

Oracle Financial Services  
Enterprise Financial Performance Analytics  
User Guide

*Release 8.0.7.0.0*  
*March, 2021*

Part No. E57807-02



Oracle Financial Services Enterprise Financial Performance Analytics User Guide, Release 8.0.7.0.0

Part No. E57807-02

Copyright © 2019 Oracle and/or its affiliates. All rights reserved.

Primary Author: Raghavacharyulu NCB

Contributor: Chris Spofford

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

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

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

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

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

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

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

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

---

# Contents

---

<b>Preface</b> .....	<b>3</b>
Intended Audience .....	3
Documentation Accessibility .....	3
Access to Oracle Support.....	3
Structure.....	3
Related Information Sources .....	4
<b>CHAPTER 1           Introduction</b> .....	<b>1</b>
Overview of Oracle Financial Services Enterprise Financial Performance Analytics (OFSEFPA).....	1
OFSEFPA Configurability in OFSAAI .....	2
Oracle Financial Services Enterprise Financial Performance Analytics Run Chart.....	3
<b>CHAPTER 2           Overview of Process Flow</b> .....	<b>5</b>
Introduction .....	5
OFSEFPA Data Flow .....	6
Data Flow: OFSPM Processing Area to OFSEFPA Reporting Area .....	6
Dimension Data Flow .....	7
Ledger Stat Data flow.....	7
Management Reporting Data Flow .....	8
OFSEFPA Data Flow: Third Party Profitability Management Applications to OFSEFPA Reporting Area .....	10
Data Model.....	10
Hierarchy.....	12
<b>CHAPTER 3           Dimension Loading Process</b> .....	<b>13</b>
Overview of Dimension Loading Process.....	13
Hierarchy Transformation.....	14
Overview of Hierarchy Flattening Process .....	14
Prerequisites .....	15
Tables Used by the Hierarchy Flattening Transformation .....	16
Executing the Hierarchy Flattening Transformation .....	16
Checking the Execution Status .....	18
Dimension Tables Population .....	18
Overview of SCD Process .....	18
Type 1 Methodology .....	19
Type 2 Methodology .....	19
Prerequisites .....	20
Tables Used by the SCD Component .....	20
Executing the SCD Component.....	24

Checking the Execution Status .....	25
Legal Entity .....	26
Parent Child Hierarchy .....	26
Multiple Hierarchies .....	28
<b>CHAPTER 4</b> <b><i>Time Dimension Population</i></b> .....	<b>29</b>
Overview of Time Dimension Population .....	29
Prerequisites.....	29
Tables Used by the Time Dimension Population Transformation .....	30
Executing the Time Dimension Population Transformation.....	30
Checking the Execution Status.....	31
<b>CHAPTER 5</b> <b><i>Modification and Mapping of Reporting Lines</i></b> .....	<b>33</b>
Overview.....	33
Summary of Configuration Required for Financial Reporting and Management Reporting .....	33
Financial Reporting (FR).....	33
Management Reporting (MR) .....	34
<i>Income Statement (MR-IS)</i> .....	34
<i>Balance Sheet (MR-BS)</i> .....	34
Addition and Modification of Reporting Lines Using AMHM.....	34
Modifying Financial Reporting Line .....	34
Modifying Management Reporting Line .....	35
Mapping of Reporting Line Items .....	35
Mapping of Income Statement Leaf Nodes to Financial Elements.....	36
Mapping of Balance Sheet Leaf Nodes to General Ledger.....	36
New Repline Hierarchies .....	39
<b>CHAPTER 6</b> <b><i>Multi-Currency Reporting</i></b> .....	<b>41</b>
Introduction .....	41
Overview of Multi-Currency Reporting.....	41
<b>CHAPTER 7</b> <b><i>Fact Ledger Population</i></b> .....	<b>43</b>
Overview of Fact Ledger Population Transformation .....	43
Prerequisites.....	43
Tables Used by the Fact Ledger Population Transformation .....	44
Map New Dimensions .....	46
Hierarchy Configuration for LOB and Legal Entity .....	46
<i>DIM_LOB</i> .....	46
<i>DIM_ORG_STRUCTURE</i> .....	46
Executing the Fact Ledger Population Transformation.....	47
Checking the Execution Status.....	49

Optional Load from Ledger Class Entities for Fact Ledger Population .....	49
Identity Codes Used in Financial Reporting .....	49
<b>CHAPTER 8</b>	
<b>Fact Management Reporting Population from LEDGER_STAT Table .....</b>	<b>51</b>
Overview of Fact Management Reporting Transformation .....	51
Prerequisites .....	52
Tables Used by the Fact Management Reporting Transformation .....	52
Map New Dimensions .....	55
Hierarchy Configuration for LOB and Legal Entity .....	55
DIM_LOB .....	55
DIM_ORG_STRUCTURE .....	55
Executing the Fact Management Reporting Transformation .....	56
Checking the Execution Status .....	57
Optional Load from Ledger Class Entities for Fact Management Reporting Population .....	58
Rollup Signage and Operational Signage .....	58
<b>CHAPTER 9</b>	
<b>Fact Management Reporting Population from FSI_D_MANAGEMENT_LEDGER Table .....</b>	<b>61</b>
Overview of Fact Management Reporting Transformation .....	61
Prerequisites .....	62
Tables Used by the Fact Management Reporting Transformation .....	62
Map New Dimensions .....	64
Executing the Fact Management Reporting Transformation .....	64
Checking the Execution Status .....	66
<b>CHAPTER 10</b>	
<b>What-If Analysis .....</b>	<b>67</b>
Introduction .....	67
Configuration for What-If Analysis .....	68
Basic Scenario .....	75
Advanced Scenario .....	77
EFPA Integration with Price Creation and Discovery (PCD) .....	78
Reporting Line Correlation Calculations .....	78
Excluding Outliers .....	79
Testing for Stationarity .....	79
Testing for Cointegration .....	79
What-If Analysis Limitation .....	80
.....	80
<b>CHAPTER 11</b>	
<b>Cube Build Process .....</b>	<b>81</b>
Introduction .....	81
Overview of Cubes .....	81

Creating Configuration Files .....	82
Building Of Cubes .....	82
Prerequisites .....	83
Tables Used by the Cube Build Component .....	84
Executing the Cube Build Task .....	84
Checking the Execution Status.....	87
Creating Multiple Ledger Essbase Cubes from OFSAA UI.....	87
<b>CHAPTER 12</b> <b>Predictive Modelling</b> .....	<b>89</b>
Introduction to Predictive Modelling.....	89
Guidelines .....	89
Files Used.....	90
Errors.....	90
<b>CHAPTER 13</b> <b>Overview of OFSEFPA Reports</b> .....	<b>91</b>
Introduction to Dashboards .....	91
Tabular Consolidated List of OFSEFPA Reports .....	91
Structure of Reporting Line Items .....	95
Hard-Coded Dimension Members.....	95
<i>Reporting Currency Calculation</i> .....	97
OBIEE: Catalogs Related Configuration .....	97
<i>Currency Label Modification</i> .....	97
<i>Amount Unit Modification</i> .....	98
Dashboards.....	98
Financial Reporting.....	98
Management Reporting.....	100
Forecast Reporting.....	112
<b>APPENDIX A</b> <b>How to Define a Dimension</b> .....	<b>117</b>
Introduction - Dimension Definition.....	117
Dimension Definition Process .....	117
<i>Create Business Hierarchy</i> .....	117
<i>Create Business Dimension</i> .....	119
<i>Modify Dataset</i> .....	119
.....	120
<b>APPENDIX B</b> <b>How to Define a Measure</b> .....	<b>121</b>
Introduction to Measure Definition .....	121
Measure Definition Process .....	121
Create Business Measure.....	121
<b>APPENDIX C</b> <b>How to Develop a New Cube</b> .....	<b>123</b>

Introduction to Developing a New Cube.....123

    Procedures to Develop a New Cube.....123

*Step 1 – Add Cube* .....123

*Step 2 – Include Dimensions* .....123

*Step 3 – Specify Variations* .....123

*Step 4 – Specify Dataset* .....123

*Step 5 – Specify Node Level Formula* .....123

*Step 6 – Save and Build* .....123

**APPENDIX D            *How to Define a Batch* ..... 125**

Introduction .....125

    Batch Creation .....125





## *List of Tables*

Table 1. Type 1 Methodology.....	19
Table 2. Type 1 Methodology.....	19
Table 3. Type 2 Methodology.....	20
Table 4. SETUP_MASTER .....	21
Table 5. SYS_TBL_MASTER.....	22
Table 6. Product Dimension .....	22
Table 7. SYS_STG_JOIN_MASTER.....	23
Table 8. Product Dimension .....	24
Table 9. Dynamic Parameters.....	25
Table 10. SETUP_MASTER .....	28
Table 11. DIM_REPORTING_LINE_ATTR .....	36
Table 12. SETUP_MASTER table.....	37
Table 13. New Replines .....	39
Table 14. FSI_BI_SETUP_TABLE .....	45
Table 15. Seeded entry of FSI_BI_SETUP_TABLE .....	47
Table 16. Updated Entry of FSI_BI_SETUP_TABLE.....	47
Table 17. FSI_BI_SETUP_TABLE .....	53
Table 18. Seeded entry of FSI_BI_SETUP_TABLE .....	56
Table 19. Updated entry of FSI_BI_SETUP_TABLE.....	56
Table 20. FSI_BI_SETUP_TABLE .....	63
Table 21. What-if Input Parameters .....	68
Table 22. Testing for Cointegration .....	80
Table 23. Consolidated List of OFSEFPA Report .....	91
Table 24. Reporting Line Dimension .....	96
Table 25. Batch Details .....	125



## *List of Figures*

Figure 1. OFSEFPA configuration .....	2
Figure 2. Reporting Line Hierarchy .....	59
Figure 3. Reporting Line Hierarchy - Rollup Signage .....	59
Figure 4. Financial Reporting - Income Statement.....	99
Figure 5. Financial Reporting - Income Statement - Time Series .....	99
Figure 6. Financial Reporting - Income Statement - Variance Analysis .....	100
Figure 7. Management Reporting - Balance Sheet - Balance Sheet.....	101
Figure 8. Management Reporting - Balance Sheet - Time Series .....	102
Figure 9. Management Reporting - Balance Sheet - Key Indicators.....	103
Figure 10. Management Reporting - Income Statement - Income Statement .....	103
Figure 11. Management Reporting - Income Statement - Time Series.....	104
Figure 12. Management Reporting - Income Statement - Key Indicators .....	105
Figure 13. Management Reporting - Income Statement - Variance Analysis.....	106
Figure 14. Management Reporting - Income Statement - Variance Analysis 1 .....	106
Figure 15. Management Reporting - Income Statement - Income Statement - Line Of Business 107	
Figure 16. Management Reporting - Key Trends - Key Trends .....	108
Figure 17. Management Reporting - Key Trends - Net Loss Analysis .....	109
Figure 18. Management Reporting - Key Trends - NII Impact Analysis .....	110
Figure 19. Management Reporting - Key Trends - Balance Trends.....	110
Figure 20. Management Reporting - Performance Measures - Performance Measures.....	111
Figure 21. Management Reporting - Performance Measures - Capital Analysis.....	112
Figure 22. Forecast Reports - Balance Sheet Reports - Forecast Balances .....	113
Figure 23. Forecast Reports - Balance Sheet Reports - Scenario Comparison.....	113
Figure 24. Forecast Reports - Balance Sheet Reports - Strategy Comparison.....	114
Figure 25. Forecast Reports - Income Statement Reports - Interest Vs. Average Balance...	115
Figure 26. Forecast Reports - Income Statement Reports - Non Interest Fee Percentage Vs. Ending Balance .....	115
Figure 27. Forecast Reports - Income Statement Reports - Spread Analysis.....	116
Figure 28. Forecast Reports - Income Statement Reports - NIM Report .....	116
Figure 29. Business Hierarchy .....	118
Figure 30. Business Hierarchy Details Page.....	119
Figure 31. Data Set Details Page .....	120
Figure 32. Business Measure Details Page .....	122



---

# Preface

## **Intended Audience**

Welcome to *Oracle Financial Services Enterprise Financial Analytics* User Guide.

This user guide is intended for the users of Oracle Financial Services Enterprise Financial Performance Analytics application.

See *Related Information Sources* 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>
- <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## **Structure**

This user guide has been segregated into the following chapters:

- [Chapter 1-Introduction](#)
- [Chapter 2-Overview of Process Flow](#)
- [Chapter 3-Dimension Loading Process](#)
- [Chapter 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). The rollup of a particular balance depending on their nature could be a simple additive rollup wherein the child member balances are added up to arrive at the parent node balance (for example, Ending Balance) or non additive rollups wherein a node formula is used to specify how to rollup the child member balances (for example, 3 month rolling average).

- [Chapter 5-Modification and Mapping of Reporting Lines](#)
- [Chapter 6-Multi-Currency Reporting](#)
- [Chapter 7-Fact Ledger Population](#)

Fact Ledger population involves populating the FCT\_LEDGER\_STAT table from the LEDGER\_STAT table.

- [Chapter 8-Fact Management Reporting Population from LEDGER\\_STAT Table](#)

Fact Management Reporting Population involves populating the FCT\_MGMT\_REPORTING table from the LEDGER\_STAT table.

- [Chapter 9-Fact Management Reporting Population from FSI\\_D\\_MANAGEMENT\\_LEDGER Table](#)

Fact Management Reporting Population involves populating the FSI\_D\_MANAGEMENT\_LEGDER from the LEDGER\_STAT table.

- [Chapter 10-What-If Analysis](#)

- [Chapter 11-Cube Build Process](#)

- [Chapter 12-Predictive Modelling](#)

- [Chapter 13-Overview of OFSEFPA Reports](#)

- [Appendix A, How to Define a Dimension](#)

- [Appendix B, How to Define a Measure](#)

- [Appendix C, How to Develop a New Cube](#)

- [Appendix D, How to Define a Batch](#)

## ***Related Information Sources***

- [Oracle Financial Services Advanced Analytical Applications Infrastructure Installation and Configuration Guide](#)
- [Oracle Financial Services Analytical Applications Data Model Document Generation White Paper](#)
- [Oracle Financial Services Profitability Management \(OFSPM\) User Guide](#)

## **Overview of Oracle Financial Services Enterprise Financial Performance Analytics (OFSEFPA)**

Oracle Financial Services Enterprise Financial Performance Analytics (OFSEFPA) (previously, Oracle Financial Services Profitability Analytics (OFSPA)) is a complete end-to-end web-based Business Intelligence solution for generating and analyzing Financial and Management Reports. It 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.

The OFSEFPA is built using:

- OFSAAI 8.0.7.0.0 for ETL, Data Integration, and Cube Build activities.
- OBIEE 12.2.1.4 for Dashboard and Reports activities
- Essbase 12.2.1.2 for multi-dimensional Cube storage.

OFSEFPA supports generating reports from both Relational Database and/or Essbase. Hence, Essbase is no longer a prerequisite for OFSEFPA.

This manual deals with OFSAAI, required for OFSEFPA activities, process, and functional details about the dashboards and reports. Also it includes subject areas which could be used for ad-hoc reporting using OBIEE Answers tool.

## OFSEFPA Configurability in OFSAAI

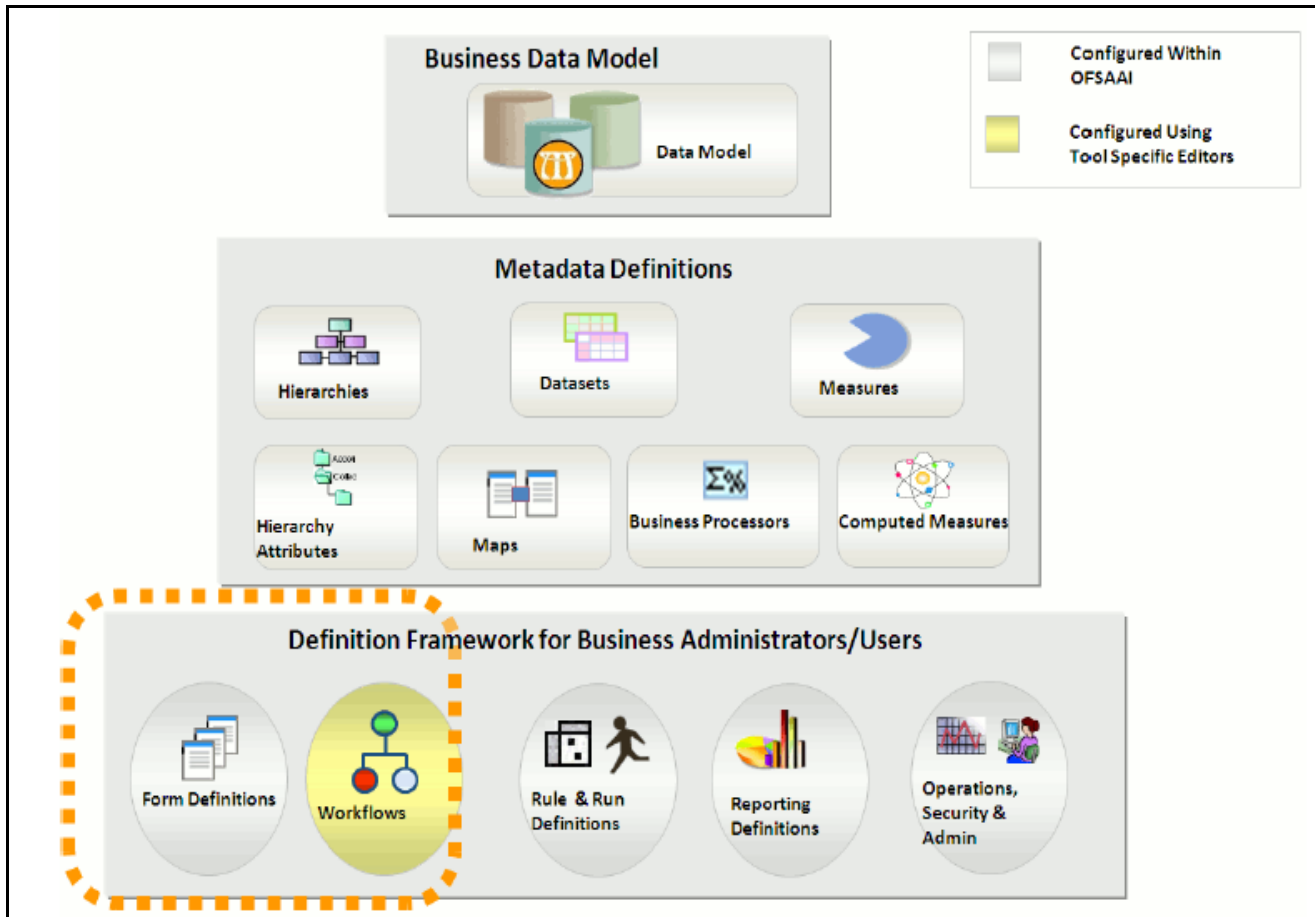


Figure 1. OFSEFPA configuration



## Oracle Financial Services Enterprise Financial Performance Analytics Run Chart

Run chart, as the name suggests helps the consultant understand the sequence of tasks to be performed to ensure the correctness of data flow in the product. It contains the details about Data loading for all mandatory tables for the product, and the sequence of execution of seeded tasks/batches.

Oracle Financial Services Enterprise Financial Performance Analytics (OFS EFPA) provides the Run Chart listing the tasks required for population of data for OFS EFPA Reports. This covers the following tasks:

- Stage Dimension Load
- Common data Population (for example Exchange Rate)
- Set up table population
  - Dimensions
  - Facts
- Model Execution
- Derived Entity Refresh
- Cubes Build

Download the OFS EFPA 8.0.7.0.0 Run Chart from the [MOS](#) page.



This chapter discusses the following topics:

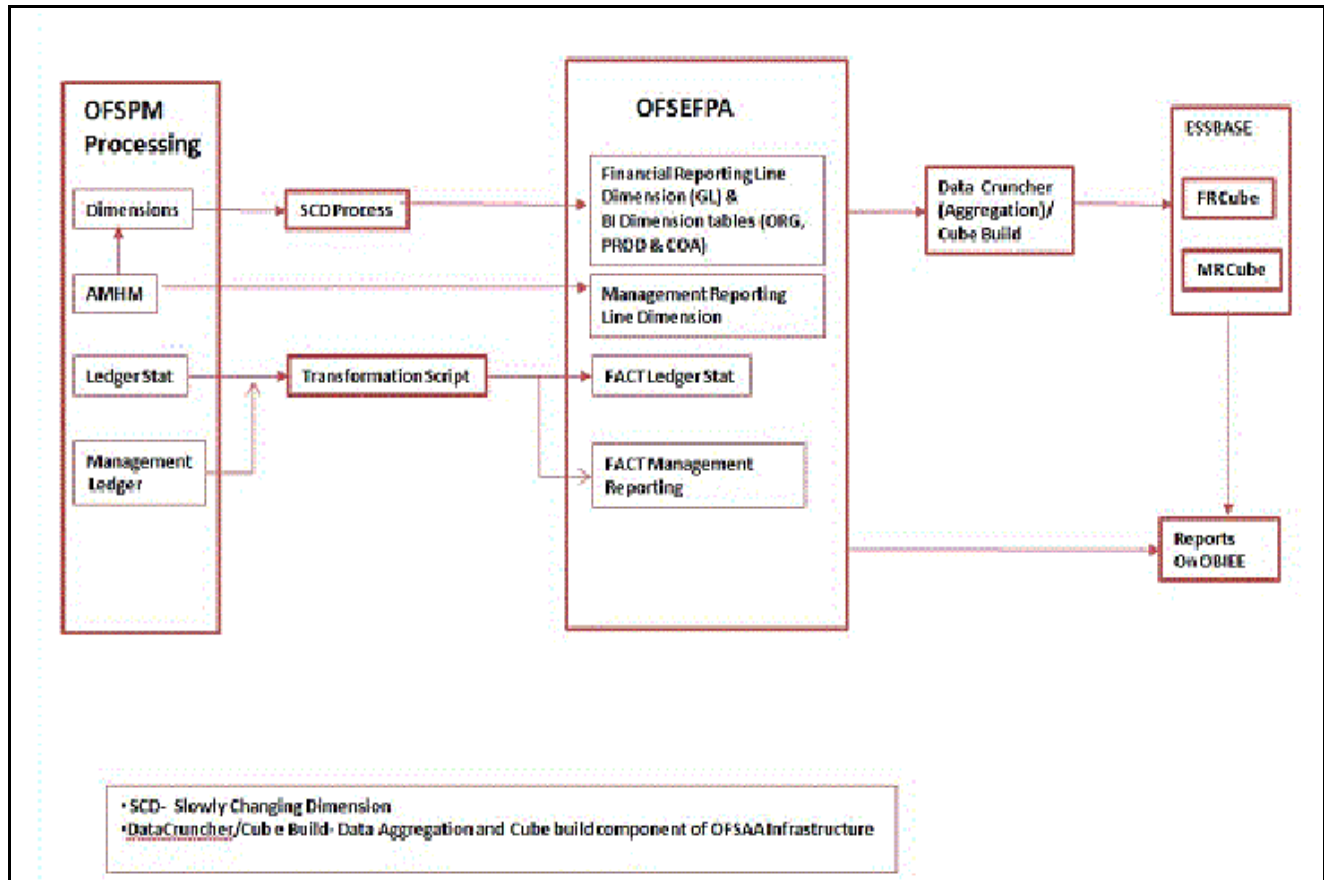
- [Introduction](#)
- [Data Flow: OFSPM Processing Area to OFSEFPA Reporting Area](#)
- [OFSEFPA Data Flow: Third Party Profitability Management Applications to OFSEFPA Reporting Area](#)
- [Data Model](#)
- [Hierarchy](#)

## **Introduction**

OFSEFPA utilizes OBIEE technology to present financial reporting and management reporting. For details on OFSEFPA reports and how OBIEE is being utilized, refer to chapter Overview of OFSEFPA Reports, page 13. The OFSEFPA data model has been designed in such a way that it can receive fact and dimension data

from the OFSAA Enterprise Performance Management (EPM) data model, which is installed through Oracle Financial Services Profitability Management (OFSPM) seeded data flow processes or from other systems. The seeded data flow processes from the processing area of OFSPM to the data model of OFSEFPA utilizes the transformation and load components of OFSAAI. OFSEFPA can be independently licensed and installed to work on top of the OFSAAI infrastructure or can be licensed along with OFSPM 8.0.7.0.0 or higher to work in an integrated manner. The following diagram depicts the high-level data flow of OFSEFPA, when both OFSPM and OFSEFPA are installed.

## OFSEFPA Data Flow



## Data Flow: OFSPM Processing Area to OFSEFPA Reporting Area

The OFSPM processing area holds dimension data and fact data. Fact data is held primarily in the management ledger table (referred to as Ledger Stat in this document). The seeded data flow processes outlined in this section are for the management ledger table. Similarly the seeded dimension data movement process definitions are for the standard processing dimensions which come with the OFSPM Attributes Members Hierarchies Module (AMHM). For more details on:

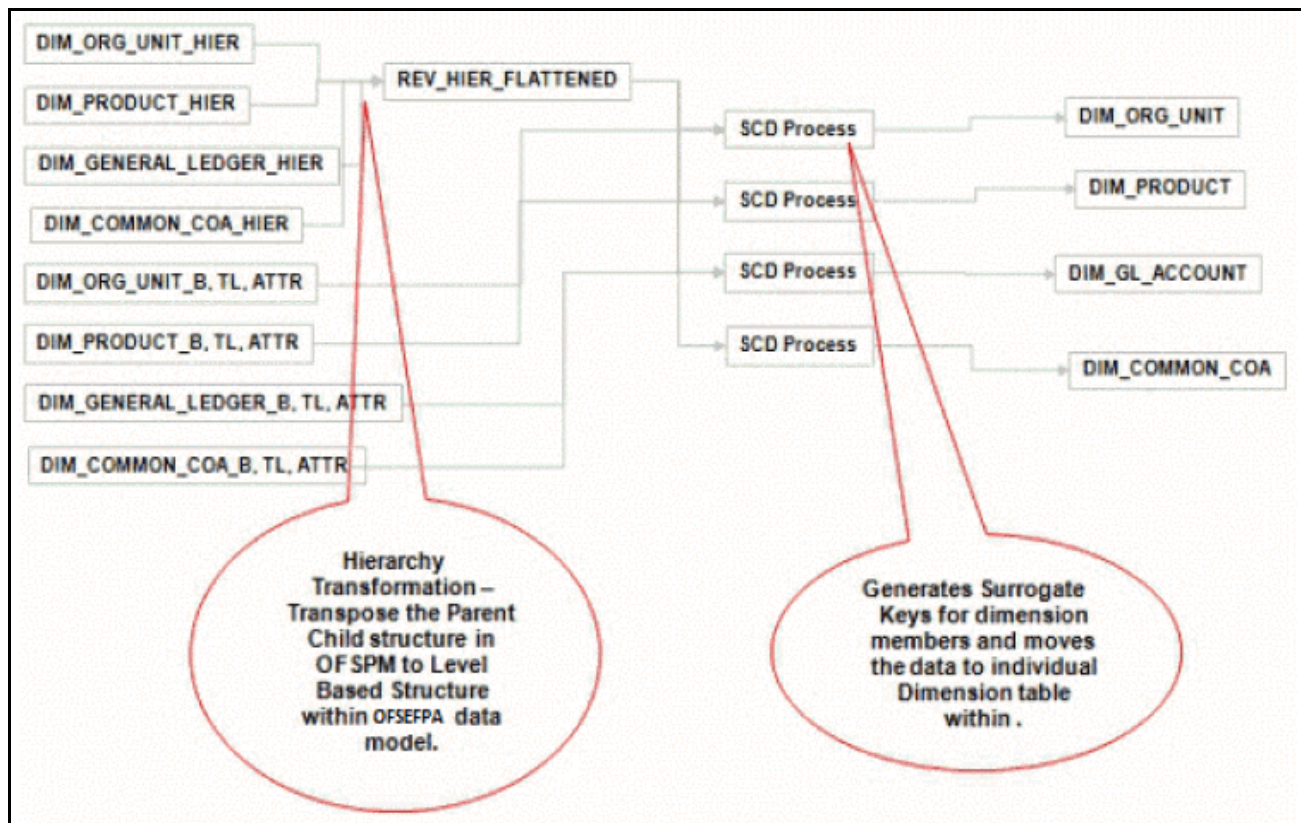
- Dimension data movement, refer to chapters [Dimension Loading Process](#) and [Time Dimension Population](#).
- Ledger Stat transformation, refer to [Fact Ledger Population](#).
- [Fact Management Reporting Population from LEDGER\\_STAT Table](#).

Similar data movement process definitions can be created for any additional user-defined dimensions or custom tables. These extensions are also described in this document.

The data movements from the OFSPM processing area to the OFSEFPA reporting area utilize the data transfer component of OFSAAI 8.0.7.0.0. Data transformation and loading is done with the Data Integrator module and is metadata driven. For more information on the Usage of the Data Integrator Component, refer to [Oracle Financial Services Analytical Applications Infrastructure User Guide](#).

## Dimension Data Flow

The Dimension data used for processing is stored in individual Member, Attribute, and Hierarchy tables for each dimension, and is part of the OFSPM AMHM data model. Hierarchies in OFSPM AMHM have a parent-child storage structure. These are flattened to a level-based structure as part of the data movement process. The flattened Hierarchies for each Dimension along with the Member and Attribute data undergo Slowly Changing Dimension (SCD) to facilitate the movement of data to OFSEFPA dimension tables. The following diagram depicts the process flow:

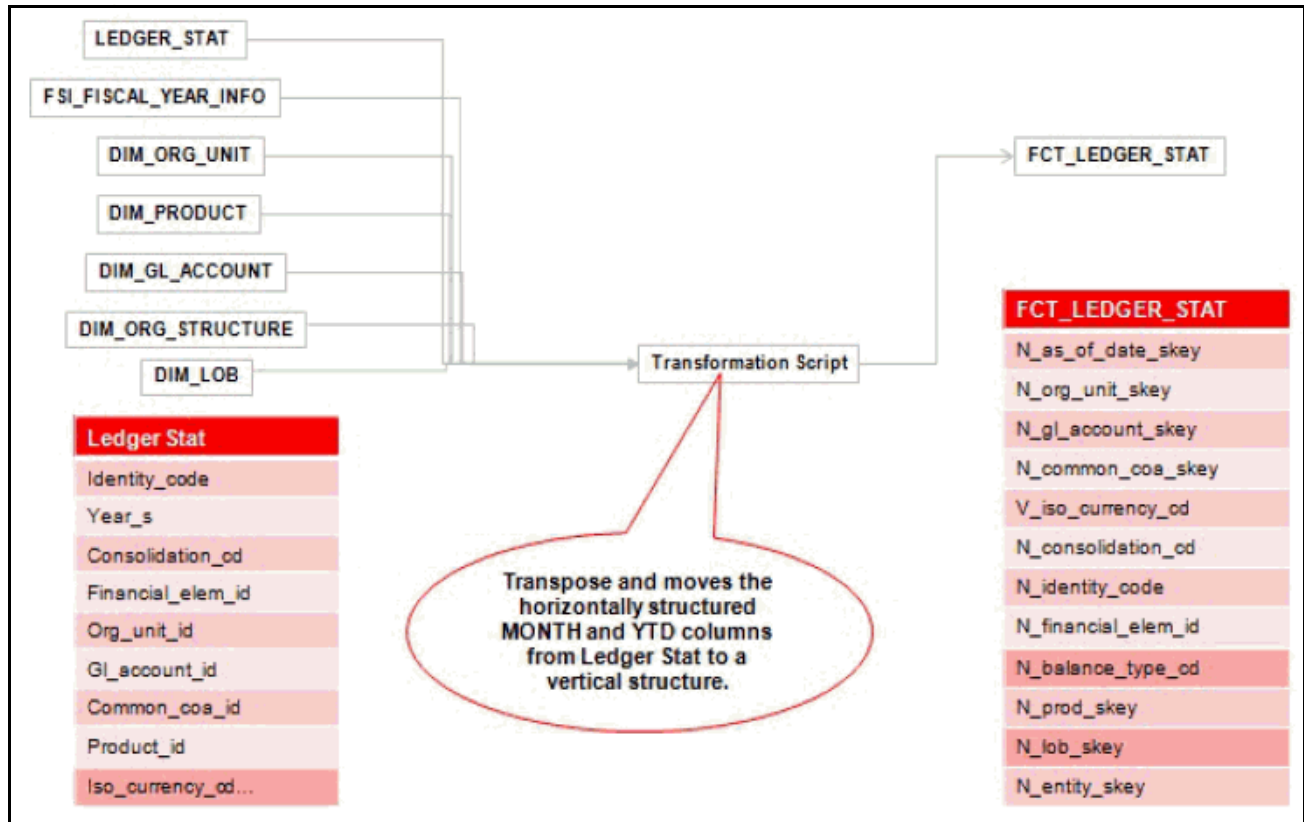


**Note:** For more information on dimension management, refer to [Oracle Financial Services Profitability Management \(OFSPM\) User Guide](#).

## Ledger Stat Data flow

The ledger transformation program moves data from Ledger Stat in the OFSPM processing area to Fact Ledger Stat Entity of OFSEFPA. The transformation program joins data for all the required dimensions. The following diagram

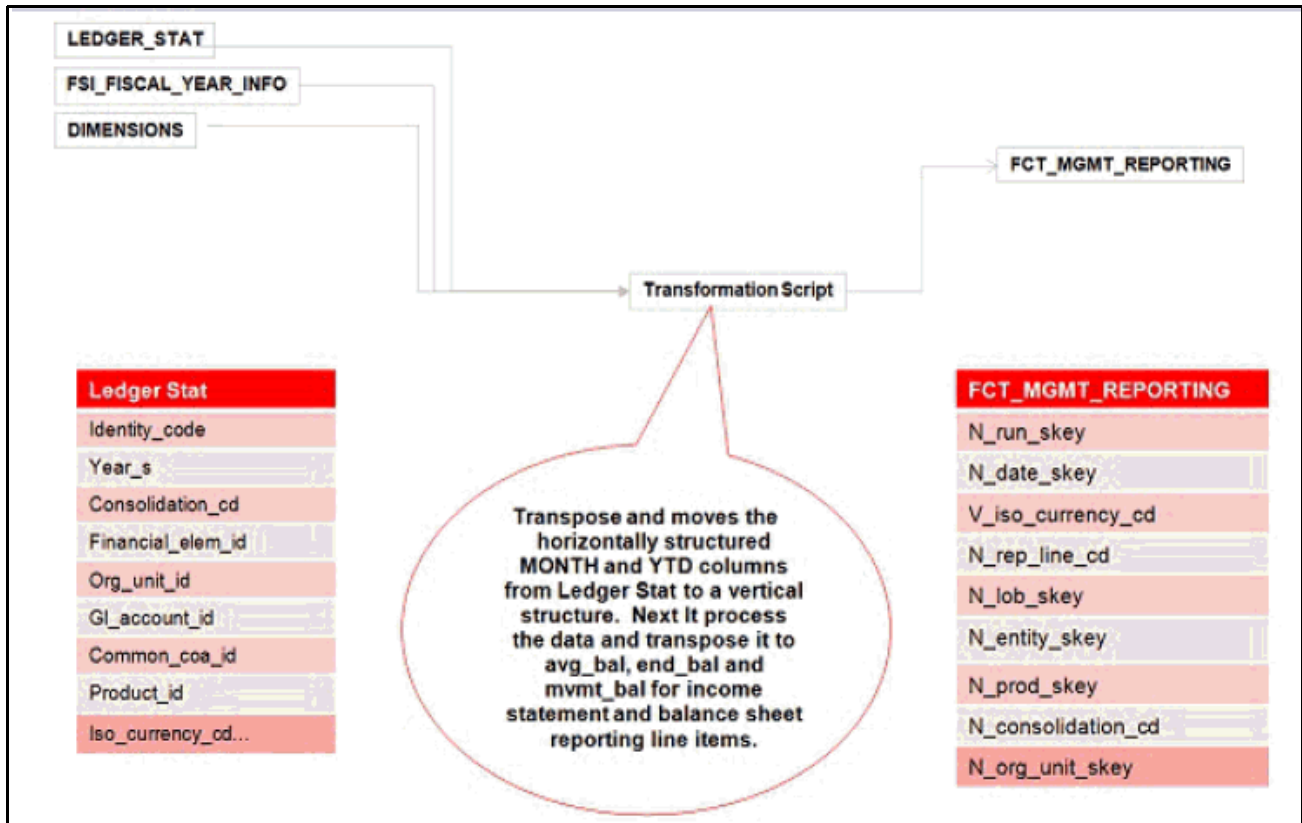
provides a high-level overview of this process. For more details on the process and its execution, refer to [Fact Ledger Population](#).



Optionally Data can be loaded from Reporting to FCT\_MGMT\_REPORTING.

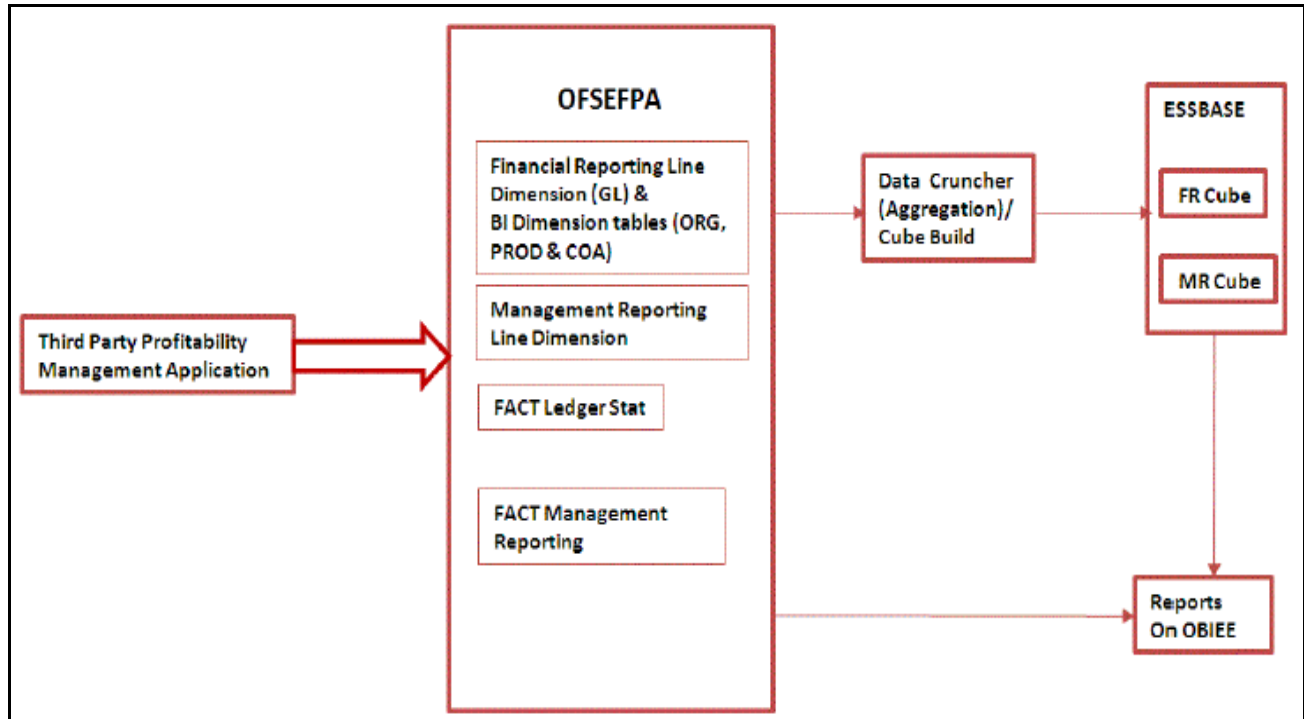
## Management Reporting Data Flow

The management transformation program moves data from the Ledger Stat in OFSPM processing area to Fact Management Reporting Entity of OFSEFPA. The transformation program joins data for all the required dimensions. The following diagram provides a high-level overview of this process. For more details on the process and its execution, refer to chapter Fact Management Reporting Population, page 8-1.



Optionally Data can be loaded from FSI\_D\_MANAGEMENT\_LEDGER to FCT\_LEDGER\_STAT.

## OFSEFPA Data Flow: Third Party Profitability Management Applications to OFSEFPA Reporting Area



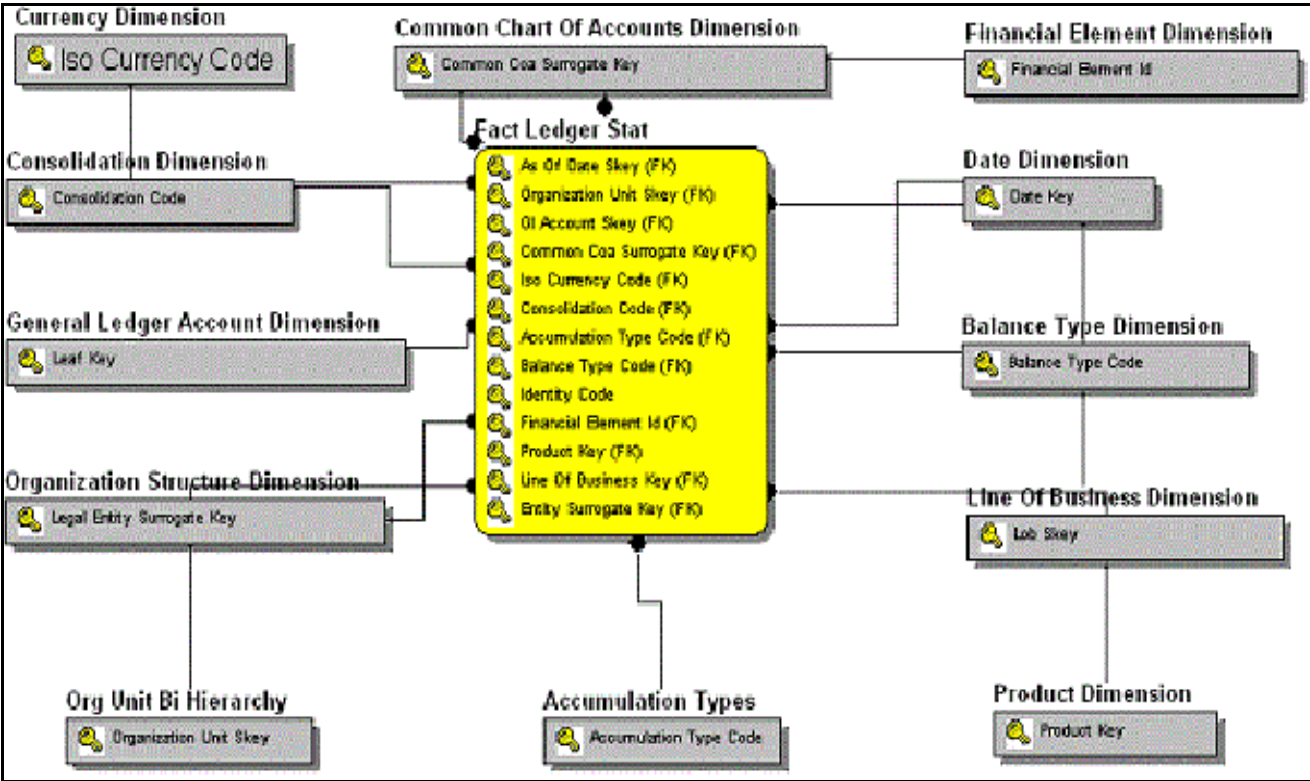
### Data Model

The data model of OFSEFPA is a star schema for the fact tables FCT\_LEDGER\_STAT and FCT\_MGMT\_REPORTING.

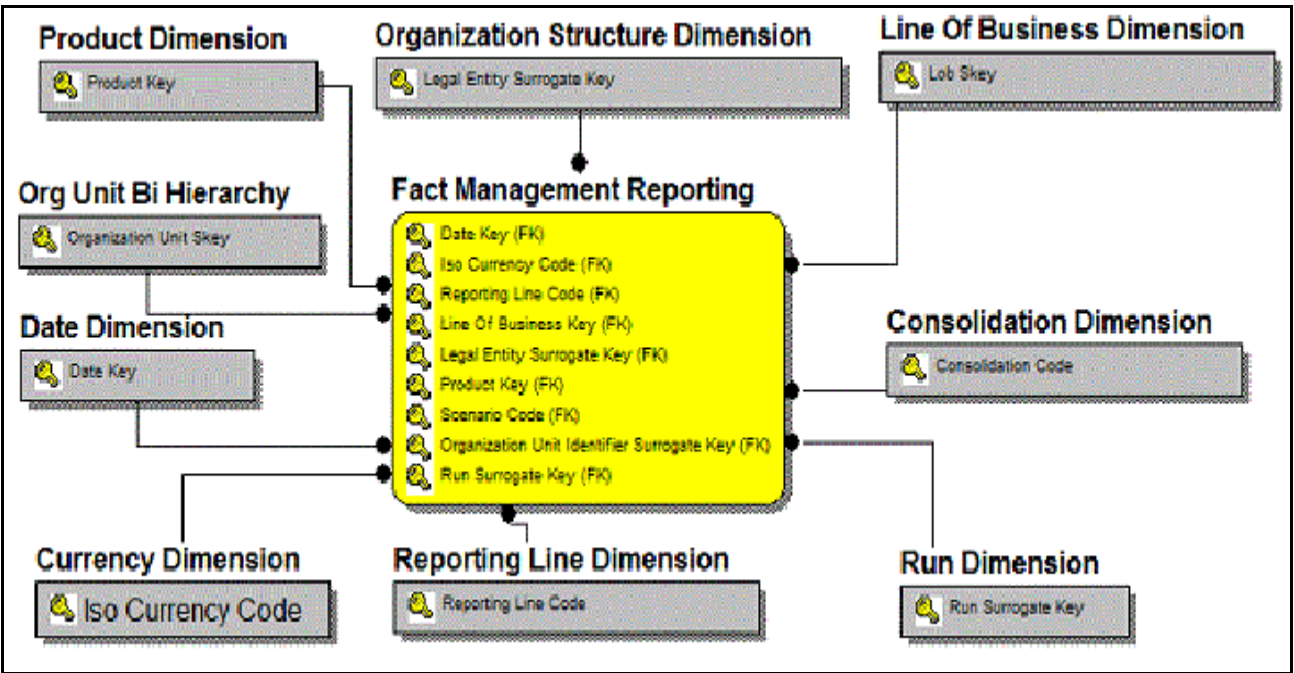
The Entity diagrams of the data model are provided below:

- Fact Ledger Stat





- Fact Management Reporting



## Hierarchy

Standard dimensions such as Product, Organization, Common COA, and General Ledger are populated through the AMHM layer of Profitability Management application (OFSPM). OFSEFPA assumes that only one hierarchy is present for each dimension in the AMHM layer. This Hierarchy data for each Dimension is populated in the Dimension Tables through SCDs.

An Entry in the SETUP\_MASTER table is required for the hierarchy for which the members are to be loaded in the dimension tables. One properly constructed Hierarchy ID should be given in the COMPONENT\_VALUE column for each dimension.

This chapter discussed the following topics:

- [Overview of Dimension Loading Process](#)
- [Hierarchy Transformation](#)
- [Dimension Tables Population](#)
- [Parent Child Hierarchy](#)
- [Multiple Hierarchies](#)

## **Overview of Dimension Loading Process**

The hierarchy data is natively stored in a parent-child structure within the *Dimension Management* component of OFSAAI. Dimension population involves the movement of dimension data from processing dimension tables (maintained by dimension management component of OFSAAI) to the reporting dimension tables used in Business Intelligence (BI) applications.

This data movement process is applicable only when OFSEFPA is installed along with OFSPM application.

Dimension loading process has the following two components:

1. [Hierarchy Transformation](#)
2. [Dimension Tables Population](#)

The Dimension loading process for the different hierarchies is discussed in the following sections:

- **General Ledger:**

The members of this hierarchy are stored within the infrastructure metadata tables for OFSEFPA. Hence the hierarchy needs to be maintained separately for OFSPM and OFSEFPA. This hierarchy need not be included as part of the hierarchy transformation process but should be part of the SCD process to move the General Ledger (GL) dimension members from OFSPM to OFSEFPA.

- **Financial Element:**

The members of this hierarchy are seeded through RDBMS insert scripts by the OFSEFPA solution installer. This hierarchy has to be maintained separately for OFSPM and OFSEFPA.

- **Organization Unit:**

This hierarchy in OFSPM first gets flattened by the Hierarchy Transformation and is then moved to the hierarchy table for Organization Unit (`DIM_ORG_UNIT`) by the Dimension table population component (as explained in chapter [Dimension Tables Population](#)). The OFSPM and OFSEFPA hierarchies can be kept in sync using the above two components.

- **Product:**

This hierarchy in OFSPM first gets flattened through Hierarchy Transformation and is then moved to the hierarchy table for Product (`DIM_PRODUCT`) by the Dimension table population component (as explained in chapter

[Dimension Tables Population](#)). The OFSPM and OFSEFPA hierarchies can be kept in sync by using the above two components.

- Time:

The hierarchy table (DIM\_DATES) for this hierarchy is loaded by the Time dimension population process (for more details, refer to chapter [Time Dimension Population](#)).

- Consolidation and Currency:

The hierarchy data for these hierarchies are seeded through RDBMS insert scripts by the OFSEFPA solution installer. These hierarchies have to be maintained separately for OFSPM and OFSEFPA.

- Legal Entity:

The hierarchy data of Legal Entity is loaded through insert scripts based on the user defined hierarchy. The above components in detail and the execution methods are explained in the following sections.

## Hierarchy Transformation

The following topics are covered in this section:

- [Overview of Hierarchy Flattening Process](#)
- [Prerequisites](#)
- [Tables Used by the Hierarchy Flattening Transformation](#)
- [Executing the Hierarchy Flattening Transformation](#)
- [Checking the Execution Status](#)

### Overview of Hierarchy Flattening Process

Hierarchy Flattening Transformation is used to move the hierarchy data from the parent child storage structure in OFSPM AMHM model to a level based storage structure in OFSEFPA. In OFSPM AMHM model, hierarchy data for any hierarchy created on seeded or user defined dimensions using the AMHM is stored within hierarchy tables of respective dimensions. These are moved to the REV\_HIER\_FLATTENED table of OFSEFPA after flattening by the Hierarchy flattening process.

#### Example

The hierarchy data of one or more Product Hierarchies created on Product dimension (a seeded dimension) are stored in DIM\_PRODUCTS\_HIER table.

The hierarchy data in the preceding example would be moved to REV\_HIER\_FLATTENED in the OFSEFPA model by the Hierarchy Flattening Process.

Database components used by this transformation are:

1. REV\_BATCHHIERFLATTEN - Oracle database function
2. REV\_HIER\_TRANSFORMATON\_BIAPPS - Oracle database Package called by the preceding function.

Some of the features of the Hierarchy Flattening Transformation are:

- The user has the choice to process a single hierarchy or all hierarchies belonging to a particular dimension as part of a single execution.

- Any changes made in the hierarchy using the *AMHM Hierarchy Maintenance* screen will change the `FLATTENED_ROWS_COMPLETION_CODE` flag in `REV_HIER_DEFINITIONS` table to 'Pending'. This improves the processing efficiency, since the Transformation process will avoid hierarchies that have not been modified.

## Prerequisites

- All the post install steps mentioned in the *Oracle Financial Services Analytical Applications Infrastructure Installation and Configuration guide* and the *Solution Installation Manuals of Profitability Management* (only if OFSPM is installed) and *Enterprise Financial Performance Analytics* have to be completed successfully.
- Seeded Hierarchies which come with the install and any hierarchy created using the OFSAAI framework should have proper data in the Tables used by the Hierarchy Flattening Transformation, page 3-5. Hierarchy is maintained in the Dimension Management component of OFSAAI. (**Financial Services Application > Master Maintenance > Dimension Management > Hierarchies** screen).

**Note:** The following debugging steps need to be performed only if the hierarchy flattening process has failed.

1. Check in the database (atomic schema) if the `FLATTENED_ROWS_COMPLETION_CODE` column of the `REV_HIER_DEFINITIONS` table has value 'Pending' for the Hierarchy ID to be processed.

This column will have the value 'Pending' for any new hierarchy created or modified using the OFSAAI Hierarchy Management User Interface.

2. Check if the `REV_DIMENSIONS_B` table has a row for the dimension that is being processed.

Execute the following query in the database to find the value and use the value in the dimension ID column for the dimension name/description to be processed.

```
Select b.dimension_id,t.dimension_name,t.description from rev_dimensions_b b inner
join rev_dimensions_tl t on b.dimension_id = t.dimension_id and t.dimension_name like
'<dimension name>'
```

3. Check if the `REV_HIERARCHIES` table has a row for the hierarchy id that is being processed.

```
SELECT * FROM rev_hierarchies rh where dimension_id = <dimension id>
```

- ◆ Map the application user to BATPRO role, that has seeded batch execution function.
- ◆ Create a Batch. For more information refer to [Executing the Hierarchy Flattening Transformation](#).
- ◆ Before executing a batch check if the following services are running on the application server.
- ◆ Iccserver
- ◆ Router
- ◆ AM Server
- ◆ Messageserver
- ◆ Olapdataserver

**Note:** 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](#).

## Tables Used by the Hierarchy Flattening Transformation

- `REV_HIERARCHIES` - This is the master table for hierarchies with one row per hierarchy.
- `REV_DIMENSIONS_B` - This is the master table for dimensions with one row per dimension.
- `REV_HIER_DEFINITIONS` - `FLATTENED_ROWS_COMPLETION_CODE` column is checked to determine whether the hierarchy is to be processed.
- `DIM_<DIMENSIONNAME>_HIER` - This table stores the hierarchy data and is the source for the transformation.  
**Example**

`DIM_PRODUCTS_HIER` - This table stores the hierarchy data of one or more Product Hierarchies created on Product dimension (a seeded dimension).

- `REV_HIER_FLATTENED` - This is the output table for the transformation into which the flattened hierarchy data gets populated.

## Executing the Hierarchy Flattening Transformation

You can execute the Data Transformation from the Operations module of OFSAAL.

The Hierarchy Flattening Transformation for OFSEFPA has been seeded with the **Batch ID** `<INFODOM>_Hierarchy_Transformation`, which can be executed from the *Batch Execution* section of OFSAAL. In the Parameter List, Enter Dimension ID and Hierarchy ID. For example, 2, 1000003710.

Queries to obtain the Dimension ID and Hierarchy ID are provided below.

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

To define a new task for a Batch definition:

- Select the check box adjacent to the newly created Batch Name in the *Batch Maintenance* window.
- Click **Add (+)** button from the Task Details grid.  
The *Task Definition* window is displayed.
- Enter the **Task ID** and **Description**.
- Select **TRANSFORM DATA** component from the drop down list.
- Select the following from the **Dynamic Parameters** list:
  - **Datastore Type** - Select the appropriate datastore type from the list
  - **Datastore Name** - Select the appropriate datastore name from the list
  - **IP address** - Select the IP address from the list
  - **Rule Name** - Select **BATCH\_HIERTRANSFORMATION** from the drop down list of available transformations. (This is a seeded Data Transformation which is installed as part of the OFSEFPA Solution Installer. If you don't see this in the list, contact Oracle Support.)
  - **Parameter List** - Enter Dimension ID and Hierarchy ID.

For the Parameter List, the values are:

- ◆ **Dimension ID** – Execute the following query in the database to find the value, and use the value in the Dimension ID column for the dimension name/description to be processed.

```
Select b.dimension_id,t.dimension_name,t.description from rev_dimensions_b b inner
join rev_dimensions_tl t on b.dimension_id = t.dimension_id and t.dimension_name like
'<dimension name>'
```

Replace <dimension name> in the preceding query with the Dimension Name you find in the UI (**Financial Services Application > Master Maintenance > Dimension Management**) for the dimension on which the

Hierarchy you want to flatten.

- ◆ **Hierarchy ID** - If all the hierarchies belonging to a dimension are to be processed, then provide **null** as the parameter value. Else, provide the System Identifier of the hierarchy that needs to be transformed.

Execute the following query in the database, only if a single hierarchy has to be processed, and use the value in HIERARCHY\_ID column as parameter for the hierarchy to be processed.

```
select b.object_definition_id,short_desc,long_desc from fsi_m_object_definition_b
b inner join fsi_m_object_definition_tl t on b.object_definition_id =
t.object_definition_id and b.id_type = 5
```

For OFSEFPA, it is assumed that only one Hierarchy is processed at a time.

#### Example

If all the hierarchies for GL Account dimension must be processed, the parameter list should be given as:

```
'2',null
```

Where '2' is the Dimension ID for the seeded dimension GL Account.

#### Example

If a particular hierarchy with code 1000018112 must be processed, the parameter list should be given as follows:

```
'2', '1000018112'
```

Where '1000018112' is the code obtained by executing the preceding query in the database.

- Click **Save**.

The Task definition is saved on the selected Batch.

You can execute the batch from *Batch Execution* window by choosing the Batch created following the steps mentioned in the preceding sections. For more details, refer to [Oracle Financial Services Analytical Applications Infrastructure User Guide](#).

Hierarchy Transformation can also be executed directly on the database through SQLPLUS. The details are:

- **Function Name:** REV\_BATCHHIERFLATTEN
- **Parameters:** BATCH\_RUN\_ID, MIS\_DATE, PDIMENSIONID, and PHIERARCHYID
- **Sample Parameter Values:** 'Batch1', '20091231', '2', '1000018112'

**Note:** Execute the Hierarchy Transformation Batch only when a new Hierarchy is defined or an existing Hierarchy is modified.



## 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 Event Log window in Batch Monitor section provides execution logs, in which the top row is the most recent. Any errors during the Batch execution are listed in the logs.

---

**Tip:** It is advisable to check the Event Log for any errors, even if the execution status returns '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.

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

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. (This is the same log you see in the Event Log window.)

## Dimension Tables Population

Dimensional data changes are handled by OFSEFPA solution using the SCD component.

The following topics are covered in this section:

- [Overview of SCD Process](#)
- [Prerequisites](#)
- [Tables Used by the SCD Component](#)
- [Executing the SCD Component](#)
- [Checking the Execution Status](#)

### 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, refer to:

- *Oracle Data Integrator Best Practices for a Data Warehouse* at <https://www.oracle.com/technetwork/middleware/data-integrator/learnmore/odi-best-practice-data-warehouse-168255.pdf>
- *Oracle Warehouse Builder Data Modeling, ETL, and Data Quality Guide* at



[https://docs.oracle.com/cd/E18283\\_01/owb.112/e10935/toc.htm](https://docs.oracle.com/cd/E18283_01/owb.112/e10935/toc.htm)

Additional online sources include:

- [http://en.wikipedia.org/wiki/Slowly\\_changing\\_dimension](http://en.wikipedia.org/wiki/Slowly_changing_dimension)
- [http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/10g/r2/owb/owb10gr2\\_gs/owb/lesson3/slowlychangingdimensions.htm](http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/10g/r2/owb/owb10gr2_gs/owb/lesson3/slowlychangingdimensions.htm)

You can also refer to *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 OFSEFPA solution are Type 1 and Type 2.

### Type 1 Methodology

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

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

**Table 1. Type 1 Methodology**

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

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:

**Table 2. Type 1 Methodology**

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 Methodology

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:

#### Example

**Table 3. Type 2 Methodology**

<code>N_PRODUCT_SKEY</code>	<code>V_PRODUCT_NAME</code>	<code>D_START_DATE</code>	<code>D_END_DATE</code>	<code>F_LATEST_RECORD_INDICATOR</code>
1	PL	5/31/2010	12/31/9999	N
1	Personal Loan	6/30/2010	12/31/9999	Y

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

## Prerequisites

1. The Hierarchy Flattening Transformation should have been executed successfully.
2. The SCD executable should be present under `<installation home>ficdb/bin`. The file name is `scd` and the user executing the SCD component should have execute rights on this file.
3. The setup tables accessed by SCD component (`SETUP_MASTER`, `SYS_TBL_MASTER`, and `SYS_STG_JOIN_MASTER`) should have the required entries. The `SETUP_MASTER` table does not come seeded with the installation; the required entries must be added manually. The required columns are mentioned in the [Tables Used by the SCD Component](#). The tables `SYS_TBL_MASTER` and `SYS_STG_JOIN_MASTER` are seeded for the Org unit, GL Account, Product, Common COA (Chart of Accounts) dimensions along with solution installation and you must only add entries in these tables, if you add new dimensions.
4. Database Views with name `DIM_<Dimension Name>_V` come seeded, for the seeded dimensions which come as part of installation. These views source data from the Profitability dimension tables as well as the flattened hierarchy data. `DIM_PRODUCT_V` is the view available for the product dimension. New views will have to be added for any new dimension, added in addition to the seeded dimensions.

## Tables Used by the SCD Component

The following are the database tables and columns used by the SCD component:

- `SETUP_MASTER`
  - `V_COMPONENT_CODE` - This column is not used by the OFSEFPA solution.
  - `V_COMPONENT_DESC` - This column value is hard coded in the database view definitions for `DIM_PRODUCT_V`, `DIM_GL_ACCOUNT_V`, `DIM_COMMON_COA_V`, and `DIM_ORG_UNIT_V` to obtain the Hierarchy ID from the `REV_HIER_FLATTENED` table. For this reason, the value for this column should be unique.

**Note:** The value in `V_COMPONENT_DESC` must exactly match with the value used in the SQL to create the `DIM_<dimension>_V` view. The View SQL contains a section referencing the `SETUP_MASTER` table. You must use the same upper and/or lower case letters in `V_COMPONENT_DESC` as used in this section of the View SQL.

- `V_COMPONENT_VALUE` - This is the hierarchy ID to be processed and this can be obtained by executing the following query:

```
select b.object_definition_id,short_desc,long_desc from fsi_m_object_definition_b
b inner join fsi_m_object_definition_t1 t on b.object_definition_id =
t.object_definition_id and b.id_type = 5
```

**Example:**

**Table 4. SETUP\_MASTER**

V_COMPONENT_CODE	V_COMPONENT_DESC	V_COMPONENT_VALUE
COMMON_COA_HEIR	COMMON_COA_HEIR1	1000063952
GL_ACCOUNT_HEIR	GL_ACCOUNT_HEIR1	200000808
ORG_HIER	ORG_UNIT_HIER1	200282
PRODUCT_HIER	PRODUCT_HIER1	1000004330

**Note:** For any newly defined Hierarchy, a row will have to be inserted to this table manually for SCD to process that Hierarchy. You can only specify one Hierarchy for each dimension.

- `SYS_TBL_MASTER`

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

SCD for GL Dimension fails if `V_COMPONENT_VALUE` is used with `DEFAULT` value. To avoid this failure, run the following query:

```
SELECT
    distinct hierarchy_id
FROM
    dim_general_ledger_hier;
```

The above query fetches the hierarchy IDs (in numbers). You can use any of the hierarchy IDs that are fetched to update the `SETUP_MASTER` table with the following query:

```
UPDATE setup_master
SET
    v_component_value = '<AMHM HierarchyID to be used for SCD>'
WHERE
    v_component_code = 'GL_ACCOUNT_HIER';
```

**Table 5. SYS\_TBL\_MASTER**

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

**Example:**

This is the row inserted by the solution installer for the product dimension.

**Table 6. Product Dimension**

MAP_REF_NUM	128
TBL_NM	DIM_PRODUCT
STG_TBL_NM	DIM_PRODUCT_V
SRC_PRTY	
SRC_PROC_SEQ	1
SRC_TYP	MASTER
DT_OFFSET	0

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

- SYS\_STG\_JOIN\_MASTER  
The solution installer populates this table for the seeded dimensions.

**Table 7. SYS\_STG\_JOIN\_MASTER**

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

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

1. PK - Primary Dimension Value (can be the multiple of the given *Mapping Reference Number*)
2. SK - Surrogate Key
3. DA - Dimensional Attribute (may be multiple for a given "Mapping Reference Number")
4. SD - Start Date
5. ED - End Date
6. LRI - Latest Record Indicator (Current Flag)
7. CSK - Current Surrogate Key
8. PSK - Previous Surrogate Key
9. SS - Source Key
10. LUD - Last Updated Date/Time
11. LUB - Last Updated By
12. NN - Not Null

**Example:**

This is the row inserted by the solution installer for the product dimension.



- **Datastore Name** - Select the appropriate datastore name from the list
- **Executable** - Enter `scd,<map ref num>`  
For example, `scd,2`
- **Wait** - Click **Yes** if you want to wait till the execution is complete or click **No** to proceed with the next task.

**Note:** Select **Yes** if you want the ICC component to wait for the process to complete the execution and update the status as either **Success** or **Failure**. If you select **No**, the component will trigger the processes and update the status as **Success**.

- **Batch Parameter** - Click **Yes** in Batch Parameter field if you want to pass the batch parameters to the executable and click **No** otherwise.

**Note:** Always select **Yes** in Batch Parameter.

6. Click **Save**.

The Task definition is saved for the selected Batch.

7. Click **Parameters**. Select the following from the Dynamic Parameters List and then click **Save**:

The map ref number values available for the **Executable** parameter are:

- **-1**, if you want to process all the dimensions. The *Executable* parameter mentioned earlier is: **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. If you want to process for Product dimension, the *Executable* parameter mentioned earlier is: **scd,6**.

**Table 9. Dynamic Parameters**

MAP_REF_NUM	TBL_NM
126	DIM_ORG_UNIT
127	DIM_GL_ACCOUNT
128	DIM_PRODUCT
129	DIM_COMMON_COA

- 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 detailed SCD component log can be accessed on the application server in the directory `$(FIC_HOME)` by accessing the following path `/ftpshare/<infodom name>/logs`.

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

The *Event Log* window in *Batch Monitor* section provides execution logs, in which the top row is the most recent. Any errors during the Batch execution are listed in the logs.

## Legal Entity

It is manual upload. Optionally it can be done by using `<INFODOM>_DIM_Org_Structure`

Batch `<INFODOM>DIM_Org_Structure` triggers the DT and loads the `DIM_Org_Structure` Table

Parameter List: NULL

DT uses data from below tables:

1. `STG_LEGAL_ENTITY_B_INTF`
2. `STG_LEGAL_ENTITY_TL_INTF`
3. `STG_LEGAL_ENTITY_ATTR_INTF`
4. `STG_LEGAL_ENTITY_HIER_INTF`

## Parent Child Hierarchy

OBIEE can handle Hierarchies in Parent - Child format as well. For this, the Level based Hierarchy of AMHM has to be converted to a Parent - Child Hierarchy to support Hierarchies in reports. Parent Child hierarchy is applicable for:

- Reporting Line (`DIM_REP_LINE`)
- General Ledger (`DIM_GL_ACCOUNT`)
- Organization Structure (`DIM_ORG_STRUCTURE`)

Once the above dimension tables are loaded, their respective Parent-Child Relation tables also have to be populated. The following are the dimension tables with their respective Parent-Child Relation tables:

- `DIM_REP_LINE` - `REP_LINE_PARENT_CHILD_RELATION`
- `DIM_GL_ACCOUNT` - `GL_PARENT_CHILD_RELATION`
- `DIM_ORG_STRUCTURE` - `ORG_STR_PARENT_CHILD_RELATION`

The following DTs populate the Parent-Child Relation tables:

1. `FN_REP_LINE_PARENT_CHILD`

The database components used to populate `REP_LINE_PARENT_CHILD_RELATION` are:

- Database function `FN_REP_LINE_PARENT_CHILDWR`



- Database function FN\_REP\_LINE\_PARENT\_CHILD, which is called by the function FN\_REP\_LINE\_PARENT\_CHILDWR mentioned above.

**Batch** <INFODOM>\_REPLINE\_PARENT\_CHILD\_UPD triggers the DT and loads the REP\_LINE\_PARENT\_CHILD\_RELATION table.

**Parameter List:** Hierarchy Name and Folder Name

**Example:** 'Repline Hierarchy','EPM61SEG'

Below query retrieves the Hierarchy Name and the Folder Name

```
select distinct FODTL.SHORT_DESC, FODB.FOLDER_NAME
from fsi_m_object_definition_tl FODTL, fsi_m_object_definition_bFODB
where FODTL.object_definition_id in
(select object_definition_id
from fsi_m_object_definition_b
where table_name = 'DIM_REPORTING_LINE_HIER')
and FOLDER_NAME = '<OFSAA segment name>';
```

There is a foreign key (FK) reference from table FCT\_MGMT\_REPORTING (N\_REP\_LINE\_CD column) to table DIM\_REP\_LINE (N\_REP\_LINE\_CD column). This foreign key, FK\_FCT\_MGMT\_REPORTING\_3 should be disabled if the Data Transformation (DT) FN\_REP\_LINE\_PARENT\_CHILD has to run again, as this DT deletes the DIM\_REP\_LINE table and reloads again. You can enable the FK after the successful execution of the DT.

#### 2. FN\_GL\_PARENT\_CHILD

The database components used to populate GL\_PARENT\_CHILD\_RELATION are:

- Database function FN\_GL\_PARENT\_CHILDWR
- Database function FN\_GL\_PARENT\_CHILD, which is called by the function FN\_GL\_PARENT\_CHILDWR mentioned above.

**Batch** <INFODOM>\_GL\_Parent\_Child\_UPD triggers the DT and loads the GL\_PARENT\_CHILD\_RELATION table.

**Parameter List:** NULL

#### 3. FN\_ORG\_PARENT\_CHILD

The database components used to populate ORG\_STR\_PARENT\_CHILD\_RELATION are:

- Database function FN\_ORG\_PARENT\_CHILDWR
- Database function FN\_ORG\_PARENT\_CHILD, which is called by the function FN\_ORG\_PARENT\_CHILDWR mentioned above.

**Batch** <INFODOM>\_ORG\_Parent\_Child\_UPD triggers the DT and loads the ORG\_STR\_PARENT\_CHILD\_RELATION table.

**Parameter List:** NULL

**Note:** Whenever there are changes to any of the three dimensions mentioned above, the corresponding Parent- Child relation DT should be executed. The OBIEE reports and the Parent- Child relation tables require the Parent ID to be null for the root node of the respective Hierarchy in the corresponding dimension table.

## Multiple Hierarchies

The current flow of hierarchy data movement from AMHM tables to Reporting dimension tables considers the following objects:

- 4 AMHM tables (<Dimension>\_B/TL/ATTR/HIER)
- REV\_HIER\_FLATTENED table
- SETUP\_MASTER table
- Dimension specific view
- SCD to load data from View to Reporting Dimension table

This out-of-the-box product has been configured to support data movement of not only one hierarchy but also to support more than one hierarchy. The information mentioned specifies the points at high level that one has to consider in order to support more than one hierarchy.

The following steps describe the data movement in two hierarchies (Product dimension is considered as an example):

1. Create two hierarchies for Product dimension in AMHM (For example: 200183, 301741)
2. Run Hierarchy Flattening transformation to load flattened structure of both in REV\_HIER\_FLATTENED.
3. Load SETUP\_MASTER table with two entries identified by V\_COMPONENT\_DESC values PRODUCT\_HIER1 and PRODUCT\_HIER2.

**Table 10. SETUP\_MASTER**

V_COMPONENT_CODE	V_COMPONENT_DESC	V_COMPONENT_VALUE
PRODUCT_HEIR	PRODUCT_HEIR1	301741
PRODUCT_HIER2	PRODUCT_HIER2	200183

4. Modify the view definition DIM\_PRODUCT\_V to include both hierarchies. Find below both the OOTB and the modified view definitions.

[DIM\\_PRODUCT\\_V](#)

[DIM\\_PRODUCT\\_\\_MULTI\\_HIER\\_V](#)

In both the hierarchies, level 10 to level 1 represent the first hierarchy and level 20 to level 11 represent the second hierarchy. Both the hierarchies share leaf nodes.

Use the [DIM\\_PRODUCT\\_\\_MULTI\\_HIER\\_V.sql](#), which is a sample view on product dimension. It can be used in the SCD process to move multiple alternate hierarchies defined in the AMHM to the flattened tables used for BI analytics.

This sample view considers two alternate hierarchies. You can extend this for additional hierarchies that may be required. Also, the sample is based on the Product dimension, but can be modified for other dimensions by replacing the relevant tables.

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

This chapter discusses the following topics:

- [Overview of Time Dimension Population](#)
- [Prerequisites](#)
- [Tables Used by the Time Dimension Population Transformation](#)
- [Executing the Time Dimension Population Transformation](#)
- [Checking the Execution Status](#)

## Overview of Time Dimension Population

The twelve month columns in LEDGER\_STAT table of OFSPM are replaced by a single N\_AS\_OF\_DATE\_KEY column in OFSEFPA's FCT\_LEDGER\_STAT table, with each month value stored in N\_VALUE column. Similarly, the YTD column value is stored in N\_VALUE\_YTD. This is done to make reporting easier considering Time is a dimension for most of the OFSEFPA reports. Time dimension population transformation is used to populate the DIM\_DATES table with values between two dates specified by the user.

The database components, used by the transformations are:

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

function FN\_DIM\_DATES mentioned earlier.

**Note:** OFSEFPA 6.1 does not use D\_FISCAL\_YEAR\_START\_DATE, D\_FISCAL\_YEAR\_END\_DATE, or N\_DAY\_OF\_WEEK columns.

## 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 of *Enterprise Financial 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:
  - Iccserver
  - Router

- AM Server
- Messageserver
- Olapdataserver

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

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

## Tables Used by the Time Dimension Population Transformation

For more details on viewing the structure of earlier tables, refer to [Oracle Financial Services Analytical Applications Data Model Data Dictionary](#) or the [OFSEFPA Erwin Data Model](#).

## Executing the Time Dimension Population Transformation

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

This component for OFSEFPA has been seeded with the Batch ID **<INFODOM>\_Dim\_Dates\_Population**, which can be executed from Batch Execution section of OFSAAI. In the Parameter List, enter the Start Date and End Date. For example, '19940101','19941231'.

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

To define a new task for a Batch definition:

1. Select the check box adjacent to the newly created Batch Name in the *Batch Maintenance* window.
2. Click **Add (+)** button from the Task Details grid.  
The *Task Definition* window is displayed.
3. Enter the **Task ID** and **Description**.
4. Select **Transform Data** component from the drop down list.
5. Select the following from the **Dynamic Parameters** list:
  - **Datastore Type** - Select the appropriate datastore type from the list.
  - **Datastore Name** - Select the appropriate datastore name from the list.
  - **IP address** - Select the IP address from the list.

- **Rule Name** - Select **Dim\_Dates\_Population** from the drop down list of available transformations. (This is a seeded Data Transformation which is installed as part of the OFSEFPA solution installer. If you don't see this in the list, contact Oracle support)
- **Parameter List** – Enter the Start Date and End Date.
  - **Start Date** – This is the starting date, from which the Transformation will populate DIM\_DATES table. This date should be specified in YYYYMMDD format.  
For example, '20081131'.
  - **End Date** - This is the end date, to which the Transformation will populate DIM\_DATES table. This date should also be specified in YYYYMMDD format.  
For example, '20091231'.

6. Click **Save**.

The Task definition is saved for the selected Batch.

7. Execute the batch.

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

The function can also be executed directly on the database through SQLPLUS Details are:

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

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

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

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

## Checking the Execution Status

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

The status messages in batch monitor are:

N - Not Started

O - On Going

F - Failure

S – Success

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

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

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



# *Modification and Mapping of Reporting Lines*

This chapter discusses the following topics:

- [Overview](#)
- [Summary of Configuration Required for Financial Reporting and Management Reporting](#)
- [Addition and Modification of Reporting Lines Using AMHM](#)
- [Mapping of Reporting Line Items](#)

## Overview

Reporting lines are configured as hierarchies within the Business Metadata of Oracle Financial Services Analytical Applications Infrastructure (OFSAAI). Reporting Line hierarchy originates in Reporting layer so it is enabled in AMHM. User needs to click on the Application Preferences screen to be able to create a new hierarchy of Reporting Line.

- **Reporting Lines for Financial Reporting**
  - Financial Reporting Line is based on General Ledger dimension. (DIM\_GL\_ACCOUNT table).
  - This reporting line hierarchy is a REGULAR parent-child hierarchy, based on the members of GL dimension.
- **Reporting Lines for Management Reporting**
  - Management Reporting Line is based on Reporting Line dimension (DIM\_REP\_LINE table).
  - This reporting line hierarchy is a REGULAR parent-child hierarchy based on the members of Reporting line dimension.

## Summary of Configuration Required for Financial Reporting and Management Reporting

### Financial Reporting (FR)

Reporting Line is based on General Ledger dimension. Therefore, all the GL codes for this subject area should functionally relate to Income Statement line items. The idea of this report is to show the un-allocated data coming in from the source system. The Income Statement General Ledger accounts in the source system need to be understood and be categorized as either income or expense GLs. The tag of income/expense can be achieved by putting in the appropriate Financial Element Code (FE) (for example, 420 for Interest, 455 for Non Interest Income, and 457 for Non Interest Expense). Each leaf level GL will be categorized as either Income or Expense with the appropriate FE mapping. Financial Reporting Income Statement hierarchy can then be constructed by using the leaf level GLs and rolling them up to higher level nodes. Assign the rollup signage (+/-) to each leaf/node

while constructing the hierarchy. Data is expected to be present only for the leaf level GL codes. Once the FR Income Statement hierarchy (can also be called as Income Statement from source ledger data) is constructed same can be made visible in Financial Reporting Dashboard by running the hierarchy transformation and SCD of GL dimension.

## **Management Reporting (MR)**

### **Income Statement (MR-IS)**

Management Reporting Income Statement (MR-IS) is seeded in the system and is available in the system using the reporting line codes present in the DIM\_REP\_LINE table. MR-IS line items are much more granular and data is created for them using the allocation engine (for example, OFSPM). The allocated data is expected to be present in the LEDGER\_STAT table. Line items of MR-IS are constructed based on Reporting line codes, but the data availability is expected into the seeded Financial Element Codes. EFPA seeds 9000 series FEs and allocation should be done to this series. Once the allocation engine provides data for these Financial Elements, the MR-IS will show results in the OBIEE dashboards. There is a mapping table DIM\_REPORTING\_LINE\_ATTR that ties up Reporting Line codes and Financial Elements. Seeded Financial Element list is provided in the install guide of OFSEFPA. The mapping between Reporting Lines and Financial Elements is 1:1 and one FE can only be mapped to a single leaf level Reporting Line code of MR Income Statement hierarchy.

### **Balance Sheet (MR-BS)**

Management Reporting Balance Sheet (MR-BS) is seeded in the system and is available in the system using the reporting line codes present in the DIM\_REP\_LINE table. Mapping of Balance Sheet General Ledger codes to Reporting Lines is explained in section [Mapping of Balance Sheet Leaf Nodes to General Ledger](#).

## **Addition and Modification of Reporting Lines Using AMHM**

### **Modifying Financial Reporting Line**

Financial Reporting line hierarchy can be modified to include new members and modify existing members. As a prerequisite, if the member to be added is a leaf, GL dimension table (DIM\_GL\_ACCOUNT) needs to have these leaf members.

#### **GL member**

When creating a new GL member in AMHM, the signage attribute should be properly mapped to the GL. This signage will be used as rollup signage in the GL hierarchy for node level calculation.

#### **GL Hierarchy**

General ledger Hierarchy is used for Financial Reporting. The hierarchy should build with proper parent and child level mapping from the topmost root to leaf level. The node level GL entries should not be present in the source table, that is LEDGER\_STAT. Only leaf level GLs are allowed to have data.

New members can be created from the *Financial Services Applications > Dimension Management* screen of OFSAAI



## Modifying Management Reporting Line

Management Reporting line hierarchy can be modified to include new members and modify existing members. As a prerequisite, if the member to be added is a leaf, Reporting Line dimension table (DIM\_REP\_LINE) needs to have these leaf members.

The table DIM\_REP\_LINE table is populated using AMHM tables, that is DIM\_REPORTING\_LINE\_B, DIM\_REPORTING\_LINE\_TL, DIM\_REPORTING\_LINE\_ATTR, and DIM\_REPORTING\_LINE\_HIER. The hierarchy has to be modified from the *Financial Services Applications > Dimension Management* section of OFSAAI, so that relevant changes are captured in the AMHM tables mentioned above. Once the hierarchy is saved, the DT has to be executed to populate DIM\_REP\_LINE and REP\_LINE\_PARENT\_CHILD\_RELATION tables.

Additional reporting line hierarchy can also be created using existing nodes using the AMHM framework of OFSAAI. If there are reporting line codes created, then you should create all such custom reporting lines using a sequence of numbers that is different from the numbers used in seeded reporting line hierarchy. It is recommended that you should create custom reporting lines with a six digit number starting with 500000, 500001, and so on.

## Mapping of Reporting Line Items

The reporting lines are seeded in the application and are used during hierarchy rollups in OBIEE reports. These reporting lines can be classified broadly into two types, that is Income Statement (IS) reporting lines and Balance Sheet (BS) reporting lines. All reporting lines are part of either IS hierarchy or BS hierarchy.

All reporting line codes of DIM\_REPORTING\_LINE\_ATTR or REP\_LINE\_GL\_MAP should be at leaf level reporting lines. When creating a new Reporting line item in AMHM, the signage attribute should be properly assigned to the Reporting line. This signage will be used as rollup signage in the Reporting Line hierarchy for node level calculation.

When a new financial element needs to be seeded into the application, you should manually seed the data into the below tables with unique FE IDs:

- DIM\_FINANCIAL\_ELEMENTS\_B
- DIM\_FINANCIAL\_ELEMENTS\_TL
- DIM\_FINANCIAL\_ELEMENTS\_ATTR

Once the above tables are loaded, DIM\_FINANCIAL\_ELEMENT has to be populated. The database components used to populate DIM\_FINANCIAL\_ELEMENT are:

- Database function FN\_DIM\_FINANCIAL\_ELEM\_UPDATEWR
- Database function FN\_DIM\_FINANCIAL\_ELEM\_UPDATE, which is called by the function FN\_DIM\_FINANCIAL\_ELEM\_UPDATEWR mentioned above.

The seeded Batch <INFODOM> **Financial\_Element\_UPD** triggers the DT FN\_DIM\_FINANCIAL\_ELEMENT that loads the DIM\_FINANCIAL\_ELEMENT table.

**Parameter List:** OFSAA User

**Example:** 'OFSAAUSER'

## Mapping of Income Statement Leaf Nodes to Financial Elements

IS reporting lines are seeded into the application and a default mapping is provided between Financial Element (FE) and Reporting Lines. The mapping of FE and IS reporting lines is present in DIM\_REPORTING\_LINE\_ATTR table. This mapping can be enhanced based on the requirements and the data need to be created as tabulated below:

**Columns:** REPORTING\_LINE\_ID, and

**Table 11. DIM\_REPORTING\_LINE\_ATTR**

V_COMPONENT_CODE	V_COMPONENT_DESC	V_COMPONENT_VALUE
REPORTING_LINE_ID	Column to stores Reporting Line Identifiers	
ATTRIBUTE_ID	Column to store the type of attribute used for Mapping	1-Financial Element 2-Signage
DIM_ATTRIBUTE_NUM ERIC_MEMBER	Column to store the actual attribute i.e. Financial Element or Signage	a) Actual Financial Element Code is used in this column in case the ATTRIBUTE_ID = 1 b) Signage value 1 or 2 is used in this column to signify positive signage or negative signage. Signage is applicable only when ATTRIBUTE_ID=2

- It is not required to process the mapping of financial elements 140 and 100 to reporting lines (Average Bal and End Bal).
- A single reporting line should be mapped only to one financial element.
- IS reporting lines are applicable for allocated data from OFSPM.
- Management Reporting - Income Statement: The Identity Codes available in LEDGER\_STAT table, having source\_type as 100 are considered for populating Income Statement reporting lines in FCT\_MGMT\_REPORTING table.  
`Select identity_code from FSI_DATA_IDENTITY where source_type = 100;`

## Mapping of Balance Sheet Leaf Nodes to General Ledger

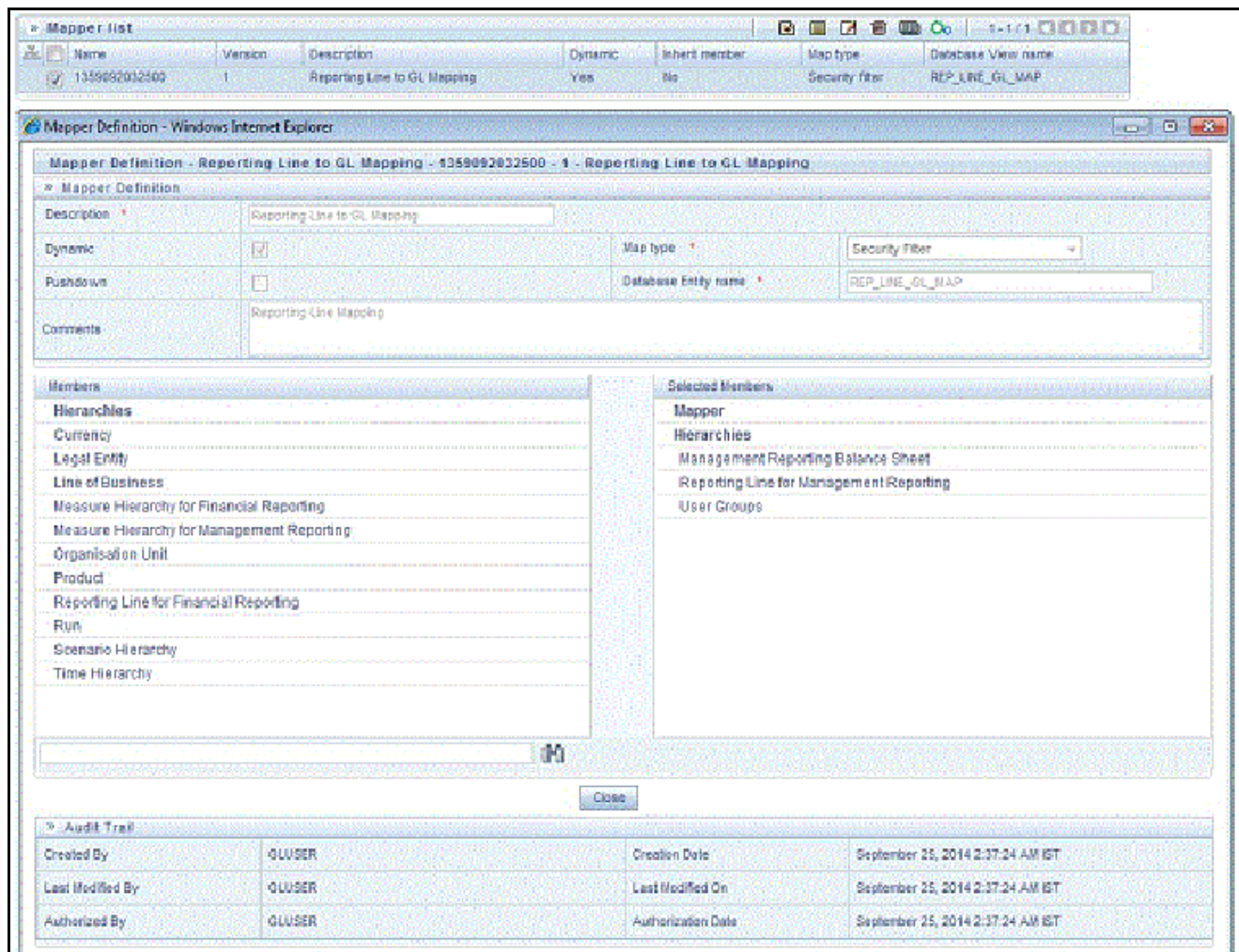
BS reporting lines are seeded in OFSEFPA application. The mapping of BS reporting lines are based on General Ledger codes, as they are available to OFSEFPA from Profitability Management. The data considered for BS hierarchy is un-allocated data, that is the data available in the OFSPM in un-allocated form (for all those IDENTITY\_CODE where SOURCE\_TYPE is '0' in the FSI\_DATA\_IDENTITY table). However, default mapping cannot be provided as the general ledger codes can differ at the customer site. The mapping of BS reporting lines and general ledger codes are done from the *Map Maintenance* section of OFSAAI. Map definition has been seeded in the OFSEFPA application for users to create the mapping of reporting lines (leaf nodes) of Reporting Line hierarchy with the general ledger hierarchy (leaf nodes). The general ledger hierarchy is

evolved from DIM\_GL\_ACCOUNT table through HEPMGLBS. In the SETUP\_MASTER table configuration (key-value pair) has been seeded to allow the mapping.

**Table 12. SETUP\_MASTER table**

V_COMPONENT_CODE	V_COMPONENT_DESC	V_COMPONENT_VALUE
REP_LINE_GL_MAPPER	REP_LINE_GL_MAPPE	R REP_LINE_GL_MAP

A new map definition can be created from the in *Map Maintenance* section of OFSAAI and the new table name against the key provided (REP\_LINE\_GL\_MAPPER) in SETUP\_MASTER table can be used.



- Mapping of more than one general ledger leaf node can be done to a single reporting line.
- V\_COMPONENT\_VALUE is the column that should be changed in case a new mapper is created on OFSAAI platform.

You can map the Balance Sheet leaf level GLs to leaf level Balance sheet Reporting Lines from the *Map Maintenance* screen of OFSAAI. However, this mapping can also be done directly in the database table that stores the mapping. Use the following table details to map leaf level codes in the map table:

- **Table Name:** REP\_LINE\_GL\_MAP
- **GL ID Column:** V\_MEMBER\_1  
Leaf level General Ledger ID that needs to be mapped to the leaf level Node of seeded Balance Sheet Reporting Line.
- **Balance Sheet Reporting Line Code:** V\_MEMBER\_2  
Leaf level Balance Sheet Reporting Line Code. The Balance Sheet Reporting line hierarchy is seeded in the application and it starts with node 107003.
- **Map ID:** V\_Map\_ID  
1359092032500 (It is the seeded ID of the mapper that is present after installation).
- **User Group Name:** V\_Member\_3  
User Group- It is the user group to which the logged in user belongs. You can refer to the below query to find it (in oonfig schema).

```
SELECT * FROM cssms_usr_group_map where v_usr_id = <>
```

#### **Example**

```
Insert into REP_LINE_GL_MAP (V_MAP_ID, N_MAP_ID, N_INHERIT_MAP_ID, V_MEMBER_1,  
V_MEMBER_2, V_MEMBER_3, V_MEMBER_4, V_MEMBER_5, V_MEMBER_6, V_MEMBER_7,  
V_MEMBER_8, V_MEMBER_9)
```

Values ('1359092032500', null, null, '12345700000149', '50100', 'TESTGRP', null, null, null, null, null)

12345700000149 - n\_gl\_account\_id in DIM\_GL\_ACCOUNT table of dev env.

50100 - Leaf level node from Balance Sheet hierarchy (Refer to the attached excel).

TESTGRP – test user group.

The Financial Elements (FE) supported for the MR- Balance Sheet are 100 and 140. Therefore, the GL IDs for which FE is either 100 or 140 will be picked up to populate Balance Sheet reporting line ID in FCT\_MGMT\_REPORTING table.

**Note:** The Identity Codes available in LEDGER\_STAT table, having source\_type as 0 are considered for populating Balance Sheet reporting lines in FCT\_MGMT\_REPORTING table.

```
Select identity_code from FSI_DATA_IDENTITY where source_type = 0;
```

## New Repline Hierarchies

Two new replines are Introduced, in order to calculate NIBT at selected and desired Repline levels. These are as follows:

**Table 13. New Replines**

Repline Code	Repline Name
81000	Other Interest Income All children of 82000 now report to 81000. 81000 reports to 82000.
81001	Other Interest Expenses All children of 82001 now report to 81001. 81001 reports to 82001.



This chapter discusses the following topics:

- [Introduction](#)
- [Overview of Multi-Currency Reporting](#)

## **Introduction**

A bank may desire to represent the reports in different currencies depending on geographic presence, regulatory requirements, and so on. The functionality of multi-currency reporting enables the reports to be displayed in multiple currencies.

## **Overview of Multi-Currency Reporting**

There are three different currency types that are supported and they are as follows:

- Local Currency
- Regional Currency
- Reporting Currency

Each currency type needs to be mapped to a currency and the reports can then be viewed in the corresponding currency. Using the exchange rate prevailing at the time the data is loaded, the conversion from the reporting currency (which is the default currency type) to another currency is done. The selection of the currency in which reports are to be viewed is done through dashboard prompts





Fact Ledger population involves populating the FCT\_LEDGER\_STAT table from the LEDGER\_STAT table. This chapter discusses the following topics:

- [Overview of Fact Ledger Population Transformation](#)
- [Prerequisites](#)
- [Tables Used by the Fact Ledger Population Transformation](#)
- [Map New Dimensions](#)
- [Executing the Fact Ledger Population Transformation](#)
- [Checking the Execution Status](#)
- [Optional Load from Ledger Class Entities for Fact Ledger Population](#)
- [Identity Codes Used in Financial Reporting](#)

## Overview of Fact Ledger Population Transformation

The LEDGER\_STAT table is optimized for processing purposes, but is not a convenient structure for reporting purposes. In generating FACT\_LEDGER\_STAT, time from LEDGER\_STAT is transformed into an explicit dimension in FACT\_LEDGER\_STAT.

Fact Ledger Population transformation is used to populate the FCT\_LEDGER\_STAT table from the Profitability LEDGER\_STAT table. The horizontally structured MONTH and YTD columns in Ledger/Stat are transposed to a vertical structure. The twelve Month Columns in LEDGER\_STAT are replaced by a single N\_AS\_OF\_DATE\_SKEY column in FCT\_LEDGER\_STAT with each month value stored in N\_VALUE column.

Similarly, the YTD column value is stored in N\_VALUE\_YTD. This is done to make reporting easier, considering Time is a dimension in most of the reports.

The database components, used by the Fact Ledger Population transformations are:

1. Database function FSI\_LEDGER\_STAT\_TRM
2. Database function LEDGER\_STAT\_TRM which is called by the function FSI\_LEDGER\_STAT\_TRM as mentioned earlier.

## 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 *Profitability Management* (only if OFSPM is installed) and *Enterprise Financial Performance Analytics* have to be completed successfully.
2. Application User must be mapped to a role that has seeded batch execution function (BATPRO).

3. Ensure that your FISCAL year information is configured properly. It has the following two columns.
  - FISCAL\_PERIOD: This gives the number of months in the given FISCAL period
  - START\_MONTH: This indicates which month of the calendar year is the FISCAL starting month. For example, a value '1' for this column means FISCAL year starts from January and value of '4' indicates that the FISCAL year starts from April.
4. Before executing a batch, check if the following services are running on the application server.
  - Iccserver
  - Router
  - AM Server
  - Messageserver
  - Olapdataserver

For more information on how to check if the services are up and how to start the services, refer to *Oracle Financial Services Analytical Applications Infrastructure User Guide*.
5. Batches will have to be created for executing the function. For more information, refer to section [How to Define a Batch](#).

## Tables Used by the Fact Ledger Population Transformation

- FSI\_FISCAL\_YEAR\_INFO - This table has the FISCAL year info. The entries required in this table are mentioned in the [Prerequisites](#).
- FSI\_BI\_SETUP\_TABLE - This table has the setup information used by the Transformation. They are :
  - TARGET\_TABLE\_NAME: This is the destination table name for transformation.
  - TARGET\_COLUMN\_NAME: This is the Destination column name in FCT\_LEDGER\_STAT table.
  - MEMBER\_COL\_NAME: This is the column Name in LEDGER\_STAT table.
  - SOURCE\_DIM\_TABLE\_NAME: This is the dimension table to which the Ledger data has to be joined to get the surrogate key value.
  - SOURCE\_COLUMN\_NAME: This is the column in the dimension table to which the LEDGER\_STAT ID column is joined.
  - SKEY\_COLUMN\_NAME: This is the column in the dimension table which has the surrogate key value.
  - JOIN\_REQUIRED: This provides the information, whether the column to be moved to FCT\_LEDGER\_STAT is directly available in LEDGER\_STAT or a join has to be taken with dimension table to get the skey.

Sample data for this table is plotted below:

Table 14. FSI\_BI\_SETUP\_TABLE

TABLE_TABLE_NAME	TABLE_COLUMN_NAME	MEMBER_COLUMN_NAME	SOURCE_DIM_TABLE_NAME	SOURCE_COLUMN_NAME	KEY_COLUMN_NAME	JOIN_REQUIRED	GROUP_BY_REQUIRED
FCT_LEDGER_STAT	N_BALANCE_TYPE_CD	BALANCE_TYPE_CD	LEDGER_STAT	BALANCE_TYPE_CD	BALANCE_TYPE_CD	N	
FCT_LEDGER_STAT	N_COMMON_COA_SKEY	COMMON_COA_ID	DIM_COMMON_COA	N_COMMON_COA_ID	N_COMMON_COA_SKEY	Y	
FCT_LEDGER_STAT	N_CONSOLIDATION_CD	CONSOLIDATION_CD	LEDGER_STAT	CONSOLIDATION_CD	CONSOLIDATION_CD	N	
FCT_LEDGER_STAT	N_ENTITY_SKEY	V_ENTITY_SKEY	DIM_ORG_STRUCTURE	-1	-1	N	
FCT_LEDGER_STAT	N_FINANCIAL_ELEM_ID	FINANCIAL_ELEMENT_ID	LEDGER_STAT	FINANCIAL_ELEM_ID	FINANCIAL_ELEM_ID	N	
FCT_LEDGER_STAT	N_GL_ACCOUNT_SKEY	GL_ACCOUNT_ID	DIM_GL_ACCOUNT	N_GL_ACCOUNT_ID	N_GL_ACCOUNT_SKEY	Y	
FCT_LEDGER_STAT	N_IDENTITY_CODE	IDENTITY_CODE	LEDGER_STAT	IDENTITY_CODE	IDENTITY_CODE	N	
FCT_LEDGER_STAT	N_LOB_SKEY	V_LOB_SKEY	DIM_LOB	-1	-1	N	
FCT_LEDGER_STAT	N_ORG_UNIT_SKEY	ORG_UNIT_ID	DIM_ORG_UNIT	N_ORG_UNIT_ID	N_ORG_UNIT_SKEY	Y	
FCT_LEDGER_STAT	N_PROD_SKEY	PRODUCT_ID	DIM_PRODUCT	N_PRODUCT_ID	N_PROD_SKEY	Y	
FCT_LEDGER_STAT	V_ACCUMULATION_TYPE_CD	ACCUM_TYPE_CD	LEDGER_STAT	ACCUM_TYPE_CD	ACCUM_TYPE_CD	N	
FCT_LEDGER_STAT	V_ISO_CURRENCY_CD	ISO_CURRENCY_CD	LEDGER_STAT	ISO_CURRENCY_CD	ISO_CURRENCY_CD	N	

- LEDGER\_STAT - This table is the source for the transformation.
- DIM\_<dimension Name> - The flattened dimension tables used in Business Intelligence (BI) reporting are accessed to obtain the surrogate key to be populated to FCT\_LEDGER\_STAT dimension columns. For example, DIM\_ORG\_UNIT, DIM\_PRODUCT, and so on.
- FCT\_LEDGER\_STAT - This is the output table for the transformation.

For more details on viewing the structure of these tables, refer to [Oracle Financial Services Analytical Applications Data Model Data Dictionary](#) or the [OFSEFPA Erwin Data Model](#).

## Map New Dimensions

For user defined dimensions, manual update to the DIM\_<DIMENSION>\_V is required:

1. Backup current DIM\_<CUSTOM\_DIM>\_V
2. Update the following line of the view:

```
INNER JOIN DIM_<CUSTOM_DIM>_TL D ON D.<CUSTOM_DIM_ID>=A.<CUSTOM_DIM_ID> AND  
A.<CUSTOM_DIM_ID> not in (-1,0)
```

With the following:

```
INNER JOIN DIM_<CUSTOM_DIM>_TL D ON D.<CUSTOM_DIM_ID>=A.<CUSTOM_DIM_ID> AND  
A.<CUSTOM_DIM_ID> not in (-1,0) and D.LANGUAGE = A.DEFINITION_LANGUAGE
```

The change required is to add the additional language condition\*\*\*\* AND D.LANGUAGE = A.DEFINITION\_LANGUAGE \*\*\*\*. "

The Custom Dimensions can be added to target table, FCT\_LEDGER\_STAT by updating the configuration table FSI\_BI\_SETUP\_TABLE. DIM\_LOB and Dim\_Org\_Structure can be loaded manually or by using the AMHM tables and batch.

**Note:** All nodes in FCT\_LEDGER\_STAT table should be part of the respective hierarchies for which the SCDs run. Otherwise, the cube will fail.

## Hierarchy Configuration for LOB and Legal Entity

### DIM\_LOB

The hierarchy to be built for LOB (Line of Business) is a level based hierarchy. All leaf nodes of this hierarchy are mapped to a single root. The branch navigation of the hierarchy starts from root node which is the Level 1 column of the DIM\_LOB table. The Leaf node has to be placed at the appropriate column between Level 1 and Level 16 of the DIM\_LOB table. This step is required only if the reports are working out of ESSBASE cubes.

### DIM\_ORG\_STRUCTURE

The hierarchy to be built for DIM\_ORG\_STRUCTURE is parent-child hierarchy. The parent node of the root has to be kept blank in the V\_PARENT\_CODE column. The hierarchy can then be built appropriately by putting a child node - parent node combination in the V\_ENTITY\_CODE and V\_PARENT\_CODE columns of the DIM\_ORG\_STRUCTURE table. This metadata setup supports only one hierarchy for Legal Entity.

Follow the below steps manually to populate data into DIM\_LOB and DIM\_ORG\_STRUCTURE custom dimensions for the fact table:

#### Procedure

1. Add the columns N\_LOB\_ID and N\_ENTITY\_ID as part of Unique Key (Ledger\_stat) in FCT\_LEDGER\_STAT table.
2. Update the columns SOURCE\_COLUMN\_NAME and SKEY\_COLUMN\_NAME for the target table FCT\_LEDGER\_STAT in FSI\_BI\_SETUP\_TABLE, with the actual source and skey column values, that is N\_LOB\_ID and N\_ENTITY\_ID.
3. Set the value to 'Y' in JOIN\_REQUIRED column of FCT\_LEDGER\_STAT table.

Table 15. Seeded entry of FSI\_BI\_SETUP\_TABLE

TABLE_TABLE_NAME	TABLE_CODE	MEMBER_CODE	SOURCE_DIM_TABLE_NAME	SOURCE_COLUMN_NAME	SOURCE_COLUMN_NAME	JOIN_REQUIRED	GROUP_BY_REQUIRED
FCT_LEDGER_STAT	N_ENTITY_SKEY	V_ENTITY_SKEY	DIM_ORG_STRUCTURE	-1	-1	N	
FCT_LEDGER_STAT	N_LOB_SKEY	V_LOB_SKEY	DIM_LOB	-1	-1	N	

Table 16. Updated Entry of FSI\_BI\_SETUP\_TABLE

TABLE_TABLE_NAME	TABLE_CODE	MEMBER_CODE	SOURCE_DIM_TABLE_NAME	SOURCE_COLUMN_NAME	SOURCE_COLUMN_NAME	JOIN_REQUIRED	GROUP_BY_REQUIRED
FCT_LEDGER_STAT	N_ENTITY_SKEY	N_ENTITY_ID	DIM_ORG_STRUCTURE	N_ENTITY_ID	N_ENTITY_SKEY	Y	
FCT_LEDGER_STAT	N_LOB_SKEY	N_LOB_ID	DIM_LOB	N_LOB_ID	N_LOB_SKEY	Y	

- The new data transformation FN\_ORG\_PARENT\_CHILD has to be re executed with respect to any change in DIM\_ORG\_STRUCTURE table. This process loads the ORG\_STR\_PARENT\_CHILD\_RELATION table data, which is sourced from DIM\_ORG\_STRUCTURE table
- The columns N\_ENTITY\_ID and N\_LOB\_ID are only required to be present in LEDGER\_STAT if user decides to move data of LOB and ENTITY dimensions to EFPA reporting area. The seeded entries provided during installation doesn't mandate that the both these columns need to present before the transformation is executed.

## Executing the Fact Ledger Population Transformation

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

This component for OFSEFPA 6.1 has been seeded with the following Batch ID

<INFODOM>\_Fact\_Table\_Transformation - Task1, which can be executed from *Batch Execution* section of OFSAAI. A single Batch triggers the transformation for both FCT\_LEDGER\_STAT and FCT\_MGMT\_REPORTING as separate tasks. You can execute these task individually, by excluding the other. In the Parameter List, include pstart\_month, pend\_month, pyears, pidentity\_code, psource\_type, pre\_run\_flg, and prcy. For example, 1,12,1994,"','Y','USD'.

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

To define a new task for a Batch definition:

1. Select the check box adjacent to the newly created Batch Name in the *Batch Maintenance* window.
2. Click **Add (+)** button from the Task Details grid.  
The *Task Definition* window is displayed.
3. Enter the **Task ID** and **Description**.
4. Select **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 list.
- **Datastore Name** - Select the appropriate datastore name from the list.
- **IP address** - Select the IP address from the list.
- **Rule Name** - Select **FSI\_LEDGER\_STAT\_TRM** from the drop down list of available transformations. (This is a seeded Data Transformation which is installed as part of the OFSEFPA solution installer. If you don't see this in the list, contact Oracle support).
- **Parameter List** - Enter pStart\_Month, pEnd\_Month, pYears, pIdentity\_Code, pSource\_Type, pRe\_Run\_Flg, and pRCY.

- **pStart\_Month** - This parameter indicates the starting Month.
- **pEnd\_Month** - This parameter indicates the ending Month.
- **pYears** - This is a mandatory parameter that indicates the Year value.
- **pIdentity\_Code** - This is an optional parameter that indicates the Identity Code.

This is the identity code in OFSPM LEDGER\_STAT table. The value '0' in this field indicates, only the rows in LEDGER\_STAT with source code '0' should get processed. Source code '0' indicates rows in LEDGER\_STAT loaded by the ledger load program. This results in movement of rows loaded by ledger load program to FCT\_LEDGER\_STAT in OFSEFPA solution. Similarly, any particular allocation output values can be moved by filtering on the identity code.

- **pSource\_Type** - This is an optional parameter that indicates the Source Type.  
Source Type indicate which process populated a row in LEDGER\_STAT of profitability solution. For example, '0' indicates it was loaded by the Ledger Load program and '100' indicates the allocation rule populated it, and so on.
- **pRe\_Run\_Flg** - This is an optional parameter that indicates Re-run Flag. If value is 'Y', the existing data in the fact table will be removed and reloaded.
- **pRCY** - This indicates the reporting currency with Default Value 'USD'.

6. Click **Save**.

The Task definition is saved for the selected Batch.

7. Execute the Batch.

You can execute a Batch definition from the *Batch Execution* section of *OFSAAI Operations* module. Ledger Stat Transformation can also be directly executed on the database through SQLPLUS.

Details are:

**Function Name:** FSI\_LEDGER\_STAT\_TRM

**Parameters:** pBatch\_Id, pAs\_of\_date, pStart\_Month, pEnd\_Month, pYears, pIdentity\_Code, pSource\_Type, pRe\_Run\_Flg, and prcy.

Sample parameter values are 'Batch1', '20091231', 1, 8, 2009, 0, 0, 'Y', and 'GBP' respectively.

## 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.

## Optional Load from Ledger Class Entities for Fact Ledger Population

**Batch Name:** <Infodom>\_MANAGEMENT\_LEDGER

**Task1:** Loads data from Fsi\_D\_Management\_Ledger to Fct\_Ledger\_stat

Parameter to be configured: [DRCY]=To\_Currency\_Code

**Example**

[DRCY]=USD

Data will be loaded for the data selected while running the batch.

## Identity Codes Used in Financial Reporting

Financial Reporting reports are used to show the custom built GL hierarchy. This GL hierarchy is treated as a Reporting Line dimension in Financial Reporting (FR) Dashboard pages. The idea of reports in FR is to show the unallocated data that has been downloaded from source system and rule out the allocated data records. In order to filter out the unallocated data a SETUP\_MASTER configuration table entry is required with V\_COMPONENT\_DESC column value as 'IDENTITY\_CODE\_PFTBI\_FR\_UNALLOCATED\_DATA'. The other two columns of SETUP\_MASTER can be populated as: V\_COMPONENT\_CODE- Identity Code corresponding to unallocated data. V\_COMPONENT\_VALUE- Identity Code corresponding to unallocated data. Refer to FSI\_DATA\_IDENTITY table to find out the IDENTITY CODES. Generally, the source\_type = 0 highlights the records that are unallocated.





# *Fact Management Reporting Population from LEDGER\_STAT Table*

Fact Management Reporting Population involves populating the FCT\_MGMT\_REPORTING table from the LEDGER\_STAT table.

This chapter covers the following topics:

- [Overview of Fact Management Reporting Transformation](#)
- [Prerequisites](#)
- [Tables Used by the Fact Management Reporting Transformation](#)
- [Map New Dimensions](#)
- [Executing the Fact Management Reporting Transformation](#)
- [Checking the Execution Status](#)
- [Optional Load from Ledger Class Entities for Fact Management Reporting Population](#)
- [Rollup Signage and Operational Signage](#)

## **Overview of Fact Management Reporting Transformation**

The LEDGER\_STAT table is optimized for processing purposes, but is not a convenient structure for reporting purposes. In generating FCT\_MGMT\_REPORTING table, time from LEDGER\_STAT table is transformed into an explicit dimension in FCT\_MGMT\_REPORTING table. Fact Management Reporting transformation is used to populate the FCT\_MGMT\_REPORTING table from the Profitability LEDGER\_STAT table.

During FCT\_MGMT\_REPORTING table population, the horizontally structured MONTH column in LEDGER\_STAT is transposed to a vertical structure. The twelve Month Columns in LEDGER\_STAT table are replaced by a single N\_DATE\_KEY column in FCT\_MGMT\_REPORTING table. The fact table is populated with reporting line codes from DIM\_REP\_LINE table.

The database components used by the Fact Management Reporting transformation are:

- Database function FN\_MGMT\_REPORTING\_TRM

Database function MGMT\_REPORTING\_TRM, which is called by the function FN\_MGMT\_REPORTING\_TRM mentioned above.

## Prerequisites

1. All the post install steps mentioned in the [Oracle Financial Services Analytical Applications Infrastructure Installation and Configuration Guide](#) and the solution installation manuals of *Profitability Management* (only if OFSPM is installed) and *Enterprise Financial Performance Analytics* have to be completed successfully.
2. Application User must be mapped to a role that has seeded batch execution function (BATPRO).
3. Ensure that your FISCAL year information is configured properly. It has the following two columns:
  - **FISCAL\_PERIOD**: This gives the number of months in the given FISCAL period.
  - **START\_MONTH**: This indicates which month of the calendar year is the FISCAL starting month. For example, a value '1' for this column means FISCAL year starts from January and value of '4' indicates that the FISCAL year starts from April.
4. Before executing a batch, check if the following services are running on the application server:
  - Iccserver
  - Router
  - AM Server
  - Messageserver
  - Olapdataserver

For more information on how to check if the services are up and how to start the services, refer to [Oracle Financial Services Analytical Applications Infrastructure User Guide](#).
5. The following seeded Batch should be available for execution:

```
<INFODOM>_Fact_Table_Transformation - Task2.
```

## Tables Used by the Fact Management Reporting Transformation

- **FSI\_FISCAL\_YEAR\_INFO** - This table has the FISCAL year info. The entries required in this table are mentioned in the Prerequisites section.
- **FSI\_BI\_SETUP\_TABLE** - This table has the setup information used by the Fact Management Reporting Transformation. They are:
  - **TARGET\_TABLE\_NAME** - This is the destination table name for transformation.
  - **TARGET\_COLUMN\_NAME** - This is the Destination column name in FCT\_MGMT\_REPORTING table.
  - **MEMBER\_COL\_NAME** - This is the column Name in source tables.
  - **SOURCE\_DIM\_TABLE\_NAME** - This is the table to which the Ledger data has to be joined to get the surrogate key value.
  - **SOURCE\_COLUMN\_NAME** - This is the column in the dimension table to which the LEDGER\_STAT\_ID column is joined.

- **SKEY\_COLUMN\_NAME** - This is the column in the dimension table which has the surrogate key value.
- **JOIN\_REQUIRED** - This provides the information, whether the column to be moved to **FCT\_MGMT\_REPORTING** table is directly available in **LEDGER\_STAT** table or a join has to be taken with dimension table to get the key.
- **GROUP\_BY\_REQUIRED** - The group by is used to find the unique records of **LEDGER\_STAT** for which the key column should exist as part of target table's primary key.

Sample data for this table is plotted below:

**Table 17. FSI\_BI\_SETUP\_TABLE**

TARGET_TABLE_NAME	TARGET_COL_NAME	MEMBER_COL_NAME	SOURCE_DIM_TABLE_NAME	SOURCE_COLUMN_NAME	SKEY_COLUMN_NAME	JOIN_REQUIRED	GROUP_BY_REQUIRED	IS_DIM_TABLE	IS_EXPRESSION
FCT_MGMT_REPORTING	N_BALANCE_TYPE_CD	BALANCE_TYPE_CD	FSI_DIM_MANAGEMENT_LEDGER	BALANCE_TYPE_CD	BALANCE_TYPE_CD	N	N	N	N
FCT_MGMT_REPORTING	N_COMMON_COA_SKEY	COMMON_COA_ID	DIM_COMMON_COA	N_COMMON_COA_ID	N_COMMON_COA_SKEY	Y	N	Y	N
FCT_MGMT_REPORTING	N_ENTITY_SKEY	N_ENTITY_ID	FSI_DIM_MANAGEMENT_LEDGER	-1	-1	N	Y	N	Y
FCT_MGMT_REPORTING	N_FINANCIAL_ELEM_ID	FINANCIAL_ELEM_ID	FSI_DIM_MANAGEMENT_LEDGER	FINANCIAL_ELEM_ID	FINANCIAL_ELEM_ID	N	N	N	N
FCT_MGMT_REPORTING	N_GL_ACCOUNT_SKEY	GL_ACCOUNT_ID	DIM_GL_ACCOUNT	N_GL_ACCOUNT_ID	N_GL_ACCOUNT_SKEY	Y	N	Y	N
FCT_MGMT_REPORTING	N_IDENTITY_CODE	IDENTITY_CODE	FSI_DIM_MANAGEMENT_LEDGER	IDENTITY_CODE	IDENTITY_CODE	N	N	N	N
FCT_MGMT_REPORTING	N_LOB_SKEY	N_LOB_ID	FSI_DIM_MANAGEMENT_LEDGER	-1	-1	N	Y	N	Y
FCT_MGMT_REPORTING	N_ORG_UNIT_SKEY	ORG_UNIT_ID	DIM_ORG_UNIT	N_ORG_UNIT_ID	N_ORG_UNIT_SKEY	Y	Y	Y	N
FCT_MGMT_REPORTING	N_PRODUCT_SKEY	PRODUCT_ID	DIM_PRODUCT	N_PRODUCT_ID	N_PRODUCT_SKEY	Y	Y	Y	N

Table 17. FSI\_BI\_SETUP\_TABLE

TARGET_TABLE_NAME	TARGET_COL_NAME	MEMBER_COL_NAME	SOURCE_DIM_TABLE_NAME	SOURCE_COLUMN_NAME	KEY_COLUMN_NAME	JOIN_REQUIRED	GROUP_BY_REQUIRED	IS_DIM_TABLE	IS_EXPRESSION
FCT_MGMT_REPORTING	N_SCENARIO_CD	CONSOLIDATION_CD	FSI_DIM_MANAGEMENT_LEDGER	CONSOLIDATION_CD	CONSOLIDATION_CD	N	Y	N	N
FCT_MGMT_REPORTING	V_ISO_CURRENCY_CD	ISO_CURRENCY_CD	FSI_DIM_MANAGEMENT_LEDGER	ISO_CURRENCY_CD	ISO_CURRENCY_CD	N	Y	N	N

- LEDGER\_STAT - This table is the source for the transformation.
- DIM\_<dimension Name> - The flattened dimension tables used in Business Intelligence (BI) reporting are accessed to obtain the surrogate key to be populated to FCT\_MGMT\_REPORTING dimension columns. For example, DIM\_ORG\_UNIT, DIM\_PRODUCT, and so on.
- FCT\_MGMT\_REPORTING - This is the output table for the transformation.

For more details on viewing the structure of these tables, refer to [Oracle Financial Services Analytical Applications Data Model Data Dictionary](#) or the [OFSEFPA Erwin Data Model](#).

## Map New Dimensions

The dimensions which you can customize during OFSEFPA implementation are known as Custom Dimensions. Custom Dimensions can be added to target FCT\_MGMT\_REPORTING table by updating the configuration table FSI\_BI\_SETUP\_TABLE. DIM\_LOB and DIM\_ORG\_STRUCTURE are two Custom Dimension tables, for which the dimension data has to be entered manually during implementation.

**Note:** All nodes in FCT\_MGMT\_REPORTING table should be part of the respective hierarchies for which the SCDs run. Otherwise, the cube will fail.

## Hierarchy Configuration for LOB and Legal Entity

### DIM\_LOB

The hierarchy to be built for LOB (Line of Business) is a level based hierarchy. All leaf nodes of this hierarchy are mapped to a single root. The branch navigation of the hierarchy starts from root node which is the Level 1 column of the DIM\_LOB table. The Leaf node has to be placed at the appropriate column between Level 1 and Level 16 of the DIM\_LOB table. This step is required only if the reports are working out of ESSBASE cubes.

### DIM\_ORG\_STRUCTURE

The hierarchy to be built for DIM\_ORG\_STRUCTURE is parent-child hierarchy. The parent node of the root has to be kept blank in the V\_PARENT\_CODE column. The hierarchy can then be built appropriately by putting a child node - parent node combination in the V\_ENTITY\_CODE and V\_PARENT\_CODE columns of the DIM\_ORG\_STRUCTURE table. This metadata setup supports only one hierarchy for Legal Entity.

V\_LCY\_CODE column needs to be entered for each legal entity. This currency column is used to calculate the amount in local currency. FSI\_EXCHANGE\_RATE\_HIST table will then guide the conversion from base currency to local currency columns in fact table.

Follow the below steps manually to populate data into DIM\_LOB and DIM\_ORG\_STRUCTURE custom dimensions for the fact table:

#### Procedure

1. Add the columns N\_LOB\_ID and N\_ENTITY\_ID as part of Unique Key (Ledger\_stat) in FCT\_LEDGER\_STAT table.
2. Update the columns SOURCE\_COLUMN\_NAME and SKEY\_COLUMN\_NAME for the target table FCT\_MGMT\_REPORTING in FSI\_BI\_SETUP\_TABLE, with the actual source and skey column values, that is N\_LOB\_ID and N\_ENTITY\_ID.

3. Set the value to 'Y' in JOIN\_REQUIRED and GROUP\_BY\_REQUIRED columns of FCT\_MGMT\_REPORTING table.

**Table 18. Seeded entry of FSI\_BI\_SETUP\_TABLE**

TABLE_TABLE_NAME	TABLE_CODE	MEMBER_CODE	SOURCE_DIM_TABLE_NAME	SOURCE_COLUMN_NAME	KEY_COLUMN_NAME	JOIN_REQUIRED	GROUP_BY_REQUIRED
FCT_MGMT_REPORTING	N_ENTITY_SKEY	V_ENTITY_SKEY	DIM_ORG_STRUCTURE	-1	-1	N	Y
FCT_MGMT_REPORTING	N_LOB_SKEY	V_LOB_SKEY	DIM_LOB	-1	-1	N	Y

**Table 19. Updated entry of FSI\_BI\_SETUP\_TABLE**

TABLE_TABLE_NAME	TABLE_CODE	MEMBER_CODE	SOURCE_DIM_TABLE_NAME	SOURCE_COLUMN_NAME	KEY_COLUMN_NAME	JOIN_REQUIRED	GROUP_BY_REQUIRED
FCT_MGMT_REPORTING	N_ENTITY_SKEY	N_ENTITY_ID	DIM_ORG_STRUCTURE	N_ENTITY_ID	N_ENTITY_SKEY	Y	Y
FCT_MGMT_REPORTING	N_LOB_SKEY	N_LOB_ID	DIM_LOB	N_LOB_ID	N_LOB_SKEY	Y	Y

**Note:** The columns N\_ENTITY\_ID and N\_LOB\_ID are only required to be present in LEDGER\_STAT if user decides to move data of LOB and ENTITY dimensions to EFPA reporting area. The seeded entries provided during installation doesn't mandate that the both these columns need to present before the transformation is executed.

## Executing the Fact Management Reporting Transformation

You can execute the function from the Operations module of OFSAAI.

This component for OFSEFPA has been seeded with the Batch ID <INFODOM>\_Fact\_Table\_Transformation, which can be executed from *Batch Execution* section of OFSAAI. A single batch triggers the transformations for both FCT\_LEDGER\_STAT and FCT\_MGMT\_REPORTING as separate tasks. You can execute these task individually, by excluding the other.

Both Task1 and Task2 are mutually exclusive flows in the runchart. Both flows can not be supported through single setup table (FSI\_BI\_SETUP\_TABLE) with seeded configuration. It is recommended that ootb flow should be management ledger as source for Task 1. So if you want to consider Task 1 for transformation, then FSI\_BI\_SETUP\_TABLE structure should be as it was prior to the Release 8.0.4.0.3.

In the Parameter List, include pstart\_month, pend\_month, pyears, prcy, and pre\_run\_flg. For example, 1,12,1994,'USD','Y'.

You can also define a new Batch and an underlying Task definition from the *Batch Maintenance* window of OFSAAI.

For more information on defining a new Batch, refer to section [How to Define a Batch](#).

A seeded batch <INFODOM>\_Fact\_Table\_Transformation - Task2 has to be executed. Parameter list has to be provided in the *Batch Maintenance* window after selecting Batch and the Task2. You can use Edit mode to provide the parameters associated with the Batch. Sample list of parameters is mentioned below.

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 list.
  - **Datastore Name** - Select the appropriate datastore name from the list.
  - **IP address** - Select the IP address from the list.
  - **Rule Name** - Select `FCT_MGMT_TRANSFORMATION` from the drop down list of available transformations. (This is a seeded Data Transformation which is installed as part of the OFSEFPA solution installer. If you don't see this in the list, contact Oracle support).
  - **Parameter List** - Enter `pstart_month`, `pend_month`, `pyears`, `prcy`, and `pre_run_flg`.
    - **pStart\_Month** - This parameter indicates the starting Month.
    - **pEnd\_Month** - This parameter indicates the ending Month.
    - **pYears** - This is a mandatory parameter that indicates the Year value.
    - **prcy** - This indicates the reporting currency with default value 'USD'.
    - **pRe\_Run\_Flg** - This is an optional parameter that indicates Re-run Flag. If value is 'Y', the existing data in the fact table will be removed and reloaded.
6. Click **Save**.

The Task definition is saved for the selected Batch.
7. Execute the Batch.

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

Fact Management Reporting Transformation can also be directly executed on the database through SQLPLUS.

Details are:

**Function Name:** `FN_MGMT_REPORTING_TRM`

**Parameters:** `pBatch_Id`, `pAs_of_date`, `pStart_Month`, `pEnd_Month`, `pYears`, `prcy`, and `pRe_Run_Flg`

Sample Parameter Values are 'Batch1', '20091231', 1, 8, 2009, 'USD', and 'Y' respectively.

## 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.

## Optional Load from Ledger Class Entities for Fact Management Reporting Population

**Batch Name:** <Infodom>\_MANAGEMENT\_LEDGER

**Task2:** Loads data from LEDGER\_STAT to Fct\_Mgmt\_Reporting

Parameter to be configured:

(pStart\_Month,pEnd\_Month,pYears,pRCY,pRe\_Run\_Flg,pTable\_Name)

Sample Parameters Configured:

(1,12,2012,'USD','Y','LEDGER\_STAT')

Add the column mapping between "Ledger Class" Entity, Dimension Column and Target Column in FSI\_BI\_SETUP\_TABLE.

**Note:** The parameters with source name have to be passed in capital letters, as the FSI\_BI\_SETUP\_TABLE was seeded with configuration in capital letters.

As part of this transformation, among the fields uploaded in FCT\_MGMT\_REPORTING, below fields are being used in EFPA Reports:

- N\_EOP\_BAL: Ending balance derived from FE 100
- N\_AVG\_BAL: Average Balance derived from FE 140
- N\_EOP\_BAL\_RCY: Ending Balance converted to Reporting Currency
- N\_AVG\_BAL\_RCY: Average Balance converted to Reporting Currency
- N\_EOP\_BAL\_LCY: Ending Balance converted to Local Currency, as defined in dim\_org\_structure.v\_lcy\_code defined against each entity and exchange rate
- N\_AVG\_BAL\_LCY: Average Balance converted to Local Currency, as defined in dim\_org\_structure.v\_lcy\_code defined against each entity and exchange rate

## Rollup Signage and Operational Signage

In the context of Reporting Lines, the significance of Signage is that it indicates whether the Reporting Line Value in question will be an addition or subtraction to the corresponding parent reporting line. It is important to note at this



point that the reporting line values that are loaded on the Fact tables like FCT\_ACCOUNT\_PROFITABILITY or FCT\_MGMT\_REPORTING are leaf level reporting lines.

For example, consider the following hierarchy:

Reporting Line Hierarchy
▽ Income before Taxes
▷ Total Revenue
▷ Net Credit Losses
▽ Operating Expenses
Deposit Insurance
▽ Advertising and Marketing
Total Brand Management Expenses
Business Promotion Expenses
▷ Other Allocated Costs
▷ Processing Expenses
▷ Sales and Marketing Expenses
▷ Product Management Expenses
▷ Business Management Expenses
Indirect Processing Expense

Figure 2. Reporting Line Hierarchy

The Fact table will not contain values for **Advertising and Marketing**, as that value is expected to be calculated based on the rollup of the underlying leaf level values – **Total Brand Management Expenses and Business Promotion Expenses**. However, all the underlying values will not be added together. Some of the values expected are positive and some negative. For example, the following are the leaf level values:

Reporting Line Hierarchy	Rollup Signage
▽ Income before Taxes	1
▷ Total Revenue	1
▷ Net Credit Losses	-1
▽ Operating Expenses	-1
Deposit Insurance	-1
▽ Advertising and Marketing	1
Total Brand Management Expenses	1
Business Promotion Expenses	1
▷ Other Allocated Costs	1
▷ Processing Expenses	1
▷ Sales and Marketing Expenses	1
▷ Product Management Expenses	1

Figure 3. Reporting Line Hierarchy - Rollup Signage

Hence, when **Deposit Insurance** rollup into **Operating Expenses**, it is considered a subtraction. This type of rollup into the immediate parent is called **Rollup Signage**.

However, on rolling up further (**Income before Taxes** in the following example), the signage of **Deposit Insurance** is dependent on the rollup signage of **Operating Expenses**.

Operating Expenses = (-1) x Deposit Insurance

Income before Taxes = (-1) x Operating Expenses

Hence, when the leaf value Deposit Insurance rollup into Income before Taxes,

Income before Taxes = (-1) x (-1) x Deposit Insurance = (+1) x Deposit Insurance.

Hence, the rollup signage of Deposit Insurance is -1 (or negative).

However, in relation to Income before Taxes, the Operational Signage of Deposit Insurance is +1 (or positive).

The effective signage of the leaf reporting line with respect to a parent reporting line is called Operational Signage.

It is important to note that the Operational Signage of a reporting line is defined in relation to a parent reporting line. However, the Rollup Signage is always in relation to the immediate parent reporting line.

# *Fact Management Reporting Population from FSI\_D\_MANAGEMENT\_LEDGER Table*

Fact Management Reporting Population involves populating the FCT\_MGMT\_REPORTING table from the FSI\_D\_MANAGEMENT\_LEDGER table.

This chapter covers the following topics:

- [Overview of Fact Management Reporting Transformation](#)
- [Prerequisites](#)
- [Tables Used by the Fact Management Reporting Transformation](#)
- [Map New Dimensions](#)
- [Executing the Fact Management Reporting Transformation](#)
- [Checking the Execution Status](#)

## **Overview of Fact Management Reporting Transformation**

This transformation populates the FCT\_MGMT\_REPORTING table from FSI\_D\_MANAGEMENT\_LEDGER.

The database components used by the Fact Management Reporting transformation are:

- Database function FN\_MGMT\_REP\_MGMT\_LEDGER\_TRM
- Database function MGMT\_REPORTING\_MGMT\_LEDGER\_TRM, which is called by the function FN\_MGMT\_REP\_MGMT\_LEDGER\_TRM mentioned above.
- Database functions MGMT\_REPORTING\_MGMT\_LEDGER\_TRM and FN\_MGMT\_REP\_MGMT\_LEDGER\_TRM allow you to specify source table name. This will allow the procedure to run multiple times to load various management Ledger sources into a single FCT table for reporting

## Prerequisites

1. All the post install steps mentioned in the [Oracle Financial Services Analytical Applications Infrastructure Installation and Configuration Guide](#) and the solution installation manuals of *Profitability Management* and *Enterprise Financial Performance Analytics* have to be completed successfully.
2. Application User must be mapped to a role that has seeded batch execution function (BATPRO).
3. Ensure that your FISCAL year information is configured properly. It has the following two columns:
  - FISCAL\_PERIOD: This gives the number of months in the given FISCAL period.
  - START\_MONTH: This indicates which month of the calendar year is the FISCAL starting month. For example, a value '1' for this column means FISCAL year starts from January and value of '4' indicates that the FISCAL year starts from April.
4. Before executing a batch, check if the following services are running on the application server:
  - Iccserver
  - Router
  - AM Server
  - Messageserver
  - Olapdataserver

For more information on how to check if the services are up and how to start the services, refer to [Oracle Financial Services Analytical Applications Infrastructure User Guide](#).

5. The following seeded Batch should be available for execution:

```
<INFODOM>_Fact_Table_Transformation - Task2.
```

## Tables Used by the Fact Management Reporting Transformation

- FSI\_FISCAL\_YEAR\_INFO - This table has the FISCAL year info. The entries required in this table are mentioned in the Prerequisites section.
- FSI\_BI\_SETUP\_TABLE - This table has the setup information used by the Fact Management Reporting Transformation. They are:
  - TARGET\_TABLE\_NAME - This is the destination table name for transformation.
  - TARGET\_COLUMN\_NAME - This is the Destination column name in FCT\_MGMT\_REPORTING table.
  - MEMBER\_COL\_NAME - This is the column Name in source tables.
  - SOURCE\_DIM\_TABLE\_NAME - This is the table to which the Ledger data has to be joined to get the surrogate key value.
  - SOURCE\_COLUMN\_NAME - This is the column in the dimension table
  - SKEY\_COLUMN\_NAME - This is the column in the dimension table which has the surrogate key value.

- JOIN\_REQUIRED - This provides the information, whether the column to be moved to FCT\_MGMT\_REPORTING table is directly available in fsi\_d\_management\_ledger table or a join has to be taken with dimension table to get the key.
- GROUP\_BY\_REQUIRED - The group by is used to find the unique records of fsi\_d\_management\_ledger for which the key column should exist as part of target table's primary key.

Sample data for this table is plotted below:

**Table 20. FSI\_BI\_SETUP\_TABLE**

TARGET_TABLE_NAME	TARGET_COL_NAME	MEMBER_COL_NAME	SOURCE_DIM_TABLE_NAME	SOURCE_COLUMN_NAME	KEY_COLUMN_NAME	JOIN_REQUIRED	GROUP_BY_REQUIRED	IS_DIM_TABLE	IS_EXPRESSION
FCT_MGMT_REPORTING	N_BALANCE_TYPE_CD	BALANCE_TYPE_CD	FSI_D_MANAGEMENT_LEDGER	BALANCE_TYPE_CD	BALANCE_TYPE_CD	N	N	N	N
FCT_MGMT_REPORTING	N_COMMON_COA_SKEY	COMMON_COA_ID	DIM_COMMON_COA	N_COMMON_COA_ID	N_COMMON_COA_SKEY	Y	N	Y	N
FCT_MGMT_REPORTING	N_ENTITY_SKEY	N_ENTITY_ID	FSI_D_MANAGEMENT_LEDGER	-1	-1	N	Y	N	Y
FCT_MGMT_REPORTING	N_FINANCIAL_ELEM_ID	FINANCIAL_ELEM_ID	FSI_D_MANAGEMENT_LEDGER	FINANCIAL_ELEM_ID	FINANCIAL_ELEM_ID	N	N	N	N
FCT_MGMT_REPORTING	N_GL_ACCOUNT_SKEY	GL_ACCOUNT_ID	DIM_GL_ACCOUNT	N_GL_ACCOUNT_ID	N_GL_ACCOUNT_SKEY	Y	N	Y	N
FCT_MGMT_REPORTING	N_IDENTITY_CODE	IDENTITY_CODE	FSI_D_MANAGEMENT_LEDGER	IDENTITY_CODE	IDENTITY_CODE	N	N	N	N
FCT_MGMT_REPORTING	N_LOB_SKEY	N_LOB_ID	FSI_D_MANAGEMENT_LEDGER	-1	-1	N	Y	N	Y
FCT_MGMT_REPORTING	N_ORG_UNIT_SKEY	ORG_UNIT_ID	DIM_ORG_UNIT	N_ORG_UNIT_ID	N_ORG_UNIT_SKEY	Y	Y	Y	N
FCT_MGMT_REPORTING	N_PRODUCT_SKEY	PRODUCT_ID	DIM_PRODUCT	N_PRODUCT_ID	N_PRODUCT_SKEY	Y	Y	Y	N

Table 20. FSI\_BI\_SETUP\_TABLE

TARGET_TABLE_NAME	TARGET_COL_NAME	MEMBER_COL_NAME	SOURCE_DIM_TABLE_NAME	SOURCE_COLUMN_NAME	KEY_COLUMN_NAME	JOIN_REQUIRE	GROUP_BY_REQUIRED	IS_DIM_TABLE	IS_EXPRESSSION
FCT_MGMT_REPORTING	N_SCENARIO_CD	CONSOLIDATION_CD	FSI_D_MANAGEMENT_LEDGER	CONSOLIDATION_CD	CONSOLIDATION_CD	N	Y	N	N
FCT_MGMT_REPORTING	V_ISO_CURRENCY_CD	ISO_CURRENCY_CD	FSI_D_MANAGEMENT_LEDGER	ISO_CURRENCY_CD	ISO_CURRENCY_CD	N	Y	N	N

- FSI\_D\_MANAGEMENT\_LEDGER - This table is the source for the transformation.
- DIM\_<dimension Name> - The flattened dimension tables used in Business Intelligence (BI) reporting are accessed to obtain the surrogate key to be populated to FCT\_MGMT\_REPORTING dimension columns.  
For example, DIM\_ORG\_UNIT, DIM\_PRODUCT, and so on.
- FCT\_MGMT\_REPORTING - This is the output table for the transformation

For more details on viewing the structure of these tables, refer to [Oracle Financial Services Analytical Applications Data Model Data Dictionary](#) or the [OFSEFPA Erwin Data Model](#).

## Map New Dimensions

The dimensions which you can customize during OFSEFPA implementation are known as Custom Dimensions. Custom Dimensions can be added to target FCT\_MGMT\_REPORTING table by updating the configuration table FSI\_BI\_SETUP\_TABLE. DIM\_LOB and DIM\_ORG\_STRUCTURE are two Custom Dimension tables, for which the dimension data has to be entered manually during implementation.

**Note:** All nodes in FCT\_MGMT\_REPORTING table should be part of the respective hierarchies for which the SCDs run. Otherwise, the cube will fail.

## Executing the Fact Management Reporting Transformation

You can execute the Fact Management Reporting Transformation function from the Operations module of OFSAAL.

This component has been seeded with the Batch ID <INFODOM>\_MANAGEMENT\_LEDGER, which can be executed from Batch Execution section of OFSAAL. A single batch triggers the transformations for both FCT\_LEDGER\_STAT and FCT\_MGMT\_REPORTING as separate tasks. You can execute these task individually, by excluding the other. In the Parameter List, include pstart\_month, pend\_month, pyears, prcy, pre\_run\_flg and pTable\_name. A detailed description of each parameter is provided under point 5 below.

For example: 1, 8, 2018, 'USD', 'Y', 'FSI\_D\_MANAGEMENT\_LEDGER'

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

To define a new task for a Batch definition:

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

The *Task Definition* window is displayed.

3. Enter the **Task ID** and **Description**.
4. Select **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 list.
  - **Datastore Name** - Select the appropriate datastore name from the list.
  - **IP address** - Select the IP address from the list.
  - **Rule Name** - Select FCT\_MGMT\_TRANSFORMATION from the drop down list of available transformations. (This is a seeded Data Transformation which is installed as part of the OFSEFPA solution installer. If you don't see this in the list, contact Oracle Support).
  - **Parameter List** - Enter pstart\_month, pend\_month, pyears, prcy, pre\_run\_flg and pTable\_name
    - ◆ **pStart\_Month** - This parameter indicates the starting Month.
    - ◆ **pEnd\_Month** - This parameter indicates the ending Month.
    - ◆ **pYears** - This is a mandatory parameter that indicates the Year value.
    - ◆ **prcy** - This indicates the reporting currency with default value 'USD'.
    - ◆ **pRe\_Run\_Flg** - This is an optional parameter that indicates Re-run Flag. If value is 'Y', the existing data in the fact table will be removed and reloaded.
    - ◆ **pTable\_name** - This is a mandatory parameter takes source table 'FSI\_D\_MANAGEMENT\_LEDGER' as a value.

6. Click **Save**.

The Task definition is saved for the selected Batch.

7. Execute the Batch.

You can execute a Batch definition from the Batch Execution section of OFSAAI Operations module. Fact Management Reporting Transformation can also be directly executed on the database through SQLPLUS.

Details are:

- **Function Name:** fn\_mgmt\_rep\_mgmt\_ledger\_trm
- **Parameters:** pBatch\_Id, pAs\_of\_date, pStart\_Month, pEnd\_Month, pYears, pRCY, pRe\_Run\_Flg and pTable\_name

Sample Parameter Values are 'Batch1', '20091231', 1, 8, 2009, 'USD', 'Y' and 'FSI\_D\_MANAGEMENT\_LEDGER' respectively.

This transformation deletes only the management ledger data which was loaded earlier for the same reporting date, keeping any other data intact in the table `FCT_MGMT_REPORTING`.

## 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.

You can access the sql queries fired as part of this transformation in `FSI_SQL_LOG` table.



This chapter discusses the following topics:

- [Introduction](#)
- [Configuration for What-If Analysis](#)
- [EFPA Integration with Price Creation and Discovery \(PCD\)](#)
- [Reporting Line Correlation Calculations](#)
- [What-If Analysis Limitation](#)

## **Introduction**

This report enables the user to account for the change in profitability owing to any probable changes in the projected components of profitability. The probable change can be defined by the user and is termed as 'Variation'. User could define the parameters to which variation is being applied and the magnitude of variation. The net effect on profitability as a result of these variations can be applied.

The effect of variations on profitability can be analyzed at differing levels of granularity like enterprise, LOB, Product, Customer, and Account. This selection is enabled to the user through dashboard prompt selections. The projected data of income statement is available at an account level. Aggregations are done based on the desired level of granularity. The projections are created based on historical data of account.

User could define the variations through the UI, which when imposed on the income statement provide the resulting net income. The resulting income statement post applying the variations is called as a 'Scenario'. The projections are by default created for a period of 5 years, but the change in projection may not necessarily be applied for the entire 5 years. The tenure for which the specified variation is applicable can also be defined while specifying the variation. The magnitude of variation being applied can be specified either in 'percentage' or 'absolute'. If the variation is specified as percentage then the value of the component to which variation is being applied changes by the corresponding percentage value for the specified time. Similarly, when variation is applied in absolute terms the value of the component to which variation is being applied changes by the corresponding absolute value for the specified time.

Certain users should have the authority to save a scenario which can later be accessed by other users for reference. The variations once applied can be applied on the income statement by either of the following two methods:

- **Basic** - The variations that are applied get simply aggregated with the modified values of components to show the resulting net income. The basic version supports variations to be applied to multiple parameters at the same time.
- **Advanced** - The variations that are applied also affect the other components it is correlated to and the modified values of all such parameters gets aggregated to show the resulting net income. In the Advanced version variation can be applied to only a single component at a time.

The scenarios that are thus created can be used to analyze the outcome on profitability of any probable change in future. Certain users should have the authority to save a scenario which can later be accessed by other users for reference. It is also possible to create a scenario on an existing scenario by applying variations to the components of income statement in the scenario.

Income Statement Variation												
Time run: 3/28/2016 12:20:16 PM												
												Amount in Millions (USD)
	2013		2014		2015		2016		2017		2018	
	Projected Movement	Revised Movement	Projected Movement	Revised Movement	Projected Movement	Revised Movement	Projected Movement	Revised Movement	Projected Movement	Revised Movement	Projected Movement	Revised Movement
Net Income Before Taxes	-246.27		-330.95		-333.85		-336.55		-339.09		-85.13	

The following input parameters are applicable for EFPA:

**Table 21. What-if Input Parameters**

V_PARAMETER_NAME	V_PARAMETER_VALUE
VAR_STRT_DT_EFPA	This input parameter indicates the start date for the variance calculation.
VAR_END_DT_EFPA	This input parameter indicates the end date for the variance calculation.
TSHLD_FCTR_EFPA	This input parameter indicates the threshold factor for replines.
EXCL_LMT_EFPA	This input parameter indicates the maximum outlier exclusion percentage.

**Note:** There is limitation on graph. Default upper limit for the graph will be based on underlying data in fact table. Variations applied above the normal can be applied using the grid. If the user wants to apply variation beyond the upper limit shown in graph, it is not possible through graph and hence, it should be applied through grid.

The same applies to Negative Values. The graph does not allow applying negative variations. That is, the graph nodes cannot be dragged below the X Axis. This change needs to be done using the grid mode.

## Configuration for What-If Analysis

The following list of configurations is to be performed for working on What-If Analysis:

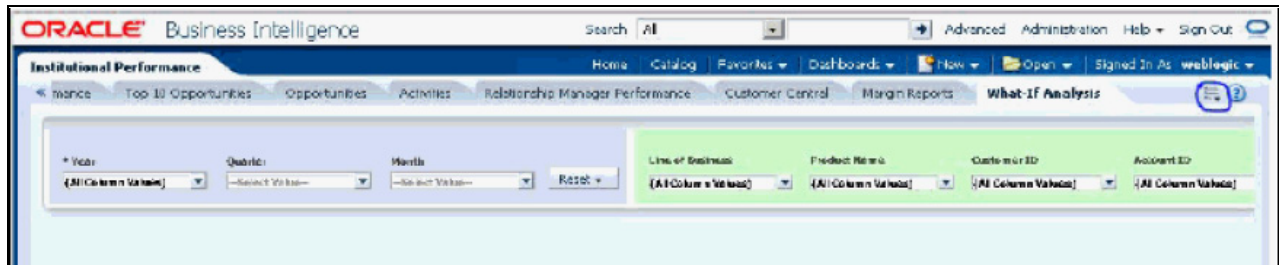
1. Configure connection to the What-If Variation application page in OFSAAI in the Create Scenario Analysis.

OFSAAI is Oracle Financial Services proprietary tool, which uses Java to enable users to apply variations on the projected data.

For example: Assume that the OFSAA hostname is 10.184.150.107 and the OBIEE analytics port is 7001, then the OBIEE analytics access URL would be: `http://bank_host:8080/PFT801`.

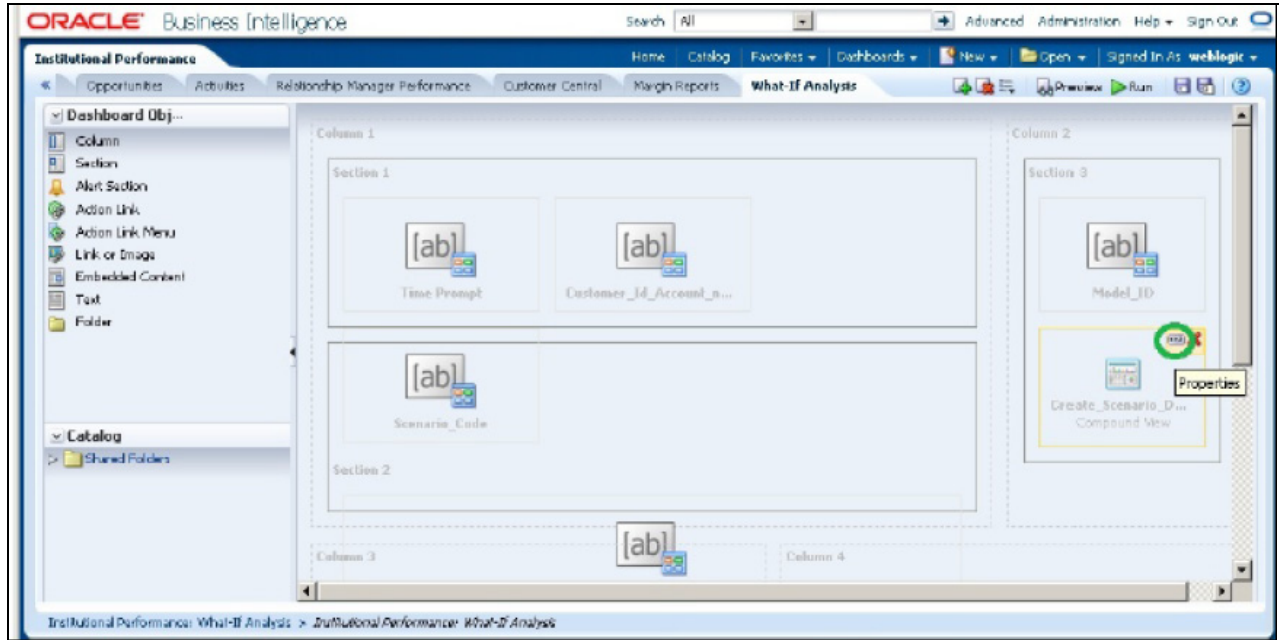
To configure the What-If analysis framework, perform the following steps:

- Navigate to What-If Analysis Dashboard Page in the Dashboard - Institutional Performance for OFSIPA, Retail Performance for OFSRPA, and Income Statement for EFPA.
- Edit Dashboard Page.

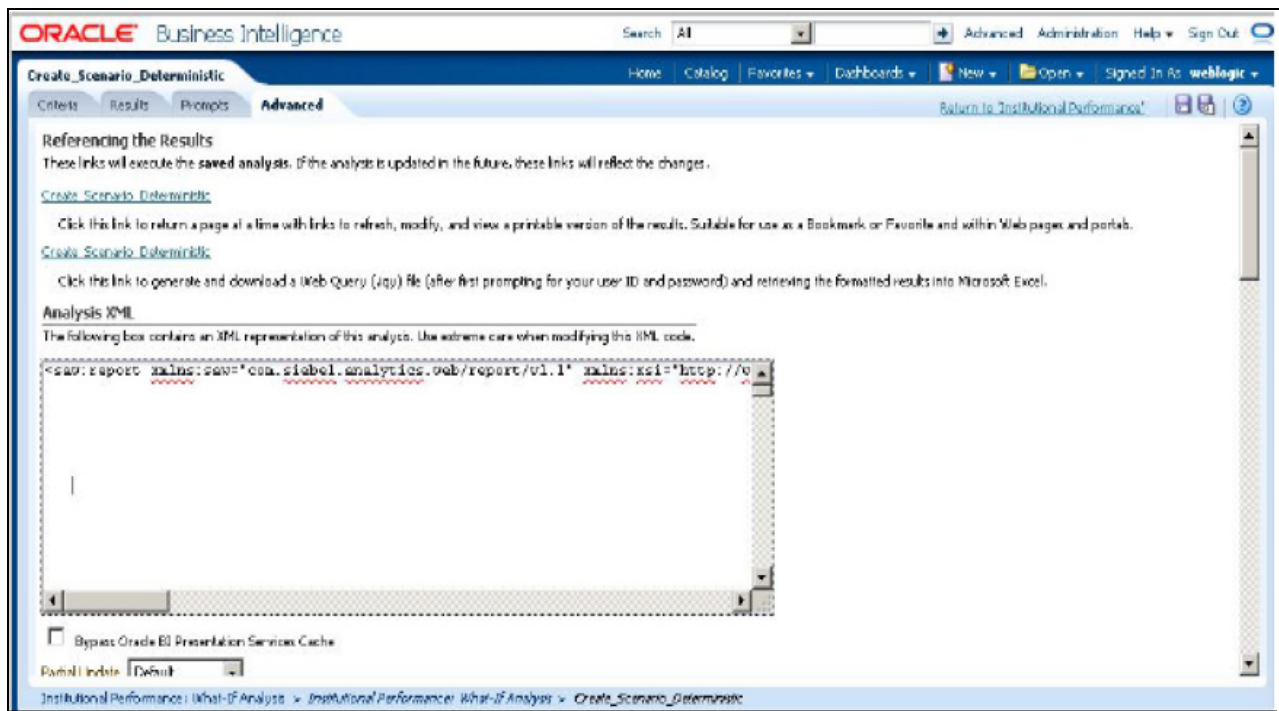


- Edit the analysis Create Scenario.

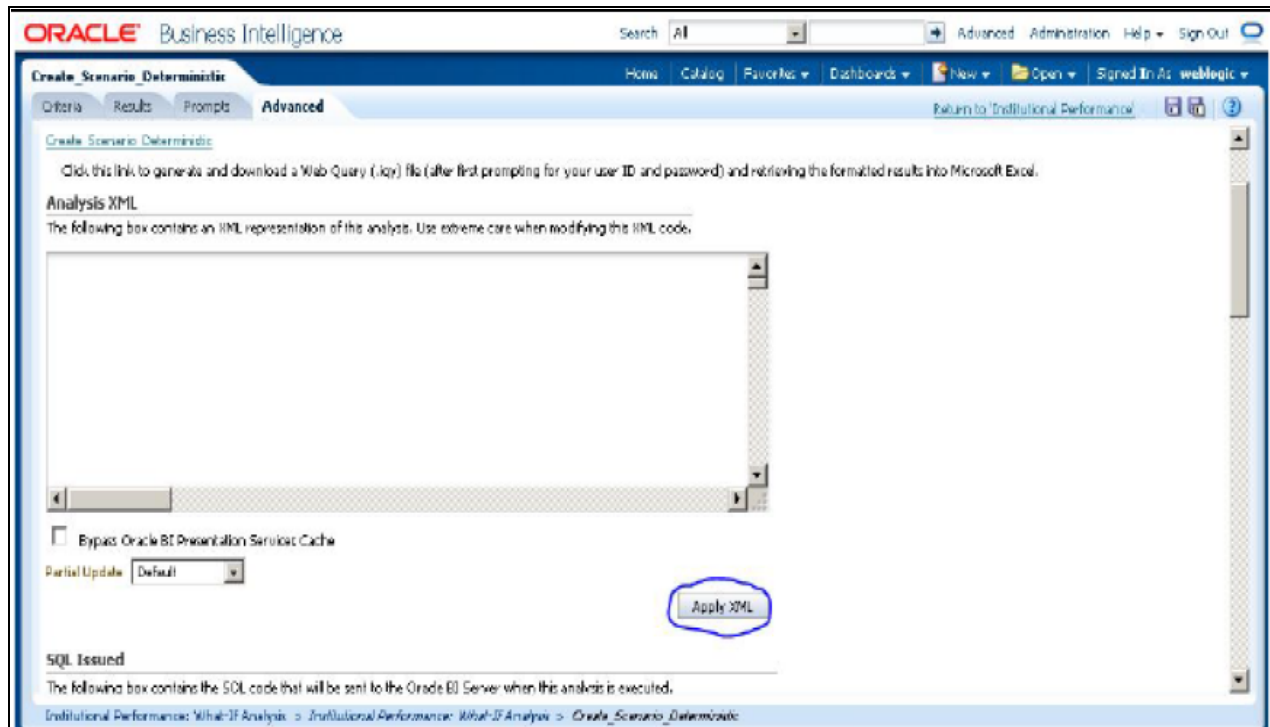
**Note:** The OBIEE users mapped to the role of BI Author or higher will be able to configure the What-If Analysis framework.



- Navigate to the Advanced XML section and edit the contents of the Analysis XML.



- Replace all occurrences as follows:
  - `##ofsaahostname##` with the OFSAAI user hostname (For example: bank\_host)
  - `##ofsaahport##` with the OFSAAI servlet port (For example: 8080)
  - `##ofsaahcontext##` with the context of the OFSAAI instance (For example: PFT801).
- Click Apply XML. Save the analysis after the occurrences of placeholders have been replaced and the XML contents have been pasted.



2. Configure the OBIEE URL in the What-If Model Definition setup tables to navigate between the OFSAAI and OBIEE screens.

**Example:**

Assume that the user hostname is 10.184.150.107 and the OBIEE analytics port is 7001, then the OBIEE analytics access URL would be `http://10.184.150.107:7001/analytics`.

To configure the details on the What-If analysis framework, the user needs to execute the following update on the atomic schema:

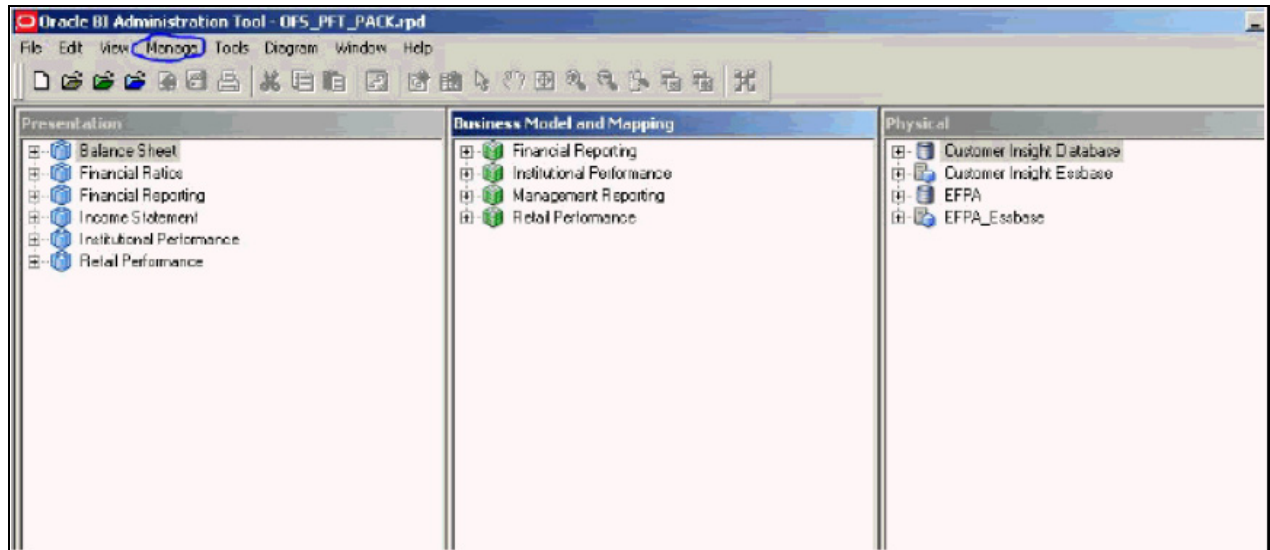
```
update fsi_m_wif_model_defn set output_page =
replace(replace(output_page,'##hostname##', '10.184.150.107' ), '##port##','7001')
/
Commit
/
```

Where the hostname and port replaced would be the user's corresponding hostname and port instead of the examples mentioned.

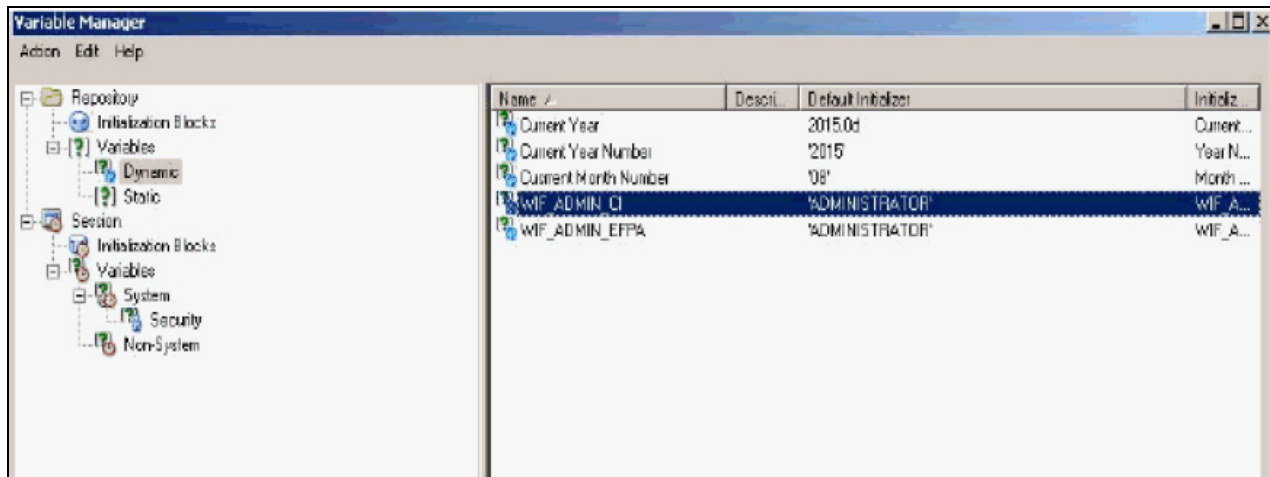
3. Configure What-If Admin Role in RPD to configure security roles to restrict Scenario Creation.

If a user is not mapped to the role of a What-If Analysis administrator, the user will only have access to Display Results. The results of this operation will not be persisted beyond one session per user.

- Open the OFS\_PFT\_PACK RPD and navigate to the Variable Definition Screen.

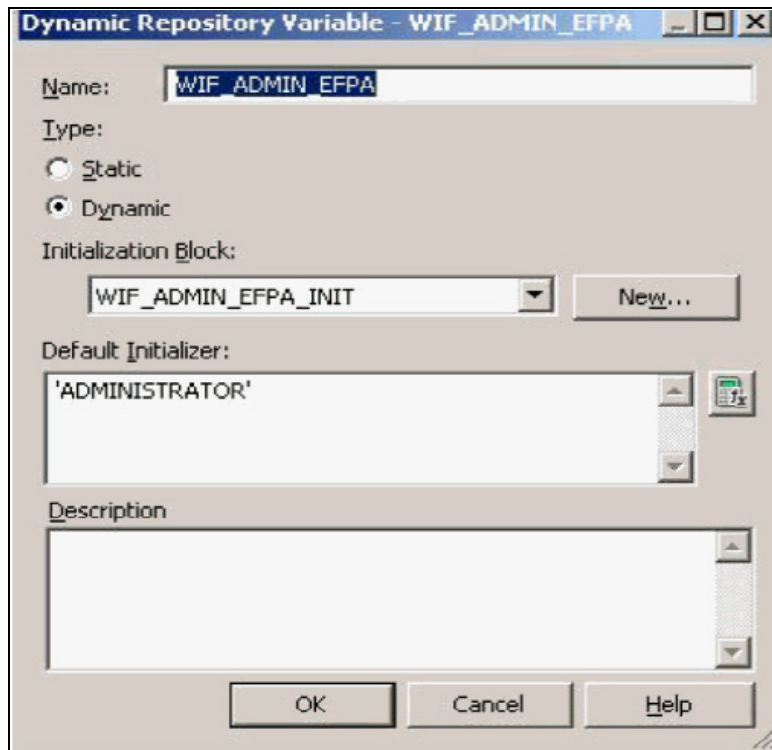


- Modify the WIF\_ADMIN\_CI Repository Dynamic Variable.



- Edit the default initializer to enter the desired What-If Administrator role.

**Note:** The user mapped to the role of Administrator has the privilege to create and save a scenario. Users who are not mapped to the role of Administrator will only be able to create a scenario, but not save it.



If the web server is Tomcat of version  $\geq 8.0.18$ , following additional configuration needs to be done to avoid performance issues while performing What-If Analysis.

1. Add the following tags in the server.xml file under tomcat\_folder/conf/:

Insert the following tag in the Context tag as the first nested tag:

```
<Loader delegate="true"/>
```

2. Insert the following attributes for all the Resource tags in the Context tag:

```
removeAbandonedOnBorrow = "true"
```

```
removeAbandonedOnMaintenance = "true"
```

Example:



```
<Context path="/PFT" docBase="/scratch/ofsaaapp/tomcat-7.0.19/webapps/PFT" debug="0" reloadable="false" crossContext="true"
  <Loader delegate="true"/>
  <resource auth="Container"
    name="jdbc/FICRSTER"
    type="javax.sql.DataSource"
    driverClassName="oracle.jdbc.driver.OracleDriver"
    username="pftconf30"
    password="ofsaa8x"
    url="jdbc:oracle:thin:@10.104.153.87:1521:DEV12C"
    maxActive="1000"
    maxIdle="30"
    maxWait="10000" removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
    removeAbandonedOnBorrow="true" removeAbandonedOnMaintenance="true"/>

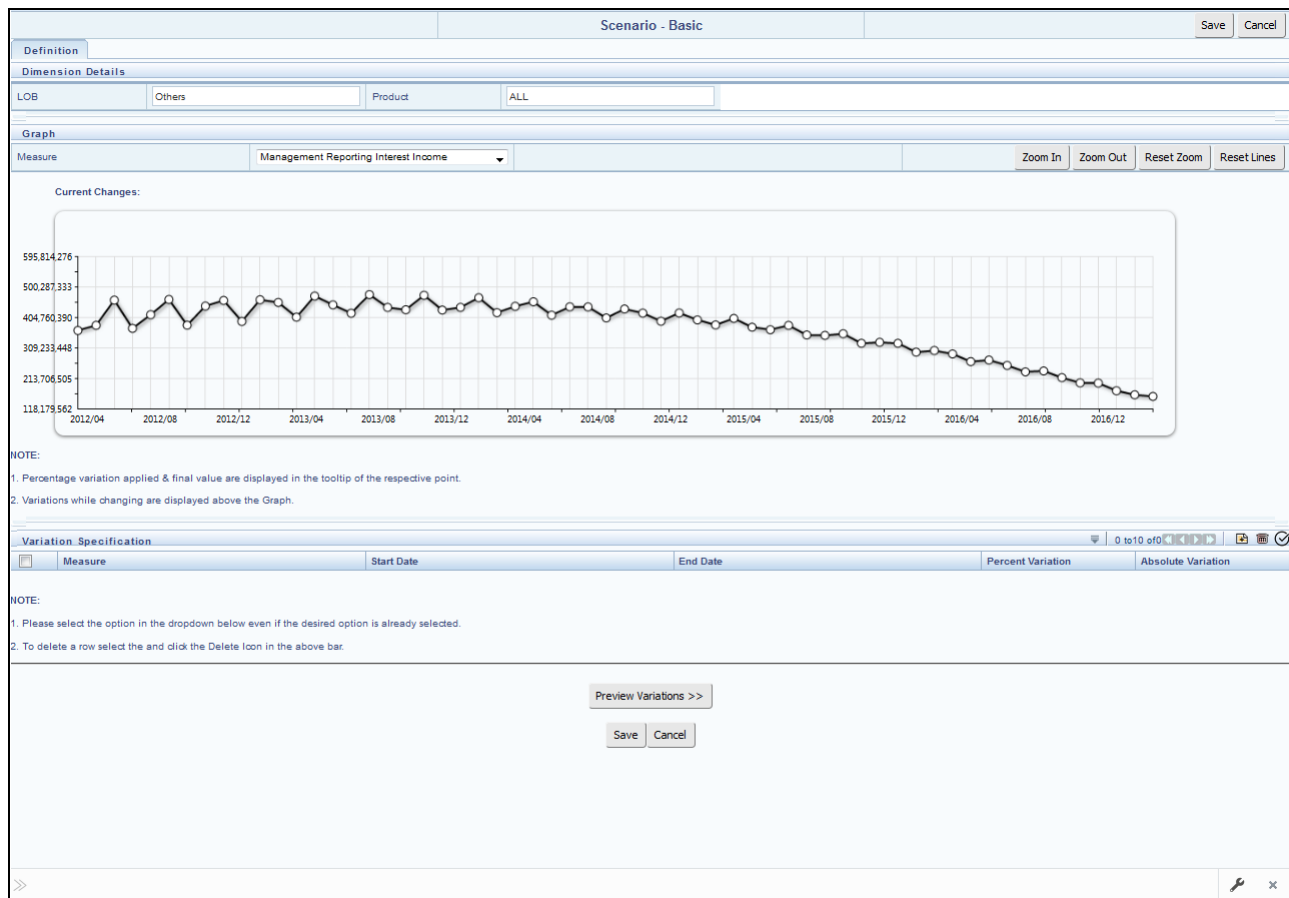
  <resource auth="Container"
    name="jdbc/DESPFTINFO"
    type="javax.sql.DataSource"
    driverClassName="oracle.jdbc.driver.OracleDriver"
    username="pftatn30"
    password="ofsaa8x"
    url="jdbc:oracle:thin:@10.104.153.87:1521:DEV12C"
    maxActive="1000"
    maxIdle="30"
    maxWait="10000" removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
    removeAbandonedOnBorrow="true" removeAbandonedOnMaintenance="true"/>
```



## Basic Scenario

The following procedure describes the steps to create a variation:

1. Navigate to the OBIEE Dashboard page of What If Analysis.
2. Click **Create Scenario** after selecting the relevant dimensions to display the **Scenario - Basic** screen.



3. Apply the necessary **Dimension Details** for the following:
  - Account
  - Customer
  - LOB
  - Product
4. Select the relevant repline **Measure** from the drop down list to which you want to apply the variation.
5. Select a point on the graph and drag to apply the desired variations. Percentage variation and applied, final values in the drop down are displayed on the graph and as a tooltip on the point that is being changed.
6. Select the relevant details for the following under **Variation Specification** section of the screen.
  - Measure

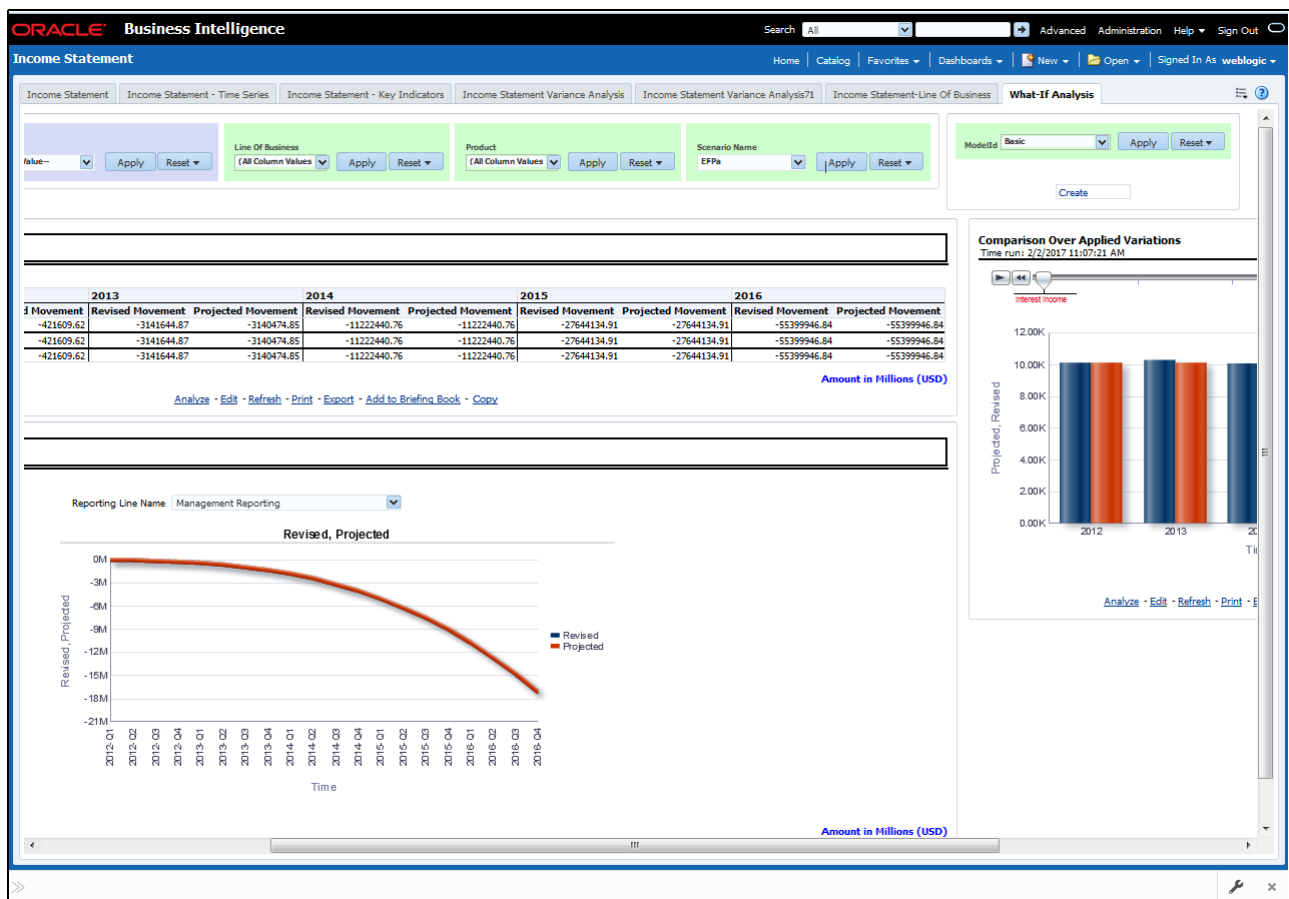
- Start Date
- End Date
- Percent Variation: Enter the % value. If you enter this, you cannot enter the Absolute Variation.
- Absolute Variation: Enter the absolute value. If you enter this, you cannot enter the Percent Variation.

If there is an overlap in dates among various variation specification rows, the same can be overridden to apply the respective change.

Always the most recent changed value is considered for variation and the other value will be cleared.

7. Click **Preview Variations** to see the results.
8. Click **Save** after confirming the variations. After this, you will be redirected to the OBIEE screen where the applied variations can be seen and analyzed further.

The OBIEE screen is displayed as shown below:



## Advanced Scenario

In the advanced scenario, when a change is applied to a repline, then all its corresponding correlated changes affecting other replines are also made. These correlated changes can be preview in the preview results view by clicking on the Preview Variation button at the bottom of the page.

The following procedure describes the steps to create a variation:

1. Navigate to the What If Definition dashboard to display the Scenario - Advanced screen.

This screen allows you to zoom in, zoom out and reset the zoom.

The screenshot displays the 'Scenario - Advanced' configuration interface. At the top, there are 'Save' and 'Cancel' buttons. Below is the 'Definition' section, followed by 'Dimension Details' with 'LOB' set to 'Others' and 'Product' set to 'ALL'. The 'Graph' section shows the 'Measure' as 'Management Reporting Interest Income'. The graph area includes 'Zoom In', 'Zoom Out', 'Reset Zoom', and 'Reset Lines' buttons. The graph itself shows a line chart with data points from 2012/04 to 2016/12. Below the graph, there are two 'NOTE' sections. The first note states: 'Percentage variation applied & final value are displayed in the tooltip of the respective point.' The second note states: 'Variations while changing are displayed above the Graph.' Below the notes is the 'Variation Specification' table, which has columns for 'Measure', 'Start Date', 'End Date', 'Percent Variation', and 'Absolute Variation'. At the bottom of the screen, there are buttons for 'Preview Variations >>', 'Save', and 'Cancel'.

2. Apply the necessary **Dimension Details** for the following:

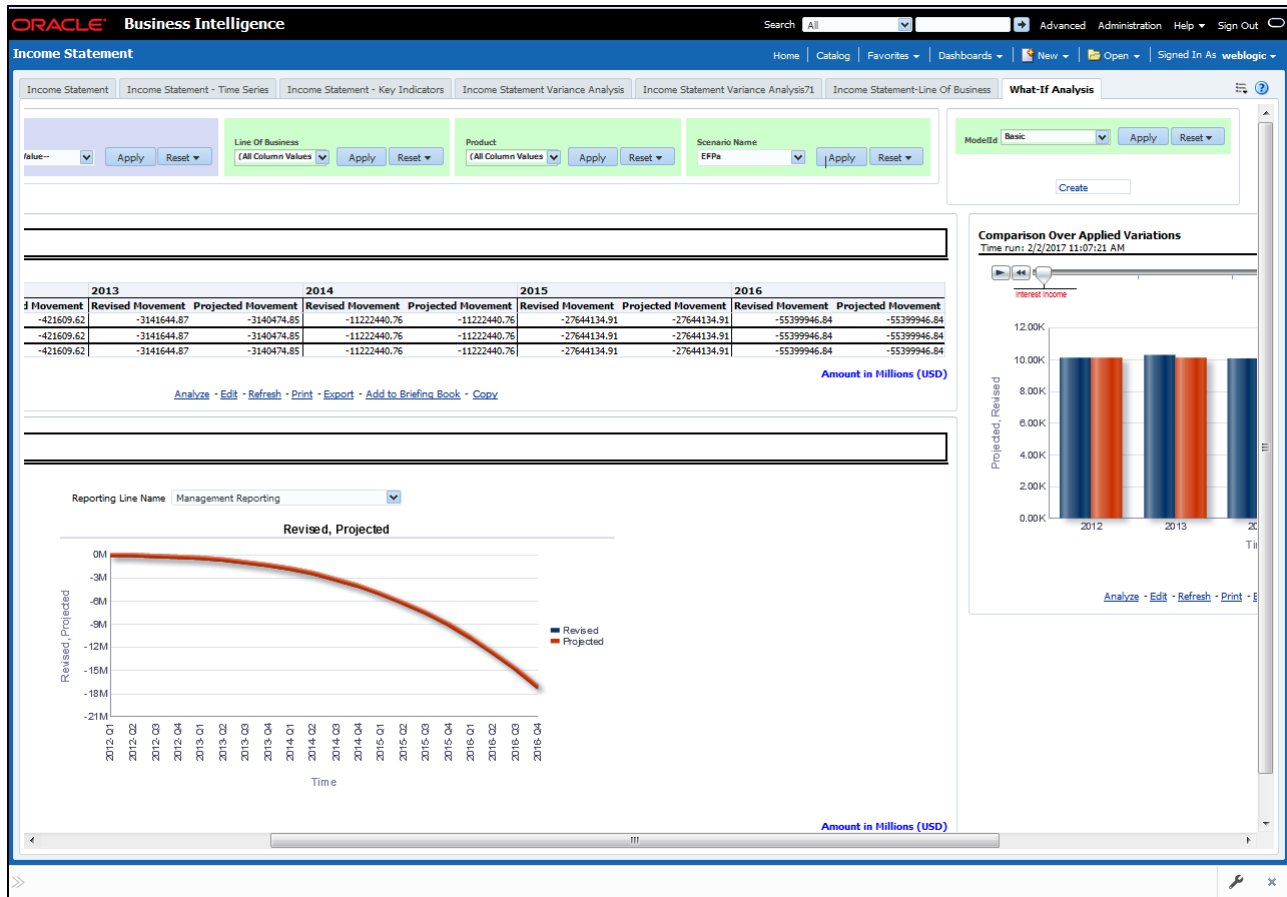
- Account
- Customer
- LOB
- Product

3. Select the relevant repline **Measure** to which you want to apply the variation.

4. Select a point on the graph and drag to apply the desired variations.

5. Click **Preview Variations** to see the results.
6. Click **Save** after confirming the variations. After this, you will be redirected to the OBIEE screen where the applied variations can be seen and analyzed further.

The OBIEE screen is displayed as shown below:



## EFPA Integration with Price Creation and Discovery (PCD)

The input from EFPA to PCD has been enhanced from the current setup with projected values for extended future periods based on the available current actual data for each customer account. Output on current and projected values have been determined for each product type through various measures such as EOP Balance, Fee Income, Other Income of Customers, Expenses, Credit utilization ratio, and credit card revolving rate.

## Reporting Line Correlation Calculations

For the reporting lines, regression co-efficients are calculated using the R-model based on the threshold values. It is considered that a pairwise relationship exists between independent and dependent reporting lines.

In what-if analysis, you can make variations to the value of a variable. Variations can be applied only to the below reporting lines in the income statement:

- Interest Income
- Interest Expenses
- Transfer Pricing Charge
- Transfer Pricing Credit
- Non Interest Income
- Operating Expenses
- Net Credit Losses
- Other Revenue

The following parameters are available in the FSI\_MODEL\_PARAMETERS table:

- Start date of the reference time period
- End date of the reference time period
- Percentage of values that lie within the threshold
- Percentage of outliers that need to be removed

The following steps are used in repline correlation calculation:

1. Excluding Outliers
2. Testing for Stationarity
3. Testing for Cointegration

### Excluding Outliers

For each variable, the sigma and mean are calculated within the reference time period as defined in the database. If the value of variable lies outside the threshold provided, the respective pairs are excluded for all associated variables.

Pairs are excluded based on the Mahalanobis distance, i.e., pairs are excluded in descending order of their absolute distance from the mean.

### Testing for Stationarity

After the outliers are excluded, the ADF test is used to check for stationarity on the time replines. The stationary is checked for each repline at two levels: I(0) and I(1). If any time repline is not found to be stationary, do a differencing of data and repeat the test.

**Note:** `add.test` is a function of the R-library. A limitation of the R-library is that the stationary value can be calculated only if the records or data points are more than or equal to 6.

Results are reported and used in the cointegration test.

### Testing for Cointegration

After the stationary test is done, the causal relations between regression variables are checked. Then cointegration is done.

The following table shows the action that is performed for pairwise stationarity and pairwise integration based on the stationarity level:

**Table 22. Testing for Cointegration**

<b>Pairwise Stationarity</b>	<b>Pairwise Cointegration</b>	<b>Action</b>
Both I(1)	Exists	Do regression without any transformation
	Does not exist	Do regression after differencing
Both(0)	NA	Do regression without any transformation
One I(1) other I(0)	NA	Do regression after differencing I(1) series

## What-If Analysis Limitation

There is limitation on graph. Default upper limit for the graph will be based on underlying data in fact table. Variations applied above the normal can be applied using the grid. If the user wants to apply variation beyond the upper limit shown in graph, it is not possible through graph and hence, it should be applied through grid.

The same applies to Negative Values. The graph does not allow applying negative variations. That is, the graph nodes cannot be dragged below the X Axis. This change needs to be done using the grid mode.

This chapter covers the following topics:

- [Introduction](#)
- [Overview of Cubes](#)
- [Creating Configuration Files](#)
- [Building Of Cubes](#)
- [Checking the Execution Status](#)
- [Creating Multiple Ledger Essbase Cubes from OFSAA UI](#)

## Introduction

OFSEFPA reports 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 for cube building process. Cubes are optional source of data in OFSEFPA 8.0.5.0.0

## Overview of Cubes

OFSEFPA application has the following seeded cubes:

- Cube for Financial Reporting - ADPAFNRE
  - **Purpose:** The purpose of this cube is to store data for reports belonging to Financial Reporting Dashboard.
  - **Dataset:** DSPAFR  
This cube is based on the FCT\_LEDGER\_STAT fact table.
- Cube for Management Reporting - ADPAMNRE
  - **Purpose:** The purpose of this cube is to store data for reports belonging to Management Reporting Dashboard.
  - **Dataset:** DSPAMR  
This cube is based on the FCT\_MGMT\_REPORTING fact table.

## Creating Configuration Files

The metadata tables of the config schema hold the details of the cube. Also, the Measure Hierarchy details are seeded in the REV\_BIHIER and the REV\_LOCALE\_HIER tables in the atomic schema.

Other Hierarchy data with their Measures are loaded into these REV tables, once following steps are performed:

- Individual re-save of each Hierarchy
- Individual re-save of each Dimension
- Metadata Authorization (If any Forms are to be authorized)
- Save Metadata

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. The configuration file is created during saving of the cube definition.

Follow the below steps:

1. In OFSAAI, navigate to **Home > Unified Metadata Manager > Business Metadata Management > Cubes**.
2. Click **Search**.

All the available cubes are displayed in the *Cube* popup.

3. Select the Cube name that needs to be built, and click **OK** to return to the **Cube Definition** window.
4. Click **Save** to save the cube.

A pop up appears with a message 'Operation Successful'.

**Note:** Cube definition is 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

OFSAAI Cube definition process is as follows:

- Generate 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.
- Create the Cube outline on Essbase server. This is performed by the CREATE CUBE component task within the Batch definition.
- Load the data to the cube. This is performed by the CREATE CUBE task within the Batch definition. Data load has to happen in sequential order for the months for which data is available in fact tables.



## Prerequisites

The following are prerequisites for creating a Cube:

- All the post install steps mentioned in the *Oracle Financial Services Analytical Applications Infrastructure Installation Guide* and Solution installation manual have to be 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 user must have the necessary function roles mapped to Resave Metadata from the **Home> Unified Metadata Manager > Business Metadata Management** screen and execute a Batch from *Batch Operations* screen.
- You can save Metadata as mentioned below:
  - Navigate to **Home > Administration > Save Metadata** section.
  - Select the available metadata under a Hierarchy and move it to the RHS pane using '>>>' button.
  - Click **Save**.
  - You can view the log by clicking **Show Details** button.

For more details, refer to *System Configuration* and *Administration* chapters in *Oracle Financial Services Analytical Applications Infrastructure User Guide*..

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
  - Messageserver
  - Olapdataserver
- Batches need to be created for executing. For more information, refer to section [How to Define a Batch](#).
- All the required Dimensions and Fact tables should be populated before executing the cube.
- 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**.
- Select the dataset in the pop up 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 **Browse** 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 *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

- Perform the same for Join/Filter Condition and Date filter.
- Frame an 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

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 section [How to Define a Batch](#).

To define a new task for a Batch definition:

### Aggregate Data Component:

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 **Aggregate Data** component from the drop down list.
5. Select the following from the **Dynamic Parameters** list:
  - **Datastore Type** - Select the appropriate datastore type from the list.
  - **Datastore Name** - Select the appropriate datastore name from the list.
  - **IP address** - Select the IP address from the list.
  - **Cube Parameter** - Choose the cube code to be built from the drop down list.
  - **Operation** - Choose **All** from the drop down list.
6. Click Save.

The Task definition is saved for the selected Batch.

### Create Cube Component:

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 **Create Cube** component from the drop down list.
5. Select the following from the **Dynamic Parameters** list:
  - **Datastore Type** - Select the appropriate datastore type from the list.
  - **Datastore Name** - Select the appropriate datastore name from the list.
  - **IP address** - Select the IP address from the list.
  - **Cube Parameter** - Choose the cube code to be built from the drop down list.
  - **Operation** - Choose **All** from the drop down list.
6. Click Save.

The Task definition is saved for the selected Batch.

7. Execute the Batch.

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

**Note:** A common issue in the Aggregate task is, Data Set not having records. You can check this as mentioned in the Prerequisites or by executing the SQL query in the Data Cruncher log file available in the path `$FIC_DB_HOME/log/dc`.

**Note:** 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 append a string to the Hierarchy member code to make it unique across the Cube or change the hierarchy data to make the node unique across the cube.

Points to be considered before executing the batches for the cubes `<INFODOM>_Financial_Reporting_Cube` and `<INFODOM>_Management_Reporting_Cube`:

- Populate dummy values against NULL (wherever applicable) for Level IDs and Level Names in the underlying DIM tables for the Hierarchies Product, Line of Business, and Org Unit.

For the successful execution of the cube, all the levels (Level IDs and Level Names used in the Hierarchy) having null in the underlying Dimension tables along with the level that holds the leaf should be loaded with some dummy values (with proper hierarchical structure followed) for these three Hierarchies. A new Level, LEVEL00 has been introduced which holds the leaf nodes for these hierarchies. The Hierarchy should always be balanced. Members across dimensions should be distinct in Code and Names.

For instance, if the leaf is present at LEVEL18 for a product, then LEVEL06 to LEVEL17 will have null values in Level IDs and Level Names after SCD. These null values should be updated to some dummy values, with a proper hierarchical structure. The Level ID and Level Name of LEVEL18 should also be updated along with this. The leaf node will be depicted by LEVEL00. A particular level value should not be repeated across Level IDs, Level Names, `v_prod_code`, `v_product_name`, and `n_product_id` in a row.

DIM\_PRODUCT has to be updated in this manner for only those product keys that are part of the Fact table for which the cube is executed. This method should be applied to DIM\_ORG\_UNIT and DIM\_LOB as well.

DIM\_PRODUCT and DIM\_ORG\_UNIT have a similar Hierarchical Level structure with root being LEVEL20. But for DIM\_LOB, the root is LEVEL1. This difference should be kept in mind while populating the dummy values.

- Levels that are provided for DIM\_PRODUCT, DIM\_ORG\_UNIT and DIM\_LOB are as follows:

- DIM\_PRODUCT: Levels will be defined from Level\_20 to Level\_6 and Level\_0 will be the leaf level. In CUBE, Dimension and Hierarchy levels get added by default. All together constituting 16 levels.
- DIM\_ORG\_UNIT: Levels will be defined from Level\_20 to Level\_6 and Level\_0 will be the leaf level. In CUBE, Dimension and Hierarchy levels get added by default. All together constituting 16 levels.
- DIM\_LOB: Levels will be defined from Level\_1 to Level\_16 and Level\_0 will be the leaf level. In CUBE, Dimension and Hierarchy levels get added by default. All together constituting 17 levels.
- Month data of cubes should be populated in proper order, else the AGO function in OBIEE will not give correct results.
  - Preference of reports is RDBMS and not cubes.

For more information on Cubes, refer to *Cubes* section under *Unified Metadata Manager* chapter in [Oracle Financial Services Analytical Applications Infrastructure User Guide](#).

Seeded batches are provided along with the Enterprise Financial Performance Analytics application installer. The below are the OFSEFPA seeded batches:

- Cube for Financial Reporting

Seeded batch <INFODOM>\_Financial\_Reporting\_Cube is provided with the installer. Execute the batch for the required MIS Date.

- Cube for Management Reporting

Seeded batch <INFODOM>\_Management\_Reporting\_Cube is provided with the installer. Execute the batch for the required MIS Date.

**Note:** "What-if Analysis" report of Management Reporting Income Statement will work out of RDBMS Data source.

a. Bug 16504664 - CUBES: NOT ABLE TO SAVE HEPMGLFR (REPORTING LINE FOR FR) HIERARCHY. Workaround – Execute the scripts available in the integrated SQL file below post SCD run.

[DIM\\_GL\\_ACCOUNT\\_update\\_EFPA.sql](#)

Seeded batches are provided along with the Enterprise Financial Performance Analytics application installer. The below are the OFSEFPA seeded batches:

- Cube for Financial Reporting

Seeded batch <INFODOM> \_Financial\_Reporting\_Cube is provided with the installer. Execute the batch for the required MIS Date.

- Cube for Management Reporting

Seeded batch <INFODOM> \_Management\_Reporting\_Cube is provided with the installer. Execute the batch for the required MIS Date.

## 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/dc` for the Task 1 above (Aggregate Data). The file name will have the Batch Execution ID.

## Creating Multiple Ledger Essbase Cubes from OFSAA UI

**Datamodel:** Identify the star schema to be used for reports

**Metadata:** For metadata, create the following:

- A dataset for the star schema
- Hierarchie(s) on the dimension tables. You are creating Level Based Hierarchy out of DIM tables.
- Dimensions that group the hierarchies from the same dimension table
- Measures on the fact tables
- Business processors (BPs) if any measures need to be derived. Business Processor is an expression built on top of Measures or other Business Processors. For more details, see *Business Processor* section in [Oracle Financial Services Analytical Applications Infrastructure User Guide](#).
- A derived entity using the dataset, hierarchies, measures and BPs (central table for RDBMS as source in reports)
- A dataset on the derived entity
- Measures on the derived entity
- Measure hierarchy that groups the measures required in cube
- A cube using this dataset, dimensions, measure hierarchy (central cube for MDDDB as source in reports)

**Runchart:**

- Transform and load the star schema (ensure all dimensions are level based)
- refresh the hierarchies (ensure Essbase restricted keywords are not there in the members)
- refresh the derived entities
- refresh (build and load) the cube



This chapter discusses the following topics:

- [Introduction to Predictive Modelling](#)
- [Guidelines](#)
- [Files Used](#)
- [Errors](#)

## Introduction to Predictive Modelling

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

The guidelines associated with respect to the execution R code are mentioned below:

- 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 will be 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,d and q. 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>
- Documentation of implementation of ARIMA in R can be found at :  
<http://stat.ethz.ch/R-manual/R-patched/library/stats/html/arima.html>

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

## Files Used

There are two files that are required to R script execution. Both the files are present at \$FIC\_DB\_HOME/bin folder. Provide execute permissions to the following files:

- RExec executable
- EFPA\_ARIMA\_MR\_IS\_PV.r

## Errors

Following are the errors:

1. Logging for the individual groups' arima model summaries will happen in a file named "out.log". This log file will be generated in \$FIC\_DB\_HOME/bin folder. If a particular group has unstable data and prediction fails, corresponding error will also be documented in the out.log file against that particular group.

2. **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.



This chapter discusses the following topics:

- [Introduction to Dashboards](#)
- [Tabular Consolidated List of OFSEFPA Reports](#)
- [Dashboards](#)

## Introduction to Dashboards

OFSEFPA offers the following dashboards that organize different kinds of reports by subject area.

- Financial Reporting
  - Financial Reporting
- Management Reporting
  - Balance Sheet
  - Income Statement
  - Key Trends
  - Performance Measures
- Forecast Reports
  - Balance Sheet Reports
  - Income statement Reports

## Tabular Consolidated List of OFSEFPA Reports

**Table 23. Consolidated List of OFSEFPA Report**

Data Source	Business Intent	Dashboard Name	Page Name	Report Name
Initial loads to the management ledger (that is, exclusive of any data generated by allocation rules and/or transfer pricing rules). Reporting line is General Ledger Account.	Represent Income Statement reporting lines across Legal Entity and Time dimension	Financial Reporting	Income Statement	Income Statemen
		Financial Reporting	Income Statement-Time Series	Income Statement-Time Series
		Financial Reporting	Income Statement Variance Analysis	Income Statement Variance Analysis

**Table 23. Consolidated List of OFSEFPA Report**

<b>Data Source</b>	<b>Business Intent</b>	<b>Dashboard Name</b>	<b>Page Name</b>	<b>Report Name</b>
Data from Fact Management table. Reporting line is Reporting Line Dimension	Represent balance sheet and show balances across dimensions	Balance Sheet	Balance Sheet	Balance Sheet
		Balance Sheet	Balance Sheet-Time Series	Balance Sheet-Time Series
		Balance Sheet	Balance Sheet-Key Indicators	Total Assets
		Balance Sheet	Balance Sheet-Key Indicators	Total Liabilities
		Balance Sheet	Balance Sheet-Key Indicators	Capital
		Balance Sheet	Balance Sheet-Key Indicators	Total Customer Assets
		Balance Sheet	Balance Sheet-Key Indicators	Total Customer Total Liabilities
Data from Fact Management table. Reporting line is Reporting Line Dimension	Represent Income Statement reporting lines across dimensions	Income Statement	Income Statement	Income Statement
		Income Statement	Income Statement-Time Series	Income Statement-Time Series
		Income Statement	Income Statement - Key Indicators	Net Interest Income
		Income Statement	Income Statement - Key Indicators	Commission
		Income Statement	Income Statement - Key Indicators	Fees
		Income Statement	Income Statement - Key Indicators	Net Credit Loss
		Income Statement	Income Statement Variance Analysis	Income Statement Variance Analysis
		Income Statement	Income Statement - Line Of Business	Income Statement across Line Of Business
		Income Statement	Income Statement - Line Of Business	Key Statistics from Balance Sheet
		Income Statement	Income Statement - Line Of Business	No. of Customers
		Income Statement	Income Statement - Line Of Business	Other Key Indicators

**Table 23. Consolidated List of OFSEFPA Report**

Data Source	Business Intent	Dashboard Name	Page Name	Report Name		
Data from Fact Management table. Reporting line is Reporting Line Dimension	Show trends for balance and movement measures for key line items relevant to the business	Key Trends	Key Trends	Break-up of Total Assets		
		Key Trends	Key Trends	Break-up of Total Assets		
		Key Trends	Key Trends	Break-up of Total Assets by Line of Business		
		Key Trends	Key Trends	Break-up of Total Liabilities by Line of Business		
		Key Trends	Key Trends	NIM Speedometer		
		Key Trends	Key Trends	Balance Sheet		
		Key Trends	Key Trends	Income Statement		
		Key Trends	Net Loss Analysis	Net Loss Rate Time Series		
		Key Trends	Net Loss Analysis	Net Loss Across Line of Business		
		Key Trends	Net Loss Analysis	Net Loss Across Products		
		Key Trends	Net Loss Analysis	Net Loss Rate Across Scenarios		
		Key Trends	NII Impact Analysis	NII at a Glance		
		Key Trends	NII Impact Analysis	Impact of FX Fluctuation		
		Key Trends	NII Impact Analysis	Impact of Volume Movement		
		Key Trends	NII Impact Analysis	Impact of Yield and Margin Rates		
		Data from Fact Management table. Reporting line is Reporting Line Dimension	Show key performance measures relevant to the business	Performance Measures	Performance Measures	Return on Total Assets
				Performance Measures	Performance Measures	Return on Equity
Performance Measures	Performance Measures			RAROC		
Performance Measures	Performance Measures			Profitability by Product over Reporting Period		
Performance Measures	Capital Analysis			Capital Analysis		

**Table 23. Consolidated List of OFSEFPA Report**

Data Source	Business Intent	Dashboard Name	Page Name	Report Name
Data is directly sourced from the BSP cube	<p>Tabular and graphical representation of forecast balances for the selected intersection points (Chart of Account, Entity, Currency, Version, Scenario, Strategy, Period, Years and Financial Element). FEs represented are:</p> <ul style="list-style-type: none"> <li>● Beginning balance</li> <li>● Current Position Beginning Balance</li> <li>● Current Position End Balance</li> <li>● Current Position Interest</li> <li>● Current Position Payment Runoff</li> <li>● Current Position Prepayment Runoff</li> <li>● Ending Balance</li> <li>● Interest</li> <li>● New Business Beginning Balance</li> <li>● New Business Ending Balance</li> <li>● New Business Interest</li> <li>● New Business Payment Runoff</li> <li>● New Business Prepayment Runoff</li> <li>● Non Interest Expense</li> <li>● Non Interest Income</li> <li>● Payment Runoff</li> <li>● Prepayment Runoff</li> </ul>	Forecast Reports	Balance Sheet Report	Forecast Reports

**Table 23. Consolidated List of OFSEFPA Report**

Data Source	Business Intent	Dashboard Name	Page Name	Report Name
	Tabular and Graphical representation of comparison between two scenarios for the selected Time period	Forecast Reports	Balance Sheet Reports	Scenario Comparison
	Tabular and Graphical representation of comparison between Base Strategy and an alternate Strategy for the selected Time period	Forecast Reports	Balance Sheet Reports	Strategy Comparison
	Tabular and Graphical representation of Interest vs. Average balances for the selected intersection points	Forecast Reports	Income Statement Reports	Interest Vs. Average Balance
	Tabular and Graphical representation of Non Interest Fee Percentage vs Ending Balance for the selected intersection points	Forecast Reports	Income Statement Reports	Non Interest Fee Percentage Vs. Ending Balance
	Tabular and Graphical representation of Spread Analysis (Yield % vs. Cost %) for the selected intersection points	Forecast Reports	Income Statement Reports	Spread Analysis

## Structure of Reporting Line Items

Reporting line items describe the financial meaning of the report.

The following spreadsheet contains the Management versus Financial Reporting Line Item Definitions.

[EFPA 8\\_0\\_2\\_FE\\_Repline\\_mapping\\_MR\\_IS.xlsx](#)

## Hard-Coded Dimension Members

The node level reporting lines (including higher level nodes) of the reporting line hierarchy that are used in the OBIEE components (Web Catalog and RPD) are mentioned below for reference.

These reporting line codes should not be modified.

**Table 24. Reporting Line Dimension**

<b>Catalog/RPD</b>	<b>Repline Code</b>	<b>Repline Name</b>
catalog	107001	Management Reporting
catalog	107002	Income Statement
catalog	107003	Balance Sheet
catalog	98000	Net Income Before Taxes
catalog	51000	Cash
catalog	54000	Trading Assets
catalog	61000	Investments
catalog	63000	Fixed Assets
catalog	64000	Other Assets
catalog	60000	Loans '&' Advances to Customers
catalog	72000	Total Liabilities
catalog	74900	Allocated Capital
catalog	77000	Shareholders Equity
catalog	72000	Total Liabilities
catalog	75000	Share Capital
catalog	76000	Reserves
catalog	60000	Loans '&' Advances to Customers
catalog	68000	Customer Liabilities
catalog	85000	Net Interest Revenue
catalog	85100	Commission
catalog	85200	Fees
catalog	91910	Credit Losses
catalog	91920	Recoveries of amounts previously written-off
RPD	95000	Operating Income before Taxes
RPD	99500	Provisions for Credit Losses
RPD	91910	Credit Losses
RPD	107100	Number of Customers
RPD	52000	Interest-bearing Assets
RPD	74900	Allocated Capital
RPD	88000	Non-Interest Revenue

**Table 24. Reporting Line Dimension**

Catalog/RPD	Repline Code	Repline Name
RPD	107120	Number of Closed Customers
RPD	83000	Credit for Float
RPD	82100	Interest Income
RPD	91920	Recoveries of amounts previously written-off
RPD	107300	Attrition Rate
RPD	60000	Loans and Advances to Customers
RPD	85000	Net Interest Revenue
RPD	83200	Interest Expenses

### Other Dimensions

- Run Dimension is hard-coded with a value zero.
- Legal Entity and Line of Business are hard-coded to -1 in the initial load. When the custom dimensions are added into `LEDGER_STAT` and `FACT` tables, the corresponding values are used in the reports.

### Reporting Currency Calculation

The base currency values are converted into reporting currency and are populated to RCY columns of Fact table using `FSI_EXCHNG_RATE_DIRECT_ACCESS` or `FSI_EXCHANGE_RATE_HIST` tables. In these tables, data should exist for the available currency combinations for the particular `FIC_MIS_DATE` column, for which user is running the fact transformations that is `FCT_LEDGER_STAT` and `FCT_MGMT_REPORTING` tables. In EFPA, by default exchange rate is calculated by referring `FSI_EXCHNG_RATE_DIRECT_ACCESS`. However, it can be configured in `SETUP_MASTER` table to use `FSI_EXCHANGE_RATE_HIST` if needed.

**Currency Type:** OFSEFPA BI reports work for Local, Base and Reporting currencies.

## OBIEE: Catalogs Related Configuration

This section explains the necessary configurations for OBIEE with respect to catalogs.

### Currency Label Modification

The currency label in the catalogs has to be configured in such a way that the amounts are displayed in a single currency unit. For Example: Amount in \$ millions.

Follow the below steps to configure the currency label:

1. Open RPD in online mode.
2. In the **Manage** menu, click **Variables** option.
3. Change the value of `RV_Reporting_Currency` variable as required. For example, **Amount in Millions (USD)**.

4. Set RV\_Base\_Currency repository variable to the appropriate Base currency.
5. Update the connection details of **OFSEFPA Connection Pool** and **OFSEFPA Init Block** in Physical Layer of RPD, to view the Local Currency label.
6. Save and check the global consistency of the RPD.

### **Amount Unit Modification**

Follow the below steps to configure the amount unit:

1. Open the RPD in online mode.
2. In the **Manage** menu, click **Variables** option.
3. Change the value of **ABBREVIATE\_VALUE** variable as required. For example: 1000000
4. Save and check the global consistency of the RPD.

## **Dashboards**

The following sections provide the dashboard screens that demonstrate the OFSEPA reports.

### **Financial Reporting**

The business intent of the Financial Reporting dashboard is to analyze the General Ledger Account based Income Statement and Balance Sheet reporting. Currency Type prompt allows data to be viewed for Reporting Currency and Base Currency for Financial Reporting reports.

- Financial Reporting - Income Statement





Figure 4. Financial Reporting - Income Statement

- Financial Reporting - Income Statement - Time Series

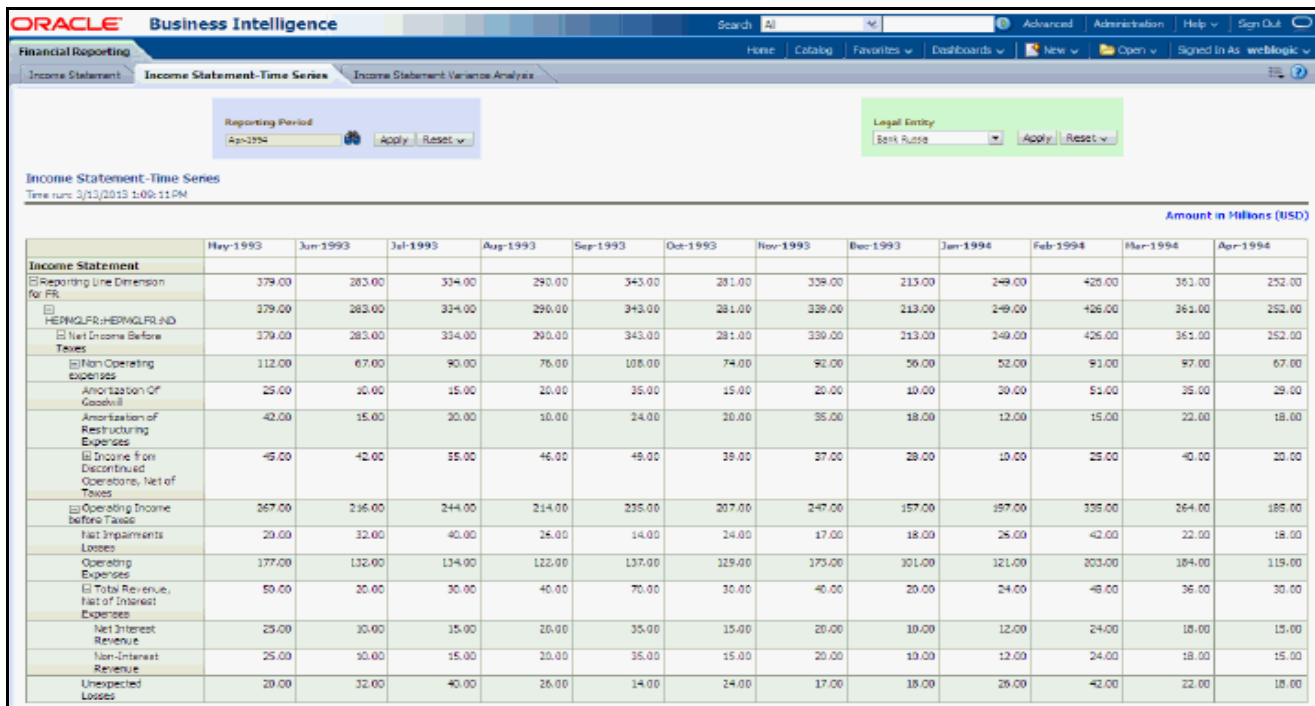


Figure 5. Financial Reporting - Income Statement - Time Series

- Financial Reporting - Income Statement - Variance Analysis

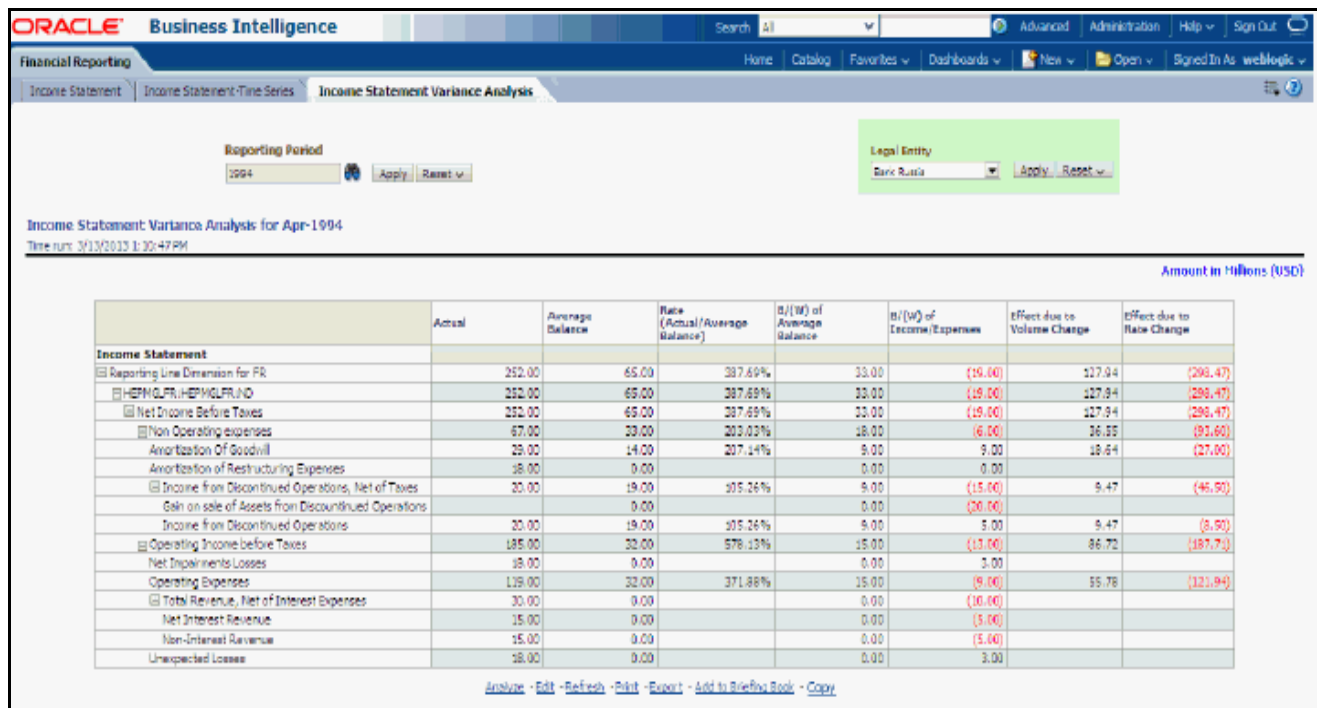


Figure 6. Financial Reporting - Income Statement - Variance Analysis

## Management Reporting

The business intent of the Management Reporting dashboard is to analyze the Financial Element based on Income Statement and Balance Sheet reporting (management). Currency Type prompt allows data to be viewed for Reporting Currency, Local Currency and Base Currency for Management Reporting reports.

- Management Reporting - Balance Sheet - Balance Sheet

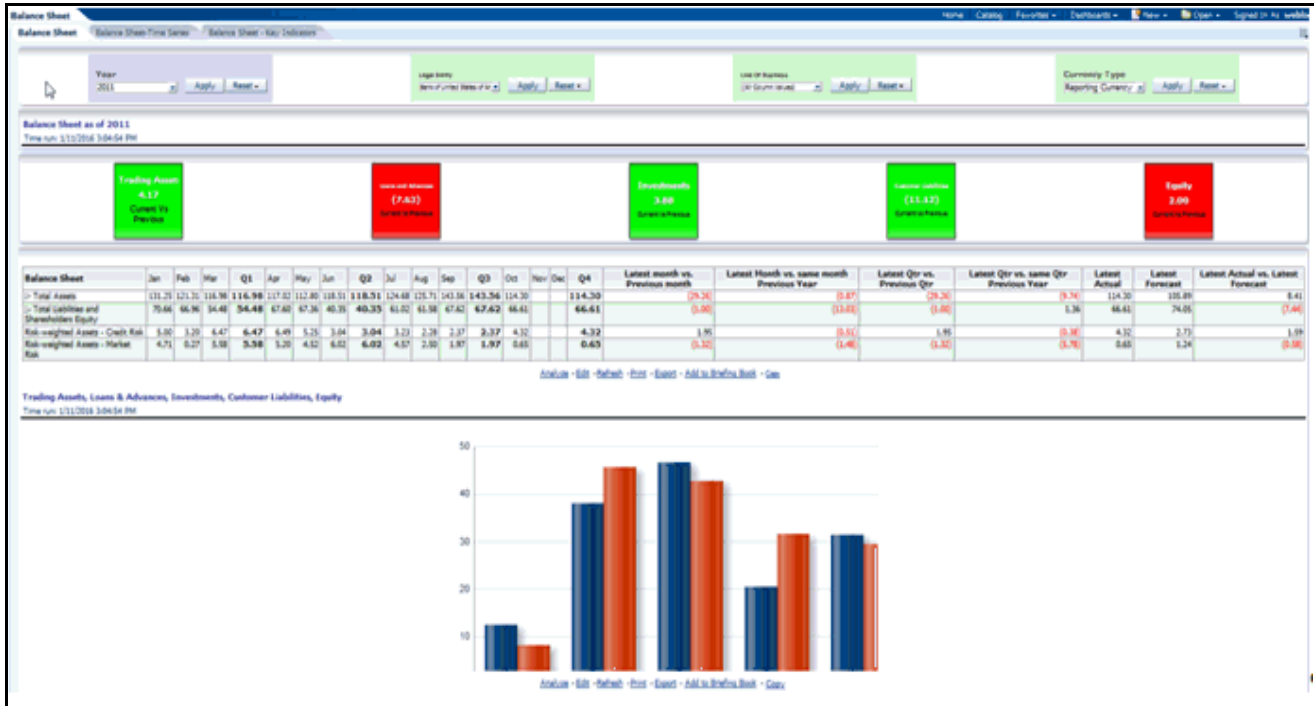


Figure 7. Management Reporting - Balance Sheet - Balance Sheet

- Management Reporting - Balance Sheet - Time Series



Figure 8. Management Reporting - Balance Sheet - Time Series

- Management Reporting - Balance Sheet - Key Indicators



Figure 9. Management Reporting - Balance Sheet - Key Indicators

- Management Reporting - Income Statement - Income Statement

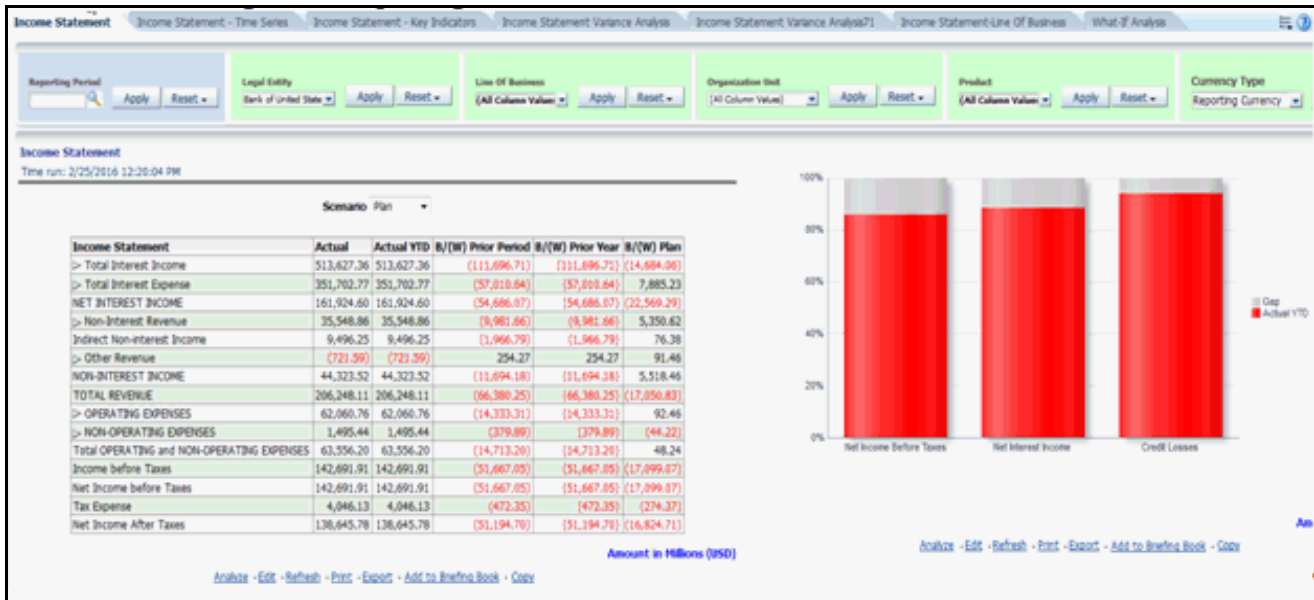


Figure 10. Management Reporting - Income Statement - Income Statement

- Management Reporting - Income Statement - Time Series

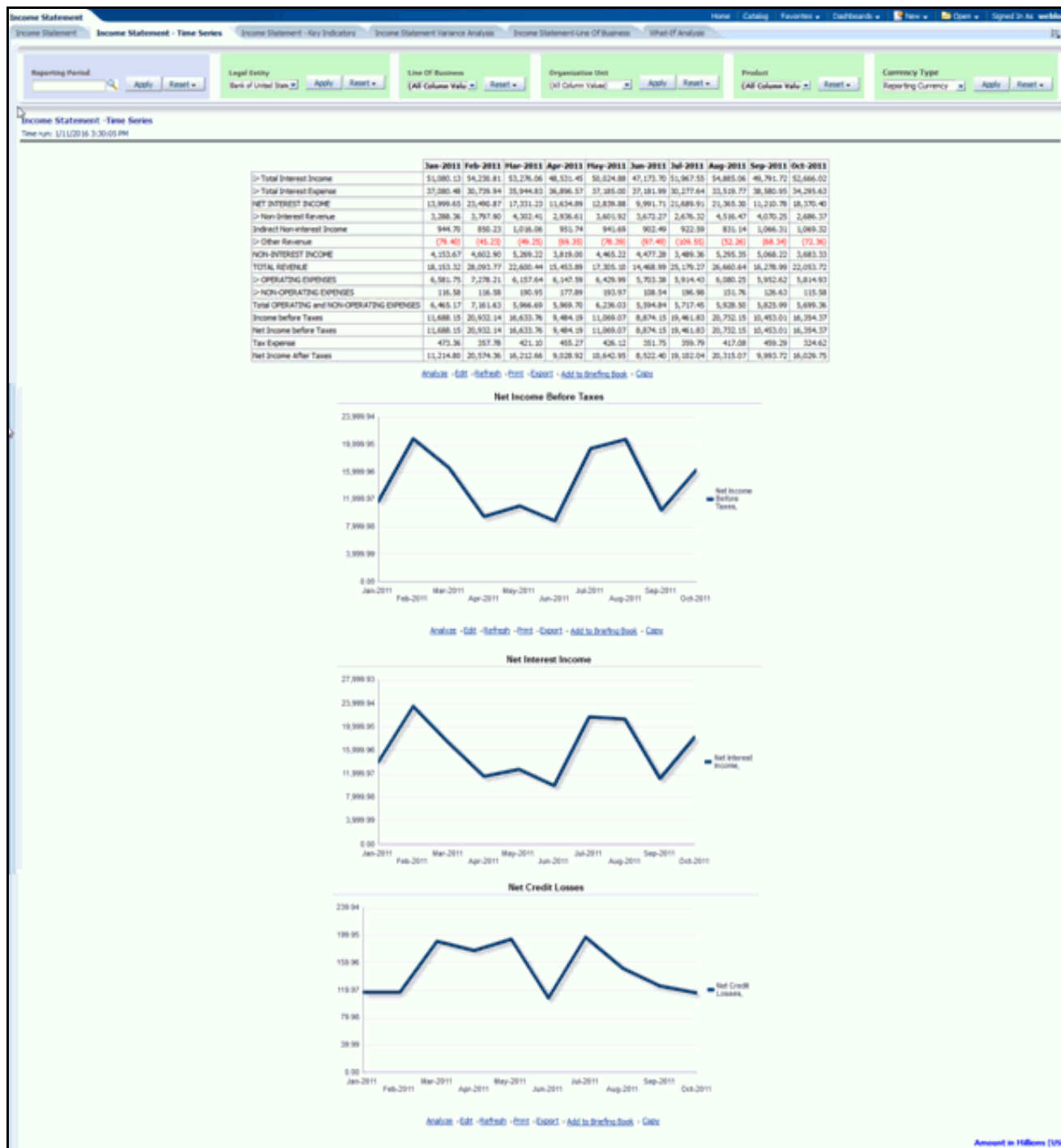


Figure 11. Management Reporting - Income Statement - Time Series

- Management Reporting - Income Statement - Key Indicators

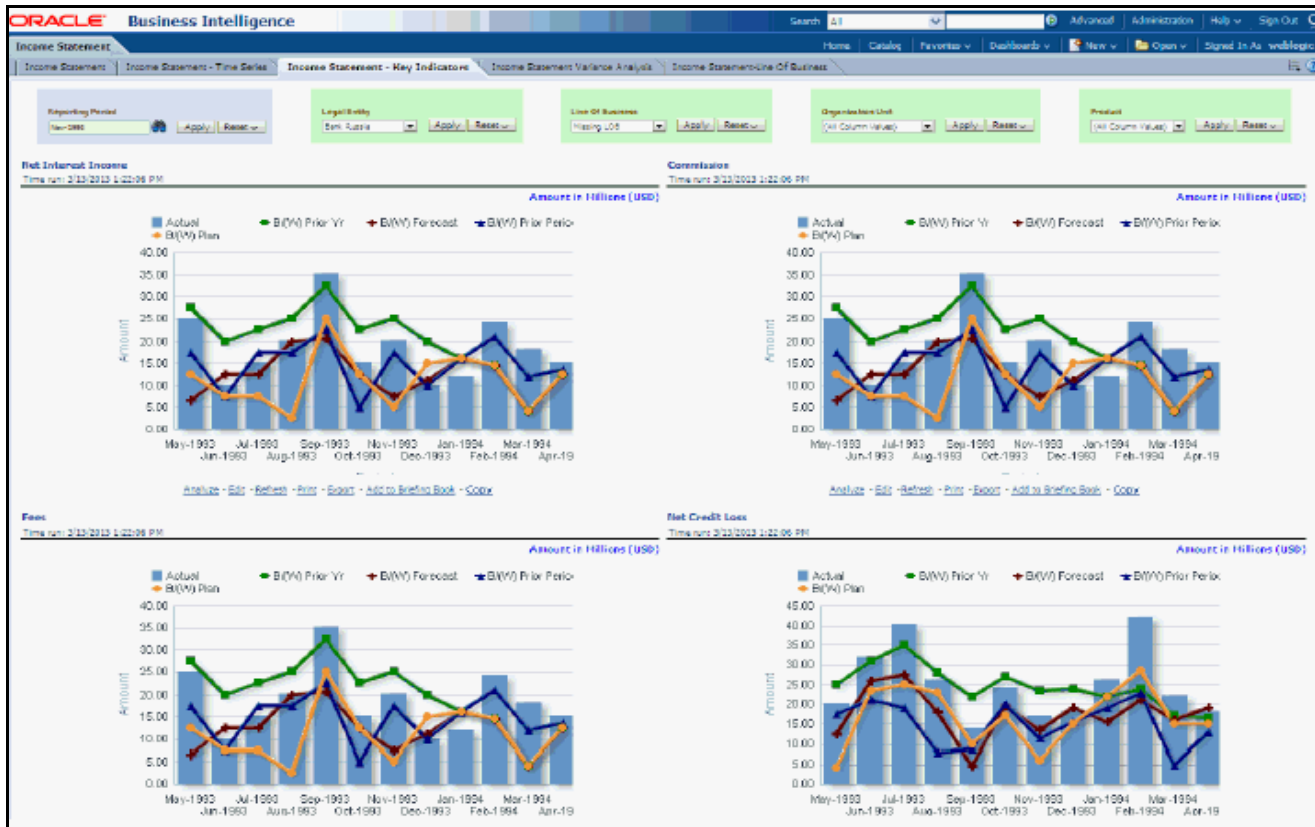


Figure 12. Management Reporting - Income Statement - Key Indicators

- Management Reporting - Income Statement - Variance Analysis





Figure 13. Management Reporting - Income Statement - Variance Analysis



Figure 14. Management Reporting - Income Statement - Variance Analysis 1

**Note:** Due to limitations in OBIEE, you may encounter errors on OBIEE version 7.1 while trying to open the enhanced Income Statement - Variance Analysis tab. Hence on OBIEE version 7.1, it is advised to use the report from the Income Statement Variance Analysis71 tab. On OBIEE version 7.1 and above, you can use the report from the Income Statement Variance Analysis tab.



You can use the **Hide Page** option to hide the tab that is not relevant. A Dashboard page can be hidden by clicking **Edit Dashboard > Dashboard Properties** and selecting the **Hide Page** option.

- Management Reporting - Income Statement - Income Statement - Line Of Business



Figure 15. Management Reporting - Income Statement - Income Statement - Line Of Business

- Management Reporting - Income Statement - What-If Analysis
- Management Reporting - Key Trends - Key Trends

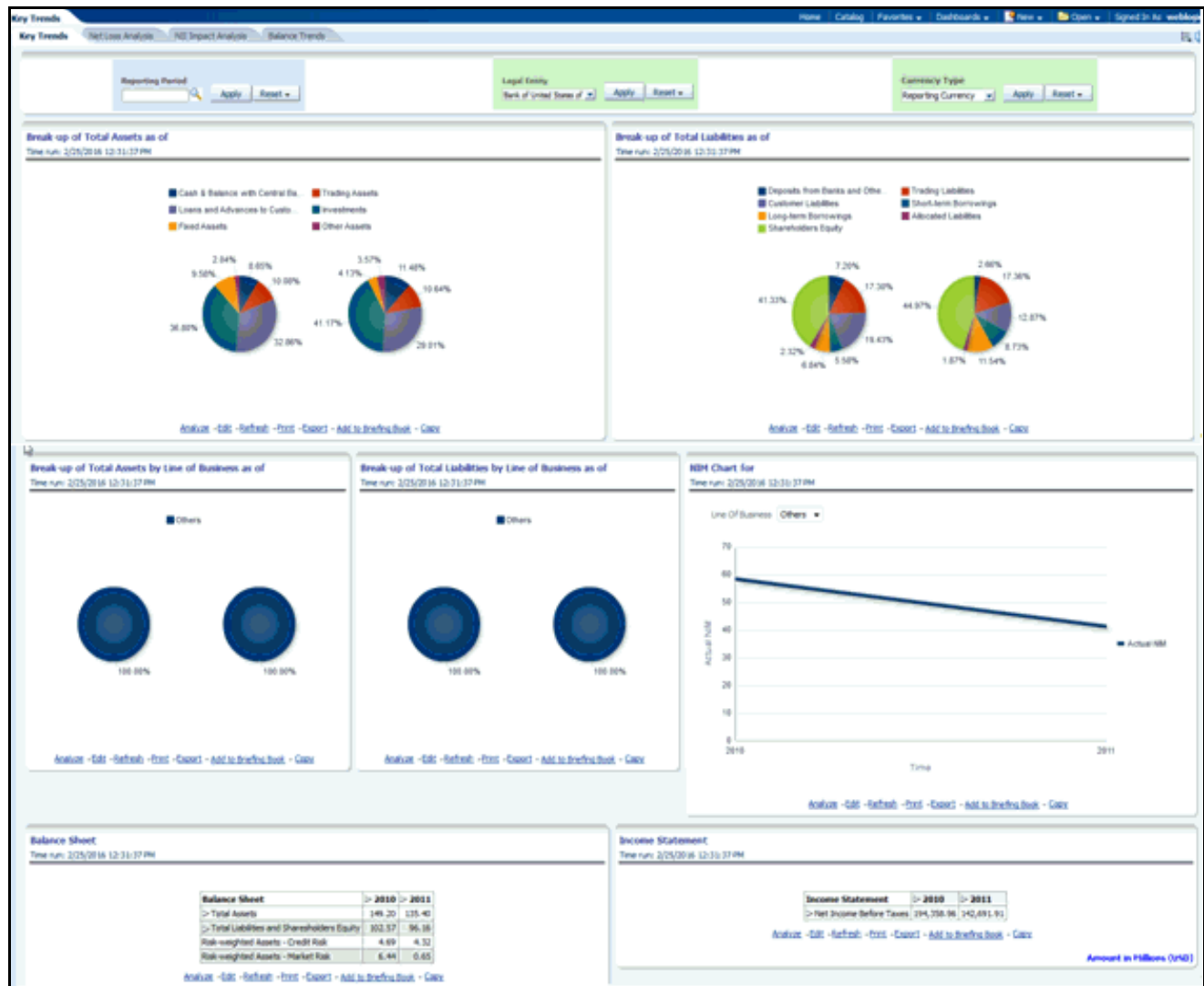


Figure 16. Management Reporting - Key Trends - Key Trends

- Management Reporting - Key Trends - Net Loss Analysis

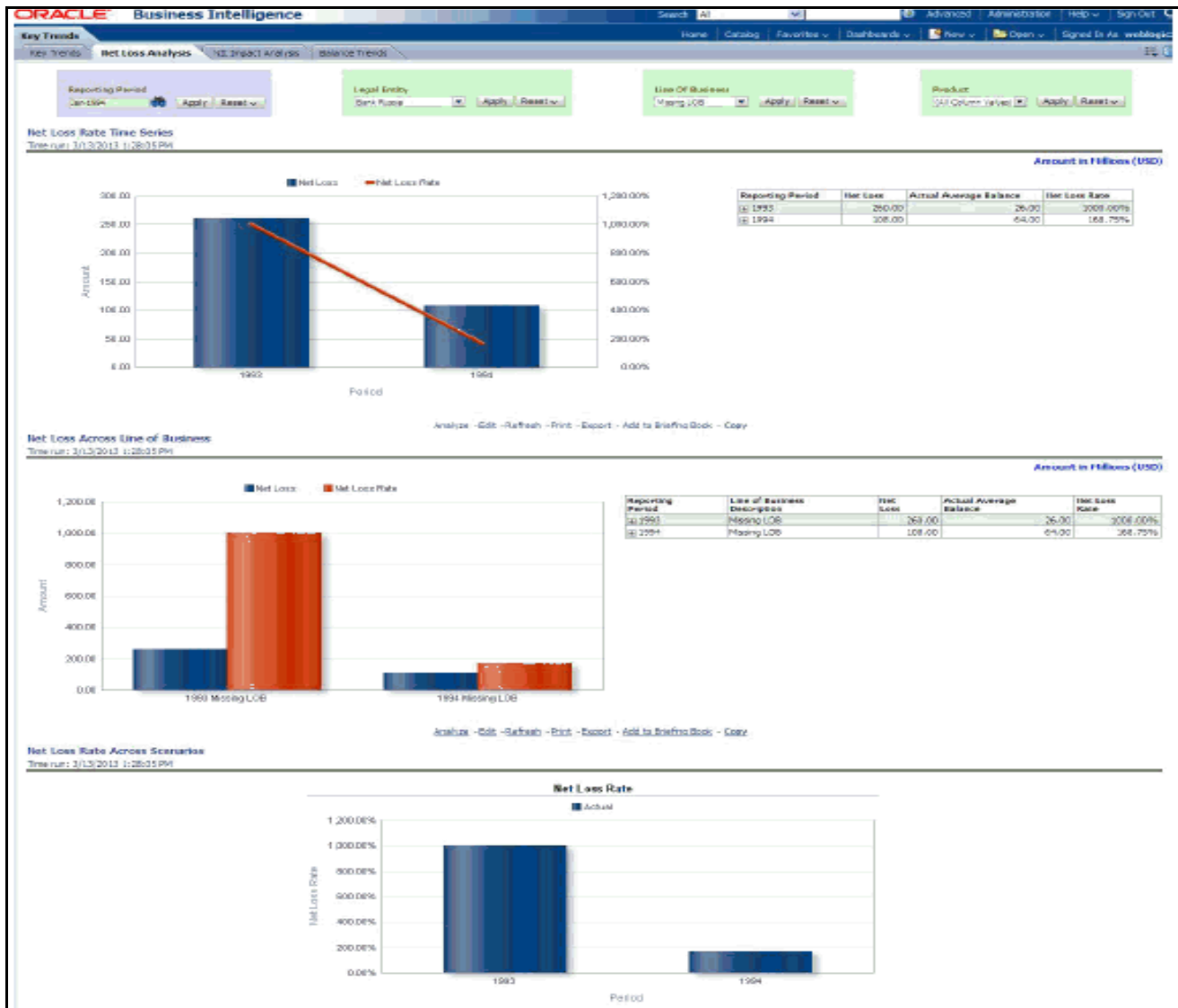


Figure 17. Management Reporting - Key Trends - Net Loss Analysis

- Management Reporting - Key Trends - NII Impact Analysis

Dashboards  
Chapter 13—Overview of OFSEFPA Reports



Figure 18. Management Reporting - Key Trends - NII Impact Analysis

- Management Reporting - Key Trends - Balance Trends

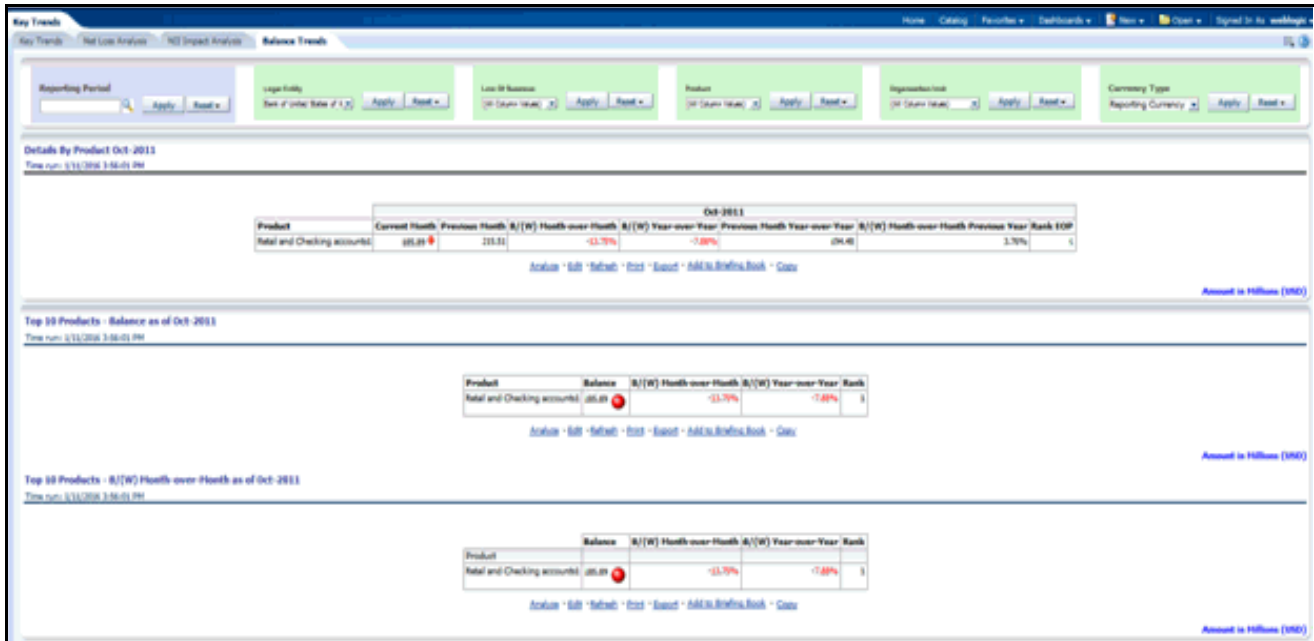


Figure 19. Management Reporting - Key Trends - Balance Trends

- Management Reporting - Performance Measures - Performance Measures

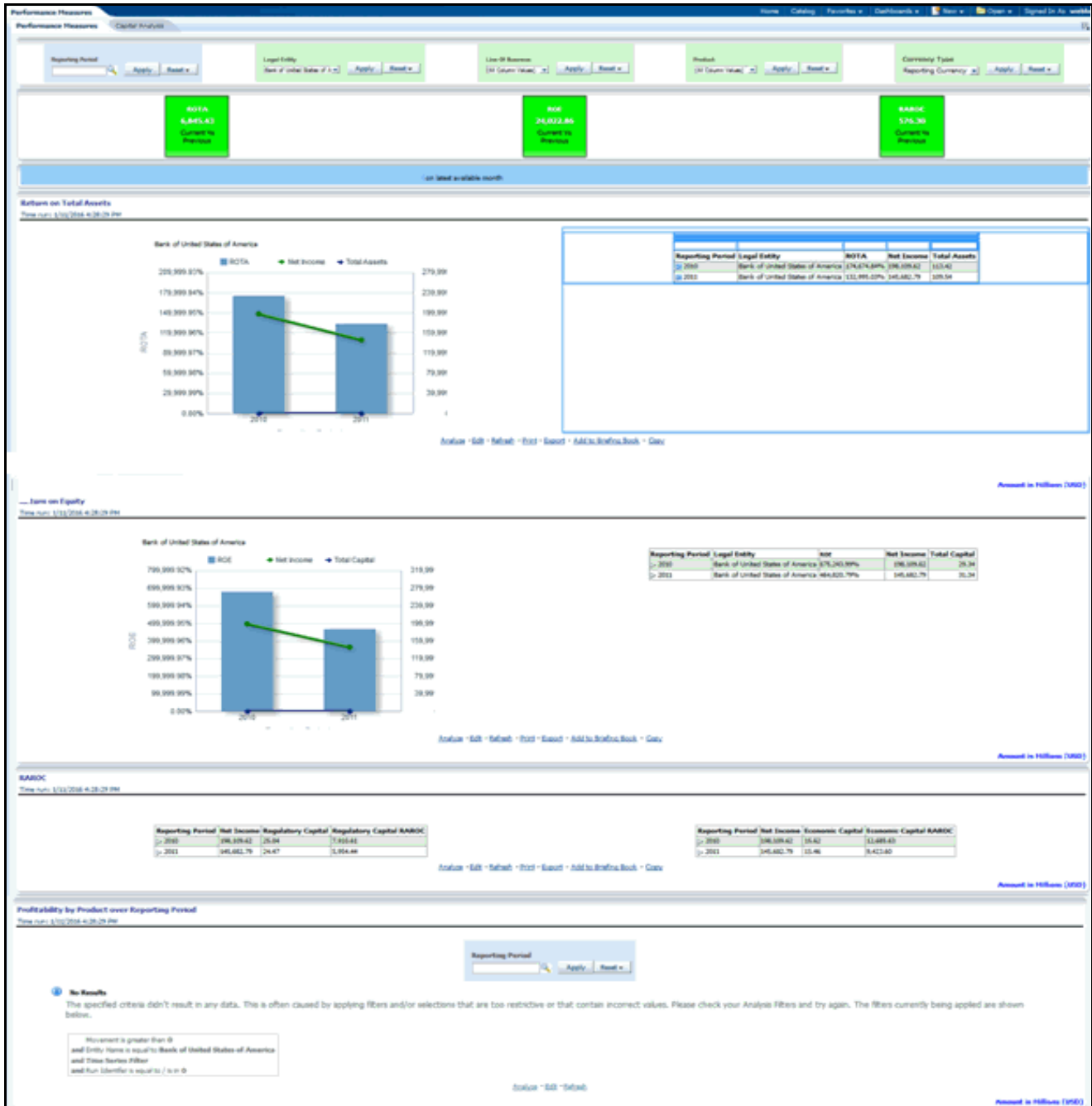


Figure 20. Management Reporting - Performance Measures - Performance Measures

- Management Reporting - Performance Measures – Capital Analysis

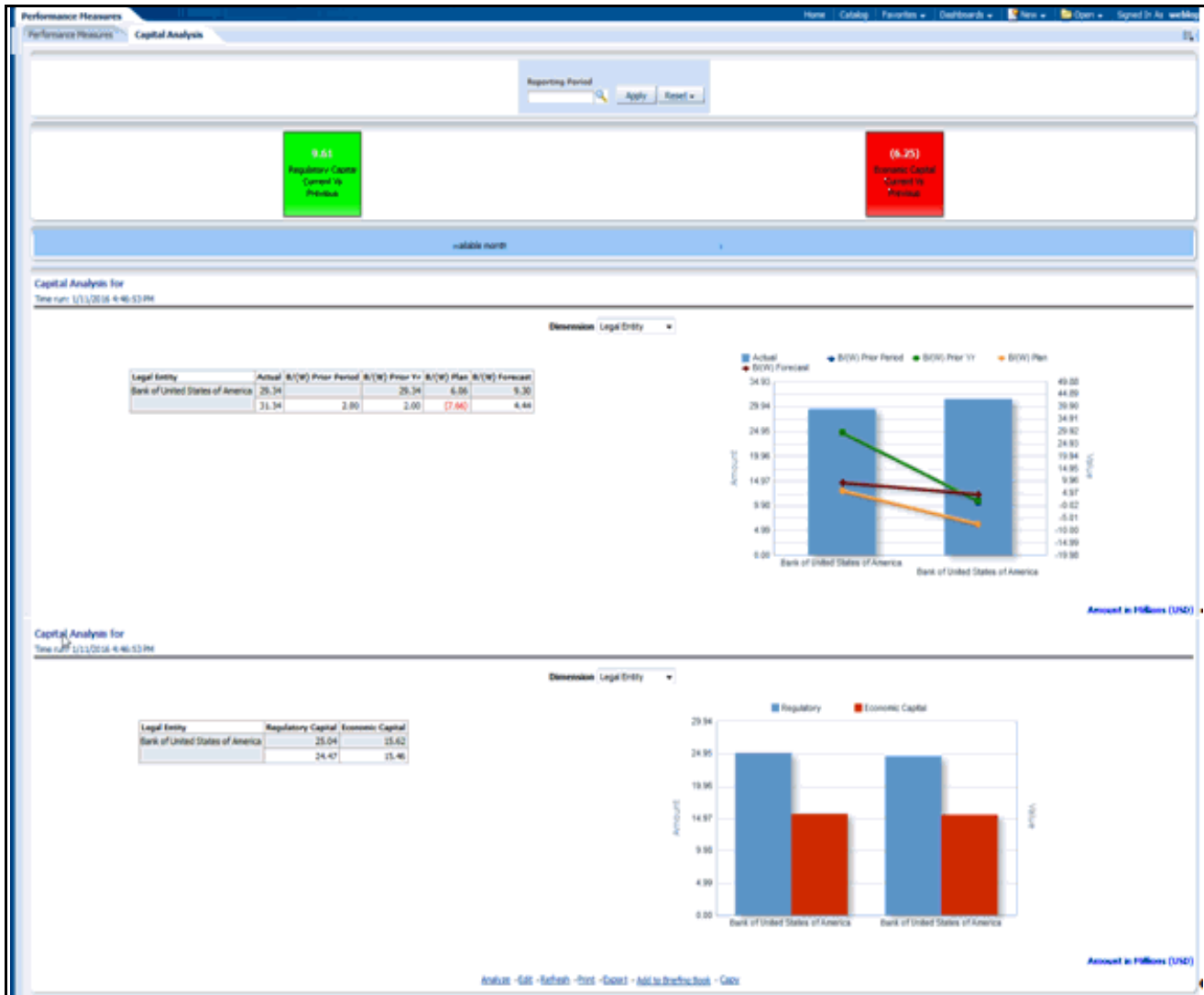


Figure 21. Management Reporting - Performance Measures – Capital Analysis

## Forecast Reporting

The business intent of the Forecast Reporting dashboard is to analyze the Financial Element based on Income Statement and Balance Sheet reporting (forecast). Currency Type prompt allows data to be viewed for Reporting Currency, Local Currency and Base Currency for Forecast Reporting reports.

- Forecast Reports - Balance Sheet Reports – Forecast Balances

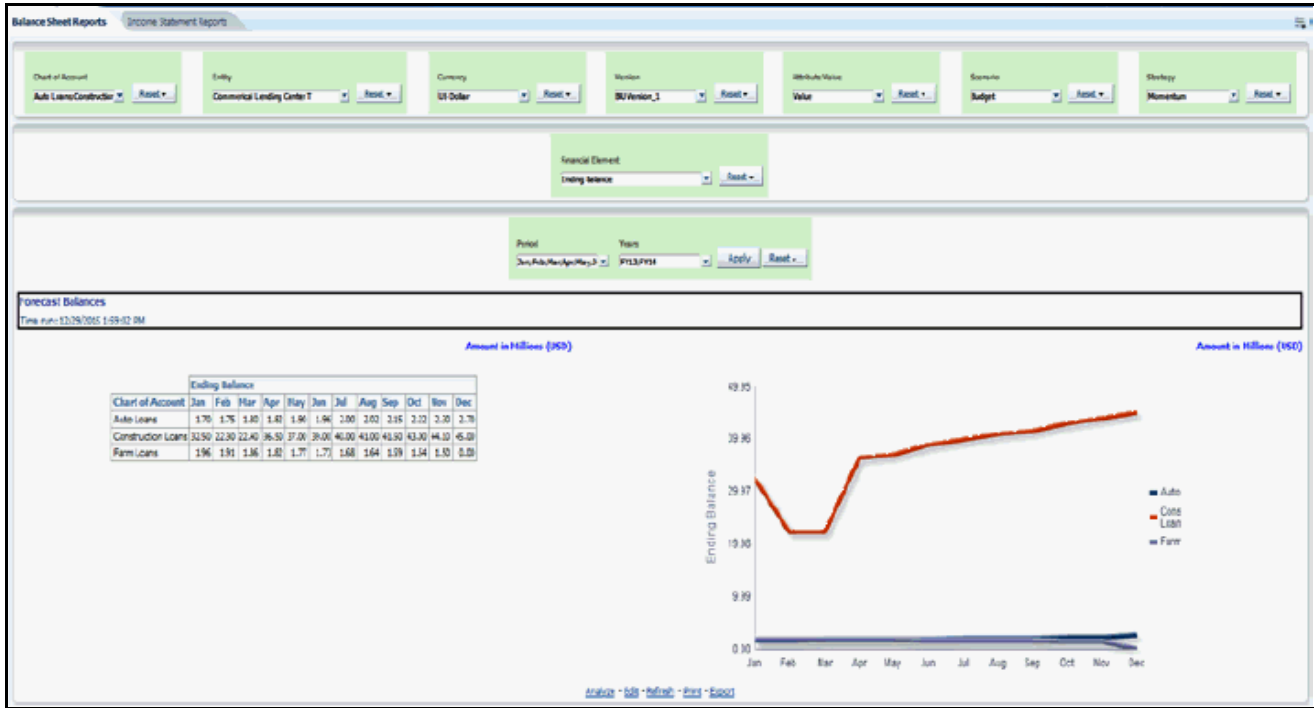


Figure 22. Forecast Reports - Balance Sheet Reports – Forecast Balances

- Forecast Reports - Balance Sheet Reports – Scenario Comparison

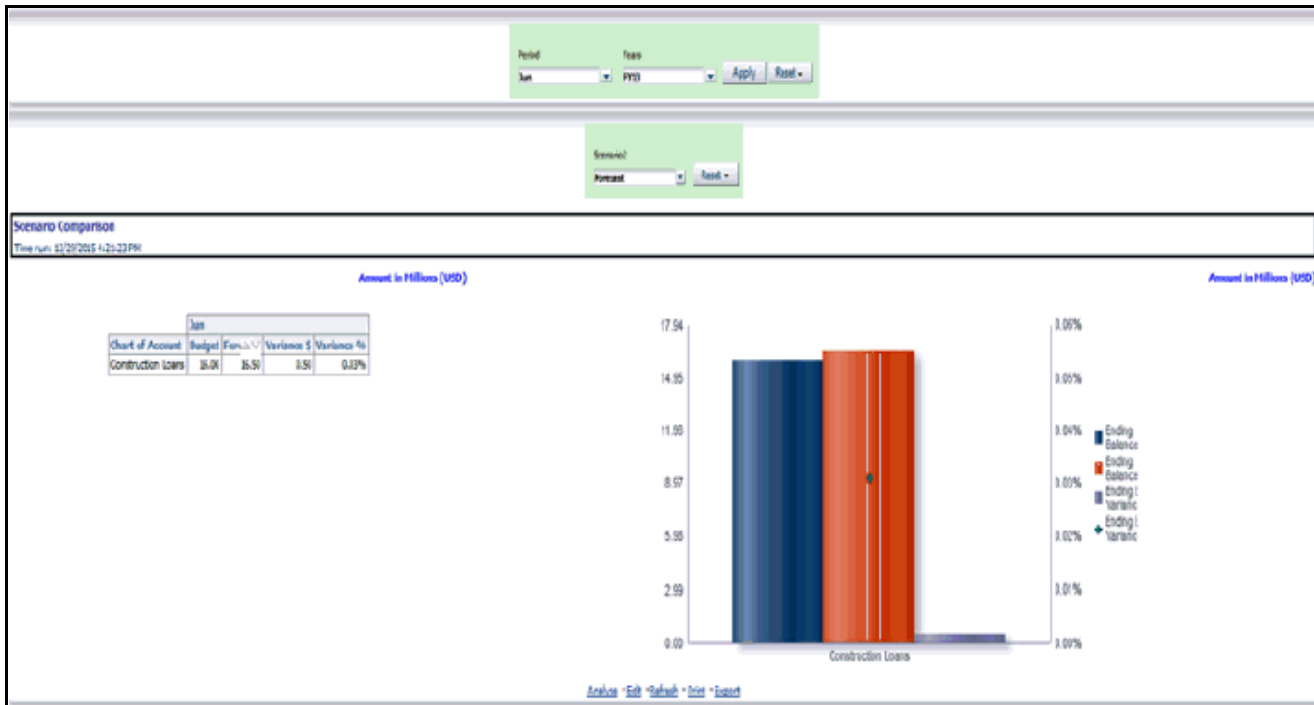


Figure 23. Forecast Reports - Balance Sheet Reports – Scenario Comparison

- Forecast Reports - Balance Sheet Reports – Strategy Comparison

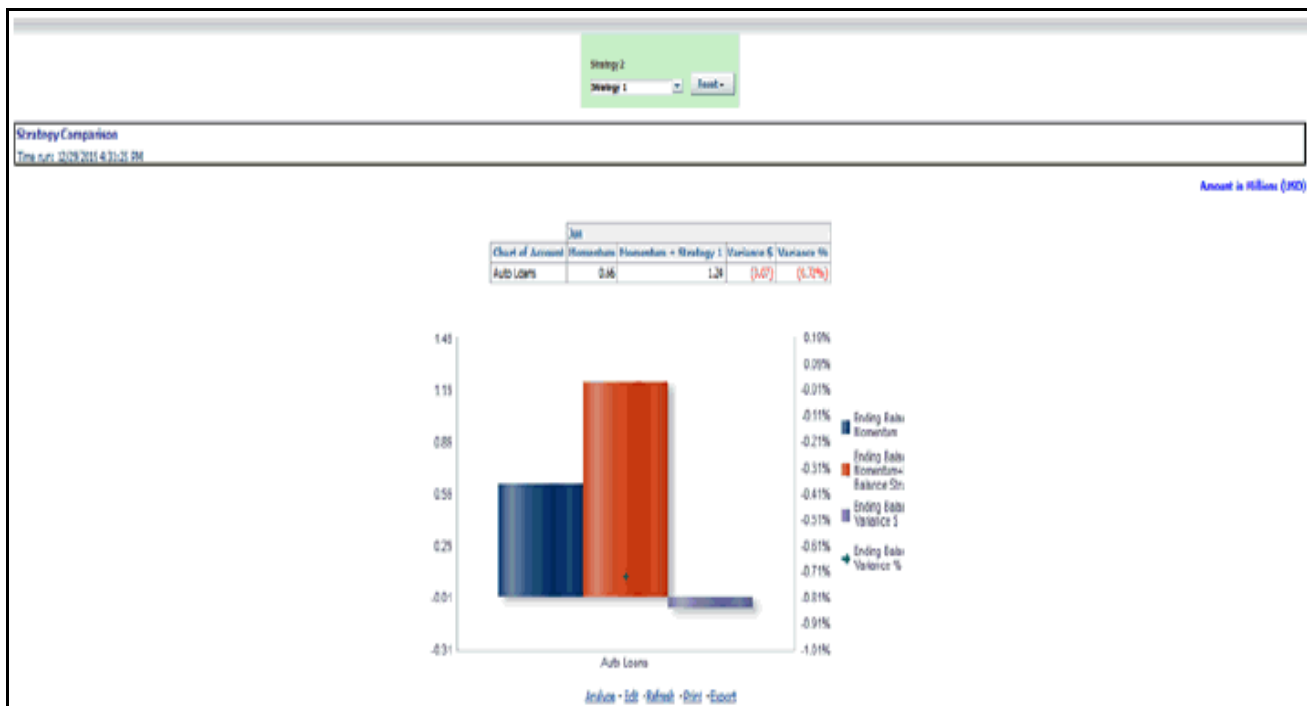


Figure 24. Forecast Reports - Balance Sheet Reports – Strategy Comparison

- Forecast Reports - Income Statement Reports – Interest Vs. Average Balance



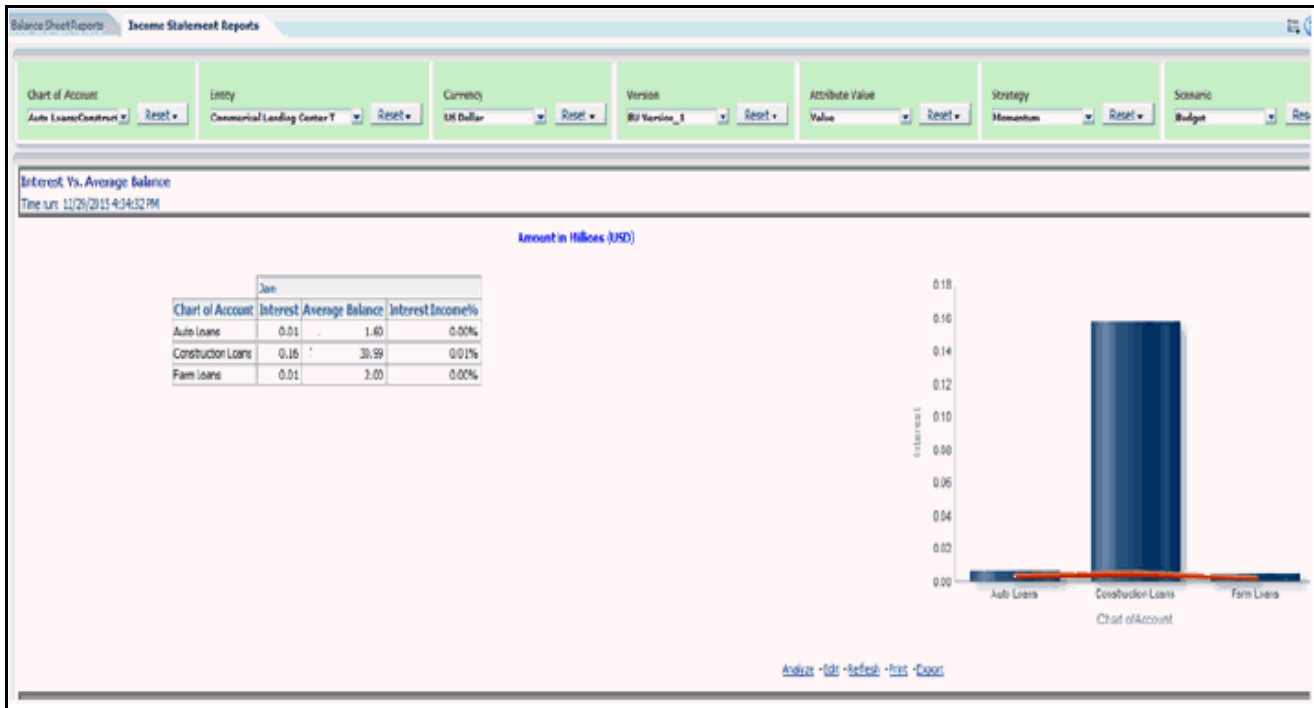


Figure 25. Forecast Reports - Income Statement Reports – Interest Vs. Average Balance

- Forecast Reports - Income Statement Reports – Non Interest Fee Percentage Vs. Ending Balance

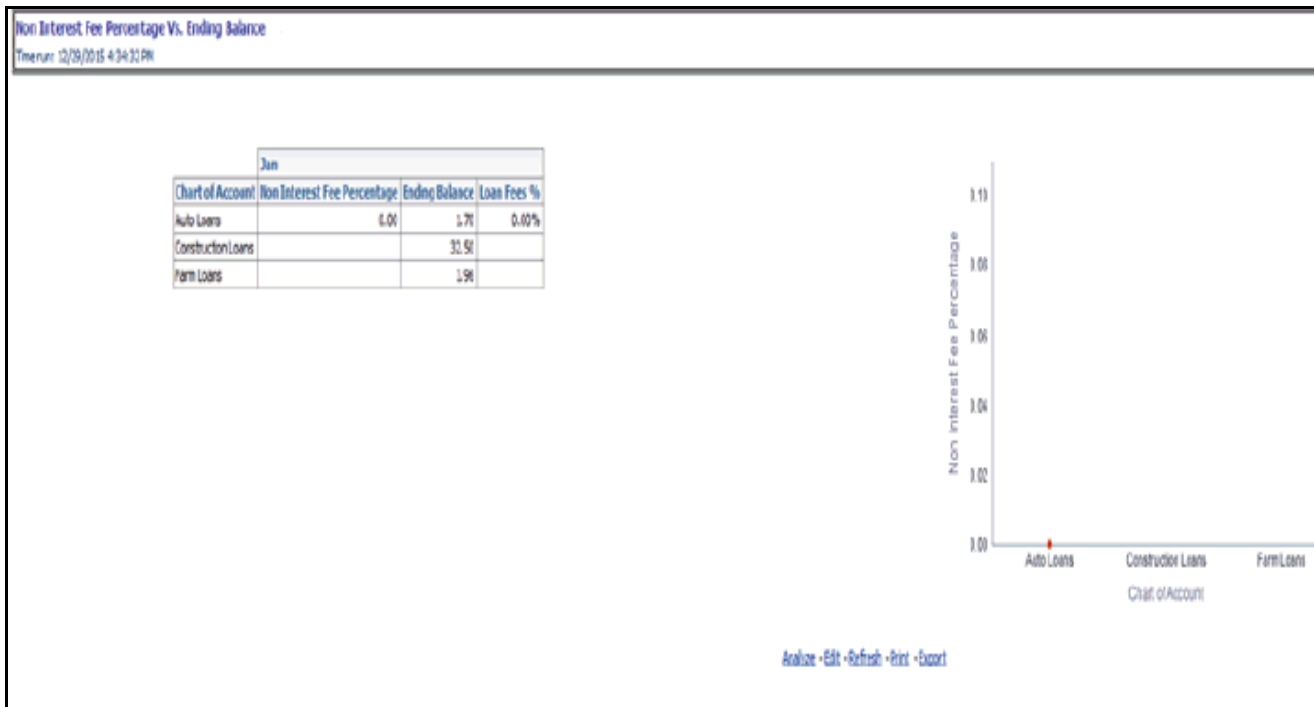


Figure 26. Forecast Reports - Income Statement Reports – Non Interest Fee Percentage Vs. Ending Balance

- Forecast Reports - Income Statement Reports – Spread Analysis

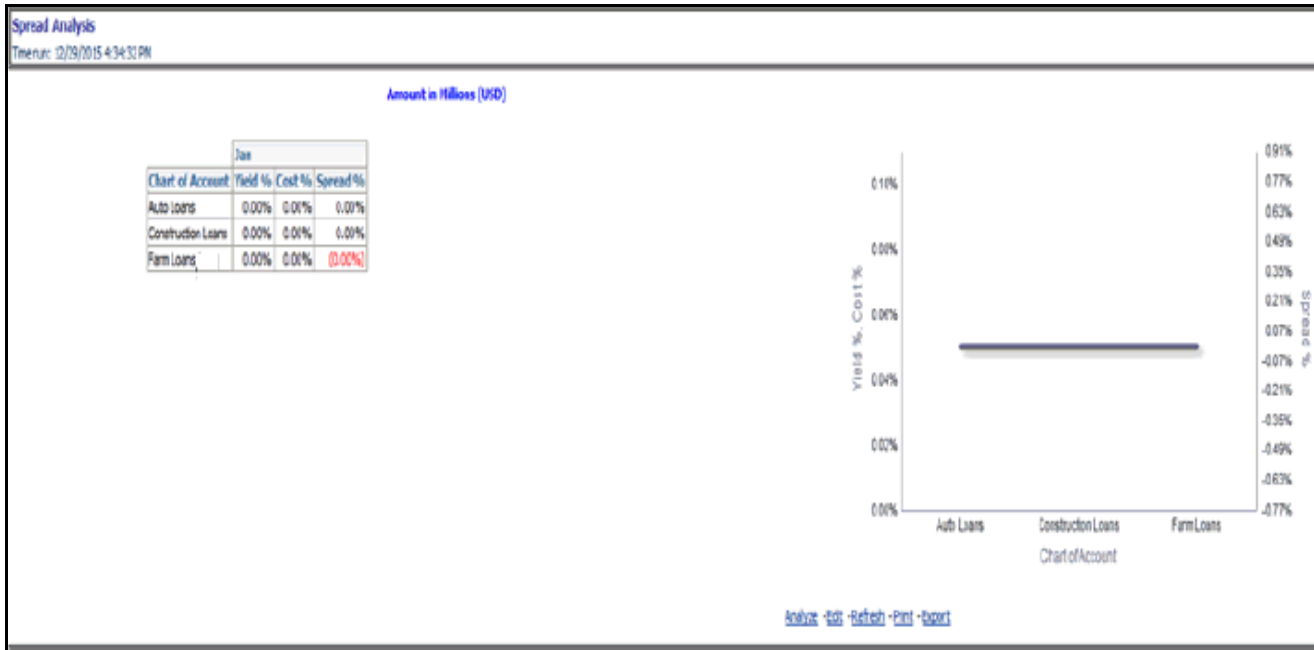


Figure 27. Forecast Reports - Income Statement Reports – Spread Analysis

- Forecast Reports - Income Statement Reports – NIM Report

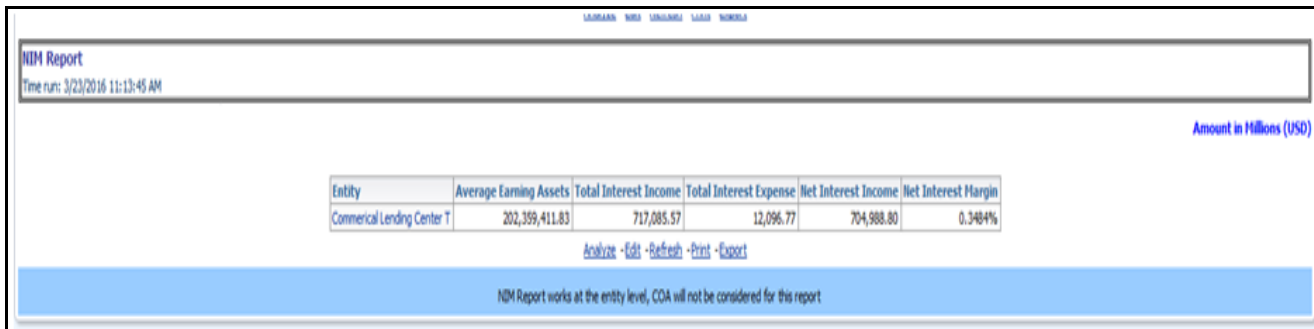


Figure 28. Forecast Reports – Income Statement Reports - NIM Report

This appendix discusses the following topics:

- [Introduction - Dimension Definition](#)
- [Dimension Definition Process](#)

## Introduction - Dimension Definition

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 hold the 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 for adding dimension tables, refer to *Oracle Financial Services Analytical Applications Data Model Utilities User Guide*. If the dimension data is fetched from OFSPM, it reads only the nodes/leaves belonging to a single hierarchy. Therefore, before fetching dimension data from OFSPM, ensure that the AMHM tables belonging to a single dimension have data present for only one hierarchy.

## Dimension Definition Process

### Create Business Hierarchy

1. From **Unified Metadata Manager**, select **Business Metadata Management**, then select **Business Hierarchy**.
2. From Business Hierarchy, click **Add** to create a Business hierarchy definition. In the *Business Hierarchy Definition (Add mode)* window, select the **Hierarchy Type**. Hierarchy Type can be :
  - **Regular** - for representing non-time and non-measure dimensions in a hierarchical format. Examples of this type are Product, Organization Unit, and so on.
  - **Measure** - for representing the measures in the hierarchical format. An example of this type is Management Reporting Line.
  - **Time** - for representing the calendar or date dimension in a hierarchical format. An example of this type is Calendar hierarchy.
3. Select **Hierarchy Sub Type**.

Hierarchy Sub Type can be:

- **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 sub type 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 sub-type is useful for modelling balanced hierarchies.

- **Parent Child** - for representing the hierarchy with underlying data store as a parent-child dimension table. This sub type is useful for modelling ragged hierarchies.
4. Select **Total Required** property, if a TOTAL is required to be included as the root node of the hierarchy.
  5. Select **List** property, if hierarchy is a flat list of members without any levels.
  6. Select the **Entity** and **Attribute** on which the hierarchy is based.

The components for hierarchy definition differ for each sub type of the hierarchy.

- If sub type is **Non Business Intelligence Enabled**, then the user can add nodes and the 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 the column that contains the child member.

For more details, refer to [Oracle Financial Services Analytical Applications Infrastructure User Guide](#).

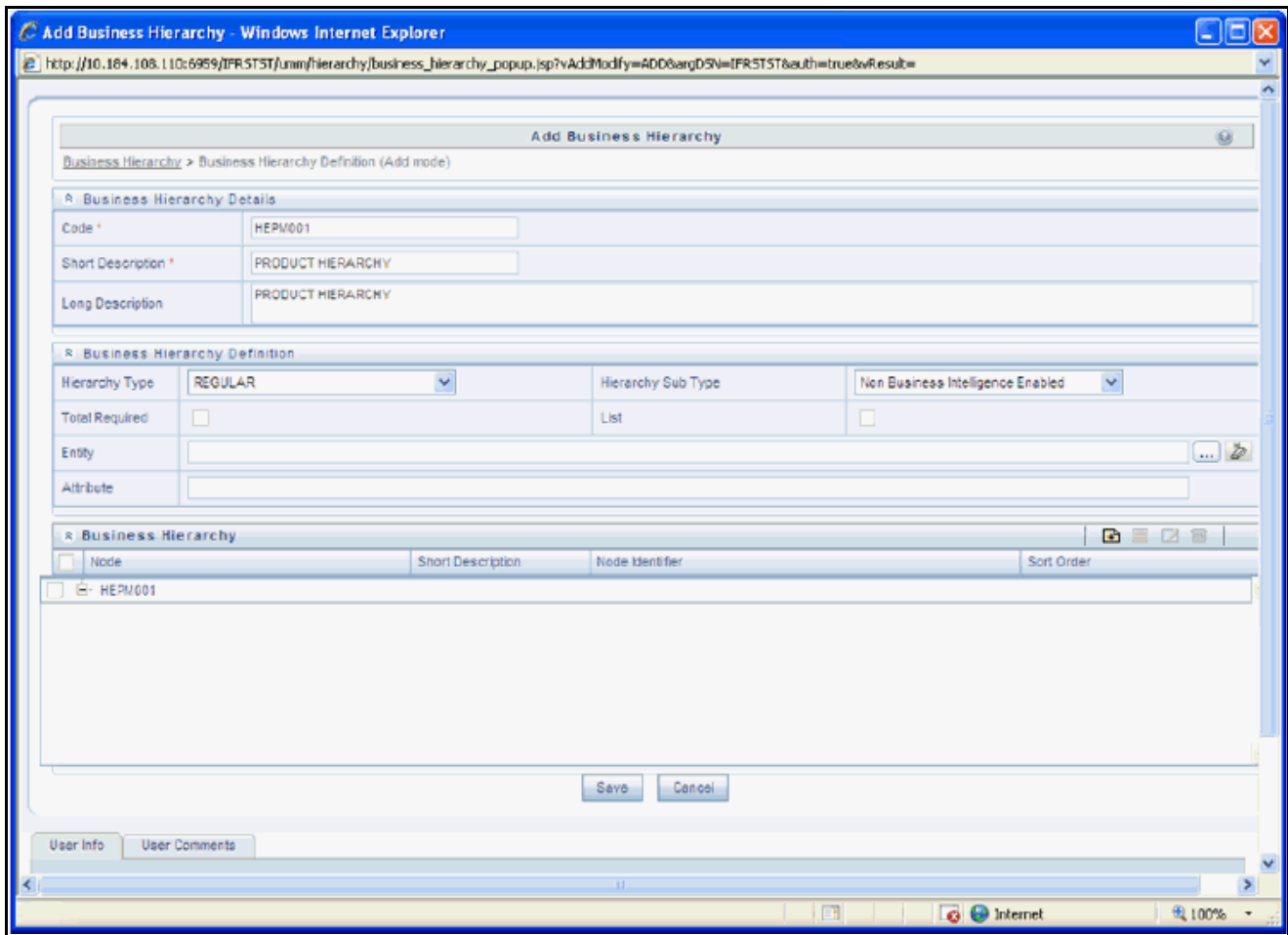


Figure 29. Business Hierarchy

## Create Business Dimension

1. From **Unified Metadata Manager**, select **Business Metadata Management**, then select **Business Dimension**.
2. From Business Dimension, click **Add** to create a Business dimension definition. In the *Business Dimension Definition (Add mode)* window, select the **Dimension Type**. Dimension type is same as Hierarchy type and helps to filter the hierarchies that will be part of the dimension.
3. A dimension will contain one or many hierarchies. Select the hierarchies that are part of the dimension.

Figure 30. Business Hierarchy Details Page

## Modify Dataset

1. From **Unified Metadata Manager**, select **Business Metadata Management**, then select **Data Sets**.
2. Identify data sets that are based on the modified fact table.
3. Edit the data set definition.

4. Include the new dimension table in the data set.
5. Modify the data set JOIN to include the join clause between the fact table and new dimension table.
6. Save the data set.

**Edit Data Sets**

[Data Sets](#) > Data Set Definition (Edit mode)

---

**^ Data Set Details**

Code *	DSFVNCP
Short Description *	Account Fair-Value Inception
Long Description	Hedge Management Inception Dataset for Account FV

---

**^ Entities** ... 1 to 5 of 5

Selected Entities
DIM_DATES
DIM_FCST_RATES_SCENARIO
DIM_HEDGE
FCT_ACCOUNT_FAIR_VALUE
FSL_HM_HEDGE_INSTRU_MAP

---

**^ Data Set Definition**

ANSI Join	
Join/Filter Condition	DIM_HEDGE.N_HEDGE_ID = FSL_HM_HEDGE_INSTRU_MAP.HEDGE_ID AND FCT_ACCOUNT_FAIR_VALUE.N_ID_NUMBER = FSL_HM_HEDGE_INSTRU_MAP.ID_NUMBER
Date Filter	
Order By	

**Figure 31. Data Set Details Page**

---

## APPENDIX B *How to Define a Measure*

This appendix discusses the following topics:

- [Introduction to Measure Definition](#)
- [Measure Definition Process](#)

### Introduction to Measure Definition

As a prerequisite, the fact table needs to have the column that holds values for the new measure.

### Measure Definition Process

#### Create 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.

**Add Business Measures**

[Business Measures](#) > Business Measure Definition (Add mode)

---

**\* Business Measure Details**

Code *	MEPM001
Short Description *	EOP Balance
Long Description	End of period balance

---

**\* Business Measure Definition**

Aggregation Function	SUM	DataType	Decimal
Roll up	<input checked="" type="checkbox"/>		
Entity			
Attribute			
Business Exclusions			
Filter Expression			

Figure 32. Business Measure Details Page



---

## **APPENDIX C** *How to Develop a New Cube*

This appendix discusses the following topics:

- [Introduction to Developing a New Cube](#)
- [Procedures 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*.



---

## APPENDIX D *How to Define a Batch*

This appendix discusses the following topics:

- [Introduction](#)
- [Batch Creation](#)

### 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.

**Table 25. Batch Details**

Field	Description
Batch Name	The <b>Batch Name</b> is auto generated by the system. You can edit to specify a Batch name based on the following conditions: <ul style="list-style-type: none"><li>● The Batch Name should be unique across the Information Domain.</li><li>● The Batch Name must be alpha-numeric and should not start with a number.</li><li>● The Batch Name should not exceed 41 characters in length.</li><li>● The Batch Name should not contain special characters "." and "-".</li></ul>
Batch Description	Enter a description for the Batch based on the Batch Name.

**Table 25. Batch Details**

Duplicate Batch	(Optional) Select the checkbox to create a new Batch by duplicating the existing Batch details. On selection, the <b>Batch ID</b> field is enabled.
Batch ID (If duplicate Batch is selected)	It is mandatory to specify the Batch ID if Duplicate Batch option is selected. Select the required <b>Batch ID</b> from the list.
Sequential Batch	Select the check box if the Batch has to be created sequentially based on the task specified. For example, if there are 3 tasks defined in a Batch, task 3 should have precedence as task 2, and task 2 should have precedence as task 1.

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

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

**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*.