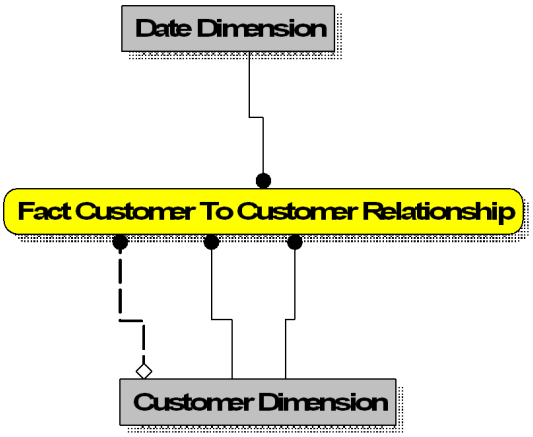
- Fact Common Customer Summary



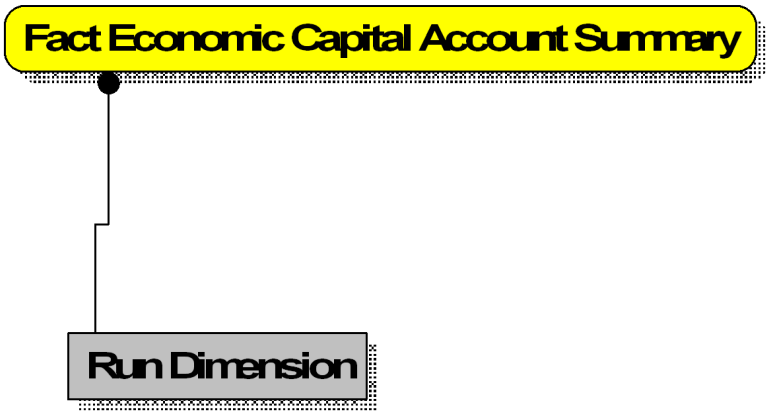
- Fact CRM Account Summary



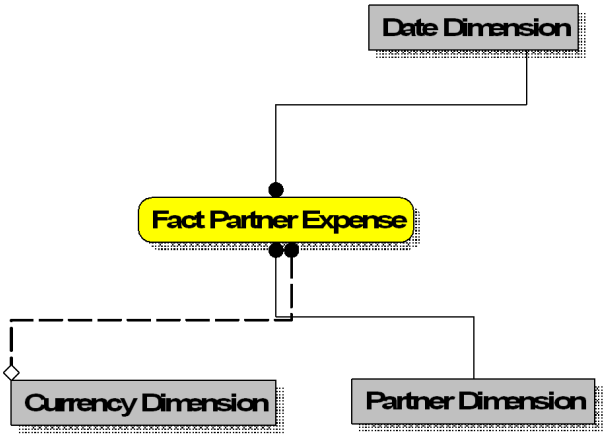
- Fact Cust Cust Relationship



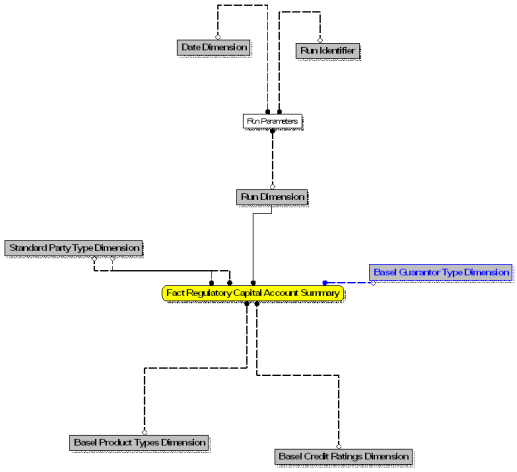
- Fact Eco Cap Account Summary



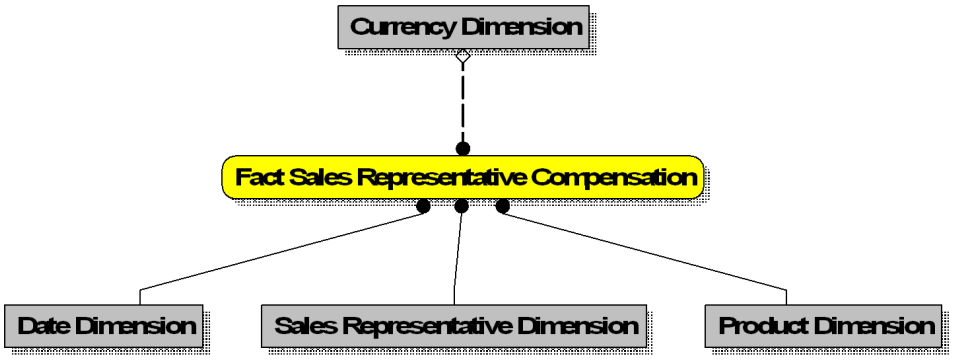
- Fact Partner Expense



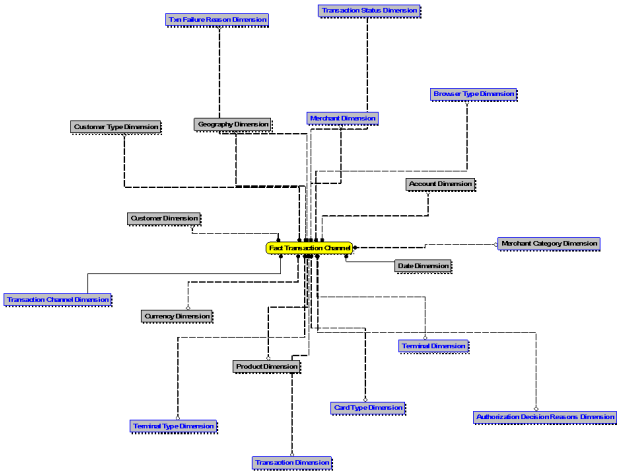
- Fact Reg Cap Account Summary



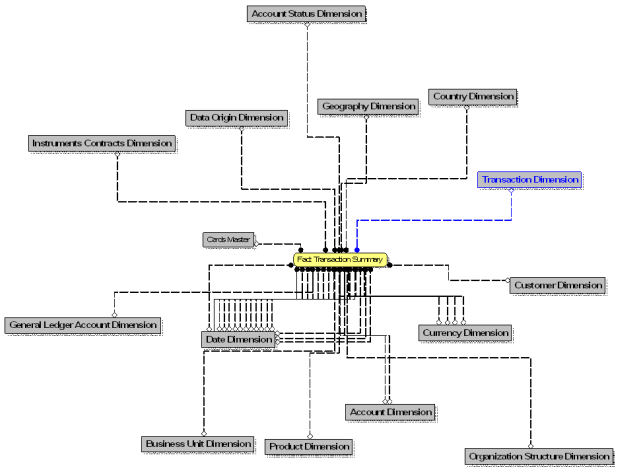
- Fact Sales Representative Compensation



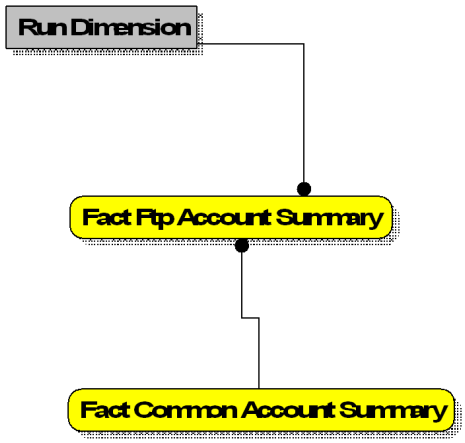
- Fact Transaction Channel



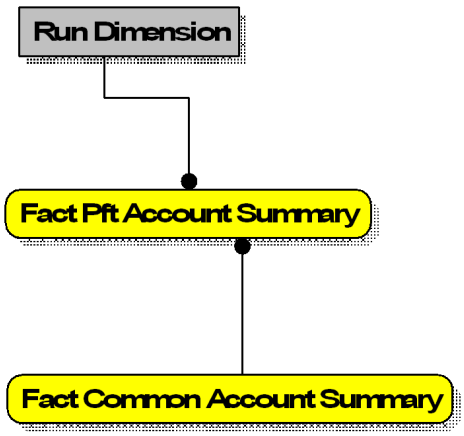
- Fact Transaction Summary



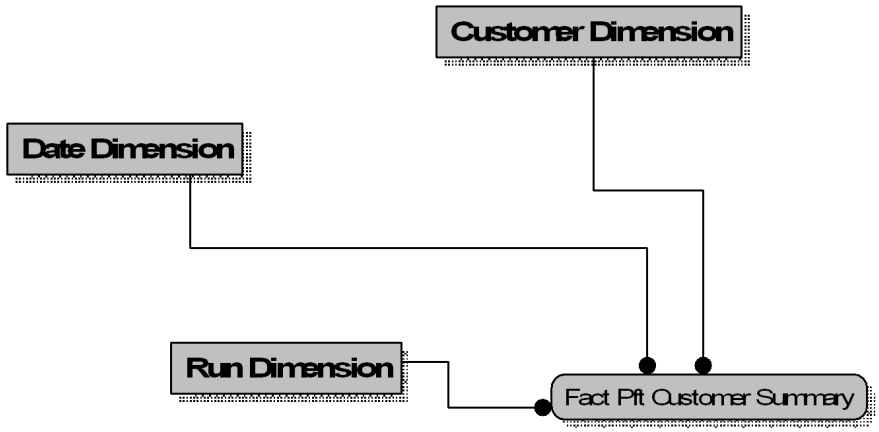
- FTP Account Summary



- PFT Account Summary



- PFT Customer Summary



Data Flow: OFSRPA BI Data Model to Essbase Cubes

Reports of OFSRPA application can be configured to work on Relational database or Hyperion Essbase Multi-dimensional databases, that is cubes. Multi-dimensional databases store aggregated data for better performance and provide mechanisms for performing non-additive rollup within a hierarchy and defining complex derived measures using cross-dimensional operations. OFSAA Infrastructure is used for defining metadata about the cube and for building the Essbase cubes. Essbase cubes can be built out of reporting fact entities to improve performance.

OFSRPA application has the following seeded cube metadata:

Cube Code	Cube Name	Fact Entities in dataset
ADCRM002	Retail Analysis	Fact Common Account Summary
		Fact CRM Account Summary
		Fact Common Customer Summary
		Fact CRM Customer Summary
		Fact FTP Account Summary
		Fact PFT Account Summary
ADCRM009	Cards Balance Summary	Fact Common Account Summary
		Fact CRM Account Summary
		Fact Common Customer Summary
		Fact CRM Customer Summary
		Fact Cards Balance Summary
ADCRM010	Account Profitability	Fact Common Account Summary
		Fact CRM Account Summary
		Fact Common Customer Summary
		Fact CRM Customer Summary
		Fact Account Profitability
ADCRM011	Customer Summary	Fact Common Customer Summary
		Fact CRM Customer Summary
ADRPARM1	RM PnL Cube for RPA	FCT_ACCOUNT_PROFITAIBILTY
		FCT_ACCOUNT_MGR_REL

3

Dimension Loading Process

Dimension Tables Population

OFSRPA solution use the SCD component to handle dimensional data changes.

Overview of SCD Process

SCDs are dimensions that have data that changes slowly, rather than changing on a time-based, regular schedule.

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/E14072_01/owb.112/e10935.pdf

Additional online sources include:

- http://en.wikipedia.org/wiki/Slowly_changing_dimension
- http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/10g/r2/owb/owb10 gr2_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=2 04800027&pgno=1
- http://www.informationweek.com/news/software/bi/showArticle.jhtml?articleID=5 9301280

Another 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 SCD component of the platform is delivered via a C++ executable. The types of SCD handled by the OFSAAI SCD component for OFSRPA solution are Type 1 and Type 2.

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

   SYS_TBL_MASTER stores the information regarding 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 regarding the source column, which is mapped to the respective target dimension table column. 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 populates one row per dimension for the seeded dimensions in this table.

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

Column Name	Data Type	Column Description
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: The following is the data put in by the solution installer for the Line of Business dimension.

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

SRC_KEY

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

- **SYS_STG_JOIN_MASTER**

  The solution installer populates this table for the seeded dimensions.

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

Column Name	Data Type	Column Description
COL_FORMAT	VARCHAR2(15) NULL	

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

1. PK - Primary Dimension Value (may be multiple for a given "Mapping Reference Number")
2. SK - Surrogate Key
3. DA - Dimensional Attribute (may be multiple for a given "Mapping Reference Number")
4. SD - Start Date
5. ED - End Date
6. LRI - Latest Record Indicator (Current Flag)
7. CSK - Current Surrogate Key
8. PSK - Previous Surrogate Key
9. SS - Source Key
10. LUD - Last Updated Date / Time
11. LUB - Last Updated By

Sample Data: The following is the data put in by the solution installer for the Line of Business dimension.

MAP_REF_NUM	6
COL_NM	V_LOB_CODE
COL_TYP	PK
STG_COL_NM	V_LOB_CODE

SCD_TYP_ID	
PRTY_LOOKUP_REQD_FLG	N
COL_DATATYPE	VARCHAR
COL_FORMAT	61

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

- 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 has 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 999999999999999999999999999
increment by 1
```

Executing the SCD Component

You can execute the function from the *Operations* (formerly Information Command Center (ICC) framework) module of OFSAAI, as mentioned below:

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 the section How to Define a Batch, page E-1.

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

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 **Run Executable** component from the drop down list.
5. Select the following from the **Dynamic Parameters** list:
   - **Datastore Type** - Select the appropriate datastore type from the drop down list
   - **Datastore Name** - Select the appropriate datastore name from the drop down list
   - **IP address** - Select the IP address from the list
   - **Executable** - Enter scd,<map ref num>

     For example:

     scd, 61

     scd, -1(If you want to process all the dimensions)

     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.
   - **Wait** - Click Yes if you want to wait till the execution is complete or click No to proceed with the next task.
   - **Batch Parameter** - Click **Yes** in Batch Parameter field if you want to pass the batch parameters to the executable and click **No** otherwise.

     Important: Always select **Y** in Batch Parameter.
6. Click **Save**.

   The Task definition is saved for the selected Batch.

You can execute a Batch definition from the *Batch Execution* section of *OFSAAI Operations* module.

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/ficgen*, where file name will have the Batch Execution ID.

The file name will have the batch execution id.

The detailed SCD component log can be accessed on the application server in the directory $FIC_HOME by accessing the path */ftpshare/<infodom name>/logs*, where file name will have the Batch Execution ID.

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

4

Time Dimension Population

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). Time dimension makes it possible to report the balances by Year, Quarter, or Month. For example, the monthly data for January, February, and March gets rolled up to Quarter 1 and the Quarter 1, 2, 3, and 4 data get rolled up to, say Year 2013. 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, three month rolling average).

This chapter covers 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

Time dimension population transformation is used to populate the DIM_DATES table with values between two dates specified by the user as a batch parameter.

The database components, used by the transformations are:

1. Database function FN_DIM_DATES
2. Database procedure PROC_DIM_DATES_POPULATION, which is called by the function FN_DIM_DATES.

Prerequisites

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 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:

    1. Iccserver

    2. Router

    3. AM Server

    4. Messageserver

    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 the section Executing the Time Dimension Population Transformation, page 4-2.

Tables Used by the Time Dimension Population Transformation

- DIM_DATES - This table stores the date details to be used for building cubes and for reporting.

For more details on viewing the structure of tables, refer to *Oracle Financial Services Analytical Applications Data Model Data Dictionary* or the *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, as mentioned below:

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 the section How to Define a Batch, page E-1.

Note: A seeded batch <INFODOM>_aCRM_CommonTasks - Task2 is

provided so that the user can just modify the parameters and execute the 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 drop down list
   - **Datastore Name** - Select the appropriate datastore name from the drop down list
   - **IP address** - Select the IP address from the list
   - **Rule Name** - Select **Dim_Dates_Population** from the list of all available transformations. (This is a seeded Data Transformation which is installed as part of the OFSRPA solution. If you don't see this in the list, contact Oracle support.)
   - **Parameter List** – Enter the **Start Date** and **End Date**.

     Explanation for the parameter list is:
     - **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.

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'

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 this path.

5

Customer Dimension Population

Overview

In the current setup, Customer Dimension is populated as part of Party Model, where DIM_CUSTOMER derives its attributes from DIM_PARTY based on the Parties that have been assigned the role of a Customer.

Populating Party Dimension

DIM_PARTY table will be populated first from stage table – STG_PARTY_MASTER using SCD. Function FN_PARTY_DENORMALIZE_DT will populate DIM_CUSTOMER from DIM_PARTY. The function is used to populate DIM_CUSTOMER table using a sequence.

The primary key for DIM_PARTY – N_PARTY_SKEY will be the surrogate key generated for the natural key - Party ID, an alphanumeric unique identifier within each staging instrument tables. This information is stored in DIM_CUSTOMER table as N_CUST_SKEY.

FSI_MERGE_SETUP_DETAILS

Customer dimension population makes use of setup table FSI_MERGE_SETUP_DETAILS. It would have seeded entries from the application installation. This table stores the mapping between source and target columns.

Column Name	Data Type	Column Description
MERGE_CODE	VARCHAR2 (50 CHAR)	This is the role of the party, that is, customer, issuer, and so on.

Column Name	Data Type	Column Description
SOURCE_TABLE	VARCHAR2 (30 CHAR)	This is the source table for Customer dimension population.
SOURCE_COLUMN	VARCHAR2 (30 CHAR)	This is the source column for Customer dimension population.
TARGET_COLUMN	VARCHAR2 (30 CHAR)	This is the target column for Customer dimension population.
DEFAULT_VALUE	VARCHAR2 (4000 CHAR)	This is the default value for some target columns.
NVL_EXPRESSION	VARCHAR2 (30 CHAR)	This is the nvl expression applied on the source column for Customer dimension population.
AGGREGATE_FUNCTION	VARCHAR2 (30 CHAR)	This is used for aggregating data for some source columns.

Here is a Sample Data:

MERGE_CODE	MI		
TABLE SOURCE	DIM_PARTY		
SOURCE_COLUMN	V_PARTY_ID	V_MIDDLE_NAME	V_LAST_NAME
TARGET_COLUMN	V_ISSUER_CODE	V_D_CUST_MIDDLE_NAME	V_D_CUST_LAST_NAME
DEFAULT_VALUE			
NVL_EXPRESSION			
AGGREGATE_FUNCTION			

FSI_MERGE_SETUP_MASTER

Customer dimension population makes use of setup table FSI_MERGE_SETUP_MASTER as well. It would have seeded entries from the application installation. This table stores the mapping between source and target tables.

Column Name	Data Type	Column Description
MERGE_CODE	VARCHAR2 (50 CHAR)	This is the role of the party, that is, customer, issuer, and so on.
SOURCE_TABLES	VARCHAR2 (4000 CHAR)	This is the list of source tables for Customer dimension population.
TARGET_TABLE	VARCHAR2 (30 CHAR)	This is the target column for Customer dimension population.
ANSI_JOIN	VARCHAR2 (4000 CHAR)	This is the join condition that results in dataset.
FILTER_CONDITION	VARCHAR2 (4000 CHAR)	This is used for filtering the values in where clause.

Here is a Sample Data:

MERGE_CODE	M1
SOURCE_TABLES	DIM_PARTY
TARGET_TABLE	DIM_CUSTOMER
ANSI_JOIN	
FILTER_CONDITION	

Executing the Customer Dimension Population

To execute the customer dimension population, create a batch by performing the

following steps:

1. From the **Home** menu, select **Operations**, then select **Batch Maintenance**.
2. Click **New Batch** ('+' symbol in Batch Name container) and enter the Batch Name and description.
3. Click **Save**.
4. Select the Batch you have created in the earlier step by clicking on the check box in the Batch Name container.
5. Click **New Task** ('+' symbol in Task Details container).
6. Enter the Task ID and Description.
7. Select **Transform Data** from the components list.
8. Select the following from the Dynamic Parameters List and then click **Save**:
   - Datastore Type - Select appropriate datastore from the list
   - Datastore Name - Select appropriate name from the list. Generally, it is the infodom name.
   - IP address - Select the IP address from the list
   - Rule Name - FN_PARTY_DENORMALIZE_DT
   - Parameter List: Surrogate Key Required Flag – Y or N

     Batch run ID and As of Date are passed internally by the ICC to the Data Transformation task.
9. Execute the batch.

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

   Note: A seeded batch<INFODOM> aCRM_CommonTasks – Task3 is provided so that the user can just modify the parameters and execute the batch.

Checking the Execution Status

The status of execution can be monitored from the *Batch Monitor* screen of OFSAAI Operations module.

Note: For a more comprehensive coverage of configuration & 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 Event Log window in Batch Monitor provides logs for execution with the top row being the most recent. If there is any error during execution, it will get listed here. Even if you see *Successful* as the status in Batch Monitor it is advisable to go through the Event Log and re-check if there are any errors. The execution log can be accessed on the application server by going to the directory*$FIC_DB_HOME/log/date*. The file name will have the Batch Execution ID.

The database level operations log can be accessed by querying the FSI_MESSAGE_LOG table. The batch run id column can be filtered for identifying the relevant log.

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

6

Account Dimension Population

Dimension Tables Population

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

Overview of SCD process

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

- Type 1 SCDs - Overwriting

  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
  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	N

  In this example, N_PRODUCT_SKEY is the surrogate key column which is a unique key for each record in the dimension table. V_PRODUCT_NAME is the product

name. D_START_DATE indicates the date from which this product record is valid. 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:

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

- 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:

N_PRODUCT_SKEY	V_PRODUCT_NAME	D_START_DATE	D_END_DATE	F_LATEST_RECORD_INDICATOR
1	PL	5/31/2010	12/31/9999	N
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 that 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/E14072_01/owb.112/e10935.pdf]

  Additional online sources include:

- http://en.wikipedia.org/wiki/Slowly_changing_dimension
- http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/10g/r2/owb/owb10 gr2_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=2 04800027 and page no.=1
- http://www.informationweek.com/news/software/bi/showArticle.jhtml?articleID=5 9301280

  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.

Prerequisites

Following are the prerequisites:

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

   SYS_TBL_MASTER stores the information like which is the source stage table and the target dimension tables. The source sometimes can be the database views which

could be simple or a complex view.

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

Tables Used by the SCD Component

The database tables used by the SCD component are:

- SYS_TBL_MASTER

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

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

Column Name	Data Type	Column Description
DT_OFFSET	NUMBER(2) NULL	The offset for calculating the Start Date based on the Functional Requirements Document (FRD).
SRC_KEY	NUMBER(3) NULL	

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

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

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

- SYS_STG_JOIN_MASTER

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

Column Name	Data Type	Column Description
MAP_REF_NUM	NUMBER(3) NOT NULL	The Mapping Reference Number for this unique mapping of a Source to a Dimension Table.
COL_NM	VARCHAR2(30) NOT NULL	Name of the column in the Dimension Table.
COL_TYP	VARCHAR2(30) NOT NULL	Type of column. The possible values are given in the following section.
STG_COL_NM	VARCHAR2(60) NOT NULL	Name of the column in the Staging Table.
SCD_TYP_ID	NUMBER(3) NOT NULL	SCD type for the column.
PRTY_LOOKUP_REQD_FLG	CHAR(1) NOT 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. CSK – Current Surrogate Key
9. SS – Source Key
10. LUD – Last Updated Date/Time
11. LUB – Last Updated By

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

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

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

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

Example
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.

```
create sequence SEQ_DIM_<DIM> minvalue 1
maxvalue 999999999999999999999999999
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.

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

- Datastore Type - Select the appropriate datastore from the list
- Datastore Name - Select the appropriate name from the list
- IP address - Select the IP address from the list
- Executable - scd, <map ref num>

Example
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.

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

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

Checking the Execution Status

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

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

Note: For a more comprehensive coverage, see *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

The status messages in Batch Monitor are:

- N - Not Started
- O - On Going
- F - Failure
- S – Success

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

The file name will have the batch execution id.

Sample

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

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

The file name will have the batch execution id.

Sample

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

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

Load DIM_ACCOUNT through SCD

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

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

- All the tasks can be executed in parallel. This might cause the N_RCV_LEG_ACCT_SKEY to have an incremental value as compared to N_ACCT_SKEY.
- Execute the SQL file with all the SESSION_DISABLE_STATEMENTs, after the successful completion of the SCD batch.
  - Once the DIM_ACCOUNT table is populated using this approach, you will not be able to use the initial approach (FN_POPDIMACCOUNT) as this will lead to skey conflict.
  - Ensure that you have set the value of the sequence seq_dim_account_scd as max (value of skey in DIM_ACCOUNT) + 1, before moving from old to new approach.
  - The F_LATEST_RECORD_INDICATOR for an existing DIM_ACCOUNT data already loaded by the function should be updated to 'Y' before running the SCD, failing which a new skey might get generated for the same account number.
  - SCD execution occurs based on the GAAP code which is configured in SETUP_MASTER table. This has been introduced to tackle the scenario of multiple GAAP codes. Whether or not there exist multiple GAAP codes, SETUP_MASTER should be manually configured as follows:

V_COMPONENT_CODE	V_COMPONENT_DESC	V_COMPONENT_VALUE
DEFAULT_GAAP	DEFAULT_GAAP	USGAAP

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

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

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

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

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

7

Exchange Rate History Population

Introduction

Exchange Rate History entity stores the exchange rates between the currencies for an effective date from one or multiple sources.

Exchange Rate History population should be executed before any fact table is populated to ensure exchange rates between currencies are available prior. Exchange Rate History entity is loaded by means of T2T Transformation process.

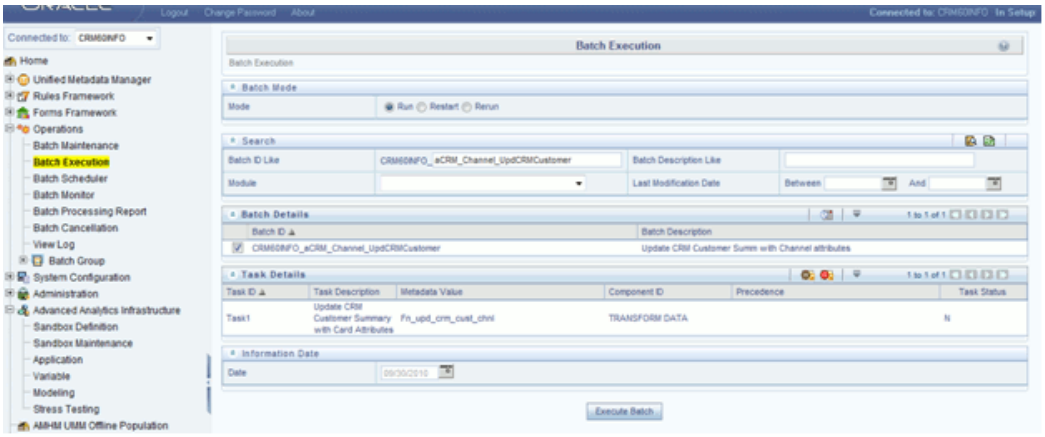
Following is the seeded T2T definition that loads data into Exchange Rate History:

T2T Definition Name	Source Table(s)	Destination Table
T2T_EXCHANGE_RATE_HIST	STG_EXCHANGE_RATE_HIST	FSI_EXCHANGE_RATE

Exchange Rate History Population

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

A seeded batch, **<INFODOM>_aCRM_CommonTasks - Task4** has to be executed for the required date.



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 the section How to Define a Batch, page E-1.

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 **Load 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 drop down list
   - **Datastore Name** - Select the appropriate datastore name from the drop down list
   - **IP address** - Select the IP address from the list
   - **Load Mode** - Select Table to Table from the list.
   - **Source Name** - Select the **T2T Source Name** from the list.
   - **File Name** - Select the T2T transformation **T2T_EXCHANGE_RATE_HIST**.

     Data file name remains blank for any Table to Table Load mode.
6. Click **Save**.

The Task definition is saved for the selected Batch.

You can execute a Batch definition from the *Batch Execution* section of *OFSAAI Operations* module.

Check T2T component logs and batch messages to check the status of load.

T2T component can fail because of following cases:

- **Unique Constraint Error** - Target table may already contain the primary keys that are part of the staging tables.
- **NOT NULL Constraint Error** - This error occurs when the transformation does not have values for NOT NULL columns in the target 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/t2t* , where file name will have the Batch Execution ID.

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 the <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.

8

Loading Multiple Load Runs in OFSAA

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 applications 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 application 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:** The load run is enabled only in the model and is defaulted to '0' in the model. This would not impact data previously available.

FSDF 8.0 staging model provides customers a flexibility to load multiple snapshots of the data in the staging tables (Product Processor's). A column named n_load_run_id was introduced as part of the primary key of the product processor tables to enable this. But the full fledged functionality to load and manage these snapshots will be part of the platform release at a later stage. Customers who would like to leverage this design in 8.0 release, the following mentioned changes will need to be done 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 need to be executed via Run Rule Framework and load needs to be filtered accordingly for each load run via the run filter. To enable execution of this run, navigate to $FIC_HOME/ficweb/webroot/conf and dit the file excludeURLList.cfg and add the following entry at the end of the file.

[SQLIA]./pr2

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

Features

Following are the features:

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

Design Details

Loading of data into OFSAA can be done 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. OFSAA data model also another entity that tracks the load run mapped to the functional key of each of the staging table. Since OFSAA processing is on snapshot of data, this entity helps users to identify set of records that are latest to be used in processing. If there is a need to load multiple sets of data within a day, customers can use the below components to manage the same.

If set of data is snapshot

- Register table that got loaded in the load run through a DT (Register_load_run_details)
- Register table that got loaded in the load run through a DT (Register_load_run_details)
- Use the load run identifier to load data into OFSAA staging. You can use the same load run identifier for all the entities loaded in the same batch/group.

- Specify run-filter during OFSAA execution to filter records for the maximum run identifier within the day

If set of data is incremental

- Register load run information in master table through a DT (Register_Load_Run_Master)
- Register table that got loaded in the load run through a DT (Register_load_run_details)
- Register table that got loaded in the load run through a DT (Register_load_run_details)
- Execute another DT (Populate_Load_Run_Map) with different parameters for each entity to maintain the latest load run for each record.
- Modify the T2T's to join with the load run map tables for identifying and filtering on the latest set of records to be used in processing.

There could be some entities that can follow snapshot and some entities incremental load. Decision of snapshot vs incremental depends on the above use-case.

Data Transformations

Function - Register Load Run

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

Steps

1. Check if the parameters are valid. Load run type can be 'B – Base, A – Adjustments, P – Backdated adjustments'.
2. Check uniqueness of load run name. Load run name is a user-specified string for easier retrievals could be MIS-DATE <Sequence> or the starting timestamp of load run.
3. Increment the load run id for a given MISDATE and insert the rest of the details.
4. Return the load run identifier (if possible, else user will query this table to get the load run id given a name)
5. Log messages accordingly
6. Return success/failure

Execution

Execute this DT before loading any fact for intra-day load. Use the registered load run identifier as a value to map to load run identifier field in staging. When one load run is ongoing (loading data to OFSAA using the load run identifier), do not register any new load runs. Else, make sure load run name is used as a filter instead of max load run identifier when querying the load run master table.

Function – Register Load Run Details

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

Steps:

1. Check if the parameters are valid. Load type can be 'S – Snapshot, I - Incremental'. Load run name or ID can be provided. If load run name is provided, we can lookup into load run master for retrieving the ID. Check if table name exist.
2. Register the information in load run details table
3. Log messages accordingly
4. Return success/failure

Execution

Execute this DT after registering load run master and before loading any fact for intra-day load. When one load run is ongoing (loading data to OFSAA using the load run identifier), do not register any new load runs.

Function – Populate Load Run Map

Parameters – batch id, mis-date, and stage table name

Steps:

1. Check if the parameters are valid.
2. Pick the corresponding load run map table from a setup table. Pick the corresponding functional key columns and their mapping to load run map table from a setup table.
3. If record do not exist in load run map table for the functional key in staging, then insert a new record with the functional key and load run identifier.
4. If record exist in load run map table for the functional key in staging, then update latest record indicator for existing rows to 'N' and then insert a new record with the functional key and load run identifier.

5. This operation has to be done in bulk mode.

6. Log messages accordingly

7. Return success/failure

Execution

Execute this DT after loading any fact for intra-day load in case the table has incremental loads.

9

Account Summary Population

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.

This chapter covers the following topics:

- Overview of Account Summary Tables
- Overview of Account Summary Population
- Prerequisites
- Executing the Account Summary Population T2T
- Checking the Execution Status
- Account Summary T2Ts

Overview of Account Summary Tables

Customer account level data from the Oracle Financial Services Analytical Applications (OFSAA) staging product processor tables must be consolidated into a standardized relational Business Intelligence (BI) data model. This consolidation is done to have all the staging product processor table data in a single Fact table.

The 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 three vertically partitioned Account Summary tables that are organized by application subject area.

- FCT_COMMON_ACCOUNT_SUMMARY - This table is shared by all OFSAA BI applications which contain dimensional values, attributes, and financial measures which are generally applicable to the individual account records. This data is sourced directly from the staging area.

- FCT_CRM_ACCOUNT_SUMMARY - This table has the measures used by all the

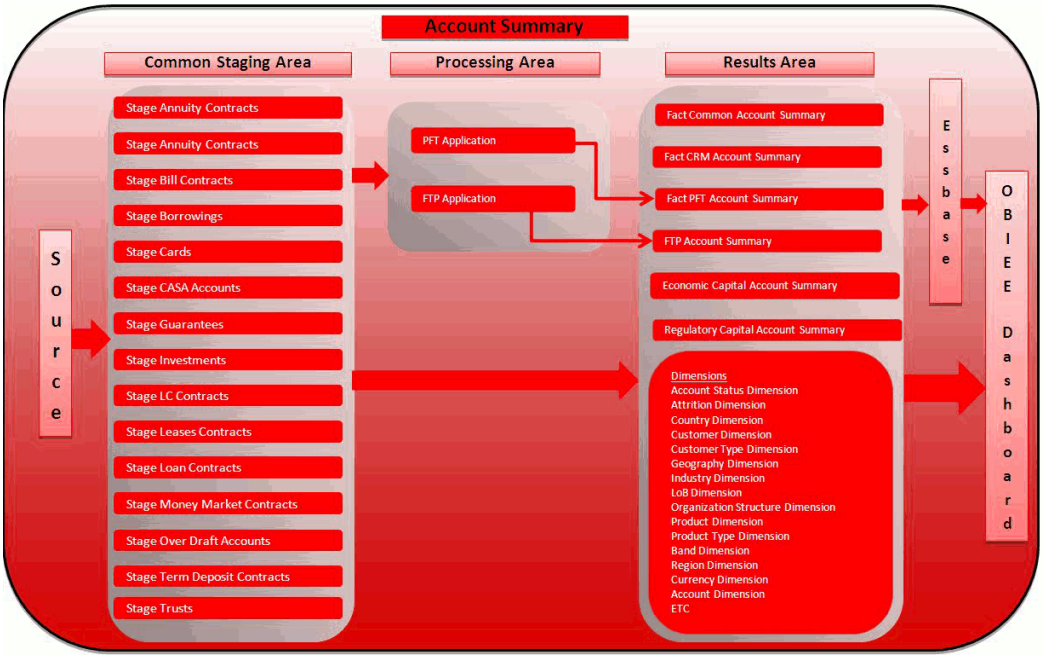
Customer Insight applications.

Yet, there are few other Account Summary tables which have been designed to store Enterprise Risk Management (ERM) data:

- FCT_PFT_ACCOUNT_SUMMARY - This table has Profitability Management (PFT) specific measures.
- FCT_FTP_ACCOUNT_SUMMARY - This table has Funds Transfer Pricing (FTP) specific measures.
- FCT_REG_CAP_ACCOUNT_SUMMARY - This table has Regulatory Capital specific measures.
- FCT_ECO_CAPITAL_ACCOUNT_SUMMARY - This table has Economic Capital specific measures.

Data Flow

The Below diagram depicts the flow of data into account summary tables:



Overview of Account Summary Population

Table to Table seeded definitions are provided for loading data into Common Account Summary and CRM Account summary tables.

Following are the lists for the same:

- **Common Account Summary**

SL No	Source Table	T2T Definition Name	Destination Table
1	STG_ANNUITY_CONTRACTS	T2T_STG_ANNUITY_CONTRACTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
2	STG_BILLS_CONTRACTS	T2T_STG_BILLS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
3	STG_BORROWINGS	T2T_STG_BORROWINGS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
4	STG_CARDS	T2T_STG_CARDS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
5	STG_CASA	T2T_STG_CASA_CAS	FCT_COMMON_ACCOUNT_SUMMARY
6	STG_GUARANTEES	T2T_STG_GUARANTEES_CAS	FCT_COMMON_ACCOUNT_SUMMARY
7	STG_INVESTMENTS	T2T_STG_INVESTMENTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
8	STG_LC_CONTRACTS	T2T_STG_LC_CAS	FCT_COMMON_ACCOUNT_SUMMARY
9	STG_LEASES_CONTRACTS	T2T_STG_LEASES_CONTRACTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
10	STG_LOAN_CONTRACTS	T2T_STG_LOANS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
11	STG_MM_CONTRACTS	T2T_STG_MM_CAS	FCT_COMMON_ACCOUNT_SUMMARY
12	STG_OD_ACCOUNTS	T2T_STG_OD_CAS	FCT_COMMON_ACCOUNT_SUMMARY
13	STG_TD_CONTRACTS	T2T_STG_TD_CONTRACTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY

SL No	Source Table	T2T Definition Name	Destination Table
14	STG_TRUSTS	T2T_STG_TRUSTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
15	STG_COMMITMENT_CONTRACTS	T2T_STG_COMMITMENT_CONTRACTS_CAS	FCT_COMMON_ACCOUNT_SUMMARY
16	STG_MUTUAL_FUNDS	T2T_STG_MUTUAL_FUNDS_CAS	FCT_COMMON_ACCOUNT_SUMMARY

- **CRM Account Summary**

Sl No.	Source Table	T2T Definition Name	Destination Table
1	STG_ANNUITY_CONTRACTS	T2T_STG_CRMAS_ANNUITY_CONTRACTS	FCT_CRM_ACCOUNT_SUMMARY
2	STG_BILLS_CONTRACTS	T2T_STG_CRMAS_BILLS_CONTRACTS	FCT_CRM_ACCOUNT_SUMMARY
3	STG_BORROWINGS	T2T_STG_CRMAS_BORROWINGS	FCT_CRM_ACCOUNT_SUMMARY
4	STG_CARDS	T2T_STG_CRMAS_CARDS	FCT_CRM_ACCOUNT_SUMMARY
5	STG_CASA	T2T_STG_CRMAS_CASA	FCT_CRM_ACCOUNT_SUMMARY
6	STG_GUARANTEES	T2T_STG_CRMAS_GUARANTEES	FCT_CRM_ACCOUNT_SUMMARY
7	STG_INVESTMENTS	T2T_STG_CRMAS_INVESTMENTS	FCT_CRM_ACCOUNT_SUMMARY
8	STG_LC_CONTRACTS	T2T_STG_CRMAS_LC_CONTRACTS	FCT_CRM_ACCOUNT_SUMMARY

Sl No.	Source Table	T2T Definition Name	Destination Table
9	STG_LEASES_CONTRACTS	T2T_STG_CRMAS_LEASES_CONTRACTS	FCT_CRM_ACCOUNT_SUMMARY
10	STG_LOAN_CONTRACTS	T2T_STG_CRMAS_LOAN_CONTRACTS	FCT_CRM_ACCOUNT_SUMMARY
11	STG_MM_CONTRACTS	T2T_STG_CRMAS_MM_CONTRACTS	FCT_CRM_ACCOUNT_SUMMARY
12	STG_OD_ACCOUNTS	T2T_STG_CRMAS_OD_ACCOUNTS	FCT_CRM_ACCOUNT_SUMMARY
13	STG_TD_CONTRACTS	T2T_STG_CRMAS_TD_CONTRACTS	FCT_CRM_ACCOUNT_SUMMARY
14	STG_TRUSTS	T2T_STG_CRMAS_TRUSTS	FCT_CRM_ACCOUNT_SUMMARY
15	STG_COMMITMENT_CONTRACTS	T2T_STG_CRMAS_TRUSTS	FCT_CRM_ACCOUNT_SUMMARY
16	STG_MUTUAL_FUNDS	T2T_STG_CRMAS_MUTUAL_FUNDS	FCT_CRM_ACCOUNT_SUMMARY

Note: Currency Exchange Rate History table has to be populated prior to loading the Account Summary tables.

Prerequisites

1. All the post install steps mentioned in the *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration guide* and the solution installation manuals of *Oracle Financial Services Retail Performance Analytics* 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.

1. Iccserver
2. Router
3. AM Server
4. Messageserver

   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 have to be created for executing. This is explained in the section Executing the Account Summary Population T2T, page 9-8.

5. Dimension Population should have been done before you execute the T2T batch.

   For more information, refer to the chapters Dimension Loading Process, and Time Dimension Population, page 4-1.

Fact Common Account Summary

Following are the lists of tables used in the population of Fact Common Account Summary and Fact CRM Account Summary tables. These Dimension tables are required to be loaded prior to executing the T2T:

- DIM_DATES
- DIM_ACCOUNT
- DIM_CUSTOMER
- DIM_PRODUCT
- DIM_CHANNEL
- DIM_BANDS
- DIM_ORG_STRUCTURE

Fact CRM Account Summary

Fact Common Account Summary entity needs to be populated before executing the Fact CRM Account Summary T2Ts.

Following are the list of tables used in the population of Fact CRM Account Summary and these tables are required to be loaded along with the staging tables, prior to running the T2T:

- DIM_DATES
- DIM_ACCOUNT
- FCT_COMMON_ACCOUNT_SUMMARY
- DIM_ACCT_STATUS
- DIM_BANDS
- DIM_CAMPAIGN
- DIM_CHANNEL
- DIM_CUSTOMER
- DIM_ORG_STRUCTURE
- DIM_LOB
- DIM_OFFER
- DIM_PRODUCT
- DIM_PROSPECT
- DIM_RETENTION_OFFER_TYPE
- DIM_SALES_REPRESENTATIVE
- DIM_TREATMENT
- DIM_VINTAGE

For details on populating dimension tables like DIM_CUSTOMER, DIM_BANDS, and so on, refer to the section Dimension Tables Population, under the chapter *Dimension Loading Process*.

For more information on populating account dimension, refer to the chapter Account Dimension Population, page 6-1.

For details on populating DIM_DATES dimension table, refer to the chapter Time Dimension Population, page 4-1.

For identifying fields required in Stage Customer Master and Stage Customer Details for the purpose of Customer Insight Application(s), refer to *Download Specification*.

For more information on the dimensions, refer to *ERwin Datamodel*.

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

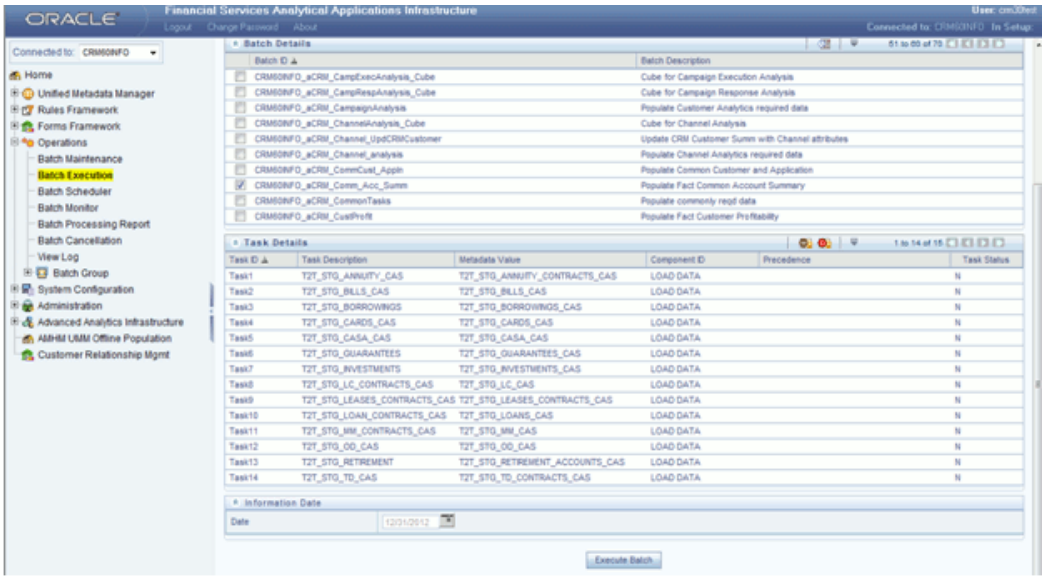
Fact Common Account Summary

A seeded batch, **<Infodom>_aCRM_Comm_Acc_Summ** has to be executed for the required MIS Date.

The Tasks associated with this batch are the following:

Batch Name - Task ID	T2T Name	Result
<INFODOM>_aCRM_Comm_Acc_Summ - Task1	T2T_STG_ANNUITY_CONTRACTS_CAS	Data from Stg_Annuity_Contracts has to be loaded in to Fct_Common_Account_Summary
<INFODOM>_aCRM_Comm_Acc_Summ - Task2	T2T_STG_BILLS_CAS	Data from STG_BILLS_CONTRACTS has to be loaded in to Fct_Common_Account_Summary
<INFODOM>_aCRM_Comm_Acc_Summ - Task3	T2T_STG_BORROWINGS_CAS	Data from STG_BORROWINGS has to be loaded in to Fct_Common_Account_Summary
<INFODOM>_aCRM_Comm_Acc_Summ - Task4	T2T_STG_CARDS_CAS	Data from Stg_Cards has to be loaded in to Fct_Common_Account_Summary
<INFODOM>_aCRM_Comm_Acc_Summ - Task5	T2T_STG_CASA_CAS	Data from Stg_CASA has to be loaded in to Fct_Common_Account_Summary
<INFODOM>_aCRM_Comm_Acc_Summ - Task6	T2T_STG_GUARANTEES_CAS	Data from Stg_Guarantees has to be loaded in to Fct_Common_Account_Summary
<INFODOM>_aCRM_Comm_Acc_Summ - Task7	T2T_STG_INVESTMENTS_CAS	Data from Stg_Investments has to be loaded in to Fct_Common_Account_Summary

Batch Name - Task ID	T2T Name	Result
<INFODOM>_aCRM_Comm_Acc_Summ - Task8	T2T_STG_LC_CAS	Data from STG_LC_CONTRACTS has to be loaded in to Fct_Common_Account_Summary
<INFODOM>_aCRM_Comm_Acc_Summ - Task9	T2T_STG_LEASES_CONTRACTS_CAS	Data from STG_LEASES_CONTRACTS has to be loaded in to Fct_Common_Account_Summary
<INFODOM>_aCRM_Comm_Acc_Summ - Task10	T2T_STG_LOANS_CAS	Data from STG_LOAN_CONTRACTS has to be loaded in to Fct_Common_Account_Summary
<INFODOM>_aCRM_Comm_Acc_Summ - Task11	T2T_STG_MM_CAS	Data from STG_MM_CONTRACTS has to be loaded in to Fct_Common_Account_Summary
<INFODOM>_aCRM_Comm_Acc_Summ - Task12	T2T_STG_OD_CAS	Data from STG_OD_ACCOUNTS has to be loaded in to Fct_Common_Account_Summary
<INFODOM>_aCRM_Comm_Acc_Summ - Task13	T2T_STG_TD_CONTRACTS_CAS	Data from STG_TD_CONTRACTS has to be loaded in to Fct_Common_Account_Summary
<INFODOM>_aCRM_Comm_Acc_Summ - Task14	T2T_STG_TRUSTS_CAS	Data from STG_TRUSTS has to be loaded in to Fct_Common_Account_Summary
##INFODOM##_aCRM_Comm_Acc_Summ - Task16	T2T_STG_COMMITMENT_CONTRACTS_CAS	
##INFODOM##_aCRM_Comm_Acc_Summ - Task17	T2T_STG_MUTUAL_FUNDS_CAS	



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 the section How to Define a Batch, page E-1.

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 **Load 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 drop down list
   - **Datastore Name** - Select the appropriate datastore name from the drop down 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.

Data file name remains blank for any T2T 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.

6. Click **Save**.

   The Task definition is saved for the selected Batch.

You can execute a Batch definition from the *Batch Execution* section of *OFSAAI Operations* module.

Fact CRM Account Summary

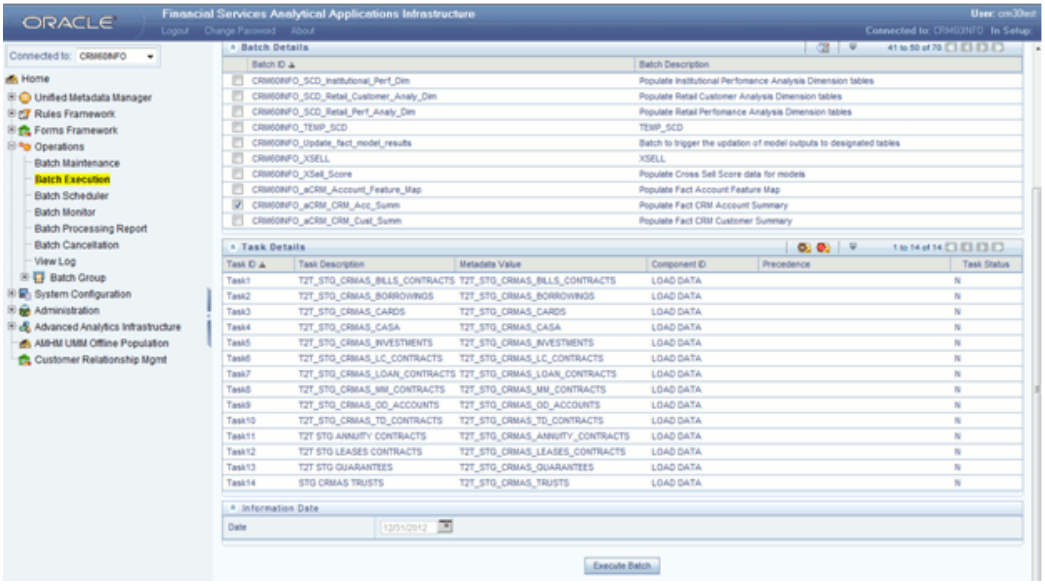
A seeded batch, **<Infodom>_aCRM_CRM_Acc_Summ** has to be executed for the required MIS Date.

The Tasks associated with this Batch are:

Batch Name - Task ID	T2T Name	Result
<INFODOM>_aCRM_CRM_Acc_Summ - Task1	T2T_STG_CRMAS_BILLS_CONTRACTS	Data from STG_BILLS_CONTRACTS has to be loaded in to FCT_CRM_ACCOUNT_SUMMARY
<INFODOM>_aCRM_CRM_Acc_Summ - Task2	T2T_STG_CRMAS_BORROWINGS	Data from STG_BORROWINGS has to be loaded in to FCT_CRM_ACCOUNT_SUMMARY
<INFODOM>_aCRM_CRM_Acc_Summ - Task3	T2T_STG_CRMAS_CARDS	Data from Stg_Cards has to be loaded in to FCT_CRM_ACCOUNT_SUMMARY
<INFODOM>_aCRM_CRM_Acc_Summ - Task4	T2T_STG_CRMAS_CASA	Data from Stg_CASA has to be loaded in to FCT_CRM_ACCOUNT_SUMMARY
<INFODOM>_aCRM_CRM_Acc_Summ - Task5	T2T_STG_CRMAS_INVESTMENTS	Data from Stg_Investments has to be loaded in to FCT_CRM_ACCOUNT_SUMMARY

Batch Name - Task ID	T2T Name	Result
<INFODOM>_aCRM_CRM_Acc_Summ - Task6	T2T_STG_CRMAS_LC_CONTRACTS	Data from STG_LC_CONTRACTS has to be loaded in to FCT_CRM_ACCOUNT_SUMMARY
<INFODOM>_aCRM_CRM_Acc_Summ - Task7	T2T_STG_CRMAS_LOAN_CONTRACTS	Data from STG_LOAN_CONTRACTS has to be loaded in to FCT_CRM_ACCOUNT_SUMMARY
<INFODOM>_aCRM_CRM_Acc_Summ - Task8	T2T_STG_CRMAS_MM_CONTRACTS	Data from STG_MM_CONTRACTS has to be loaded in to FCT_CRM_ACCOUNT_SUMMARY
<INFODOM>_aCRM_CRM_Acc_Summ - Task9	T2T_STG_CRMAS_OD_ACCOUNTS	Data from STG_OD_ACCOUNTS has to be loaded in to FCT_CRM_ACCOUNT_SUMMARY
<INFODOM>_aCRM_CRM_Acc_Summ - Task10	T2T_STG_CRMAS_TD_CONTRACTS	Data from STG_TD_CONTRACTS has to be loaded in to FCT_CRM_ACCOUNT_SUMMARY
<INFODOM>_aCRM_CRM_Acc_Summ - Task11	T2T_STG_CRMAS_ANNUITY_CONTRACTS	Data from Stg_Annuity_Contracts has to be loaded in to FCT_CRM_ACCOUNT_SUMMARY
<INFODOM>_aCRM_CRM_Acc_Summ - Task12	T2T_STG_CRMAS_LEASES_CONTRACTS	Data from STG_LEASES_CONTRACTS has to be loaded in to FCT_CRM_ACCOUNT_SUMMARY
<INFODOM>_aCRM_CRM_Acc_Summ - Task13	T2T_STG_CRMAS_GUARANTEES	Data from Stg_Guarantees has to be loaded in to FCT_CRM_ACCOUNT_SUMMARY
<INFODOM>_aCRM_CRM_Acc_Summ - Task14	T2T_STG_CRMAS_TRUSTS	Data from STG_TRUSTS has to be loaded in to FCT_CRM_ACCOUNT_SUMMARY

Batch Name - Task ID	T2T Name	Result
##INFODOM##_aCRM_CRM_Acc_Summ - Task15	T2T_STG_CRMAS_TRUSTS	
##INFODOM##_aCRM_CRM_Acc_Summ - Task17	T2T_STG_CRMAS_MUTUAL_FUNDS	



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 the section How to Define a Batch, page E-1.

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 **Load 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 drop down list
   - **Datastore Name** - Select the appropriate datastore name from the drop down 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.

   Data file name remains blank for any T2T 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.

6. Click **Save**.

   The Task definition is saved for the selected Batch.

You can execute a Batch definition from the *Batch Execution* section of *OFSAAI Operations* module.

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 be accessed on the application server in the directory *$FIC_DB_HOME/log/t2t*, where the file name will have the Batch Execution ID.

The following tables can be queried for errors:

- FCT_COMMON_ACCOUNT_SUMMARY$

- FCT_CRM_ACCOUNT_SUMMARY$

Account Summary T2Ts

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

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

10

Fact Transaction Summary

Overview

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.

Table to Table

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

Sl No.	Source Table Name	T2T Definition Name	Target Table Name
1	STG_TRUSTS_TXNS	STG_TRUSTS_TXNS_FT S	FCT_TRANSACTIO N_SUMMARY
2	STG_ANNUITY_TX NS	STG_ANNUITY_TX N_F TS	FCT_TRANSACTIO N_SUMMARY
3	STG_BILL_CONTRA CTS_ TXNS	STG_BILL_CONTRA CTS_TXNS_FTS	FCT_TRANSACTIO N_SUMMARY
4	STG_BORROWING_ COM MITMENT_TXNS	STG_BORROWING_ COMMITMENT_TX NS_FTS	FCT_TRANSACTIO N_SUMMARY
5	STG_BORROWINGS _TXNS	STG_BORROWINGS _TXNS_FTS	FCT_TRANSACTIO N_SUMMARY

Sl No.	Source Table Name	T2T Definition Name	Target Table Name
6	STG_CARDS_PAYMENT_ TXNS	STG_CARDS_PAYMENT_TXNS_FTS	FCT_TRANSACTION_SUMMARY
7	STG_CARDS_SETTLEMENT_TXNS	STG_CARDS_SETTLEM ENT_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
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

Sl No.	Source Table Name	T2T Definition Name	Target Table Name
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_OD_ACCOUNTS_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
29	STG_RETIREMENT_ACCOUNTS_TXNS	STG_RETIREMENT_ACCOUNTS_TXNS_FTS	FCT_TRANSACTION_SUMMARY
30	STG_SALAM_TXNS	STG_SALAM_TXNS_FTS	FCT_TRANSACTION_SUMMARY
31	STG_SUKUK_TXNS	STG_SUKUK_TXNS_FTS	FCT_TRANSACTION_SUMMARY
32	STG_SWAP_ACCOUNT_TXNS	STG_SWAP_ACCOUNT _TXNS_FTS	FCT_TRANSACTION_SUMMARY

Sl No.	Source Table Name	T2T Definition Name	Target Table Name
33	STG_TERMDEPOSITS_TXNS	STG_TERMDEPOSITS_TXNS_FTS	FCT_TRANSACTION_SUMMARY
34	STG_TRADING_ACCOUNT_TXNS	STG_TRADING_ACCOUNT_TXNS_FTS	FCT_TRANSACTION_SUMMARY
35	STG_FUTURES_TXNS	STG_FUTURES_TXNS_F TS	FCT_TRANSACTION_SUMMARY
36	STG_MUDARABAH_TXNS	STG_MUDARABAH_TXNS_FTS	FCT_TRANSACTION_SUMMARY

Executing the Fact Transaction Summary

Fact Transaction 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_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.

11

Customer Summary Population

This chapter explains the process flow for populating Fact Common Customer Summary table.

This chapter covers the following topics:

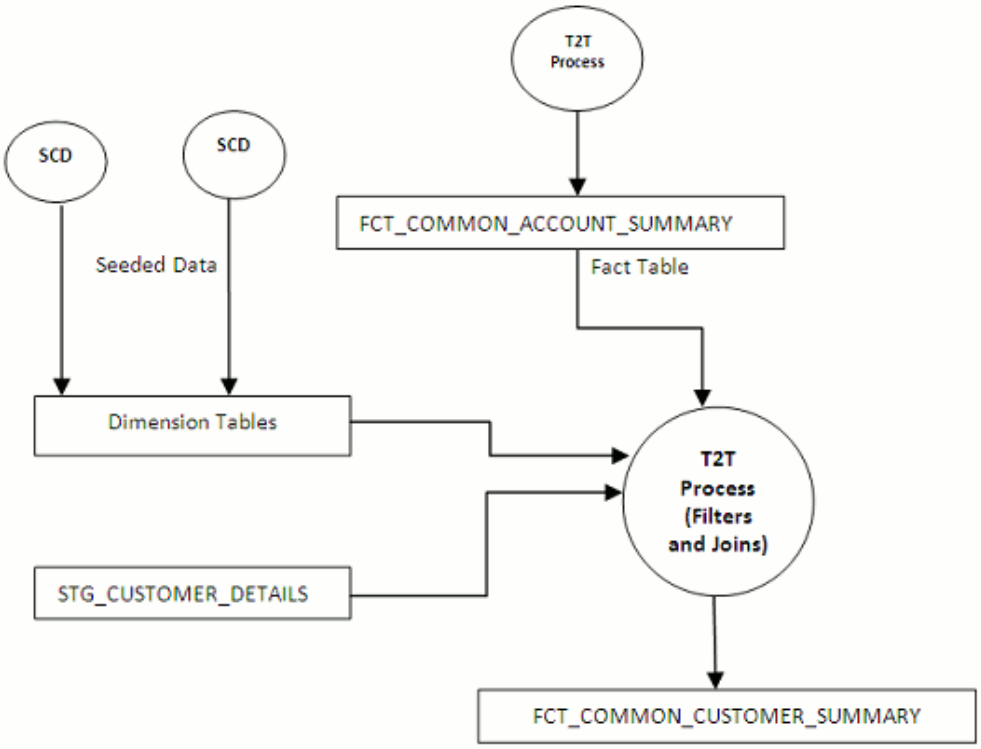
- Overview of Common Customer Summary Tables
- Prerequisites
- Executing the Customer Summary Population T2T

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.

Following data flow diagram explains the process flow for populating Fact Common Customer Summary table:



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_CUSTOMER
- DIM_BANDS
- DIM_EDUCATION
- DIM_CUSTOMER_TYPE
- DIM_GENDER
- DIM_INDUSTRY
- DIM_CHANNEL
- DIM_GEOGRAPHY
- DIM_MARITAL_STATUS

- DIM_MANAGEMENT
- DIM_PROFESSION
- DIM_CREDIT_RATING
- DIM_VINTAGE
- DIM_MIGRATION_REASONS
- FCT_COMMON_ACCOUNT_SUMMARY
- FCT_LIMITS_SUMMARY
- STG_CUSTOMER_DETAILS
- STG_PARTY_RATING_DETAILS
- STG_PARTY_FINANCIALS

Dimensions tables are loaded through the SCD process. The fact tables such as FCT_COMMON_ACCOUNT_SUMMARY is loaded from their respective T2T processes.

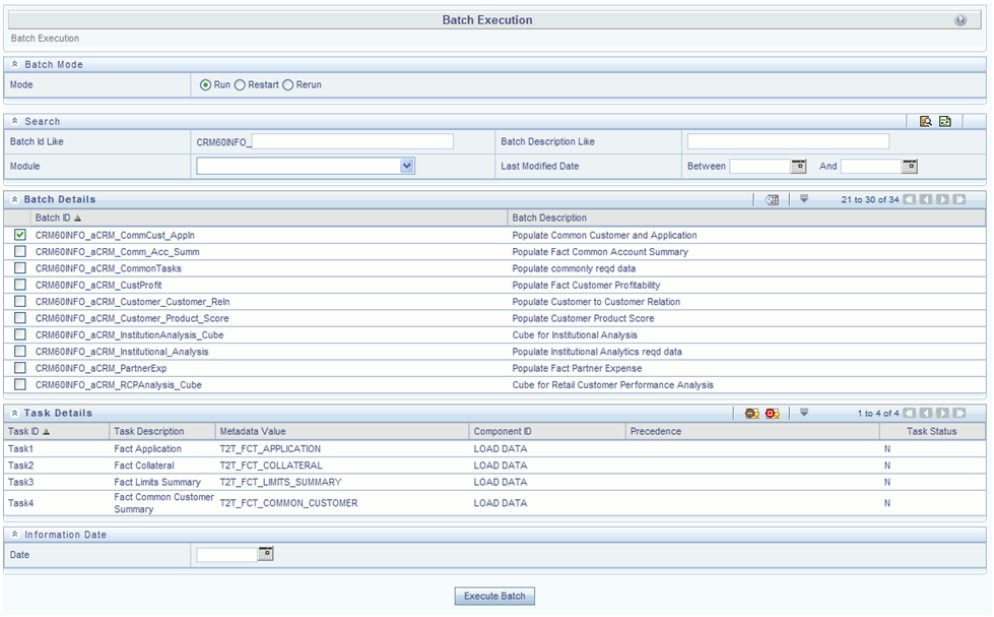
For more information on SCDs, refer to the chapter Dimension Loading Process, .

Executing the Customer Summary Population T2T

Fact Common Customer Summary T2T can be executed by executing **Task 4** - Fact Common Customer Summary, present in the seeded Batch **<INFODOM>_aCRM_CommCust_Appln**.

Perform the following to execute the Batch:

1. Navigate to *OFSAAI Home* > *Operations* > *Batch Execution* section.
2. Select the **Task 4 - Fact Common Customer Summary** of the seeded Batch **<INFODOM>_aCRM_CommCust_Appln**, where INFODOM is the information domain on which the 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. You can monitor the status of the Batch execution from the *Batch Monitor* section.



Error Messages

Following is 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 to load or reload existing records for any of the already executed AS_OF_DATE.

12

Fact Data Population

Introduction

This chapter explains all the fact tables which within describe about the seeded T2T Definitions with related Source Table and Destination tables. Prerequisites needed in population of the Fact table and tables required to be loaded prior to running the T2T.

Each fact table contains a section on how to execute the T2T component from OFSAA Infrastructure ICC framework and access the execution log to check the execution status.

Fact CRM Customer Summary

Fact CRM Customer Summary entity captures different derived/computed customer attributes pertaining to Customer Insight. Fact Common Customer Summary stores the generic application-agnostic source/raw customer attributes. Fact CRM Customer Summary is a vertical partitioned entity and has relationship to Fact Common Customer Summary.

Load Data into Fact CRM Customer Summary

Customer balances in the Fact CRM Customer Summary entity are derived from account summary. Customer relationship entity drives the relationship between accounts and customers.

Following is the seeded Table-to-Table definitions that loads data related to Fact CRM Customer Summary:

T2T Definition Name	Source Table(s)	Destination Table
T2T_FCT_CRM_CUSTOMER_	STG_CUSTOMER_MASTER	FCT_CRM_CUSTOMER_S

T2T Definition Name	Source Table(s)	Destination Table
SUMMARY	STG_CUSTOMER_DETAILS	UMMARY
	DIM_DATES	
	DIM_CUSTOMER	
	FCT_COMMON_ACCOUNT_ SUMMARY	
	FCT_CRM_ACCOUNT_SUM MARY	

To view the detailed structure of the tables, refer to *Oracle Financial Services Analytical Applications Data Model Data Dictionary* or the *Erwin Data Model*.

Prerequisites

Fact Common Customer Summary entity needs to be populated before executing the Fact CRM Customer Summary T2T. Refer to the chapter Fact Common Account Summary, page 9-8 for details on Fact Common Customer Summary T2Ts.

Following tables are used in the population of Fact CRM Customer Summary and these tables should be populated with relevant data prior to executing the T2T:

- STG_CUSTOMER_MASTER - Mandatory
- STG_CUSTOMER_DETAILS - Mandatory
- DIM_DATES - Mandatory
- DIM_CUSTOMER - Mandatory
- FCT_COMMON_ACCOUNT_SUMMARY - Mandatory
- FCT_CRM_ACCOUNT_SUMMARY - Mandatory
- DIM_BANDS - Optional

For details on populating dimension tables like DIM_CUSTOMER, DIM_BANDS, and so on, refer to the section Dimension Tables Population, under the chapter *Dimension Loading Process*.

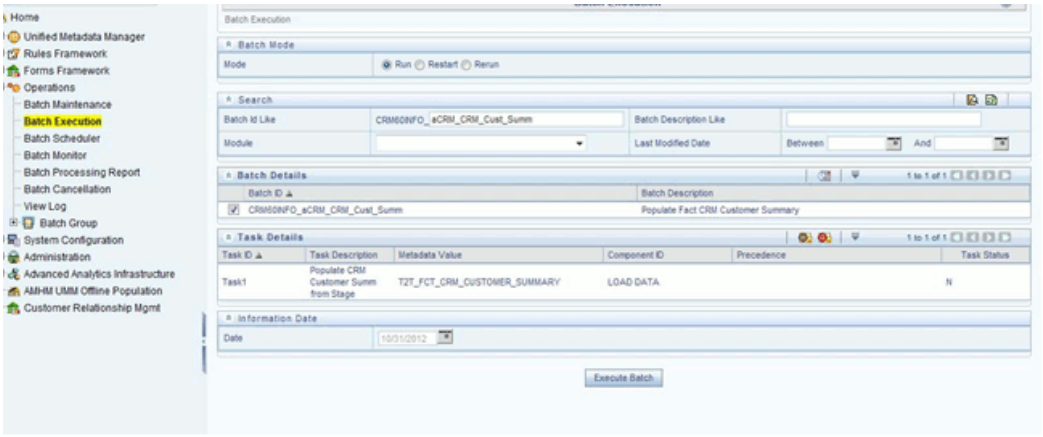
For details on populating DIM_DATES dimension table, refer to the chapter Time Dimension Population, page 4-1.

For identifying fields required in Stage Customer Master and Stage Customer Details for the purpose of Customer Insight Application(s), refer to *Download Specification*.

For details on populating these fact tables, refer to the sections Population of Fact CRM Customer Summary, page 12-1 and Fact CRM Account Summary, page 9-11.

Executing the Fact CRM Customer Summary Population T2Ts

You can execute the function from the *Operations* (formerly Information Command Center (ICC) framework) module of OFSAAI. A seeded batch, **<Infodom>_aCRM_CRM_Cust_Summ** has to be executed for the required MIS Date.



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 **Load 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 drop down list
   - **Datastore Name** - Select the appropriate datastore name from the drop down list
   - **IP address** - Select the IP address from the list
   - **Load Mode** - Select **Table to Table** from the list.

- **Source Name** - Select the **<T2T Source Name>** from the list.
- **File Name** - Select the T2T name "T2T_FCT_CRM_CUSTOMER_SUMMARY" to process.

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

Default value refers to any parameter that has to be passed to T2T. It has to be blank.

6. Click **Save**.

   The Task definition is saved for the selected Batch.

You can execute a Batch definition from the *Batch Execution* section of *OFSAAI Operations* module.

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 be accessed on the application server in the directory *$FIC_DB_HOME/log/t2t*, where the file name will have the Batch Execution ID.

The following tables can be queried for errors:

- FCT_CRM_CUSTOMER_SUMMARY

Update Fact CRM Customer Summary with Transaction Attributes

A seeded Data Transformation is provided with the installer which updates the entity Fact CRM Customer Summary with transaction attributes of customer such as ATM usage, Branch usage, net usage, POS usage, Number of ATM transactions, transacted amount, and so on.

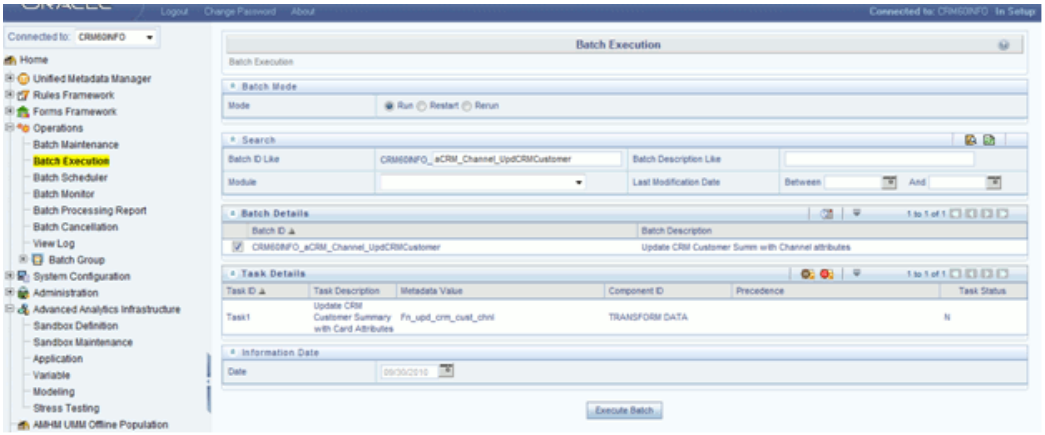
The following table lists the seeded Post Load Transformation Definition with related Source Table and Destination tables:

DT Definition Name	Source Tables	Destination Table
FN_UPD_CRM_CUST_CHNL	FCT_TXN_CHANNEL	FCT_CRM_CUSTOMER_SUMMARY

You can execute the function from the *Operations* (formerly Information Command Center (ICC) framework) module of OFSAAI, as mentioned below:

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 the section How to Define a Batch, page E-1.

A seeded batch, **<Infodom>_aCRM_Channel_UpdCRMCustomer** has to be executed for the required MIS Date.



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.

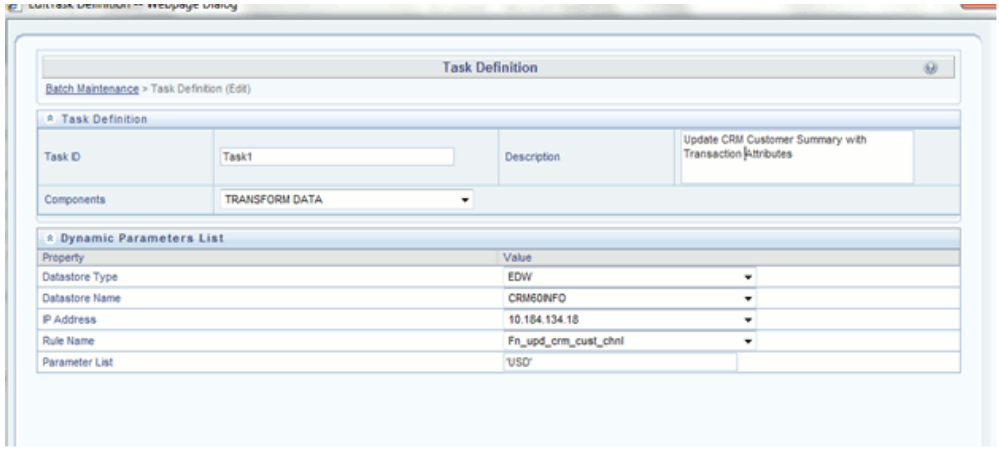
   The Task Definition window is displayed.
3. Enter the **Task ID** and **Description**.
4. Select **TRANSFORM DATA** component from the drop down list.
5. Select the following from the **Dynamic Parameters** list:
   - **Datastore Type** - Select the appropriate datastore type from the drop down list
   - **Datastore Name** - Select the appropriate datastore name from the drop down

list

- **IP address** - Select the IP address from the list
- **Rule Name** - Select **FN_UPD_CRM_CUST_CHNL** from the drop down list.
- Enter the Parameter List details as mentioned below:
  - **Reload Account Profitability table for the given MIS Date flag** - Enter Y or N within single quotes.
  - **Reporting Currency code** - This has to be enclosed within single quotes.

    For Example, if reporting currency is in US Dollar, then '**USD**' has to be specified.

    Note: Batch run ID and **As Of Date** are passed internally by the batch to the Data Transformation task.

6. 

7. Click **Save**.

   The Task definition is saved for the selected Batch.

You can execute a Batch definition from the *Batch Execution* section of *OFSAAI Operations* module.

Checking the Execution Status

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

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/date.

The file name will have the batch execution id.

> **Note:** For more information on configuration and execution of a batch, refer to *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

Fact Account Feature Map

A product might be facilitated with its own features. Fact Account Feature Map entity stores the mapping between the Account and Product Feature that is the features of the product availed by the customer account. Product processor tables in staging have information related to customer accounts.

Following table lists the seeded T2T Definitions with related Source Table and Destination tables:

T2T Definition Name	Source Staging Table	Destination Table
T2T_FCT_ACCOUNT_FEATURE_MAP	STG_ACCT_FEATURE_MAP	FCT_ACCOUNT_FEATURE_MAP

To view the detailed structure of the tables, refer to *Erwin Data Model*.

Prerequisites

Following are the lists of tables used in the population of Fact Account Feature Map and these tables are required to be loaded prior to executing the T2T:

- DIM_DATES
- DIM_PRODUCT_FEATURE
- DIM_ACCOUNT
- DIM_CUSTOMER

- DIM_PRODUCT
- DIM_VENDOR
- DIM_CAMPAIGN
- DIM_CHANNEL

For details on populating dimension tables like DIM_CUSTOMER, DIM_BANDS, and so on refer to the section Dimension Tables Population, under the chapter *Dimension Loading Process*.

For details on populating DIM_DATES dimension table, refer to the chapter Time Dimension Population, page 4-1.

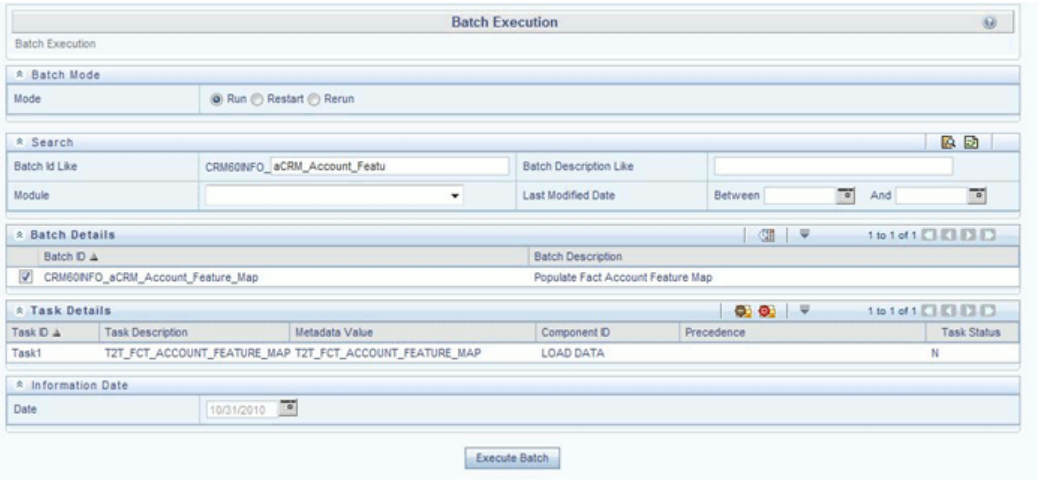
For identifying fields required in Stage Customer Master and Stage Customer Details for the purpose of Customer Insight Application(s), refer to *Download Specification*.

Executing the Fact Account Feature Map Population T2T

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

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 the section How to Define a Batch, page E-1.

A seeded batch, **<Infodom>_aCRM_Account_Feature_Map** has to be executed for the required MIS Date.



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 **Load 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 drop down list
   - **Datastore Name** - Select the appropriate datastore name from the drop down 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 "T2T_FCT_ACCOUNT_FEATURE_MAP" you want to process.

     Data file name remains blank for any T2T Load mode.

     Default value refers to any parameter that has to be passed to T2T. This should be blank.

6. Click **Save**.

   The Task definition is saved for the selected Batch.

You can execute a Batch definition from the *Batch Execution* section of *OFSAAI Operations* module.

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 be accessed on the application server in the directory *$FIC_DB_HOME/log/t2t*, where the file name will have the Batch Execution ID.

The following tables can be queried for errors:

- FCT_ACCOUNT_FEATURE_MAP$

Fact Cards Balance Summary

Fact Cards Balance Summary entity stores the balance details across various balance categories like Balance, Interest rate, Current payment, and others for each card account.

Following table lists the seeded T2T Definitions with related Source Table and Destination tables:

T2T Definition Name	Source Staging Table	Destination Table
T2T_FCT_CARDS_BALANCE_SUMM	STG_CARDS_BALANCE_SUMMARY	FCT_CARDS_BALANCE_SUMMARY

To view the detailed structure of the tables, refer to *Erwin Data Model*.

Prerequisites

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

- DIM_DATES
- DIM_BALANCE_CATEGORY
- DIM_ACCOUNT
- STG_CARDS_BALANCE_SUMMARY

For details on populating dimension tables like DIM_CUSTOMER, DIM_BANDS, and so on refer to the section Dimension Tables Population, under the chapter *Dimension Loading Process*.

For details on populating DIM_DATES dimension table, refer to the chapter Time Dimension Population, page 4-1.

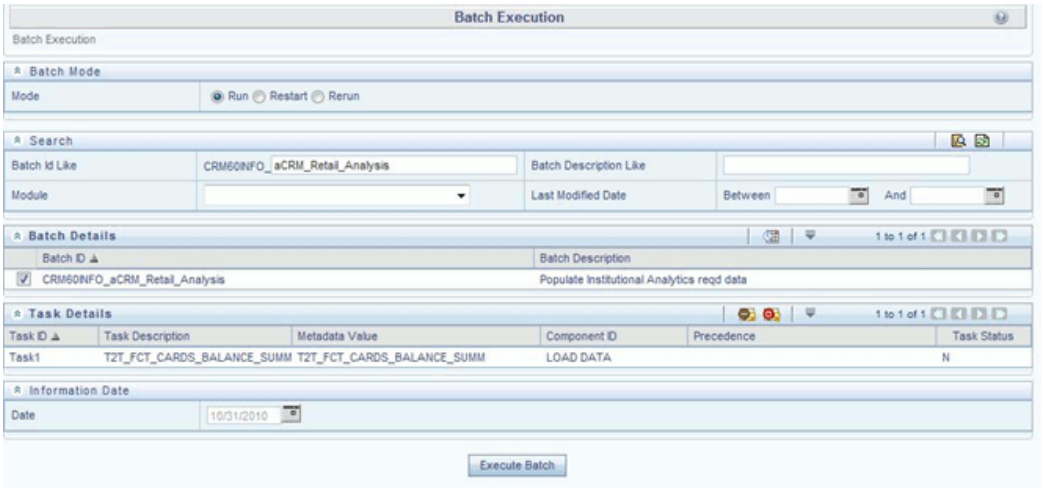
For identifying fields required in Stage Customer Master and Stage Customer Details for the purpose of Customer Insight Application(s), refer to *Download Specification*.

Executing the Fact Cards Balance Summary Population T2T

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

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 the section How to Define a Batch, page E-1.

A seeded batch, **<Infodom>_aCRM_Retail_Analysis - Task1** has to be executed for the required MIS Date.



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 **Load 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 drop down list
   - **Datastore Name** - Select the appropriate datastore name from the drop down 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 "T2T_FCT_CARDS_BALANCE_SUMM" you want to process.

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

  Default value refers to any parameter that has to be passed to T2T. This should be blank.

6. Click **Save**.

   The Task definition is saved for the selected Batch.

You can execute a Batch definition from the *Batch Execution* section of *OFSAAI Operations* module.

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 be accessed on the application server in the directory *$FIC_DB_HOME/log/t2t*, where the file name will have the Batch Execution ID.

The following tables can be queried for errors:

- FCT_ CARDS_BALANCE_SUMMARY$

Fact Customer to Customer Relationship

Fact Customer to Customer Relationship entity stores the relationship between the customers. Example of relationship amongst customers could be Employer, Employee, Children, Parent, Spouse, and so on.

Following table lists the seeded T2T Definitions with related Source Table and Destination tables:

T2T Definition Name	Source Staging Table	Destination Table
T2T_CUST_CUST_RELATION	STG_CUST_CUST_RELATIONSHIP	FCT_CUST_CUST_RELATIONSHIP

To view the detailed structure of the tables, refer to *Erwin Data Model*.

Prerequisites

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

- DIM_DATES
- DIM_CUSTOMER
- STG_CUST_CUST_RELATIONSHIP

For details on populating dimension tables like DIM_CUSTOMER, DIM_BANDS, and so on refer to the section Dimension Tables Population, under the chapter *Dimension Loading Process*.

For details on populating DIM_DATES dimension table, refer to the chapter Time Dimension Population, page 4-1.

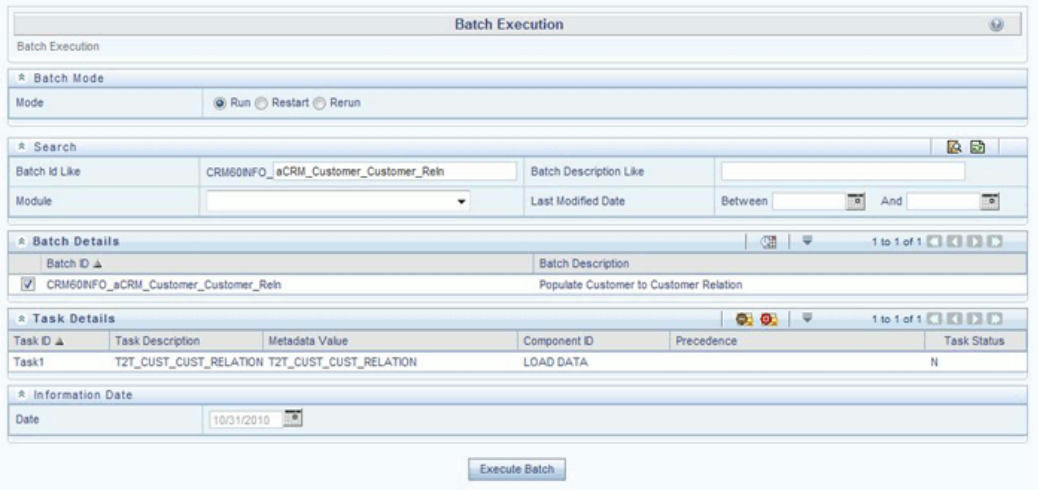
For identifying fields required in Stage Customer Master and Stage Customer Details for the purpose of Customer Insight Application(s), refer to *Download Specification*.

Executing the Fact Customer to Customer Relationship Population T2T

You can execute the T2T component from the *Operations* (formerly Information Command Center (ICC) framework) module of OFSAAI.

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 the section How to Define a Batch, page E-1.

A seeded batch, **<Infodom>_aCRM_Customer_Customer_Reln - Task1** has to be executed for the required MIS Date.



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 **Load 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 drop down list
   - **Datastore Name** - Select the appropriate datastore name from the drop down 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 "T2T_CUST_CUST_RELATION" you want to process.

     Data file name remains blank for any T2T Load mode.

     Default value refers to any parameter that has to be passed to T2T. This should be blank.

6. Click **Save**.

   The Task definition is saved for the selected Batch.

You can execute a Batch definition from the *Batch Execution* section of *OFSAAI Operations* module.

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 be accessed on the application server in the directory *$FIC_DB_HOME/log/t2t*, where the file name will have the Batch Execution ID.

The following tables can be queried for errors:

- FCT_CUST_CUST_RELATIONSHIP$

Fact Transaction Channel

Fact Transaction Channel entity stores the details of all transactions (successful and failed) done through any of the transaction channels offered by the Financial Institutions. This fact entity is loaded from multiple source staging tables.

The following table lists the seeded T2T Definitions with related Source Table and Destination tables:

T2T Definition Name	Source Staging Table	Destination Table
T2T_TEL_FCT_TXN_CHANNEL	STG_SRC_TB_TXNS	FCT_TXN_CHANNEL
T2T_POS_FCT_TXN_CHANNEL	STG_SRC_POS_TXNS	
T2T_NET_FCT_TXN_CHANNEL	STG_SRC_NET_TXNS	

T2T Definition Name	Source Staging Table	Destination Table
T2T_BRA_FCT_TXN_CHANNEL	STG_SRC_BRANCH_TXNS	
T2T_ATM_FCT_TXN_CHANNEL	STG_SRC_ATM_TXNS	

To view the detailed structure of these tables, refer to *Erwin Data Model*.

Prerequisites

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

- DIM_DATES
- DIM_TXN_CHANNEL
- DIM_ACCOUNT
- DIM_AUTH_DECISION_REASONS
- DIM_BANDS
- DIM_BROWSER_TYPE
- DIM_CARD_TYPE
- DIM_CURRENCY
- DIM_CUSTOMER
- DIM_CUSTOMER_TYPE
- DIM_GEOGRAPHY
- DIM_MERCHANT
- DIM_MERCHANT_CATEGORY
- DIM_PRODUCT
- DIM_TERMINAL
- DIM_TERMINAL_TYPE

- DIM_TRANSACTION
- DIM_TXN_FAILURE_REASON
- DIM_TXN_STATUS
- STG_SRC_ATM_TXNS
- STG_SRC_BRANCH_TXNS
- STG_SRC_NET_TXNS
- STG_SRC_POS_TXNS
- STG_SRC_TB_TXNS

For details on populating dimension tables like DIM_CUSTOMER, DIM_BANDS, and so on refer to the section Dimension Tables Population, under the chapter *Dimension Loading Process*.

For details on populating DIM_DATES dimension table, refer to the chapter Time Dimension Population, page 4-1.

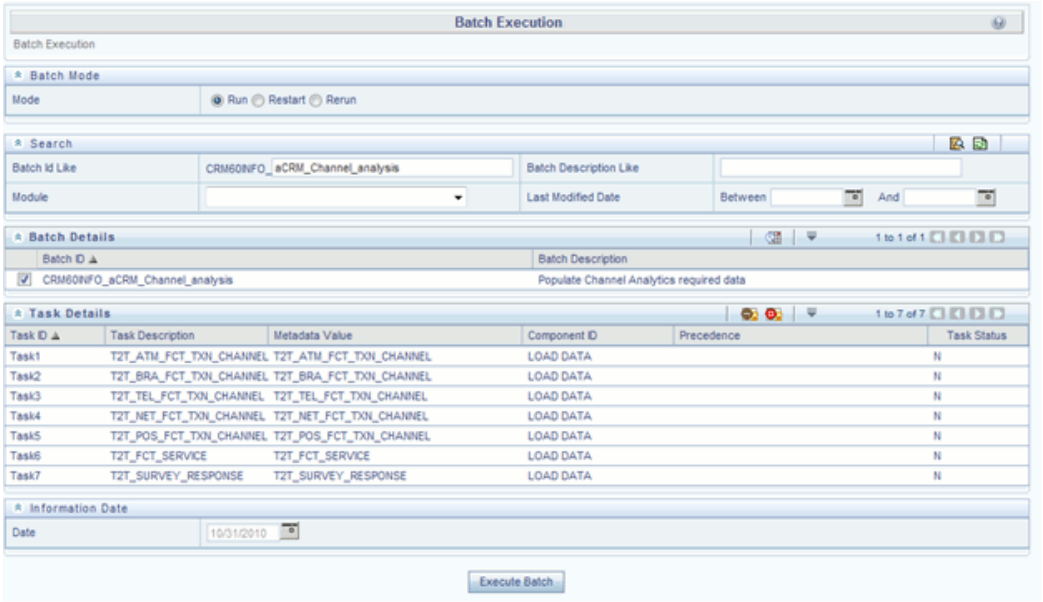
For identifying fields required in Stage Customer Master and Stage Customer Details for the purpose of Customer Insight Application(s), refer to *Download Specification*.

Executing the Fact Transaction Channel Population T2Ts

You can execute the T2T component from the *Operations* (formerly Information Command Center (ICC) framework) module of OFSAAI.

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 the section How to Define a Batch, page E-1.

A seeded batch, **<Infodom>_aCRM_Txn_Channel Task1 to Task5** has to be executed for the required MIS Date.



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 **Load 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 drop down list
   - **Datastore Name** - Select the appropriate datastore name from the drop down 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.

Data file name remains blank for any T2T 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, if default value is [DRCY]='USD', [DLCY]='USD', here 'USD' acts as currency parameter to T2T.

6. Click **Save**.

   The Task definition is saved for the selected Batch.

You can execute a Batch definition from the *Batch Execution* section of *OFSAAI Operations* module.

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 be accessed on the application server in the directory *$FIC_DB_HOME/log/t2t*, where the file name will have the Batch Execution ID.

The following tables can be queried for errors:

- FCT_TXN_CHANNEL$

Fact Application

Fact Application entity stores the fact data of applications like application details, current stage, status, rejection reason, time-taken in each stage, and so on.

The following table lists the seeded T2T Definitions with related Source Table and Destination tables:

T2T Definition Name	Source Staging Table	Destination Table
T2T_FCT_APPLICATION	STG_APPLICATION	FCT_APPLICATION

To view the detailed structure of this table, refer to *Erwin Data Model*.

Prerequisites

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

- DIM_DATES
- DIM_APPLICATION_TYPE
- DIM_PRODUCT
- DIM_CREDIT_OFFICER
- DIM_CUSTOMER
- DIM_CHANNEL
- DIM_CREDIT_CENTER
- DIM_DECISION_STATUS
- DIM_GEOGRAPHY
- DIM_INDUSTRY
- DIM_PROFESSION
- DIM_HOME_OWNERSHIP
- DIM_EDUCATION
- DIM_MARITAL_STATUS
- DIM_APPLICATION_REJECT_REASONS
- DIM_DEVIATION_REASONS
- DIM_SALES_REPRESENTATIVE
- DIM_CAMPAIGN
- DIM_ACCOUNT
- DIM_PROSPECT
- DIM_BANDS
- STG_APPLICATION

For details on populating dimension tables like DIM_CUSTOMER, DIM_BANDS, and so on, refer to the section Dimension Tables Population, under the chapter *Dimension Loading Process*.

For details on populating DIM_DATES dimension table, refer to the chapter Time Dimension Population, page 4-1.

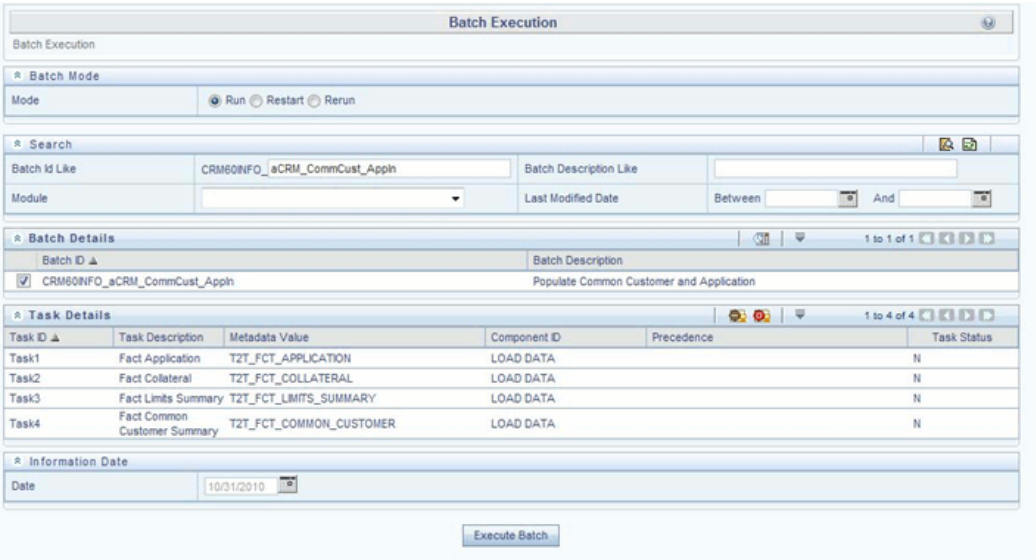
For identifying fields required in Channel Transaction tables in staging for the purpose of Customer Insight Application(s), refer to *Download Specification*.

Executing the Fact Application Population T2T

You can execute the T2T component from the *Operations* (formerly Information Command Center (ICC) framework) module of OFSAAI.

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 the section How to Define a Batch, page E-1.

A seeded batch, **<Infodom>_aCRM_CommCust_Appln – Task1** has to be executed for the required MIS Date.



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 **Load 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 drop down list
   - **Datastore Name** - Select the appropriate datastore name from the drop down 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 "T2T_FCT_APPLICATION" to process.

     Data file name remains blank for any T2T 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, if default value is [DRCY]='USD', [DLCY]='USD', here 'USD' acts as currency parameter to T2T.

6. Click **Save**.

   The Task definition is saved for the selected Batch.

You can execute a Batch definition from the *Batch Execution* section of *OFSAAI Operations* module.

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 be accessed on the application server in the directory *$FIC_DB_HOME/log/t2t*, where the file name will have the Batch Execution ID.

The following tables can be queried for errors:

- FCT_APPLICATION$

Fact Account Profitability

Fact Account Profitability entity stores fact data for reporting line items of revenue, costs, and expense related to each customer account. The data into this table is populated from other fact tables like FCT_COMMON_ACCOUNT_SUMMARY, FCT_PFT_ACCOUNT_SUMMARY, FCT_FTP_ACCOUNT_SUMMARY, FCT_REG_CAP_ACCOUNT_SUMMARY, and FCT_ECO_CAP_ACCOUNT_SUMMARY.

The following table lists the seeded Post Load Transformation Definition with related Source Table and Destination tables:

DT Definition Name	Source Tables	Destination Table
FN_FCT_ACCOUNT_PFT	FCT_COMMON_ACCOUNT_SUMMARY	FCT_ACCOUNT_PROFITABILITY
	FCT_PFT_ACCOUNT_SUMMARY	
	FCT_FTP_ACCOUNT_SUMMARY	
	FCT_REG_CAP_ACCOUNT_SUMMARY	
	FCT_ECO_CAP_ACCOUNT_SUMMARY	

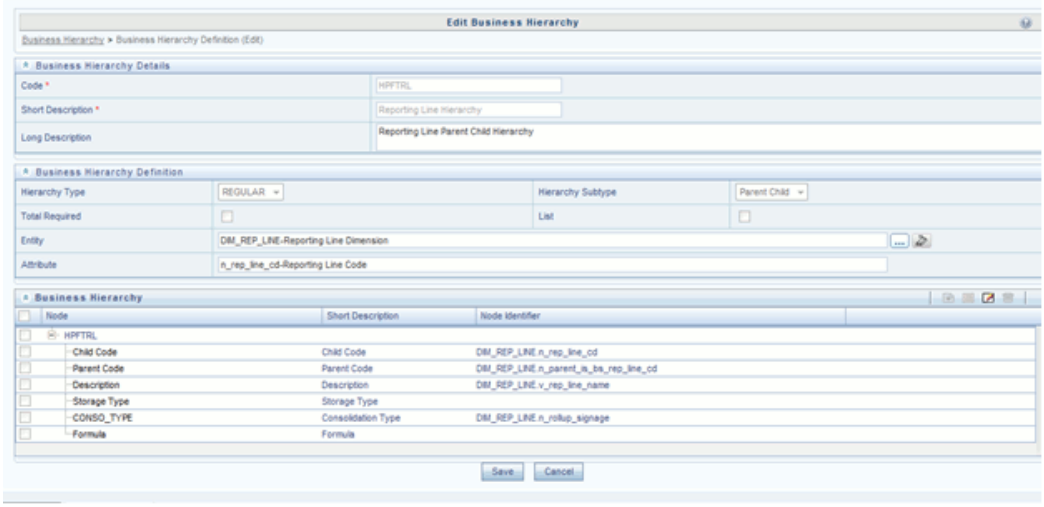
To view the detailed structure of the tables, refer to *Oracle Financial Services Analytical Applications Data Model Data Dictionary* or the *Erwin Data Model*.

Information from account summary fact tables are populated to Fact Account Profitability through a mapping process. Reporting line dimension is mapped to measures present in account summary. A PL/SQL procedure then populates the fact table by reading the mapping definition.

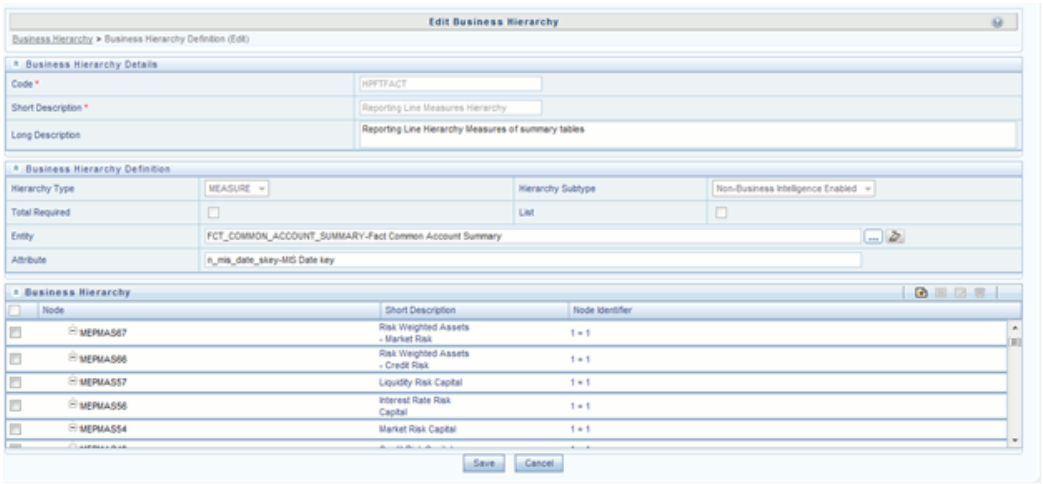
Reporting line dimension is created/maintained from Attribute Member Hierarchy Maintenance (AMHM) component of OFSAAI. A Reporting line item represents a revenue, costs, or expenses. Rollup signage is set as an attribute for a reporting line item. To know more about AMHM, refer to *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

The Account summary tables contain the revenue, costs, or expenses measures pertaining to an Account. Map Maintenance component of OFSAAI is used to map the measures of account summary tables (represented in a measure hierarchy) to reporting line hierarchy. A pre-defined mapping "Reporting Line Mapping" is seeded along with the application installer. Reporting Line Hierarchy and Reporting Line Measure

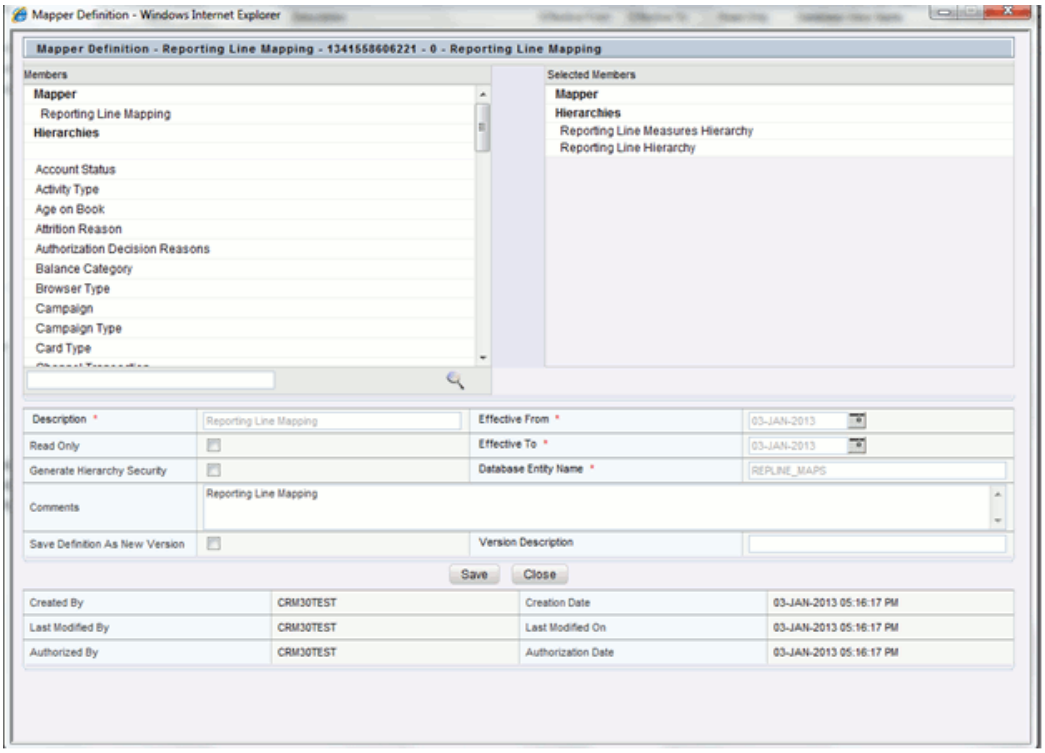
Hierarchy are the two hierarchies which are used for the mapping. Reporting Line Hierarchy is a parent child hierarchy which is based on Reporting Line Dimension entity.

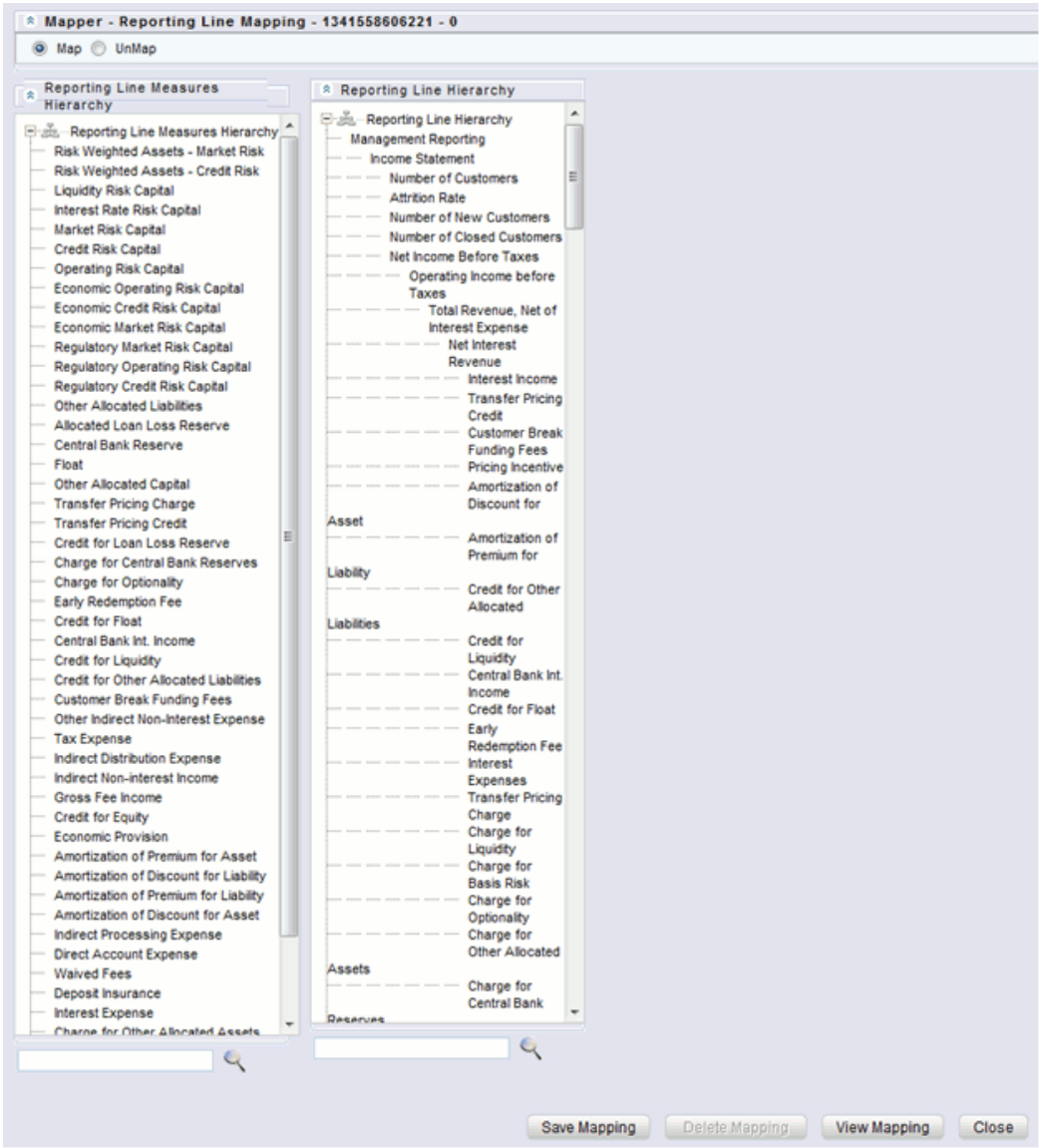


Reporting Line Measure hierarchy is a Non Business Intelligence Enabled Hierarchy which is based on measures from the Account Summary tables.



A seeded map is configured between the Reporting Line Hierarchy and Reporting Line Measure Hierarchy from Map Maintenance of OFSAAI.


Mapper Definition - Windows Internet Explorer
Mapper Definition - Reporting Line Mapping - 1341558606221 - 0 - Reporting Line Mapping
Members
Mapper
Reporting Line Mapping
Hierarchies
Account Status
Activity Type
Age on Book
Attrition Reason
Authorization Decision Reasons
Balance Category
Browser Type
Campaign
Campaign Type
Card Type
Selected Members
Mapper
Hierarchies
Reporting Line Measures Hierarchy
Reporting Line Hierarchy
Description *
Reporting Line Mapping
Effective From *
03-JAN-2013
Read Only
Effective To *
03-JAN-2013
Generate Hierarchy Security
Database Entity Name *
REPLINE_MAPS
Comments
Reporting Line Mapping
Save Definition As New Version
Version Description
Save
Close
Created By
CRM30TEST
Creation Date
03-JAN-2013 05:16:17 PM
Last Modified By
CRM30TEST
Last Modified On
03-JAN-2013 05:16:17 PM
Authorized By
CRM30TEST
Authorization Date
03-JAN-2013 05:16:17 PM



For more information on defining/maintaining Mapper, refer to *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

Steps to Define Mapping for Custom Reporting Line Items

Follow the below steps to define mapping for Custom Reporting Line items:

- Add Custom Reporting Line or Modify existing Reporting Line.
- Add Custom Reporting Line Hierarchy or modify existing seeded reporting line hierarchy.
- Execute the seeded Batch **<INFODOM>_ Repline_Dimension_Update** specifying

the Reporting line hierarchy as parameter to Batch.

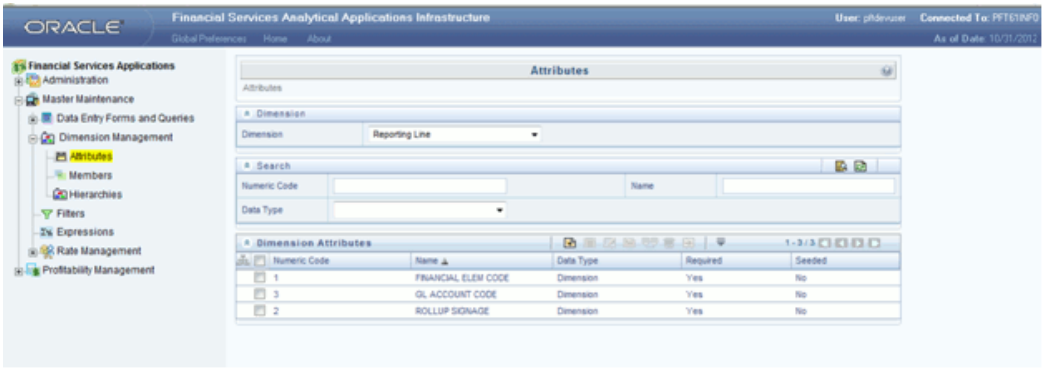
- Modify the seeded Business Metadata.
- Map Maintenance.

Add Custom Reporting Line or Modify existing Reporting Line

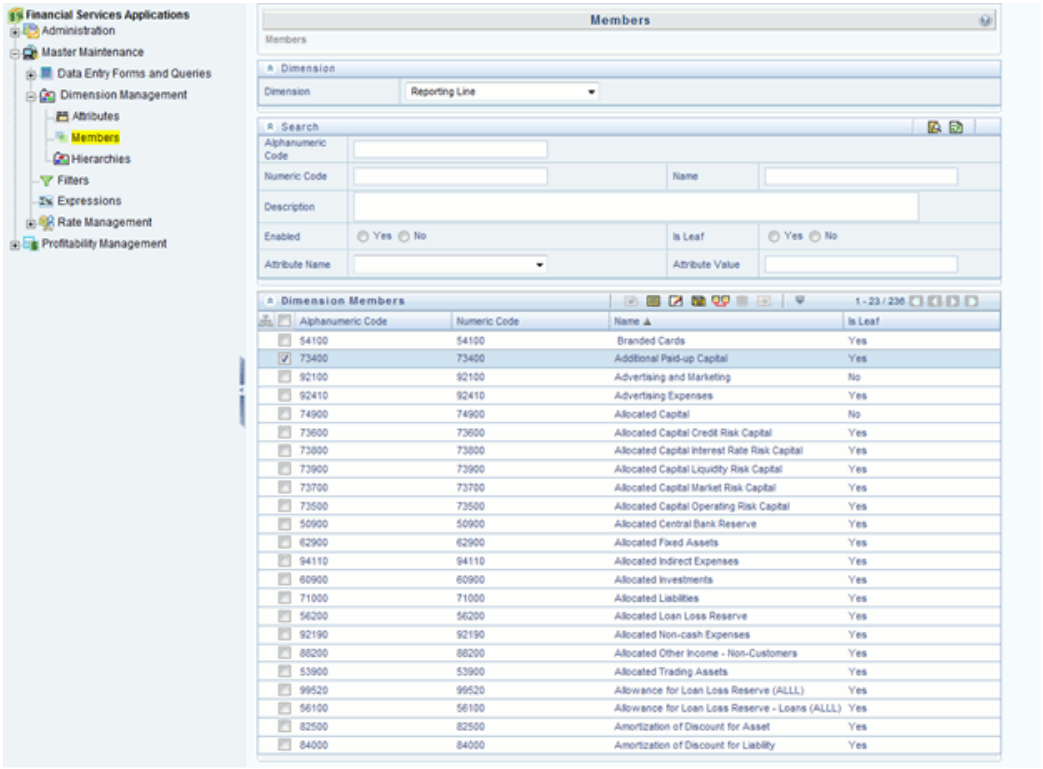
Custom Reporting Lines can be added or modified from AMHM.

Following are the seeded attributes of Reporting Line Dimension:

- Financial Element Code
- GL Account Code
- Rollup Signage



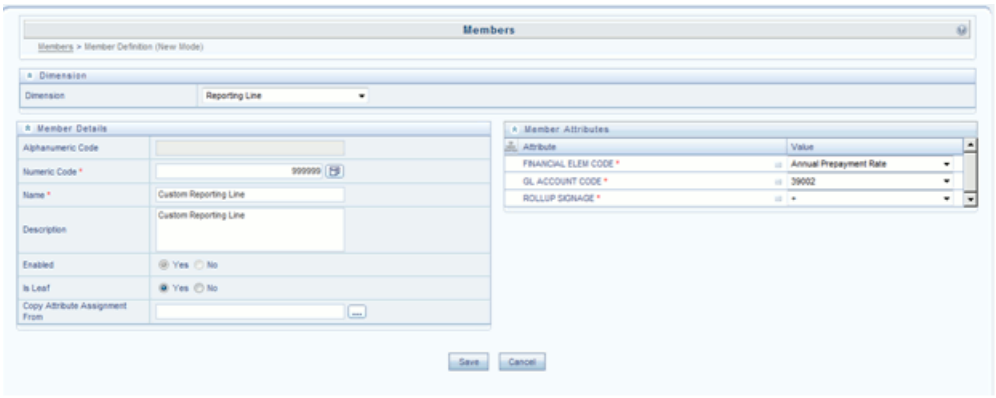
A Reporting line can be added or modified from the *Members* screen as shown below. To modify the existing reporting line, select the member by selecting the adjacent check box and select the **Edit** button on the menu bar.



To add a new reporting line:

1. Select **Add** button from the *Members* screen.

   The *Member Definition (New Mode)* screen is displayed.



2. In the *Member Definition (New Mode)* screen:

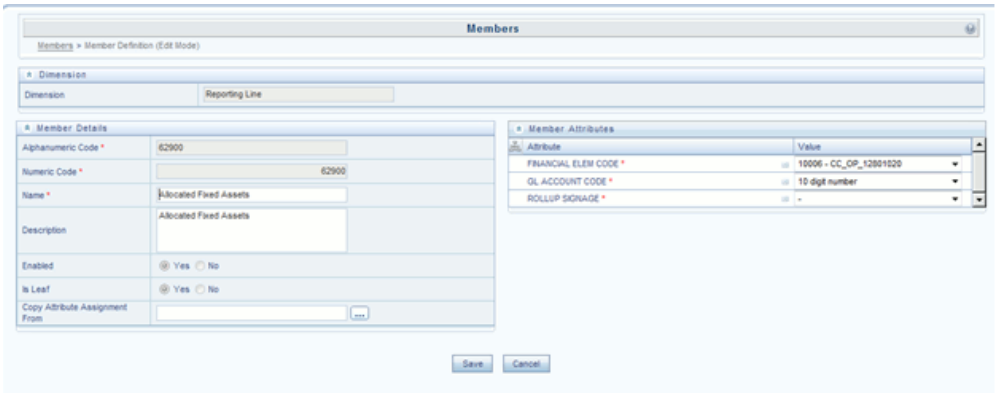
   - Enter **Numeric Code**.
   - Enter the **Name** of the custom reporting line.

- Enter the **Description** of the custom reporting line.
- Select Yes, if the custom reporting line has to be **Enabled** or not.
- Select Yes, if the custom reporting line **Is Leaf** or not.
- Select the Attributes for the reporting line member.
- Save the Member definition.

To modify a reporting line:

1. Click **Edit** button from the *Members* screen.

   The *Member Definition (Edit Mode)*screen is displayed.



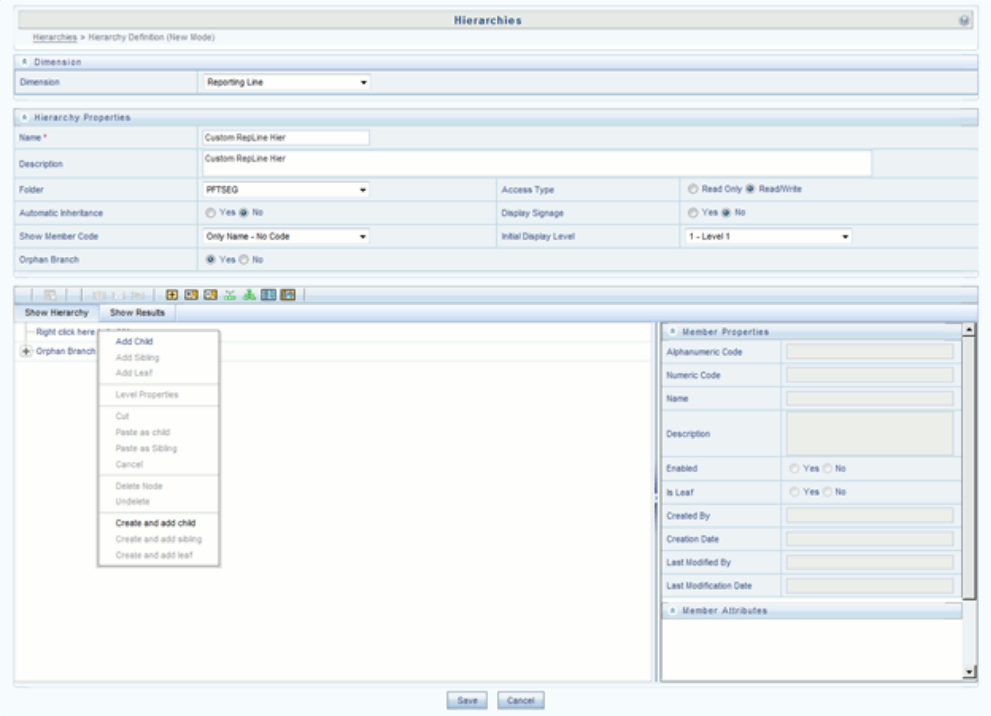
2. In the *Member Definition (Edit Mode)* screen, perform the following as required:
   - Modify the **Name** of the custom reporting line.
   - Modify the **Description** of the custom reporting line.
   - Modify the selection of the radio button in the **Enabled** field.
   - Modify the selection of the radio button in the **Is Leaf** field.
   - Modify the Attributes for the reporting line member.
   - Save the Member definition.

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

Add Custom Reporting Line Hierarchy or Modify Existing Seeded Reporting Line Hierarchy

To create a new Reporting Line Hierarchy:

1. Click **Add** button from the menu. The *Hierarchy Definition (New Mode)* screen is displayed.

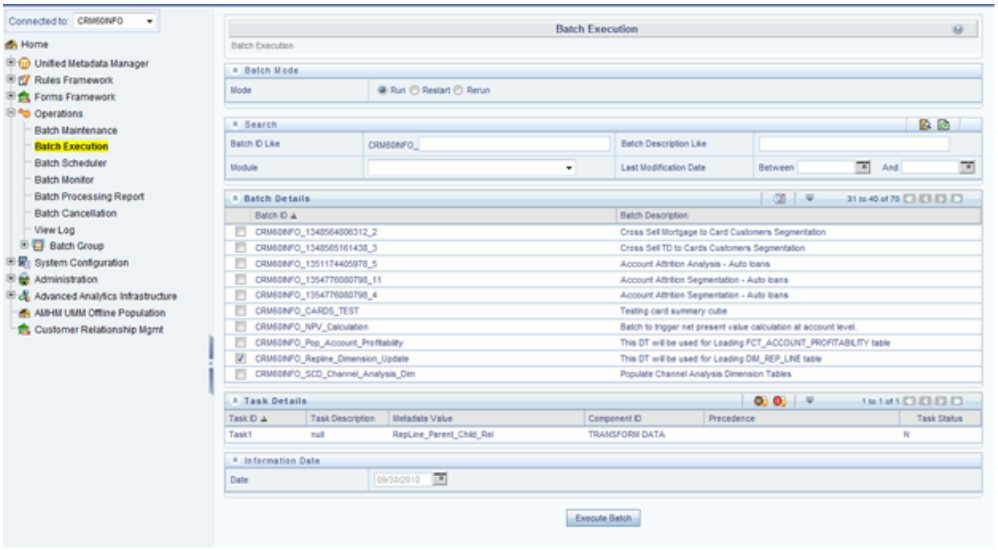


2. Enter the details in the required fields, and click **Save**.

> **Note:** Alternatively, insert scripts and update scripts can be prepared into tables DIM_REPORTING_LINE_B, DIM_REPORTING_LINE_TL, DIM_REPORTING_LINE_ATTR, and DIM_REPORTING_LINE_HIER for adding any new custom reporting lines or modifying an existing reporting line.

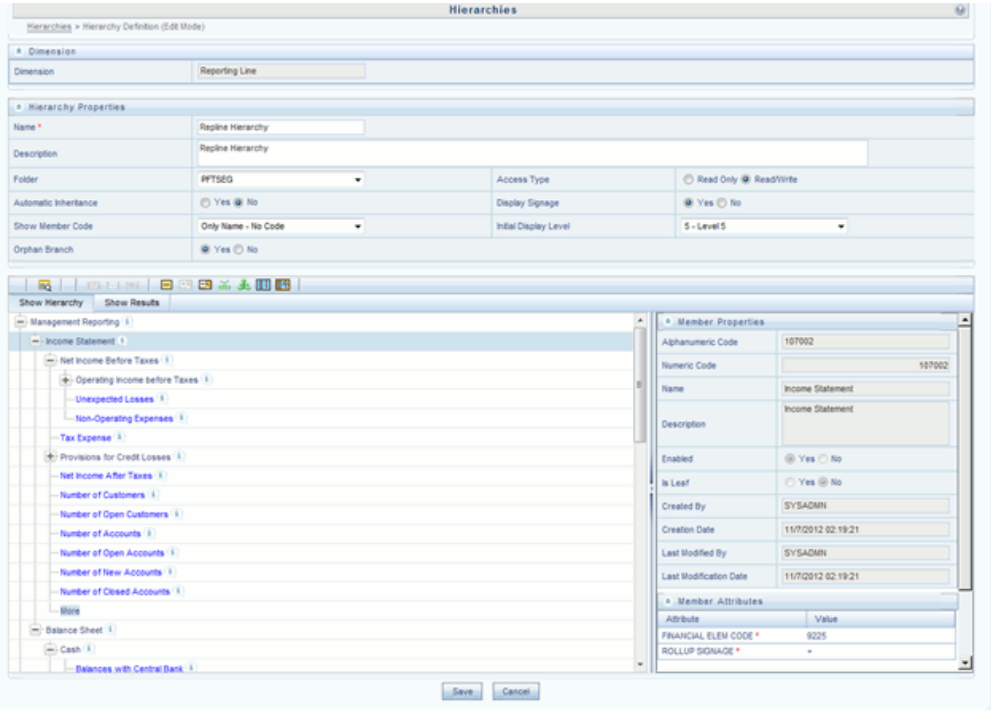
Execute the seeded batch **<Infodom>_ Repline_Dimension_Update** specifying the Reporting line hierarchy as parameter to batch.

Execute the seeded batch **<Infodom>_ Repline_Dimension_Update** . It populates data into DIM_REP_LINE table.



To modify existing seeded Reporting Line Hierarchy:

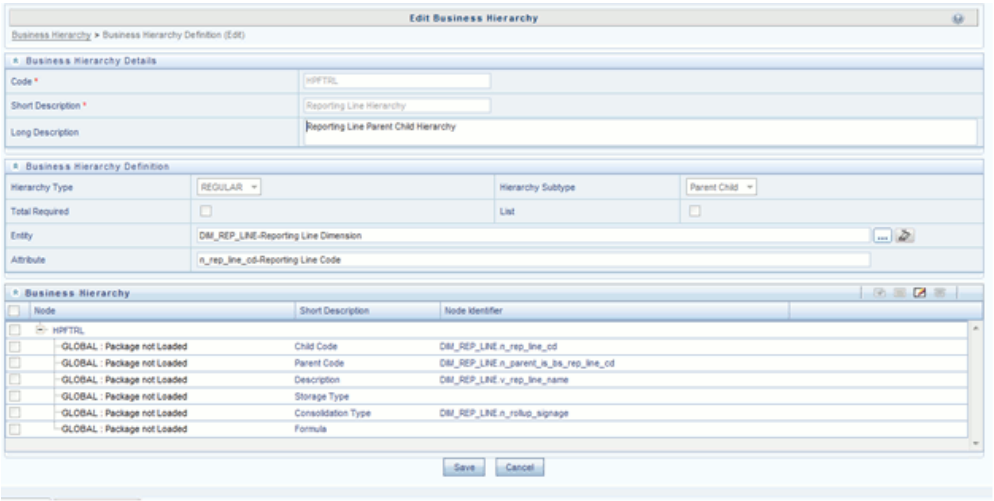
1. Select the check box adjacent to the Reporting Line Hierarchy to be modified.
2. Click **Edit** button from the menu.
3. Modify the Hierarchy as required and click **Save**.



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

Modify the Seeded Business Metadata

1. Resave the Seeded Business Metadata parent child hierarchy, "Reporting Line Hierarchy" (HPFTRL), so that the changes done are consolidated in the hierarchy as well.



2. (Optional) Create the Business Measures for the newly added reporting lines.
3. Attach and Save the defined Business Measures to the hierarchy "Reporting Line Measures".
4. Save the metadata.

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

Map Maintenance

Once all the above steps are done, the seeded map configured between the Reporting Line Hierarchy and Reporting Line Measure Hierarchy has to be modified if required from Map Maintenance of OFSAAI.

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

Prerequisites

Following are the lists of tables used in the population of Fact Account Profitability and these tables are required to be loaded prior to running the DT.

- DIM_DATES - Mandatory
- DIM_REP_LINE - Mandatory
- FCT_COMMON_ACCOUNT_SUMMARY
- FCT_PFT_ACCOUNT_SUMMARY

- FCT_FTP_ACCOUNT_SUMMARY
- FCT_REG_CAP_ACCOUNT_SUMMARY
- FCT_ECO_CAP_ACCOUNT_SUMMARY

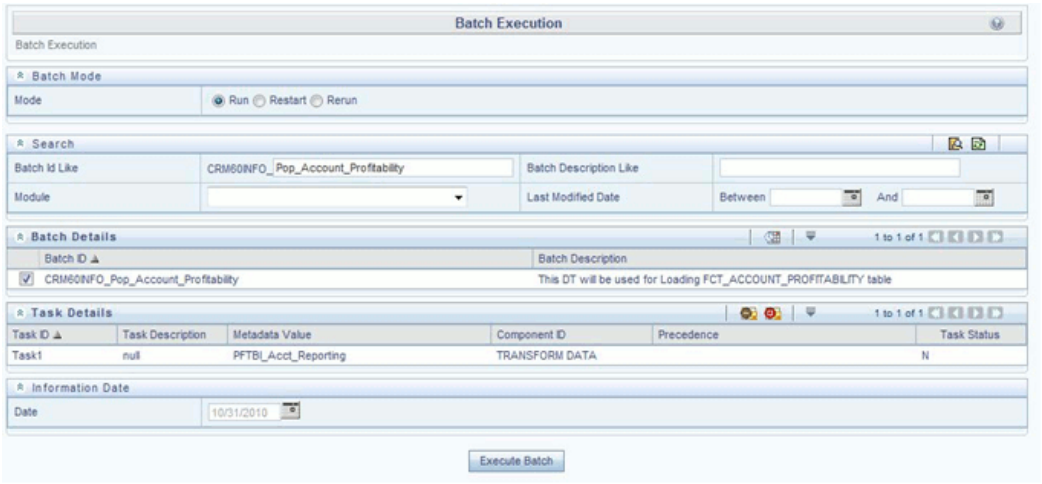
For more information on SCD, refer to the chapter Dimension Loading Process, .

Executing the Fact Account Profitability Population DT

You can execute the T2T component from the *Operations* (formerly Information Command Center (ICC) framework) module of OFSAAI.

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 the section How to Define a Batch, page E-1.

A seeded batch, **<Infodom>_Pop_Account_Profitability** has to be executed for the required MIS Date.



To define a new task for a Batch definition:

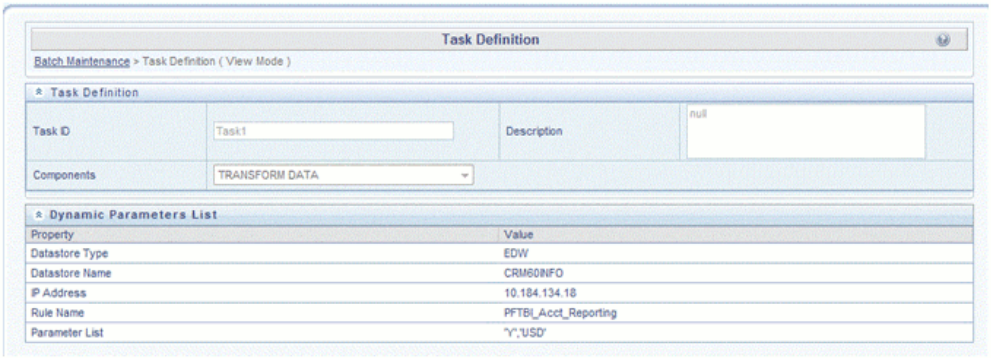
1. Select the check box adjacent to the newly created Batch Name in the *Batch Maintenance* window.
2. Click Add (+) button from the *Task Details* grid.

   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 drop down list
   - **Datastore Name** - Select the appropriate datastore name from the drop down list
   - **IP address** - Select the IP address from the list
   - **Rule Name** - Select **FN_FCT_ACCOUNT_PFT** from the list.
6. Enter the Parameter List details as mentioned below:
   - **Reload Account Profitability table for the given MIS Date flag** - can be Y or N within single quotes.
   - **Reporting Currency Code** - This has to be enclosed within single quotes.

   For Example, if reporting currency is in US Dollar, then '**USD**' has to be specified.

   > **Note: Batch run ID** and **As Of Date** are passed internally by the batch to the Data Transformation task.



7. Click **Save**.

   The Task definition is saved for the selected Batch.

You can execute a Batch definition from the *Batch Execution* section of *OFSAAI Operations* module.

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 be accessed on the application server in the directory *$FIC_DB_HOME/log/date*, where the file name will have the Batch Execution ID.

13

Cube Build Process

Introduction

Reports of OFSRPA application can be configured to work on Relational database or Essbase cubes. Source of data for the reports is determined by the priority set for each Logical Table Source (LTS) in OBIEE RPD. Multi-dimensional databases store aggregated data for better performance and provide mechanisms for performing non-additive rollup within a hierarchy and defining complex derived measures using cross-dimensional operations. OFSAA Infrastructure is used for defining metadata about the cube and for building the cubes. Cubes are optional source of data for Retail Performance application.

The chapter contains the following sections:

- List of cubes seeded within the application
- Process for building cubes

Overview of Cubes

OFSRPA application has the following seeded cubes:

- Retail Analysis
  - Purpose

    The purpose of this cube is to provide analysis of various Account related measures across dimensions like Product, Line of Business, Vintage, and so on.

  - Dataset

    This cube is based on the FCT_COMMON_ACCOUNT_SUMMARY and FCT_CRM_ACCOUNT_SUMMARY fact tables.

- Customer Summary
  - Purpose

    The purpose of this cube is to provide analysis of various Customer related measures across customer profile dimensions like Income Band, Age, and so on.

  - Dataset

    This cube is based on the FCT_COMMON_CUSTOMER_SUMMARY and FCT_CRM_CUSTOMER_SUMMARY fact tables.

- Cards Balance Summary
  - Purpose

    The purpose of this cube is to provide analysis of various Cards Balance Category related measures across dimensions like Time, Customer Profile by Income, Age, Gender Age on Book, Vintage, and so on.

  - Dataset

    This cube is based on the FCT_CARDS_BALANCE_SUMMARY fact table.

- Account Profitability
  - Purpose

    The purpose of this cube is to provide analysis of Financial Reporting Lines related measures across dimensions like Time, Line of Business, Customer Profile by Income, Age, Gender Age on Book, Vintage, and so on.

  - Dataset

    This cube is based on the FCT_ACCOUNT_PROFITABILITY, FCT_COMMON_ACCOUNT_SUMMARY, FCT_CRM_ACCOUNT_SUMMARY, FCT_COMMON_CUSTOMER_SUMMARY, and Fct_CRM_Customer_SUMMARY fact tables.

- RM PnL Cube for RPA
  - Purpose

    The purpose of this cube is to provide details of Profit and Loss statement of a Relationship Manager across dimensions like Line of Business, Product, Organizational Unit, and so on.

  - Dataset

    This cube is based on the FCT_ACCOUNT_PROFITAIBILTY and

FCT_ACCOUNT_MGR_REL fact tables.

If there is an error with the Relationship Manager cube saving or execution for the first time after the installation, the parent child hierarchies must be saved by editing the hierarchies individually and re-saving them after which the cube needs to be saved successfully and re-executed.

Creating Configuration Files

Each cube has a configuration file that contains the details of dimensions and measures which are part of the cube. Essbase outline is created using the configuration file. Configuration files for seeded cubes are available as part of the installer. However, if there are any changes to cube definition then configuration files are recreated during saving of the cube definition.

Follow these steps:

1. On the LHS menu of OFSAAI, go to **Home > Unified Metadata Manager > Business Metadata Management > Cubes**.
2. Click **Search** and check if you can see the cubes in the pop up window that opens.
3. Click on the cube that needs to be built and click **OK** to return to the *Cube Definition* Screen.
4. Click **Save** to save the cube. A pop up appears saying 'Operation Successful'.

> **Note:** Cube definition will be saved only when the UI component detects any change event. In order to trigger the change event, type a blank space in 'Long Description' text-box and remove the same. Or a dimension can be removed from selected list, again the same dimension re-selected, variation applied for the dimension, and saved.

Building Of Cubes

The Cube build process in OFSAA Infrastructure contains the following steps:

- Generating an aggregate DATA file containing the measure values for each dimension leaf that are part of the cube definition. This is performed by the **AGGREGATE DATA** component task within the batch definition.
- Creating the cube outline on Essbase server. This is performed by the **CREATE CUBE** component task within the batch definition.

- Loading the data to the cube. This is performed by the **CREATE CUBE** task within the batch definition.

This section covers the following topics:

- Prerequisites
- Tables used by the Cube build component
- Executing the Cube build task
- Checking the execution status

Prerequisites

The following are prerequisites for creating a cube:

- All the post install steps mentioned in the OFSAA Infrastructure installation guide & Solution installation manual have been completed successfully.
- Parentage files need to be created for BI hierarchies after dimension data is loaded. 'Resave Metadata' process is used to create the parentage files.
- OFSAAI application user needs to have the required functions mapped to the user for doing Resave Metadata and accessing the Home> Unified Metadata Manager > Business Metadata Management screens and executing a batch from Application batch operations screen.
- Execute Save Metadata by navigating to the following screen on the OFSAAI framework LHS Menu.
- Go to Home > Administration > Save Metadata.
- Choose all the available metadata under Hierarchy and move it to the right by using the '>>' button.
- Click Save and might take a few minutes for the saving to complete.
- Click Show Details to view the log for the Save operation.

  Refer to System Configuration & Administration chapters in OFSAAI User Manual for details on the Resave metadata feature. Saving metadata creates all the parentage files required for building cubes.
- Ensure that the following services are running on the application server before doing a cube build:
  - Iccserver

- Router
- AM
- Messageserver
- Olapdataserver

- Batches need to be created for executing, which is explained in the Executing the Cube build section.
- All the required tables for dataset need to be populated before you execute the cube batches, such as Dimension Population, Time Dimension population, Account Summary Population, and Fact Ledger Population.
- The dataset for the cube should return some rows in the database for the cube build to happen.

  To check the same, perform the following steps:

  - Navigate to Home > Unified Metadata Manager > Business Metadata Management > Data Sets.
  - Click Search
  - Click any dataset in the pop up which opens and click OK to return to the data set screen.
  - Click the button on right of ANSI Join text box. Enter the required expression or click the below button to define an expression using the Expression screen.
  - Click OK to return to the data set screen.

    For more information, refer to *Create Expression* section in *OFSAA Infrastructure User Guide*.

- Perform the same for Join/Filter Condition and Date filter.
- Frame a SQL query like this:

  ```
  SELECT COUNT(1) FROM <ENTER THE PART YOU OBTAINED FROM ANSI JOIN
  PART ABOVE>WHERE<ENTER THE PART YOU OBTAINED FROM JOIN/FILTER
  CONDITION & DATE FILTER PARTS>
  ```

  This query should show record count greater than zero when you fire this from SQL prompt in the database.

Tables Used by the Cube Build Component

Tables that are part of the dataset need to be populated before executing the cube build

component. In addition, REV_BIHIER table in atomic database schema stores the hierarchy data for Business Intelligence-enabled hierarchies for cube build. This table gets populated when a hierarchy is saved using *Save Metadata* screen.

Executing the Cube Build Task

To execute the cube build process from OFSAAI ICC framework (accessed through the application Batch Operations screen), create a new Batch with two tasks – one for performing Data crunching (component is Aggregate Data) operations and another for building cube (component is Build Cube). The above batch needs to be created for each of the cubes.

- Aggregate Data Task
    - From OFSAAI Home menu, select Operations > 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 on the check box in the Batch Name container.
    - Click New Task ('+' symbol in Task Details container).
    - Enter the Task ID and Description.
    - In the Component drop down, choose Aggregate Data.
    - 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.
        - Cube Parameter - Choose the cube code to be built from the drop down list.
        - Operation - Choose All from the drop down list.

- Create Cube Task
    - In the batch created in Aggregate Data task above, click New Task ('+' symbol in Task Details container).
    - Enter the Task ID and Description.

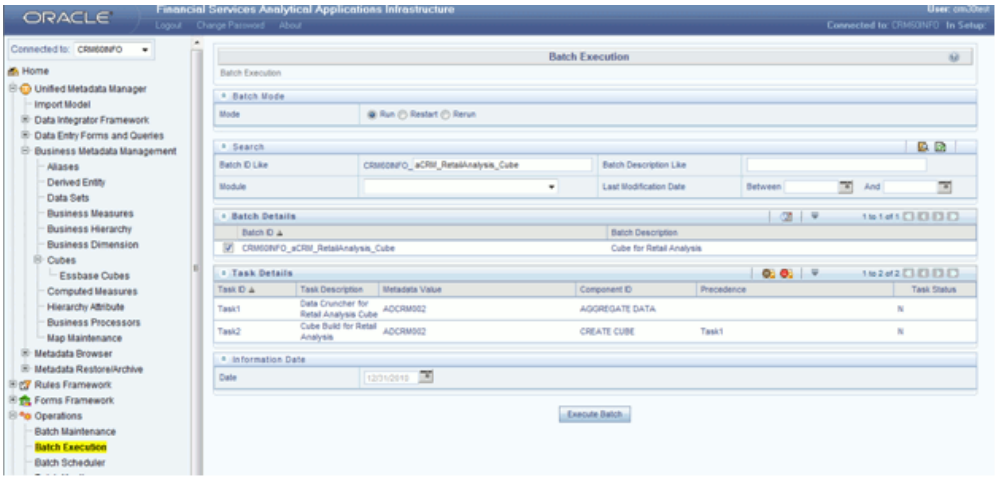
- In the Component drop down, choose Create Cube.
- the following from the Dynamic Parameters List and then click Save:
  - Datastore Type - Select the appropriate datastore from the list.
  - Datastore Name - Select the appropriate name from the list.
  - IP address - Select the IP address from the list.
  - Cube Parameter - Choose the cube code to be built from the drop down list.
  - Operation - Choose All from the drop down list.
- Execute the batch created in the above step.

  Note: A common issue in the Aggregate task is Data Set not having records for which the steps mentioned in the prerequisites have to followed or the SQL query in Data Cruncher log file has to be checked on the database (Location of log file mentioned in the 'Checking the Execution Status' section below). In the Create Cube task one common error is the hierarchy member being the same for two different dimensions which are part of the same cube (Error message: 'Duplicate Alias' in the Create Cube log file). In this case, you can try appending a string to the Hierarchy member code so that it is unique across the cube or changing the hierarchy data to make the node unique across the cube.

Seeded batches are provided along with the RPA application installer. The below described are the OFSRPA seeded batches:

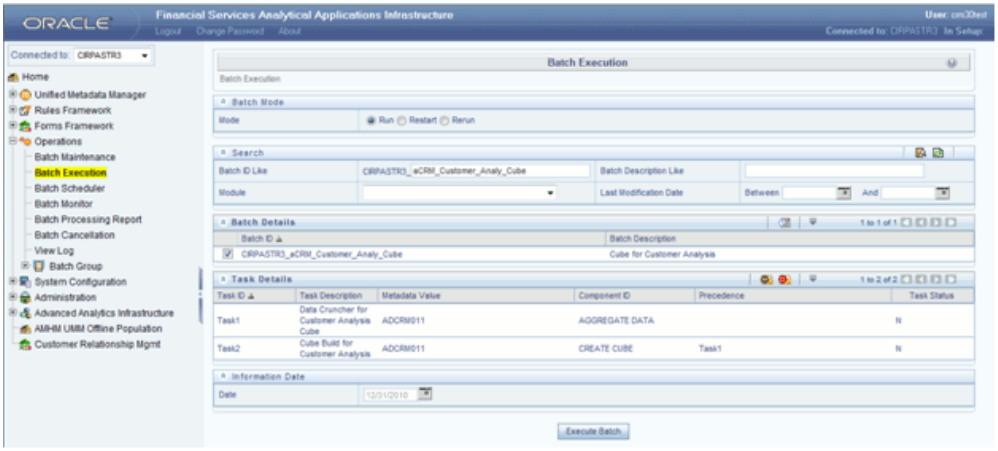
- Retail Analysis

  Seeded batch **<INFODOM>_aCRM_RetailAnalysis_Cube** is provided with the installer. Execute the batch for the required MIS Date.



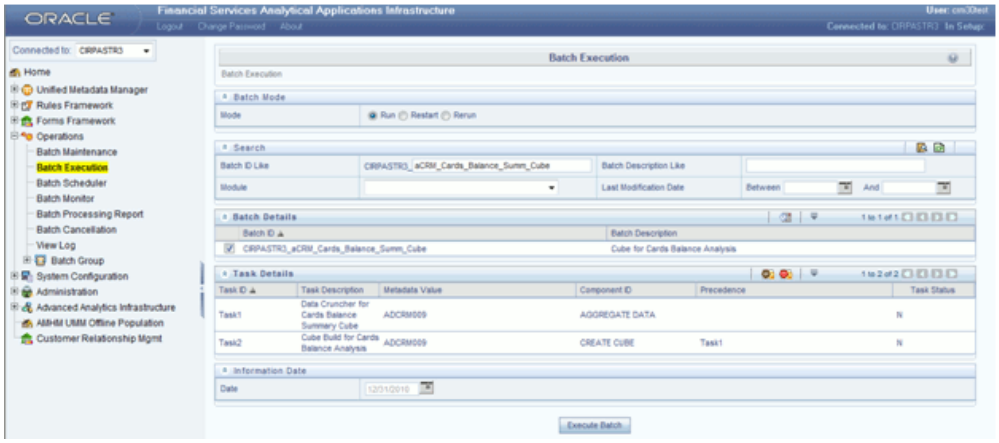
- Customer Summary

  Seeded batch **<INFODOM>_aCRM_Customer_Analy_Cube** is provided with the installer. Execute the batch for the required MIS Date.



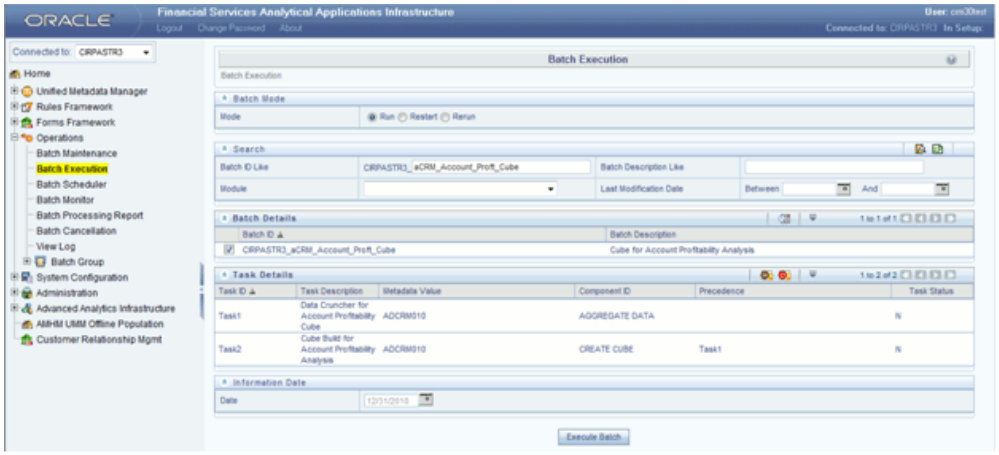
- Cards Balance Summary

  Seeded batch **<INFODOM>_aCRM_Cards_Balance_Summ_Cube** is provided with the installer. Execute the batch for the required MIS Date.



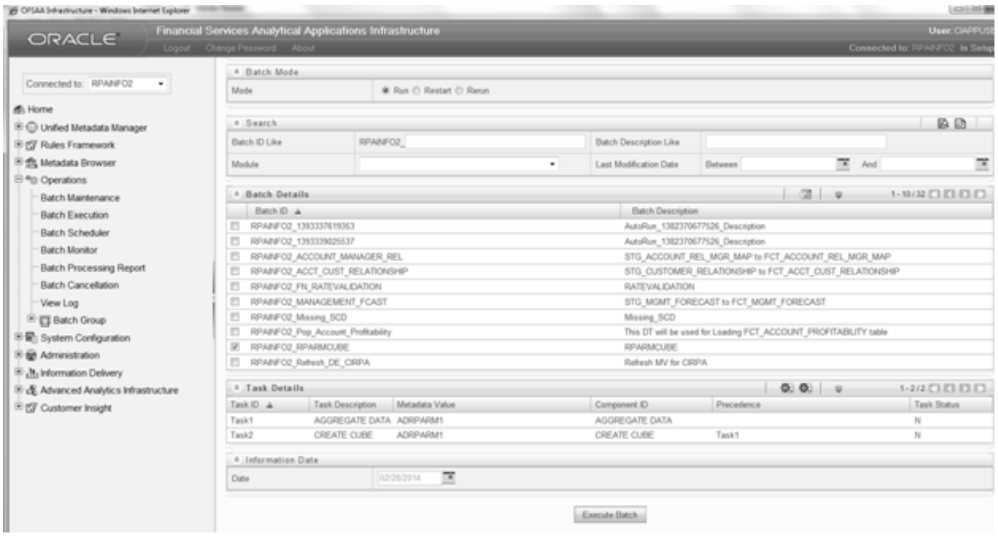
- Account Profitability

  Seeded batch **<INFODOM>_aCRM_Account_Proft_Cube** is provided with the installer. Execute the batch for the required MIS Date.



- RM PnL Cube for RPA

  Seeded batch <INFODOM>_RPARMCUBE is provided with the installer. Execute the batch for the required MIS Date.



Checking the Execution Status

The status of execution can be monitored using the *Batch Monitor* screen. From OFSAAI Home menu, select Operations > Batch Monitor.

> **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 directory *$FIC_DB_HOME/log/dc* for the Task 1 above (Aggregate Data). The file name will have the Batch Execution ID.

The execution log can be accessed on the application server by going to the following directory *$FIC_DB_HOME/log/olap* for the Task 2 above (Create Cube). The file name will have the Batch Execution ID.

> **Note:** Refer to Appendix on how to add a New cube or modifying existing ones. For any new cube added using the OFSAAI framework

Cube screen , the tasks for execution are the same as mentioned above.

14

Predictive Modeling

Introduction

What-if analysis reports use the reporting line forecast values that are generated using the Arima Algorithm in the R code seeded with the application. R has a base package called "stats" which provides the function called as "arima()". This function enables the usage of ARIMA technique for time series forecasting.

Guidelines

Following are the guidelines associated with respect to the execution R code:

- Data should be generated for at least one group for more than 12 continuous mis dates. 12 is the parameter n. Consider where we are setting how many records is significant to be considered for prediction.
- Assumption is that the data is chronological for consecutive end of month dates. By default, prediction is done for 60 months starting with the immediate month after the last available MIS Date.
- If the data provided is not for chronological end of month dates, results generated will not be accurate.
- ARIMA is a statistical technique used for time series predictions. It accepts a host of parameters of which the basic parameters are p, q, and d. p is the order of Autoregressive Process, q is the parameter for Moving Average process, and d is the number of differences after which the data can be considered stationary with a desired confidence level. It has more parameters that can be customized.
- Detailed documentation of the technique can be found at: http://www.dms.umontreal.ca/~duchesne/chap7.pdf [http://http://www.dms.umontreal.ca/~duchesne/chap7.pdf]

- Documentation of implementation of ARIMA in R can be found at: http://stat.ethz.ch/R-manual/R-patched/library/stats/html/arima.html [http://http://stat.ethz.ch/R-manual/R-patched/library/stats/html/arima.html]

Logging for the individual groups' arima model summaries happens in a file named "out.log". This log file is generated in $FIC_DB_HOME/bin folder. If a particular group has unstable data and prediction fails, corresponding error is also documented in the out.log file against that particular group.

Files Used

Two files are required for R script execution. Both the files are present at $FIC_DB_HOME/bin folder and need execute permissions. Following are the files used:

- RExec executable
- ARIMA_AVF.r

Errors

Following are the errors:

- Subscript out of bounds usually means that sufficient data has not gone in. Model is trying to apply an algorithm on a dataset that is returning null chunk.
- Error: Error in if ((lv >nr) || (lv == 0L && nr > 0L) || (nr%%lv != 0L)) stop(gettextf("replacement data has %s rows, data has %s",:
    - missing value where TRUE/FALSE needed
    - means an if condition is receiving null and is unable to evaluate true or false

15

Overview of OFSRPA Reports

Introduction to Dashboards

Oracle Financial Services Retail Performance Analytics (OFSRPA) offers dashboards to users that organize different kinds of reports by subject area.

These reports provide the following:

- Gain deep insight into customer engagements across target segments and products/LOB including lending, credit cards, and so on.
- Perform Wallet share analysis and Customer Profitability.
- Understand the efficiency of investments (like marketing, branch, and channel and so on) over time.
- Monitor customer distribution across credit and delinquency bands and related exposures.
- Perform an enterprise-wide revenue analysis across customer segments, products, and reporting lines including fee income, interest, and interchange.
- Summary performance of the LOBs, overall Profitability, and Portfolio mix.
- Customer trends across performance drivers like Sales, Balances, Deposits, Product subscriptions (revenue services), Credit scores, Delinquency bands, Losses, and so on.
- LOB specific performance reports can be analyzed against key dimensions like customer segments, product family, region, branch, risk scores, and so on.
- Analyze expenses across customer segments, products, and channels to understand ROI.

Dashboards

Following tabs are present in the Retail Performance dashboard:

- **Customer Summary**
- **Performance Summary**
- **Customer Summary**
- **All Products**
- **Cards**
- **Retail Bank**
- **Mortgage**
- **New Business Analysis**
- **Revenue Analysis**
- **Expense Analysis**
- **Credit Loss Summary**
- **Customer Profit and Loss**
- **Relationship Manager Performance**
- **What-If Analysis**
- **Margin Reports**

The following screenshots display the essential nature of the available reports as per each tab:

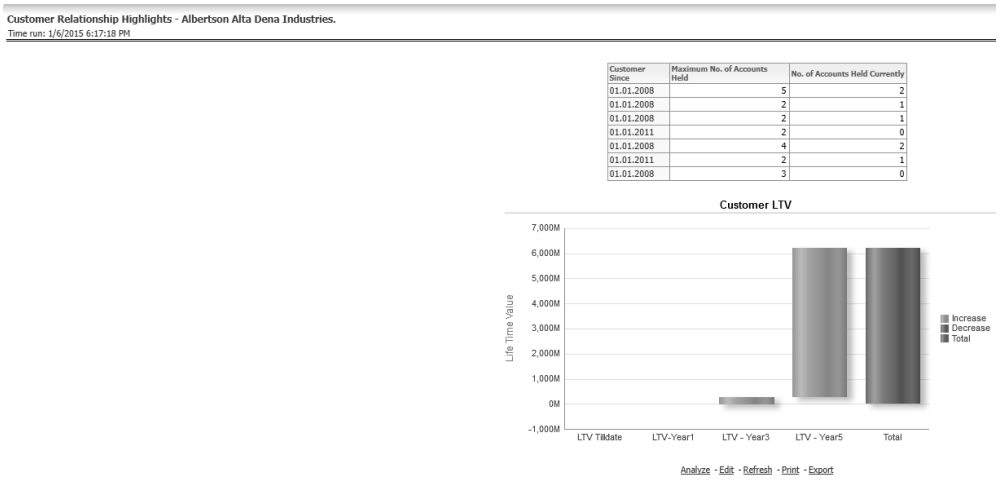
Customer Central

- Customer Details - Albertson Alta Dena Industries

  This report highlights the demographics of a particular customer.

- Customer Relationship Highlights - Albertson Alta Dena Industries.

  This report highlights the Life-time value of a particular customer through all his engagements.



- Holding Summary and Key Drivers

  This report provides the details for all the accounts that have been held by the customer in past or present.

Holding Summary & Key Drivers:

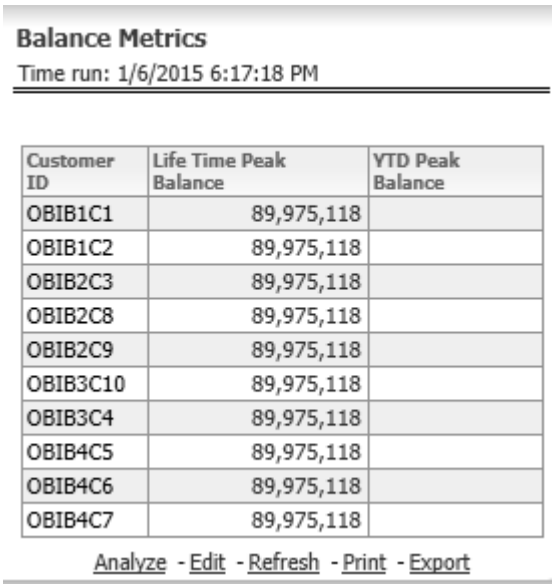
Time run: 12/22/2014 3:04:02 PM

Product Type	Product Name	Account Number	Member Since	Relationship Manager Code	Account Status	Account Attrition Score	End-of Period Balance	Average Balance	Total Sales	Net Revenue	Total Expenses	Net Loss	Net Income
	Retail & Checking Accounts	CA158150	01-Jan-01	A05	PERFORMING	0.01	29,023.07	29,023.07		2,045.90	315,345.46	1,761.80	(313,300)
		CA388978	01-Jan-01	A05	PERFORMING	0.01	12,377.32	12,377.32		5,327.20	314,888.68	1,755.47	(309,561)
		CA400228	01-Jan-01	A05	PERFORMING	0.01	1,489.66	1,489.66		3,345.70	315,607.19	1,590.57	(312,261)
		CA533816	01-Jan-01	A05	PERFORMING	0.01	597.82	597.82		2,758.65	314,696.63	2,049.51	(311,938)
		CA98115	01-Jan-01	A05	PERFORMING	0.01	7,425.81	7,425.81		3,070.60	315,093.80	2,329.93	(312,023)

- Balance Metrics

  This report denotes the peak balance for an account in its life-time and during the

current year.



Balance Metrics

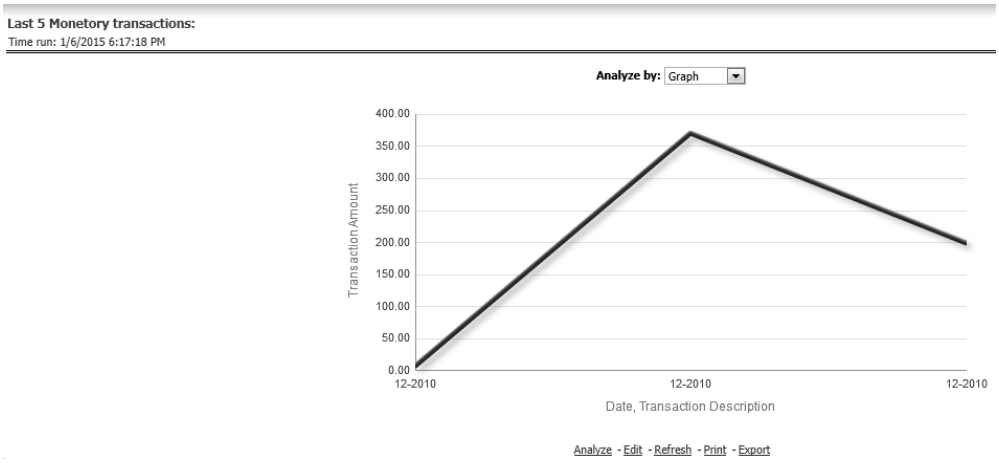
Time run: 1/6/2015 6:17:18 PM

Customer ID	Life Time Peak Balance	YTD Peak Balance
OBIB1C1	89,975,118	
OBIB1C2	89,975,118	
OBIB2C3	89,975,118	
OBIB2C8	89,975,118	
OBIB2C9	89,975,118	
OBIB3C10	89,975,118	
OBIB3C4	89,975,118	
OBIB4C5	89,975,118	
OBIB4C6	89,975,118	
OBIB4C7	89,975,118	

Analyze - Edit - Refresh - Print - Export

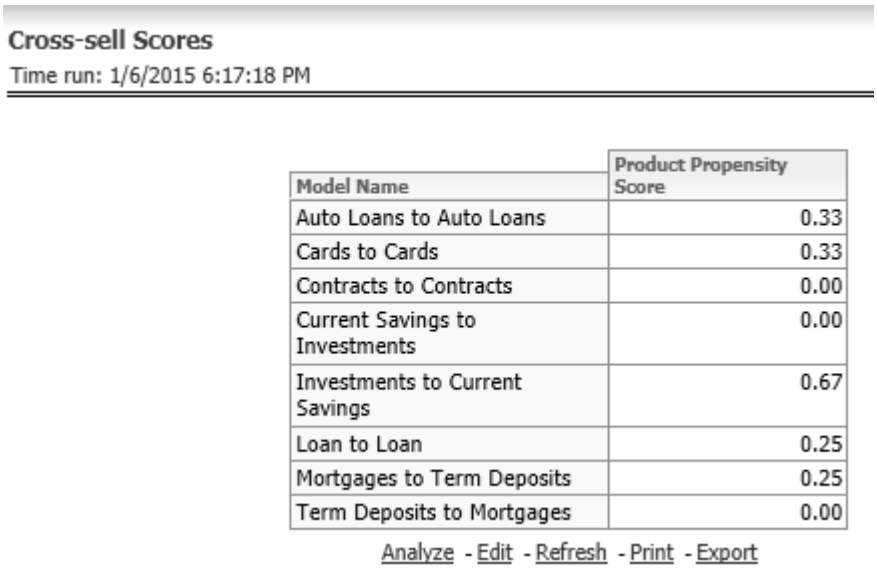
- Last 5 Monetory transactions

  This report plots the value and channel for last 5 transactions of the customer.



- Cross-sell Scores

  This report specifies the score for probability of a customer having one product purchasing another one.



Cross-sell Scores

Time run: 1/6/2015 6:17:18 PM

Model Name	Product Propensity Score
Auto Loans to Auto Loans	0.33
Cards to Cards	0.33
Contracts to Contracts	0.00
Current Savings to Investments	0.00
Investments to Current Savings	0.67
Loan to Loan	0.25
Mortgages to Term Deposits	0.25
Term Deposits to Mortgages	0.00

Analyze - Edit - Refresh - Print - Export

Performance Summary

- Portfolio Mix

  This report displays growth of key metrices such as No. of New Accounts, No. of New Customers, End of Period balance, and Sales across various Lines of Businesses within the bank.

- Revenue Distribution

  This report provides the Revenue spread across the different Lines of Businesses within the bank.

- Cross LOB Holding

  This report outlines the product holding patterns of bank customers across Lines of Business. It shows the relationships the customer has across the enterprise.

- Top 10 Products by Customers and Revenue

  This report outlines the top 10 products ranked by Number of Customers or Revenue.

- No. of Accounts by Region and Product

  This report displays the concentration of Accounts across various Regions and Products within the bank.

- Summary of New Customers

  This report displays the growth in customer base across the various products over

time.

- New Business Summary by Channel

  This report displays summary of new accounts opened across various bank channels.

- New Business Summary by Product

  This report displays summary of new customers on-boarded by product.

- Summary of Closed Accounts

  This report displays accounts closed as a distribution across product and attrition reason.

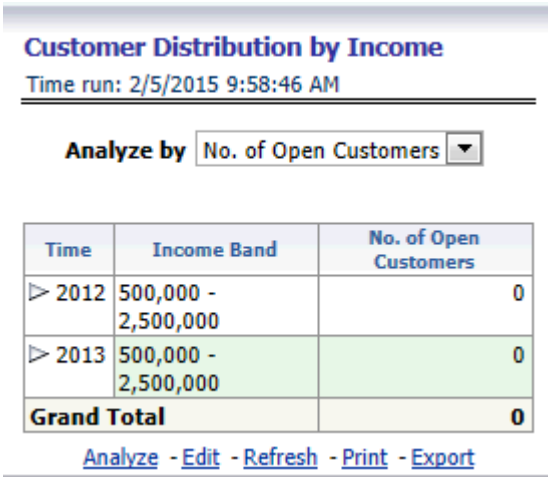
Customer Summary

- Customer Distribution by LoB

  This report displays Distribution of Open Customers and the corresponding Revenue across each Line of Business and its constituent products.

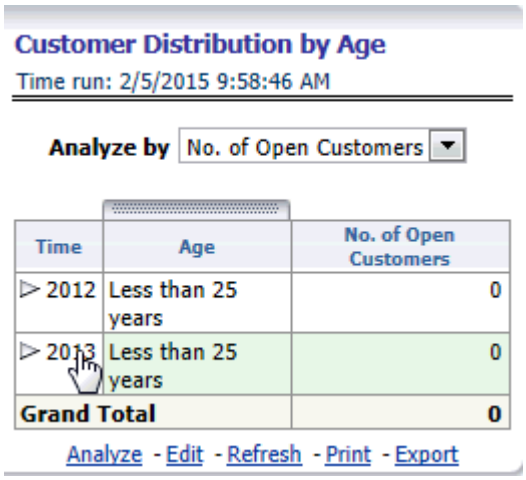
- Customer Distribution by Income

  This report displays Distribution of Open Customers and Open Accounts across Income bands.



- Customer Distribution by Age

  This report displays Distribution of Open Customers and Open Accounts across Age bands.



- Customer Distribution by Region

  This report displays Top 10 Products by Customers and Revenue Distribution of Open Customers and Open Accounts across Regions and Products.

- Relationship Depth

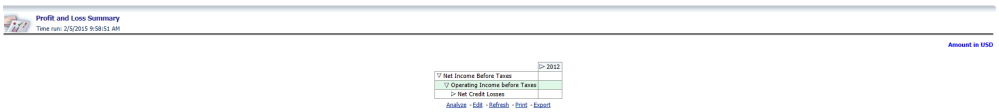
  This report outlines the depth of the customers' relationship with the bank. It distributes the Number of Open customers (customers who still have at least one relationship with the bank that is in open status) and Number of Open Accounts across the unique product combinations they currently have as open relationships with the bank.

  For example: A Customer who only has a Credit Card account will appear against Cards, while another customer who has both a Cards and a Savings Account will appear against Cards, CASA.

All Products

- Profit and Loss Summary

  This report displays Profit and Loss statement for all products which the user can then limit to specific Line of Business (LOB) or slices of the P&L for a specific LOB across individual customer segments like age, gender, and Income.



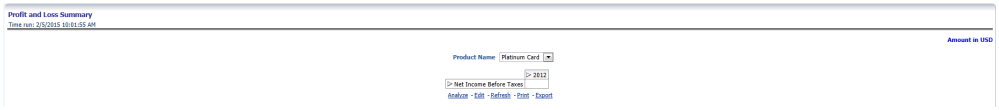
- Risk Adjusted Performance Metrics

  This report helps you to determine the ratio of risk-adjusted Net Income against the Economic Capital. This metric is also called Risk Adjusted Return On Capital (RAROC). It helps in determining the efficiency of Economic Capital corresponding to every customer.

Cards

- Profit and Loss Summary

  This report displays Profit and Loss statement for Cards products that can be viewed as slices of individual customer segments like age, gender and Income.



- Risk Adjusted Performance Metrics

  This report helps you to determine the ratio of risk-adjusted Net Income against the Economic Capital. This metric is also called Risk Adjusted Return On Capital (RAROC). It helps in determining the efficiency of Economic Capital corresponding to every customer.

- Performance by Card Type

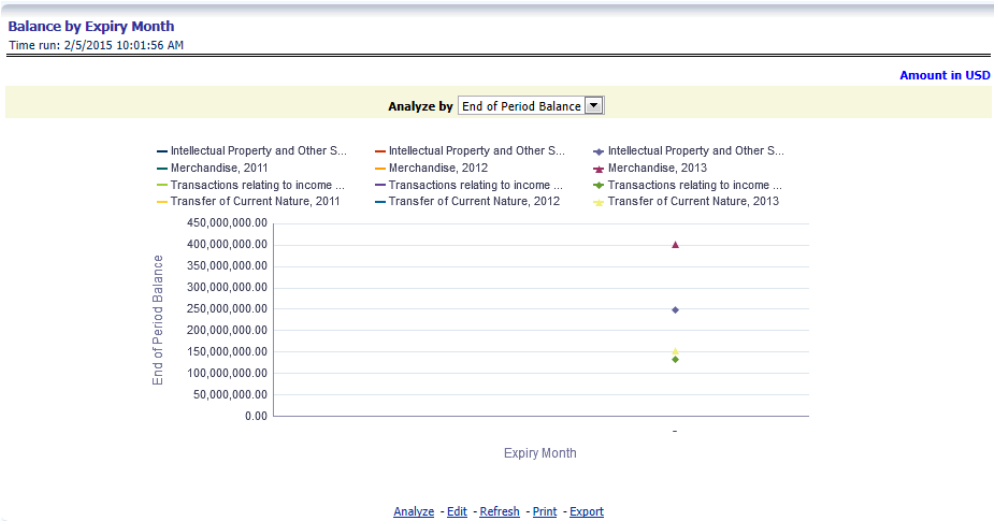
  This report displays Key Metrics such as Number of New accounts, Number of Open customers and so on reported across various card types demonstrating the mix across the card types.

- Standard Non-Cash Balance

  This report displays growth (or decline) of customer non-cash balance over time.

- Balance by Expiry Month

  This report displays balance distribution across offer expiry dates gives the user an indication of the balance that is scheduled to flip to a different balance bucket.

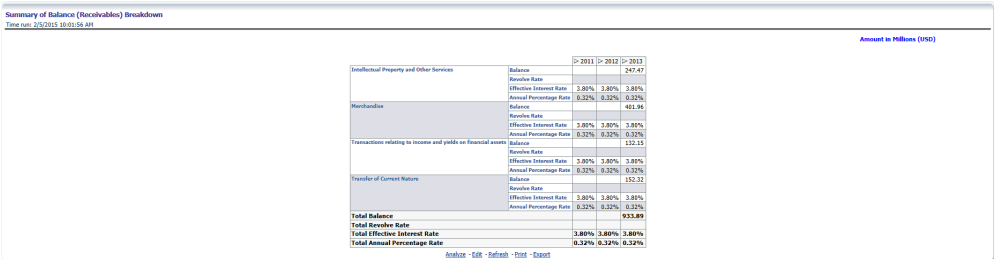


- ANR Summary

  This report displays the average growth (or decline) of balances per open customer in relation to how the standard non-cash balance is growing (or declining).

- Summary of Balance (Receivables) Breakdown

  This report summarizes the Average Net Receivable (ANR) mix across the different balance buckets available within the bank and the effective interest rate across each balance bucket.



Retail Bank

- Product Performance

  This report displays Profit and Loss statement for Retail Bank products or slices of the P&L across individual customer segments like age, gender, and Income.

- Risk Adjusted Performance Metrics

  This report helps you to determine the ratio of risk-adjusted Net Income against the Economic Capital. This metric is also called Risk Adjusted Return On Capital (RAROC). It helps in determining the efficiency of Economic Capital corresponding

to every customer.

- Total Deposit Analysis

  This report displays the distribution of deposits booked by the bank across various product categories.

Mortgage

- Product Performance

  This report displays Profit and Loss statement for Mortgage products or slices of the P&L across individual customer segments like age, gender, and Income.

- Risk Adjusted Performance Metrics

  This report helps you to determine the ratio of risk-adjusted Net Income against the Economic Capital. This metric is also called Risk Adjusted Return On Capital (RAROC). It helps in determining the efficiency of Economic Capital corresponding to every customer.

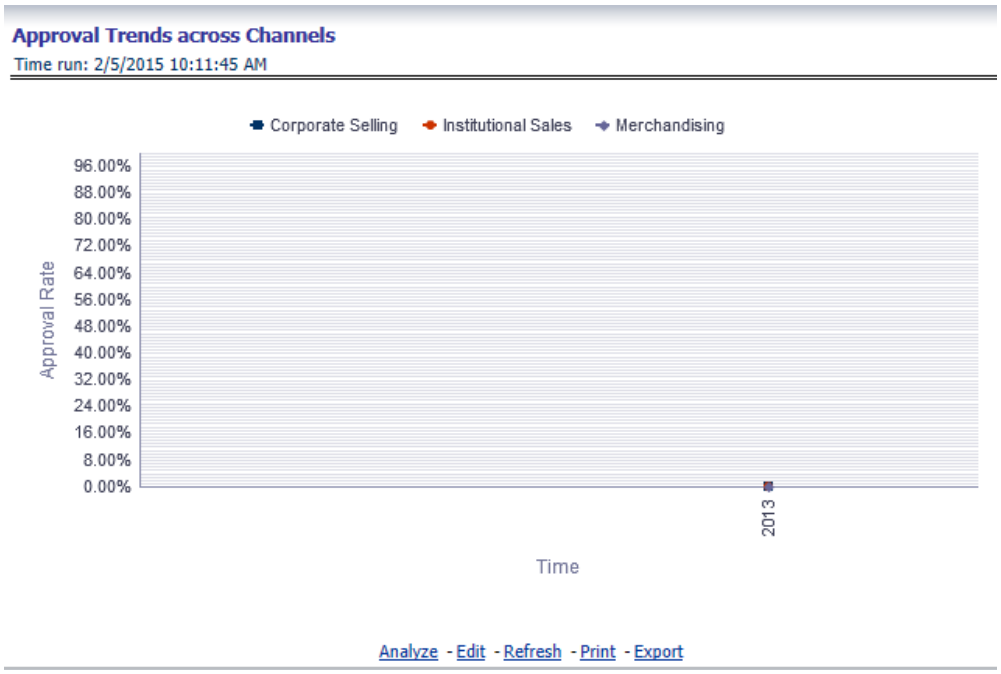
New Business Analysis

- Acquisitions Over last 5 Years

  This report displays growth of accounts and customers over a period of time. The dashboard prompts allows the user to narrow this analysis down to a specific LOB, Product Family, or Product.

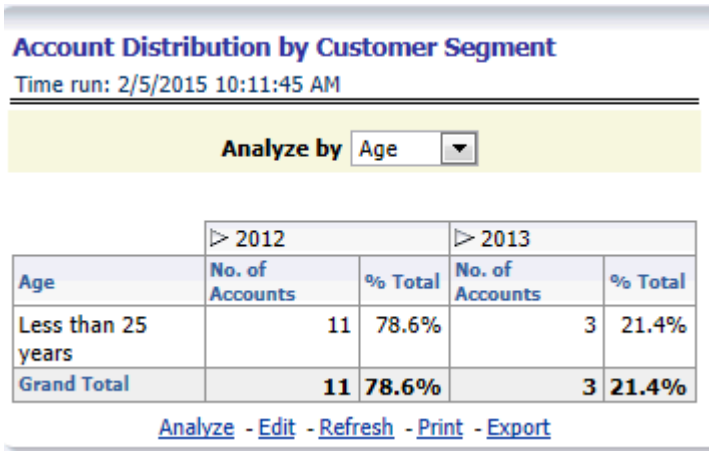
- Approval Trends across Channels

  This report displays variance in approval rates over time across the various bank channels and enables you to track how these rates have gone up or down in specific channels or compare a rise/fall in one channel vs performance in others. The dashboard prompts allows the user to narrow this analysis down to a specific LOB, Product Family or Product.



- Account Distribution by Customer Segment

  Provides a composition of the customers across key customer segments like Age, Gender, Income.



- Acquisition by Channel

  This report displays how acquisitions have migrated from one channel to another over a period of time.

- Attrition Over last 5 Years

  This report enables the user to have a view of the outflow (closed customers) or

attrition over the last 5 years in order to fully understand how customers are flowing in and out of a certain product or product family or LOB.

- Marketing Expenses Over last 5 Years

  Marketing expenses incurred by the bank over the last 5 years.

- Top 10 Reject Reasons

  This report displays variance analysis of the various reasons why an application was rejected. This can also be viewed as a time series to see trends over a period of time.

- Account Distribution by Credit Band

  This report outlines the number of accounts booked across various credit score bands. The credit score corresponds to the customer holding the account.

Revenue Analysis

- Revenue Analysis

  This report displays variance analysis of the various Revenue components like Fee Income, Net Interest Revenue, and so on by Product with ability to further understand this distribution across key customer segments like Gender, Age, and Income.

Expense Analysis

- Expense Analysis

  This report displays variance analysis of the various Expense components like Direct Expenses, Operating Expenses, and so on by Product with ability to further understand this distribution across key customer segments like Gender, Age, and Income.

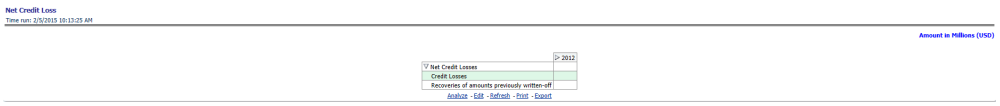
Credit Loss Summary

- Risk Summary

  This report gives a snapshot of the number of accounts in each delinquency bucket and the corresponding balance in each of these buckets. This can further be refined to limit this data to a specific LOB or a product within a LOB.

- Net Credit Loss

  This report displays Net Credit Loss (Gross Credit Loss adjusted for any recoveries) booked by the bank.



Customer Profit and Loss

- Profit and Loss Summary

  Enables you to view a P&L statement for a specific customer within the bank. This can be viewed at a customer level or for each individual account that the customer owns.

- Risk Adjusted Performance Metrics

  This report helps you to determine the ratio of risk-adjusted Net Income against the Economic Capital. This metric is also called Risk Adjusted Return On Capital (RAROC). It helps in determining the efficiency of Economic Capital corresponding to every customer.

Relationship Manager Performance

Relationship Manager Org Performance

Time run: 2/5/2015 10:15:06 AM

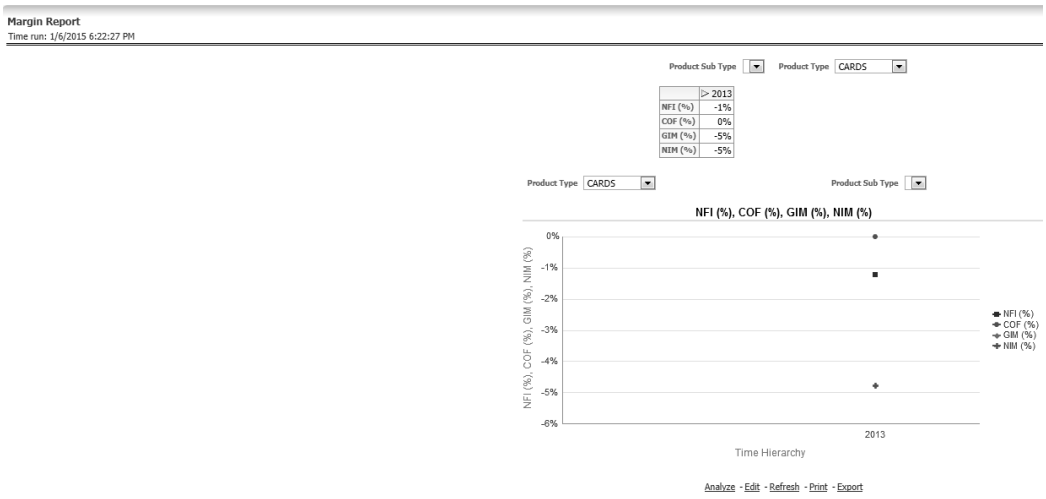
Amount in USD

Relationship Manager Hierarchy	Product	Customer	Account ID	Primary Officer Flag	Total Revenue	Percentage Contribution	Direct Contribution	Indirect Revenue	Overall Revenue Contribution
▷ USHA SHETTY	MF Regular	Reliance Capital Ltd	RBIB3C4A4	Y		100	0.00	0.00	

Analyze - Edit - Refresh - Print - Export

Margin Reports

This report tracks the margin of profitability that has been achieved by the customer over a period of time.



Product Type	Product Sub Type	GIM (%)	COF (%)	NIM (%)	NFI (%)
CARDS		-4.78	0.00	-4.78	-1.23
CASA		0.00	0.00	0.00	0.00
CONTRACTS		0.00	0.00	0.00	0.00
INVEST		0.00	0.00	0.00	0.00
LOAN		0.00	0.00	0.00	0.00
MORTGAGE		0.00	0.00	0.00	0.00
RB		0.00	0.00	0.00	0.00
TD		0.00	0.00	0.00	0.00

16

Segmentation

Introduction

The objective of segmentation is to define a framework that will score accounts at MIS Date and Run level and correspondingly create clusters based on the scores.

These scores and segments will be stored into a new fact table. The inputs for this model will come from the following tables:

- Location
- Credit Rating
- Currency
- Geography
- Industry
- Product
- Age
- Income
- Gender

Average of the measures present in Account Profitability for Account Segments across Reporting Lines and Month on Book for the account are reported into the following table.

FCT_ACCOUNT_SEGMENT_SCORE

Column Name	Logical Name
N_ACCT_SKEY	Account Key
D_ACCT_START_DATE	Account Start Date
N_ACCT_SEGMENT	Account Segment
N_ACCT_SEGMENT_SCORE	Account Segment Score
N_RUN_SKEY	Run Key
N_MIS_DATE_SKEY	Date Key

The above table will act as the input for another table that stores facts of Account Profitability (Movement, Average Balance, and so on) at the level of Month on Book, Account Segment, Run, and Reporting Line. Following is the structure of this table.

FCT_ACCT_SEGMENT_MOB_SUMMARY

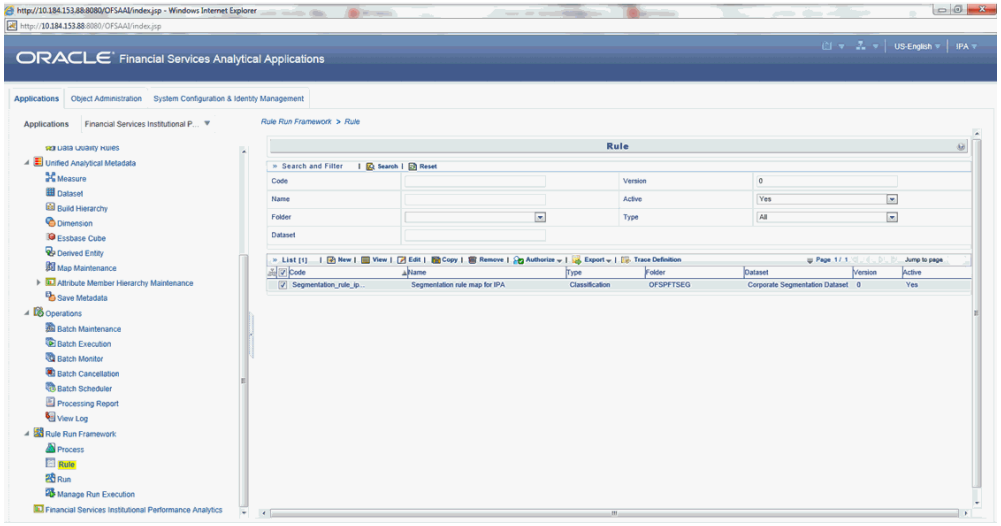
Column Name	Logical Name
N_AVG_BAL_RCY	AccountAverageBalanceReportingCurrency
N_END_BAL	AccountEndingBalance
N_AVG_BAL	AccountAverageBalance
N_END_BAL_RCY	AccountEndingBalanceReportingCurrency
N_MOVEMENT	Movement
N_MOVEMENT_RCY	MovementReportingCurrency
N_REP_LINE_CD	ReportingLineCode
N_RUN_SKEY	Run Key
N_MONTH_ON_BOOK	Month on Book

Column Name	Logical Name
N_ACCT_SEGMENT	Account Segment

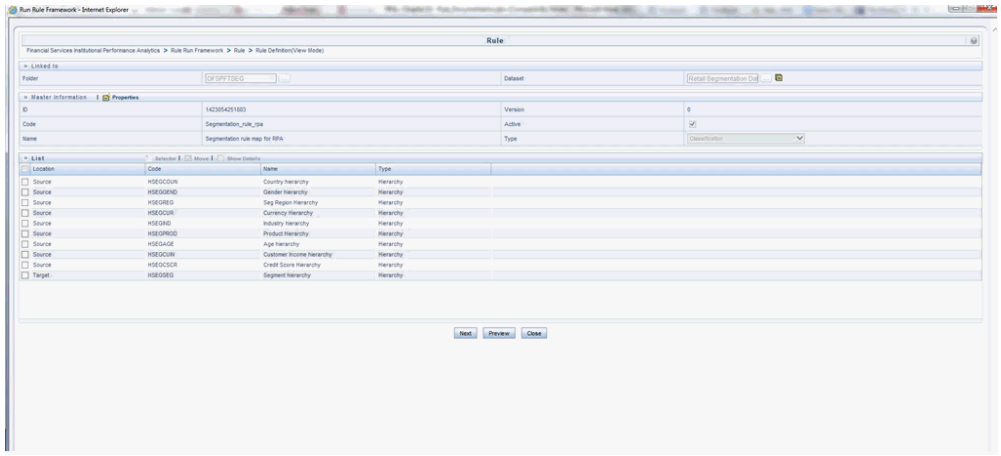
Creating a rule

To define a rule, follow these steps:

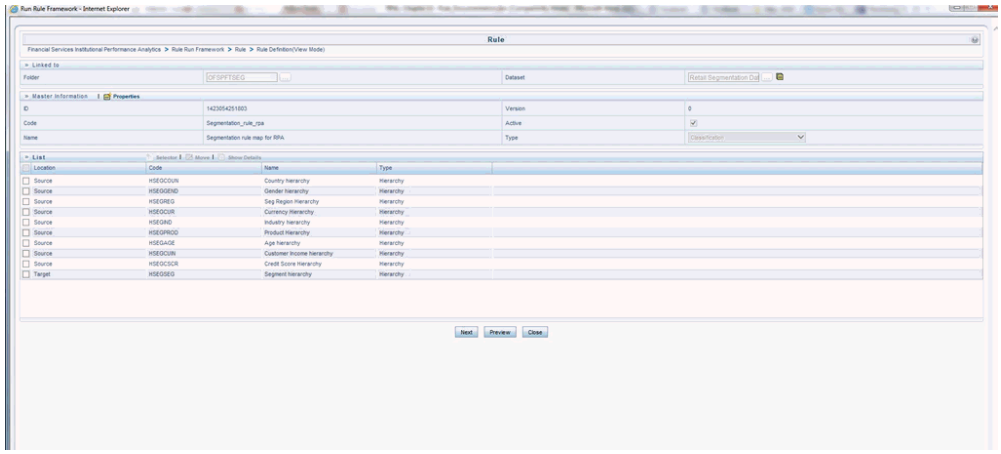
1. Click **Rule** and the following rule appears.



2. Select the rule and click **View**. The following screen appears.



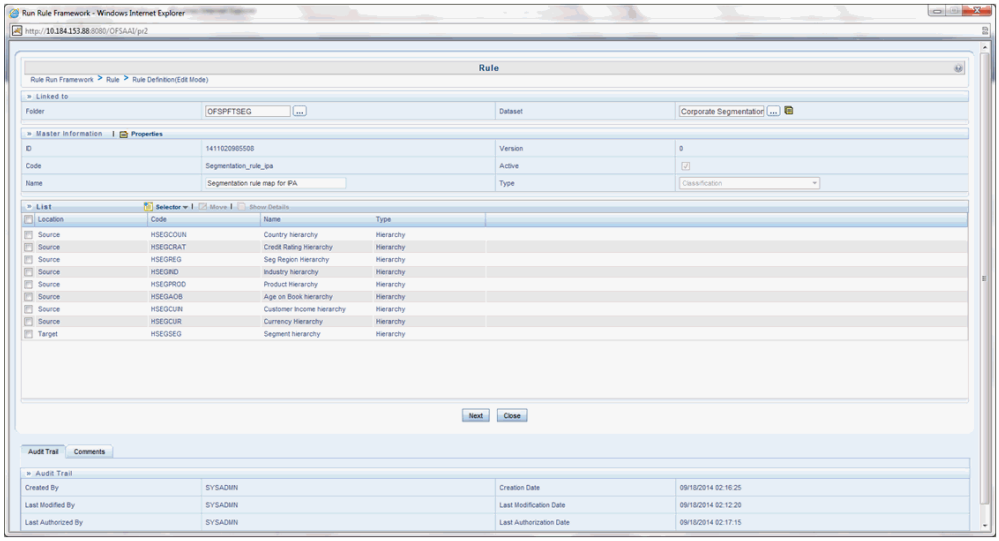
3. On clicking **Next**, the rule defined comes up. For the first time when the rule is not defined, only default seeded node rule should appear as shown in the following screen:



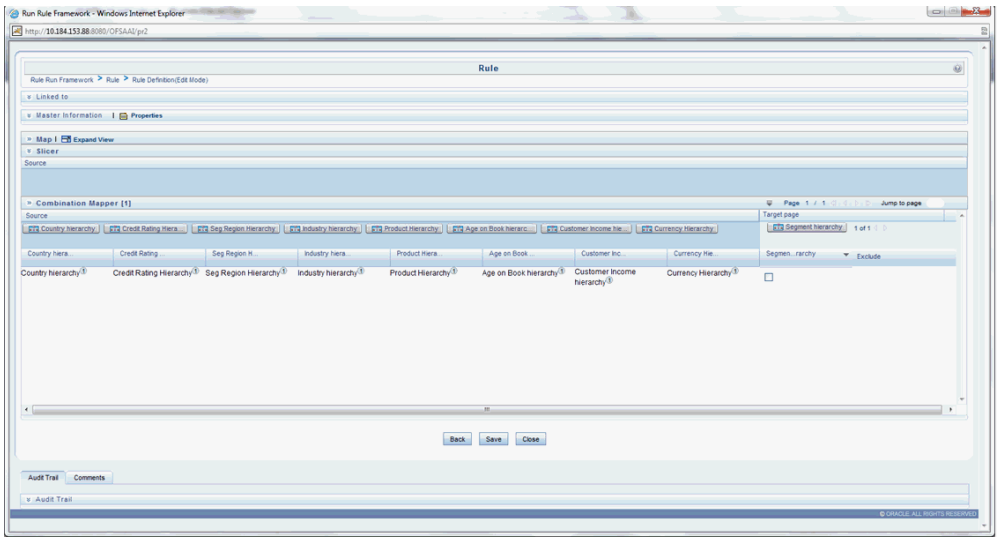
Editing a rule

To edit a rule, follow these steps:

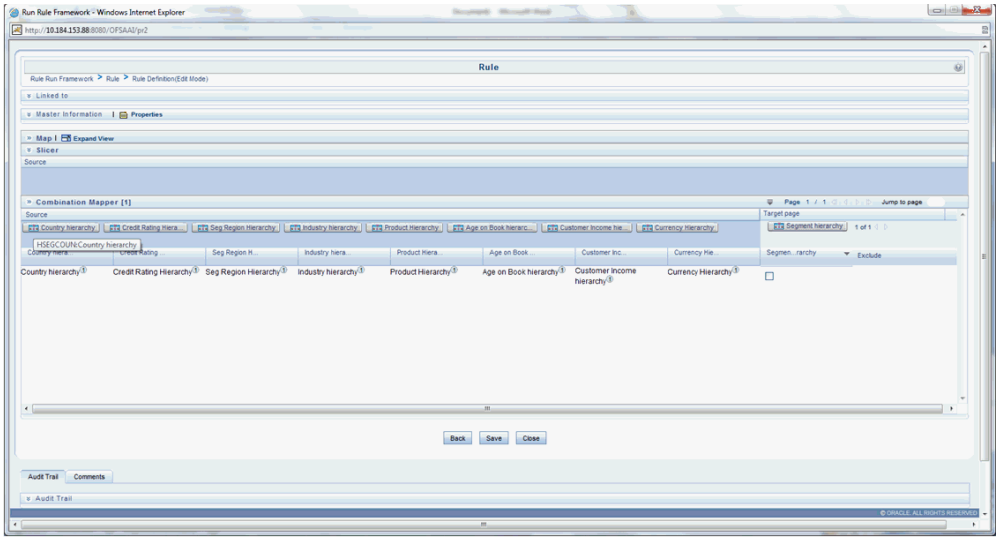
1. Select the rule and click **Edit**. The following screen appears:

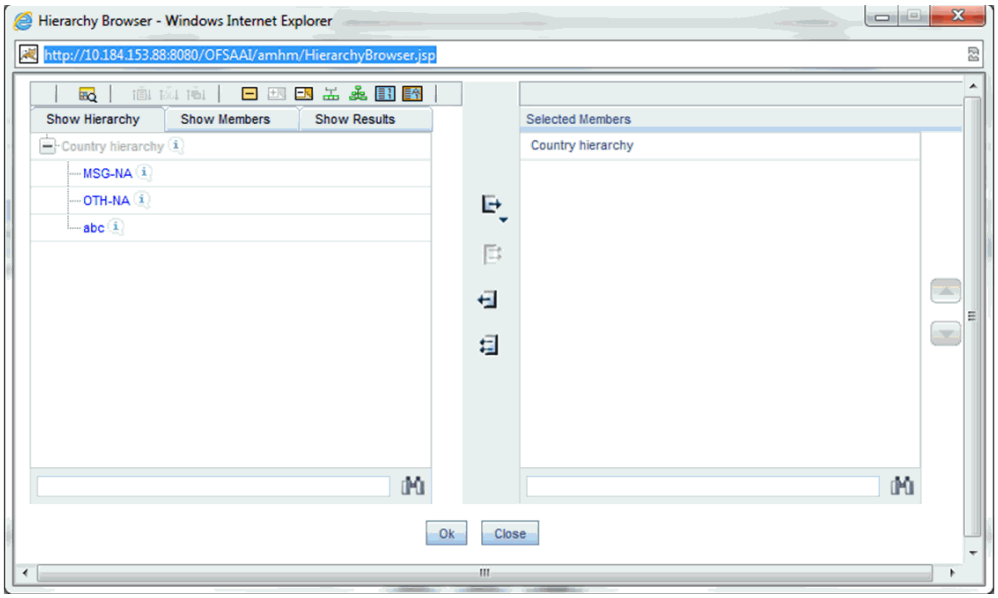


2. Click **Next**. First time default node defined as rule will show up in the following figure:

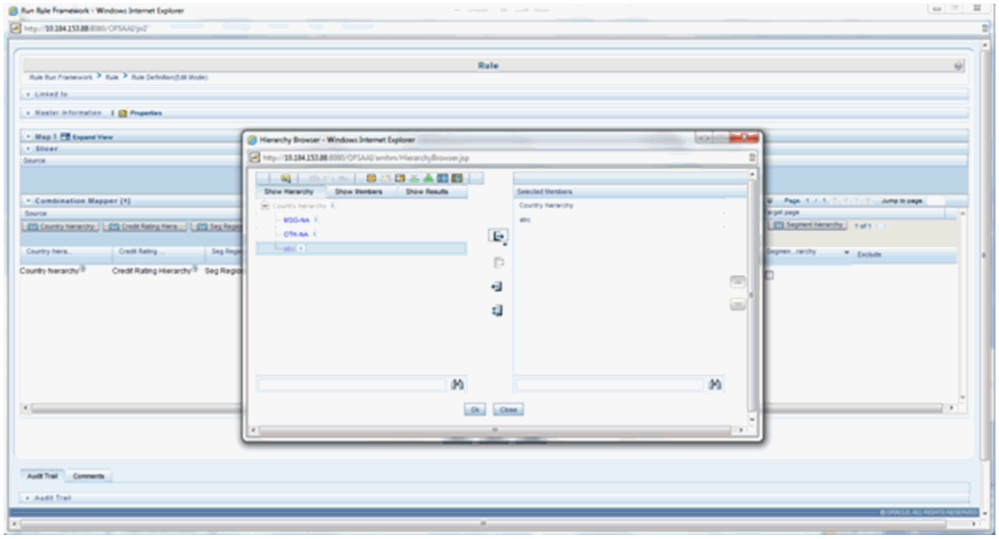


3. Click **Hierarchy** as shown in the following figure and the hierarchy screen opens up.

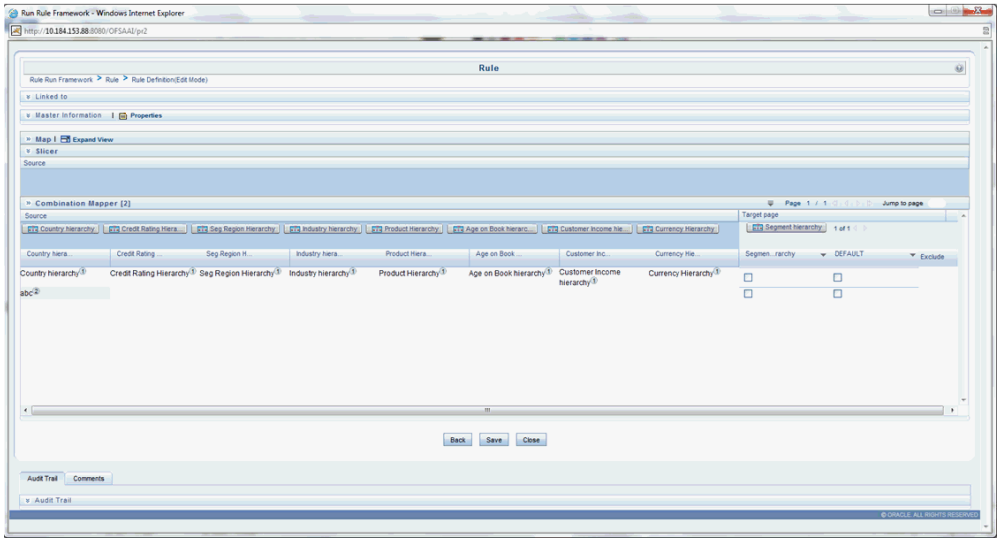




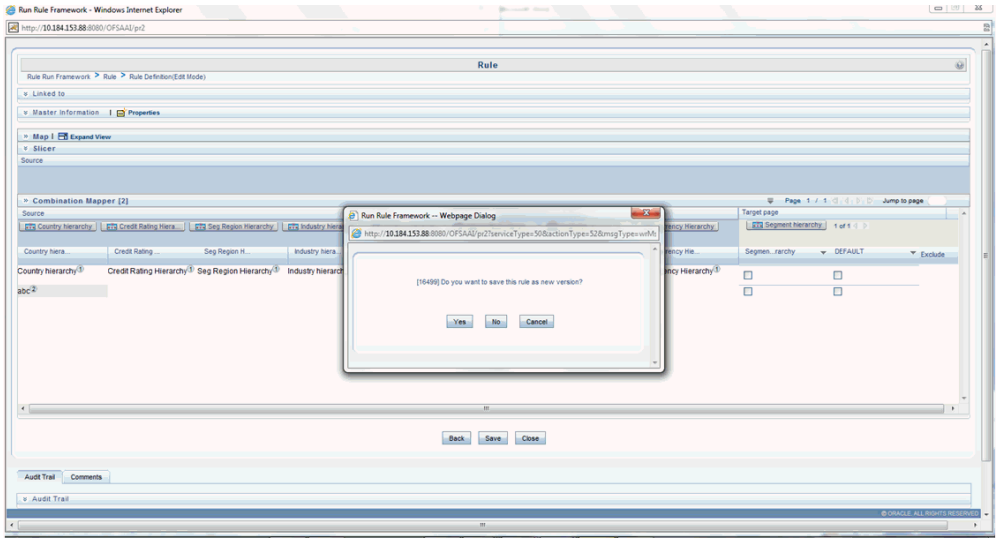
4. Select the hierarchy and click **OK**.



5. The selected node appears in the rule.

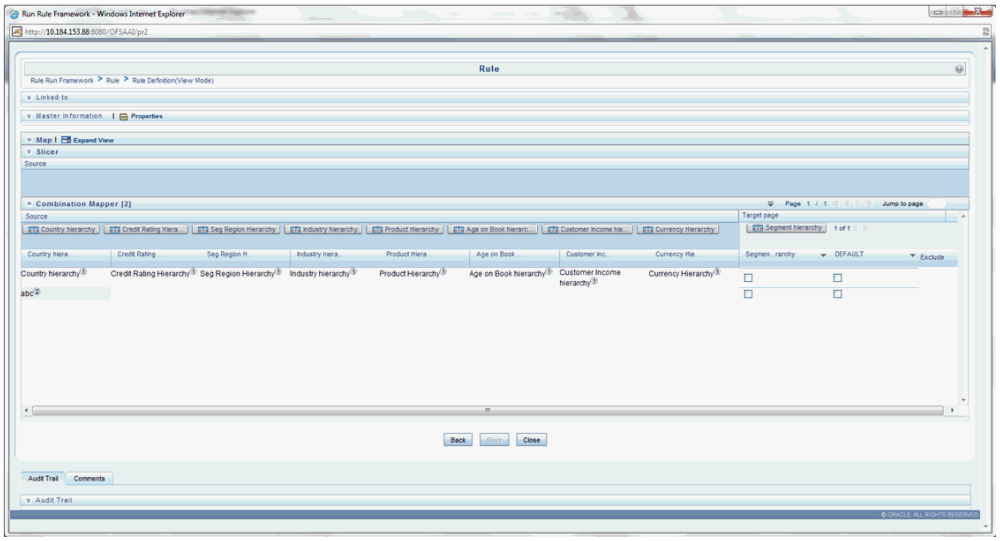


6. Similarly, select all the nodes that need to be considered for the rule and assign it to the target hierarchy. Click **Save**. A confirmation message appears as shown in the following.



7. On clicking **Yes**, the following message appears: *Save action with authorize was successful on following definitions Segmentation_rule_rpa*.

8. Navigate back to the main screen and click the view rule. The rule saved is shown.


Run Rule Framework - Windows Internet Explorer
Rule
Rule Run Framework > Rule > Rule Definition(View Mode)
Linked to
Master Information | Properties
Map | Expand View
Slicer
Source
Combination Mapper [2]
Source
Target page
Country hierarchy
Credit Rating Hiera...
Seg Region Hierarchy
Industry hierarchy
Product Hierarchy
Age on Book hierarc...
Customer Income hie...
Currency Hierarchy
Segment hierarchy
Country hiera...
Credit Rating ...
Seg Region H...
Industry hiera...
Product Hiera...
Age on Book ...
Customer Inc...
Currency Hie...
Segmen...rarchy
DEFAULT
Exclude
Country hierarchy
Credit Rating Hierarchy
Seg Region Hierarchy
Industry hierarchy
Product Hierarchy
Age on Book hierarchy
Customer Income hierarchy
Currency Hierarchy
abc
Back
Save
Close
Audit Trail
Comments
Audit Trail

17

Service Calls to RPA

Introduction

Customer insight web service is designed to get consumed by other applications in order to get the profitability details. This web service will work at two different levels: customer level and account level.

To fetch the customer details, set the request level as customer level and the customer id for that particular customer must be part of input. To fetch account details, set the request level as account level and the account number for that particular account must be part of input.

Within one request you can request for either one or multiple customers details by sending the customer id as an input in structured input xml. Similarly, it works for accounts as well. If one customer id is invalid, then the request to fetch data for multiple customers will get completely discarded. This way is similar for account numbers at account level.

Server side settings

In the server side, there is a file CUSTINSconfig.properties which allows server side user to configure web-service. The following attributes can be configured:

- Infodom

  Currently, CI web service will be enabled for one infodom and one user only at a time for a setup. That information domain has to be provided here.

- UserId

  Currently, CI web service will be enabled for one infodom and one user only at a time for a setup. The server side user has to provide this information as the web service do not validate user/password.

- Locale

  Provide 'en_US'

- runId

  Provide 'VIEW_PROF_WS_RUN' as this is a generic web service which allows any package to be called at run-time.

- threadWaitTime

  Provide a number here. Unit of the value will be milliseconds. This is a time that a web service call waits for an output to be generated. If an output is ready within that time, it is sent to the user else a system generated Reference Id is sent to the user, using which the user can get the data later as explained in the Input Structure.

Client Side Settings

In the server side, there is a file CUSTINSClientConfig.properties which allows server side user to configure web-service. The following attributes can be configured:

- wsdlSchemaLocation

  This is the url of wsdl. To generate it, take contextURl, for example, http://10.241.32.163:9085/OFSAAI73new and append " /CustomerProfitabilityService?wsdl" to it. In this case, the wsdlSchemaLocation will be "http://10.241.32.163:9085/OFSAAI73new/CustomerProfitabilityService?wsdl".

- targetNamespaceURI

  Provide this as http://webservice.customerinsight.custIns.fsapps.ofs.com/ at all time.

- serviceName

  Provide this as "CustomerProfitabilityService".

Input Structure

Input for this web service is an xml file. The required information is embedded into suitable xml tags. Input xml structure, that is, request xml will vary based on request level and the type of request. If the request is new, then it will have one input format and for polling the same request the structure will be different.

Examples

1. Input xml for new request to get particular customer's detail

   <?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>

<REQUEST>

<REQUEST_TYPE>VIEW_PROFITABILITY</REQUEST_TYPE>

<REFERENCE_NUMBER></REFERENCE_NUMBER>

<REQUEST_LEVEL>CUSTOMER_LEVEL</REQUEST_LEVEL>

<PARTY_ID>OBIB1C47</PARTY_ID>

<REQUEST>

Here for a new request, the request type must be VIEW_PROFITABILITY otherwise it throws an error saying invalid request type. Here REFERENCE_NUMBER must be blank as it is new request. As you would like to fetch customer data you must put the request level as CUSTOMER_LEVEL. And the value for the tag PARTY_ID specifies the customer id whose detail you need to fetch.

When you need to fetch data for multiple customers the request xml will be as mentioned below.

<?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>

<REQUEST>

<REQUEST_TYPE>VIEW_PROFITABILITY</REQUEST_TYPE>

<REFERENCE_NUMBER></REFERENCE_NUMBER>

<REQUEST_LEVEL>CUSTOMER_LEVEL</REQUEST_LEVEL>

<PARTY_ID>OBIB1C47</PARTY_ID>

<PARTY_ID>OBIB1C49</PARTY_ID>

<PARTY_ID>OBIB1C49</PARTY_ID>

<PARTY_ID>OBIB1C50</PARTY_ID>

<REQUEST>

Each PARTY_ID tag contains the customer id for one customer. In this way, you can request for multiple customers data.

2. Input xml for polling request to get customer's detail

When you send a new request, sometimes the request takes more processing time. So in that case, the output will be one reference number which you can use for polling for the same request. The input xml structure will remain same irrespective of your request, whether for one customer or multiple customers. It is as follows:

<?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>

<REQUEST>

<REQUEST_TYPE>RE_REQUEST</REQUEST_TYPE>

<REFERENCE_NUMBER>56</REFERENCE_NUMBER>

<REQUEST_LEVEL></REQUEST_LEVEL>

<REQUEST>

Here the REQUEST_TYPE tag must have the value RE_REQUEST. Provide the number which you have received as an output initially inside REFERENCE_NUMBER tag.

3. Input xml for new request to get particular account's detail

<?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>

<REQUEST>

<REQUEST_TYPE>VIEW_PROFITABILITY</REQUEST_TYPE>

<REFERENCE_NUMBER></REFERENCE_NUMBER>

<REQUEST_LEVEL>ACCOUNT_LEVEL</REQUEST_LEVEL>

<ACCOUNT_NUMBER>OBIB2C19A1</ACCOUNT_NUMBER>

<REQUEST>

Here for a new request, the request type must be VIEW_PROFITABILITY otherwise it throws an error saying invalid request type. Here REFERENCE_NUMBER must be blank as it is a new request. To fetch account data, provide the request level as ACCOUNT_LEVEL. And the value for the tag ACCOUNT_NUMBER specifies the account number whose detail you need to fetch.

When you need to fetch data for multiple accounts data the request xml will be as mentioned as follows:

<?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>

<REQUEST>

<REQUEST_TYPE>VIEW_PROFITABILITY</REQUEST_TYPE>

<REFERENCE_NUMBER></REFERENCE_NUMBER>

<REQUEST_LEVEL>ACCOUNT_LEVEL</REQUEST_LEVEL>

<ACCOUNT_NUMBER>OBIB2C19A1</ACCOUNT_NUMBER>

<ACCOUNT_NUMBER>OBIB2C19B1</ACCOUNT_NUMBER>

<REQUEST>

Each ACCOUNT_NUMBER tag contains the account number for one account. In this way, you can request for multiple accounts data.

4. Input xml for polling request to get account's detail

When you send a new request it may happen that your request may take some more processing time. So in that case the output will be one reference number which you can use for polling for the same request. The input xml structure will remain same irrespective of your request was for one account or multiple accounts.

It is as follows:

<?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>

<REQUEST>

<REQUEST_TYPE>RE_REQUEST</REQUEST_TYPE>

<REFERENCE_NUMBER>56</REFERENCE_NUMBER>

<REQUEST_LEVEL></REQUEST_LEVEL>

<REQUEST>

Here the REQUEST_TYPE tag must have the value RE_REQUEST. Provide the number which you have received as an output initially inside REFERENCE_NUMBER tag.

Output Structure

1. Output xml when the request is still processing at both customer and account level

   <?xml version = "1.0" encoding = "UTF-8"?>

   <RESPONSE>

   <STATUS>TIMEOUT</STATUS>

   <REFERENCE_NUMBER>36</REFERENCE_NUMBER>

   <RESPONSE>

2. Output xml structure when you send invalid customer id

   <?xml version = "1.0" encoding = "UTF-8"?>

   <RESPONSE>

   <STATUS>ERROR</STATUS>

   <ERROR_MESSAGE>FAILED TO FETCH CUSTOMER DETAILS</ERROR_MESSAGE>

   <RESPONSE>

3. Output xml structure when you send invalid account number

   <?xml version = "1.0" encoding = "UTF-8"?>

   <RESPONSE>

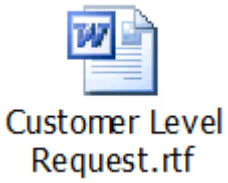
   <STATUS>ERROR</STATUS>

   <ERROR_MESSAGE>FAILED TO FETCH ACCOUNT DETAILS</ERROR_MESSAGE>

   <RESPONSE>

4. In case of successful response for customer or account level request, the output will be in the form of structured xml document

   Output for successful customer level request



   Output for successful account level request

Execute Service

To process the customer/account/re-request level request, pass one argument.

File name which contains request Input XML.

A

How to Add a New Dimension

Introduction

This section explains the steps to be performed by the user for adding a new dimension to the cube.

As a prerequisite, dimension tables should be added in the data model and the fact table needs to have the referential key with the dimension table. These dimension tables will hold dimension members and can be level-based or parent-child.

Level based dimension tables contain columns for each level of the hierarchy, while parent-child dimension tables contain columns for storing the relationship between the parent and child members. These dimension tables can be loaded from external systems or can be maintained within the Dimension Management component of OFSAAI.

If user intends to maintain the dimension within OFSAAI, see Data Model Utilities Guide for adding dimension tables under "Object Management" chapter.

Procedures to Add a New Dimension

Step 1 – Add Business Hierarchy

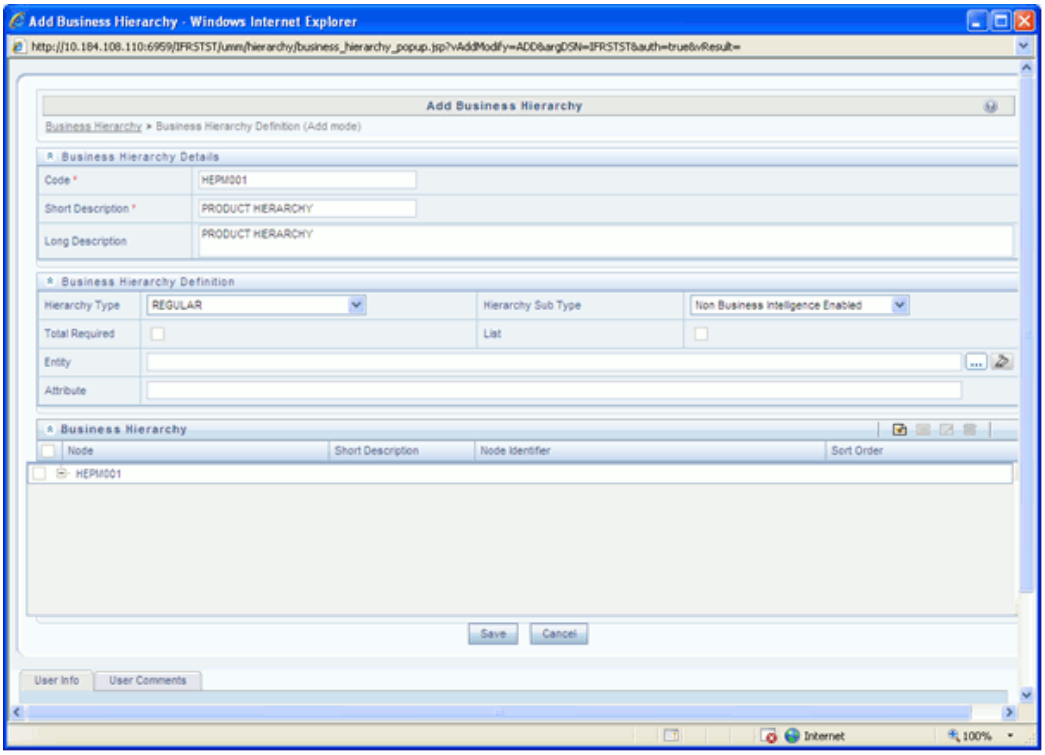
To define a new **Business Hierarchy**, go to **Unified Metadata Manager**, select **Business Metadata Management** and choose the type of hierarchy.

Hierarchy Types are:

- **Regular** – For representing non-time and non-measure dimensions in a hierarchical format. For example, this type are Product, Organization Unit, and so on.

- **Measure** – For representing the measures in the hierarchical format. This corresponds to a ACCOUNT hierarchy within the ESSBASE. An example of this type is Management Reporting Line.

- **Time** – For representing the calendar or date dimension in a hierarchical format. Time hierarchy corresponds to a TIME hierarchy within Essbase and this can be leveraged to pull data from the relational database. This An example of this type is Calendar hierarchy.



Choose Hierarchy subtype. Hierarchy SubTypes are:

- **Non Business Intelligence Enabled** – For representing the hierarchy with underlying data store containing just leaves and nodes are built within the metadata of the hierarchy. This subtype is useful for modelling bucket/range, ragged, and non-additive hierarchies.

- **Business Intelligence Enabled** – For representing the hierarchy with underlying data store as level-based dimension table. This subtype is useful for modelling balanced hierarchies.

- **Parent Child** – For representing the hierarchy with underlying data store as a parent-child dimension table. This subtype is useful for modelling ragged hierarchies.

Select the "Total Required" property, if a TOTAL is required to be included as the root node of the hierarch and select the "List" property, if hierarchy is a flat list of members without any levels.

Choose the entity and attribute on which the hierarchy is based. The components for hierarchy definition differ for each subtype of the hierarchy.

If subtype is "Non Business Intelligence Enabled", then the user can add nodes and order in which the node should appear in the hierarchy (sort-order). Node identifiers are SQL expressions that are specified for leaf members and data is classified based on the node identifiers.

If sub-type is "Business Intelligence Enabled", then the user can specify the levels and SQL expression for each level within the hierarchy.

If sub-type is "Parent Child", then the user can specify the column that contains the parent member and that contains the child member.

For more details, see *Oracle Financial Services Analytical Applications Infrastructure 7.3 User Guide.*

Step 2 – Add Business Dimension

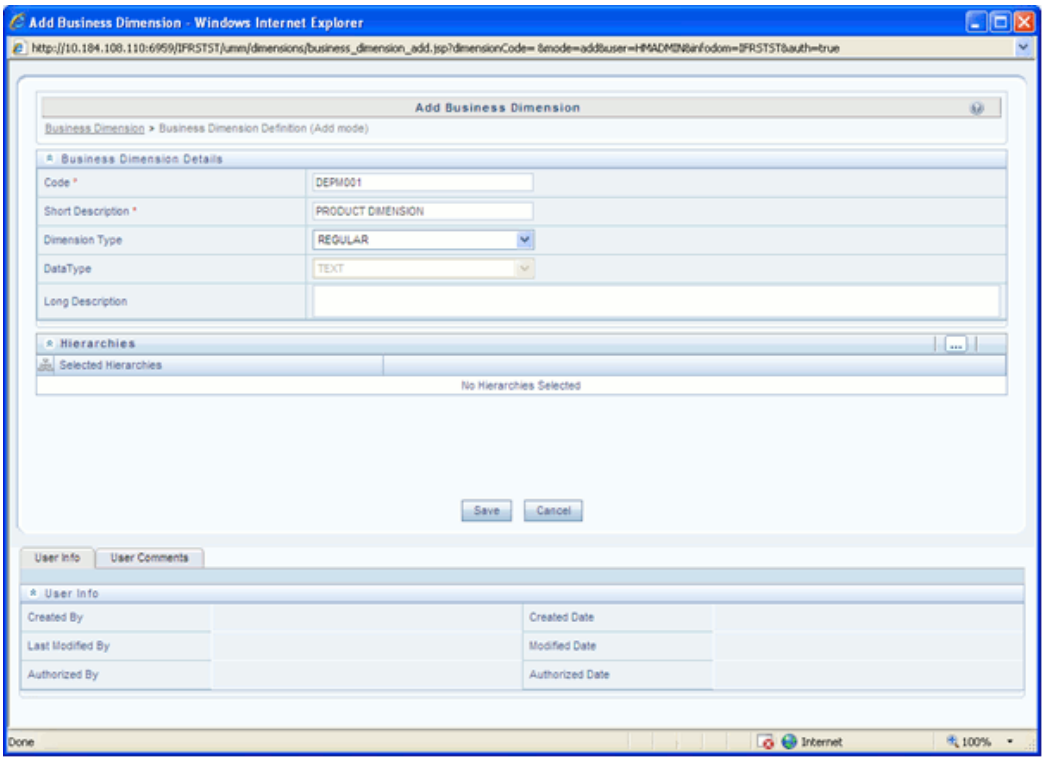
A Business Dimension is a structure of one or more logical grouping (hierarchies) that classifies data. It is the categorization across which measures are viewed. A dimension can have one or more hierarchies. Business Dimension facilitates you to create a logical connection with measures. It gives you various options across which you can view measures.

To define a new **Business Dimension,** go to **Unified Metadata Manager**, select **Business Metadata Management**.

Choose the **Dimension Type**. Dimension Type is same as Hierarchy Type and helps to filter the hierarchies that will be part of the dimension. A dimension will contain one or many hierarchies. Choose the hierarchies that are part of the dimension.

The User Info grid at the bottom of the screen displays the metadata information about the Business Dimension created along with the option to add comments.

Click **Save** in **Add Business Dimension** screen to save the details.

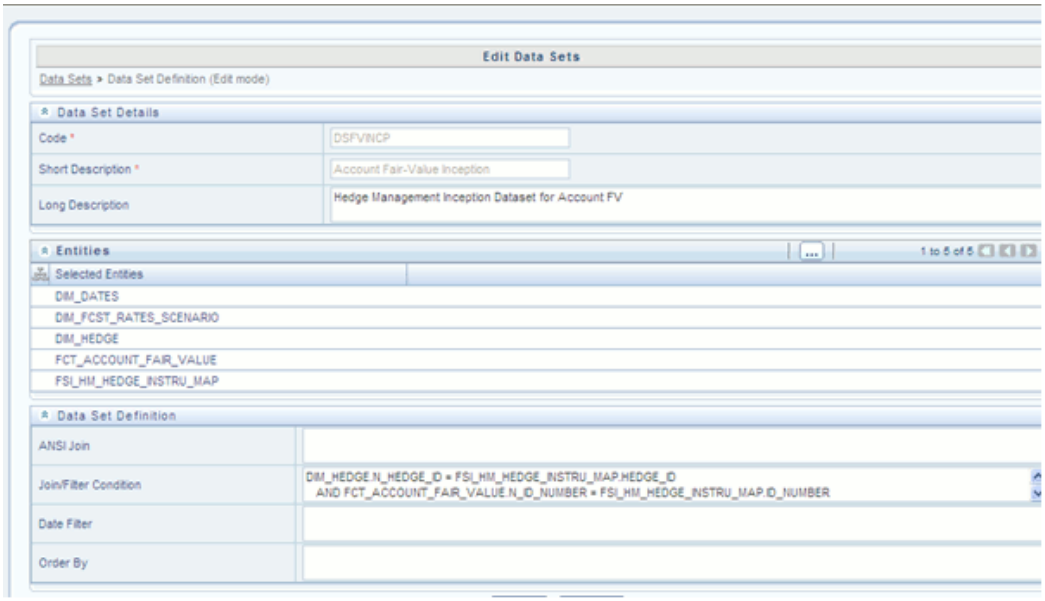


For more details, see *Oracle Financial Services Analytical Applications Infrastructure 7.3 User Guide.*

Step 3 – Modify Data Set

To modify **Data Sets**, go to **Unified Metadata Manager --> Business Metadata Management.**

Identify data sets that are based on the modified fact table. Open the data set definition. Include the new dimension table in the data set. Modify the data set JOIN to include the join clause between the fact table and new dimension table. Save the data set.



For more details, see *Oracle Financial Services Analytical Applications Infrastructure 7.3 User Guide.*

Step 4 – Modify Cube Definition

Modify "Cubes" in **Unified Metadata Manager -> Business Metadata Management**. Identify the cube that needs to be modified. Open the cube definition. Add the new dimension. Map the measures to the newly added dimension and **Save** the cube definition.

For more details, see *Oracle Financial Services Analytical Applications Infrastructure 7.3 User Guide.*

Step 5 – Build Cube

Assuming that the dimension table and fact table is loaded with relevant data, cube can be built. Define batch to execute the CREATE CUBE component that will build the outline and load data in ESSBASE.

For more information on executing batch, see *Oracle Financial Services Analytical Applications Infrastructure 7.3 User Guide*.

B

How to Add a New Measure

Introduction

This section details the steps to be performed by the user for adding a new measure to the cube. As a prerequisite, the fact table needs to have the column that holds values for the new measure.

Business Measure refers to a uniquely named data element of relevance which can be used to define views within the data warehouse. It typically implies aggregated information as opposed to information at a detailed granular level that is available before adequate transformations. Business Measure facilitates you to create measures based on the area of analysis. While creating a measure, you can choose the aggregation type and apply business exclusion rules based on your query/area of analysis.

Dimension Definition Process

Step 1 – Add Business Measure

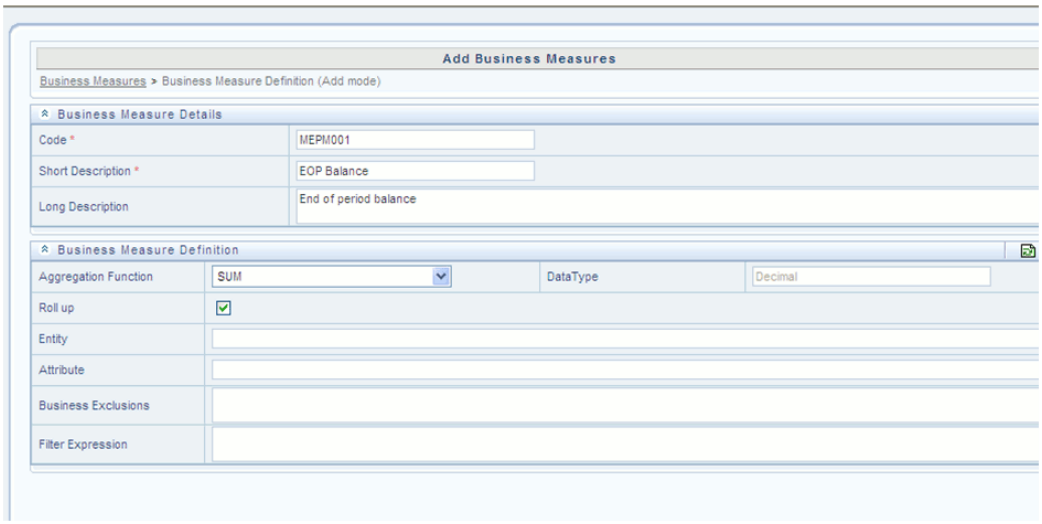
1. From **Unified Metadata Manager**, select **Business Metadata Management**, then select **Business Measures**.

2. From Business Measures, click **Add** to create a Business measure definition. In the Business Measure Definition (Add mode) window, Select **Aggregation Function**. Aggregation Function can be:

   - SUM – for summing up the values in the column of the fact table.
   - COUNT – for determining the number of records in the fact table.
   - MAXIMUM – for identifying the maximum value of a column in the fact table.
   - MINIMUM – for identifying the minimum value of a column in the fact table.

- COUNT DISTINCT – for determining the distinct count of records in the fact table.

3. Specify if this measure needs to be rolled up against hierarchies.
4. Select the fact table as part of the Entity.
5. Select the column of the fact table as part of the Attribute. This column will hold the value of the measure.
6. Specify Business Exclusions and Filters, if required.
7. Save the measure.



For more information on Business Measures, see Business Measures under Unified Metadata Manager chapter in *Oracle Financial Services Analytical Applications Infrastructure 7.3 User Guide*.

Step 2 – Modify Cube Definition

1. From **Unified Metadata Manager**, select **Business Metadata Management**, then select **Cube**.
2. Identify the cube that needs to be modified.
3. Edit the cube definition.
4. Add the new measure.
5. Map the measures to the to the required dimensions.

6. Save the cube definition.

Build Cube

Assuming that the dimension table and fact table is loaded with relevant data, cube can be built.

Define batch to execute the CREATE CUBE component that will build the outline and load data in ESSBASE.

For more information on executing the batch, see *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

C

How to Develop a New Cube

Introduction to Developing a New Cube

This section details the steps to be performed by the user for developing a new cube. Make sure that the existing cubes do not provide the required analytics / reporting coverage before deciding to define a new cube. In case user would like to see measures against a new dimension that is not part of the existing seeded metadata, then suggest including the new dimension as part of the existing cubes instead of creating a new cube. As a prerequisite, user should have defined datasets, measures, hierarchies and dimensions before defining a cube.

Procedures to Develop a New Cube

Step 1 – Add Cube

From **Unified Metadata Manager**, select **Business Metadata Management**, then select **Cube**. Specify the MDB details that will be created in ESSBASE.

Step 2 – Include Dimensions

Include dimensions that are part of the cube definition. Users mandatorily need to include TIME and MEASURE dimensions.

Step 3 – Specify Variations

Specify variations between each of the measures to the respective dimensions. All the measures that are part of the cube need not vary against all of the dimensions. Depending on business needs, variations can be specified to control the rollup of measures against a set of dimensions.

Step 4 – Specify Dataset

Specify dataset corresponding to the selected dimensions and measures. Data set will supply required data to the cube.

Step 5 – Specify Node Level Formula

If node level formula's are required to be specified for the nodes within the hierarchy, then they can be specified in this UI.

Step 6 – Save and Build

Save the cube. Define and execute batch in ICC to build the cubes.

For more information on Cubes, see Cubes under Unified Metadata Manager chapter in *Oracle Financial Services Analytical Applications Infrastructure 7.3 User Guide*.

D

List of Members

List of Hard-Coded Members

Following are the dimension members that are hard-coded within the application:

Table Name	Column Name	Expected Values
DIM_CUSTOMER_TYPE	V_CUST_CATEGORY	R
	F_LATEST_RECORD_INDICATOR	Y
FCT_CRM_ACCOUNT_SUMMARY	V_SCENARIO_CODE	PLAN,
		BUDGET
DIM_BANDS	V_BAND_TYPE	AGE
		INCOME
		AGEONBOOK
		ACCT_ATTRITION_SCORE
		CUST_CR_RISK_SCORE
		NO_OF_ACCOUNTS
		RESPRATE

Table Name	Column Name	Expected Values
		DELQBAND
FCT_TXN_CHANNEL	V_F_CHNL_TYPE	MONETARY,NONMONETARY
	F_F_TXN_DR_CR_IND	C,D
DIM_PRODUCT	V_PROD_TYPE	CARDS
		RB
		DEPOSITS
		CASA
		AUTOLOAN
		TD
		MORTGAGE
FCT_ACCOUNT_PROFITABILITY	N_REP_LINE_CD	98000 - Net Income Before Taxes
		98500 - Tax Expense
		99000 - Net Income After Taxes
		107100 - Number of Customers
		107130 - Number of Open Customers
		107200 - Number of Accounts
		107230 - Number of Open Accounts
		107300 - Attrition Rate
DIM_PRODUCT_TYPE	V_ACCT_PROD_TYPE	CARDS

Table Name	Column Name	Expected Values
		RB
		DEPOSITS
		CASA
		AUTOLOAN
		TD
		MORTGAGE

E

How to Define a Batch

Introduction

Batch refers to a set of executable processes based on a specified rule. Batch Maintenance framework within OFSAAI facilitates you to create and maintain the Batch Definitions. You can process the Batch scheduled for execution from Batch Maintenance and also from other modules.

You need to have Data Centre Manager function role mapped to access the Operations framework within OFSAAI. You can access Batch Maintenance by expanding Operations section within the tree structure of LHS menu. The *Batch Maintenance* window displays a list of Batches scheduled for maintenance with the other details such as Batch ID, Batch Description, and the editable state of the Batch.

Batch Creation

You can create a batch from the *Batch Maintenance* screen as mentioned below:

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

   The *New Batch Definition* window is displayed.
3. Enter the Batch details as tabulated.

Field	Description
Batch Name	The **Batch Name** is auto generated by the system. You can edit to specify a Batch name based on the following conditions: • The Batch Name should be unique across the Information Domain. • The Batch Name must be alpha-numeric and should not start with a number. • The Batch Name should not exceed 41 characters in length. • The Batch Name should not contain special characters "." and "-".
Batch Description	Enter a description for the Batch based on the Batch Name.
Duplicate Batch	(Optional) Select the checkbox to create a new Batch by duplicating the existing Batch details. On selection, the **Batch ID** field is enabled.
Batch ID (If duplicate Batch is selected)	It is mandatory to specify the Batch ID if Duplicate Batch option is selected. Select the required **Batch ID** from the list.
Sequential Batch	Select the check box if the Batch has to be created sequentially based on the task specified. For example, if there are 3 tasks defined in a Batch, task 3 should have precedence as task 2, and task 2 should have precedence as task 1.

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

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

> **Note:** For a more comprehensive coverage of configuration and execution of a batch, refer to the *Operations* Chapter in *Oracle Financial Services Analytical Applications Infrastructure User Guide.*

F

Run Rule Framework

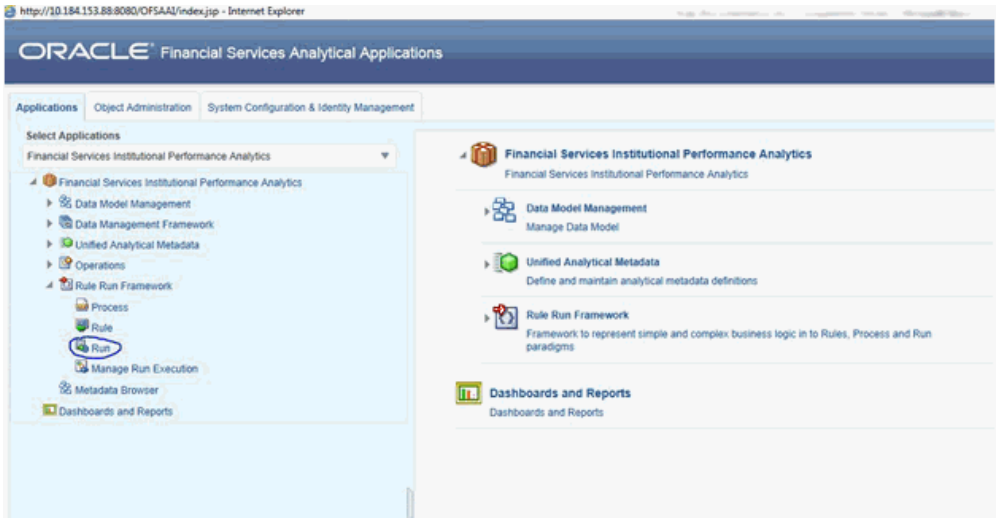
Introduction

In cases where data is required to be loaded for fact tables in multiple runs, the OFSAAI Rule Run Framework comes in handy. For example, for population of FCT_CRM_ACCOUNT_SUMMARY, the parameters of the batch execution include a parameter $RUNSK = -1. (Refer to the parameters of the batch ##INFODOM##_aCRM_CRM_Acc_Summ, Task1).

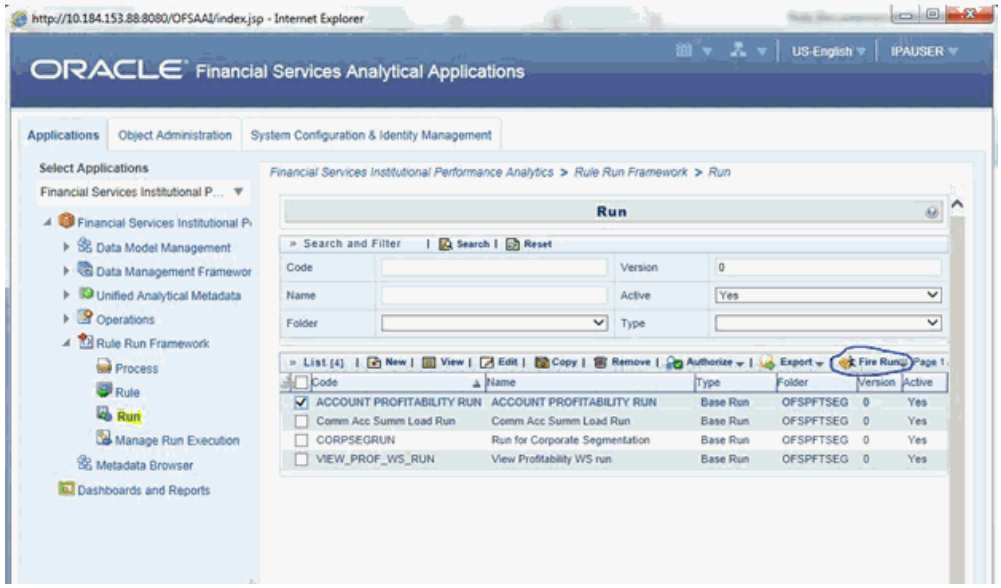
This batch execution loads the column N_RUN_SKEY in FCT_CRM_ACCOUNT_SUMMARY as -1. This will be a default run from the seeded batch. In order to be able to enter data for multiple runs, the batch tasks can be defined in Rule Run Framework. This will then create a batch internally which will load data for a different run into the fact table.

Executing a seeded run

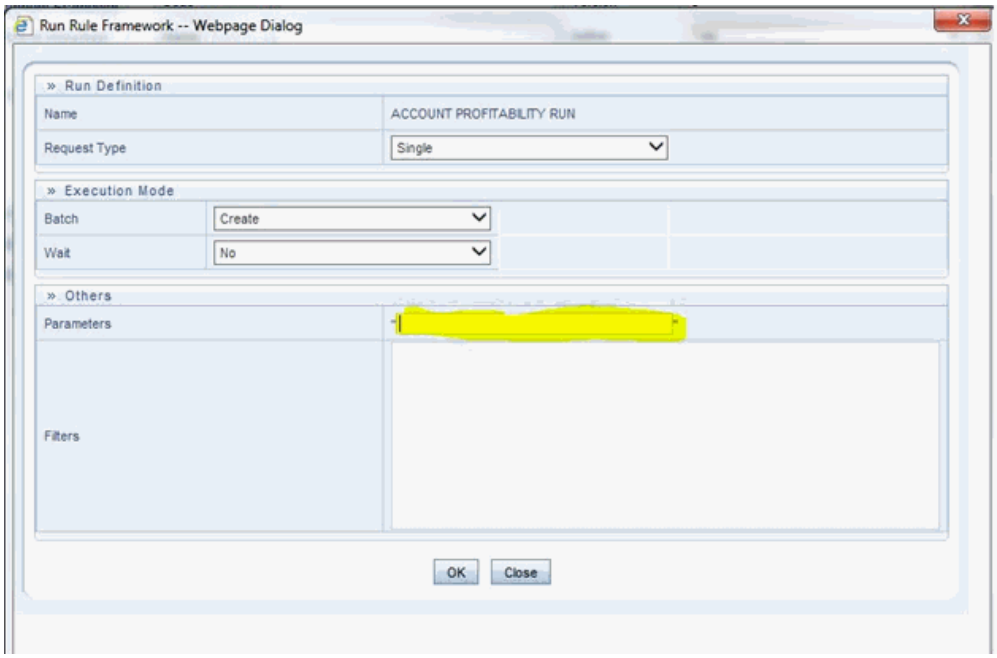
1. Navigate to**Rule Run Framework>Run**



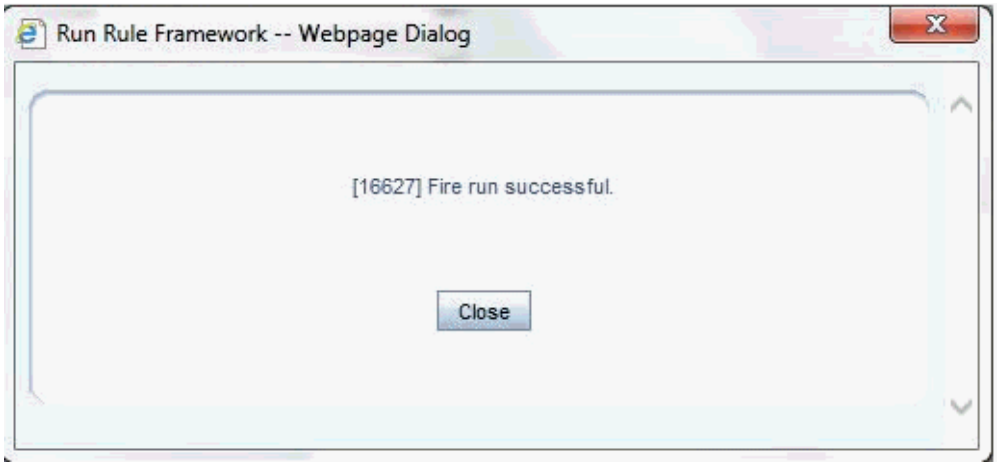
2. Choose a Run by checking the box before it and click **Fire Run**.



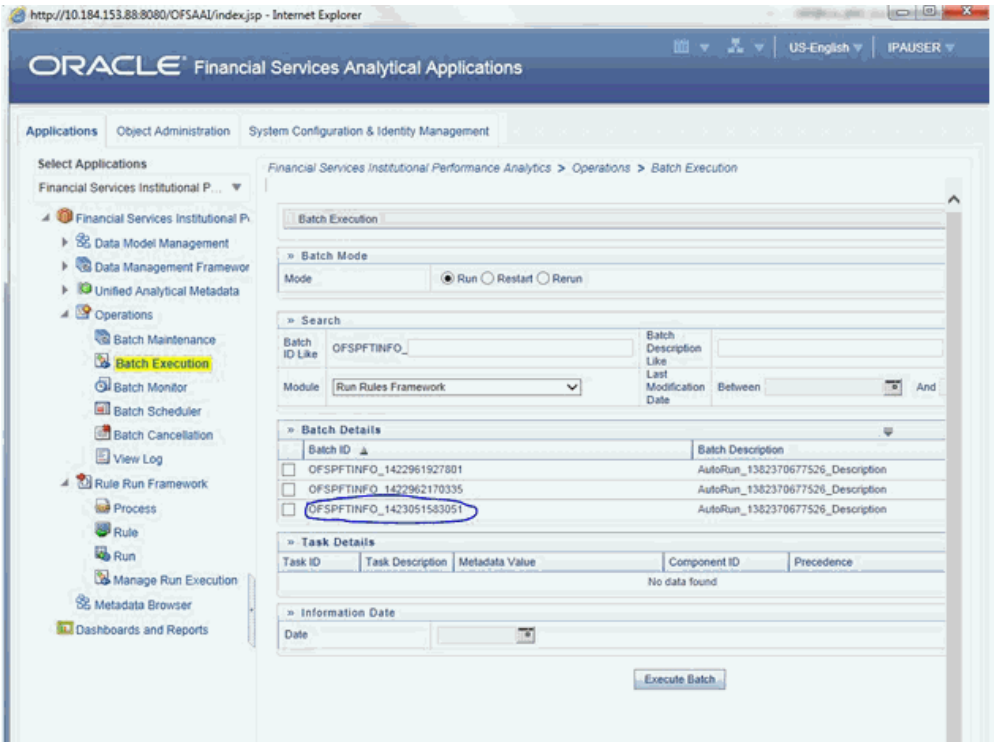
3. Enter the parameters required to execute the run (refer to details of individual runs) and click **OK**.



The following message will be displayed: *Fire run successful*



4. Execute the batch by navigating to **Operations>Batch Execution** and select the latest batch created in the Run Rules Framework module.



Runs available for RPA

Following are the runs available for RPA:

1. ACCOUNT PROFITABILITY RUN – Run for loading FCT_CRM_ACCOUNT_SUMMARY and FCT_ACCOUNT_PROFITABILITY

   Parameters to be entered as follows:

   ##RCY##", "<INFODOM>", "FCT_ACCOUNT_PROFITABILITY

   > **Note:** There are no quotes at the beginning and the end of the string. Consider as if the quotes are enclosing the delimiter (comma).

2. Comm Acc Summ Load Run – Run for loading FCT_COMMON_ACCOUNT_SUMMARY

3. VIEW_PROF_WS_RUN – Run for executing Web Service

4. RETSEGRUN – Run for executing Retail Segmentation Rule