

# **Oracle Financial Services Profitability Management User Guide**

Release 8

Part Number E55504-01

August 2022

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

Primary Author: Raghavacharyulu NCB

Other Author: Tim Braman

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 failsafe, 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.

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 or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

# Contents

<b>Preface .....</b>	<b>9</b>
----------------------	----------

## CHAPTER 1

<b>Introduction.....</b>	<b>11</b>
--------------------------	-----------

Introduction to Profitability Management.....	11
Oracle Financial Services Enterprise Performance Management.....	11
Oracle Financial Services Enterprise Risk Management .....	13
Getting Started.....	13
Navigating to the Financial Services Applications Home Page .....	17
<i>Financial Services Applications Home Page.....</i>	<i>17</i>
<i>Global Preferences .....</i>	<i>18</i>

## CHAPTER 2

<b>Overview of OFSAA Infrastructure .....</b>	<b>19</b>
---	-----------

OFSAA Infrastructure .....	19
Overview of Dimensionality in OFSAA .....	20
Referential Integrity.....	20
Dimension Attributes.....	20
Hierarchies.....	21
Rollup Members for Dimensions Supporting Hierarchies .....	22
Dimension Types .....	22
<i>Key Processing Dimensions.....</i>	<i>22</i>
<i>Seeded Key Processing Dimensions .....</i>	<i>22</i>
<i>User Defined Key Processing Dimensions .....</i>	<i>23</i>
Standard Dimensions.....	23
Simple Dimensions.....	23
Overview of Attributes .....	24
Overview of OFSAA Rules.....	25
OFSAA Filters .....	26
OFSAA Expressions.....	27
Folders .....	27
Access Type.....	28
Global Preferences.....	28
System Configuration.....	29

## CHAPTER 3

<b>OFSAA Administrative Reporting .....</b>	<b>31</b>
---	-----------

Introduction .....	31
Access Permissions .....	33
Data Quality.....	33
Data Quality Dashboard .....	34
Object Registration.....	41
Security.....	42

Adhoc Reporting – Special Instructions.....	47
---	----

CHAPTER 4

**OFSAA Data Entry Forms and Queries ..... 49**

Introduction to Data Entry Forms and Queries .....	49
Acronyms .....	49
Designing Data Entry Forms .....	50
Creating a New Form .....	50
The Grid Layout .....	51
The Single Record Layout.....	52
The Edit/View Layout .....	52
The Multi Column Layout.....	53
The Wrapping Row Layout .....	54
The Tree View Layout.....	54
Joining Tables .....	56
<i>For Grid, Single Record, Edit/View, Wrapping Row, Multicolumn and Tree View Layouts.</i> .....	57
<i>For Grid, Single Record, Edit/View, Wrapping Row &amp; Multicolumn Layouts</i> .....	59
Applying Rules.....	60
Committing the Batch .....	61
Defining the Form Filter.....	61
Data Versioning .....	62
Specifying Message Details .....	63
Altering an Existing Form.....	69
Copying an Existing Form .....	73
Deleting a Form.....	73
Assigning Rights to a Form.....	74
Message Type Maintenance.....	76
Entering Data.....	78
Viewing Form Details.....	79
<i>Single Record layout</i> .....	79
<i>Grid Layout</i> .....	81
<i>Multi Column Layout</i> .....	81
<i>Wrapping Row Layout</i> .....	82
<i>Editing a Form</i> .....	83
<i>Roll Back</i> .....	83
<i>Adding Rows to a Form</i> .....	84
<i>Deleting a Record in a Form</i> .....	84
<i>Authorizing Data</i> .....	84
<i>Re-authorizing Records</i> .....	86

CHAPTER 5

**OFSAA Attributes ..... 87**

Overview .....	87
Summary & Detail Screens .....	87
Navigation within the Summary Screen.....	88
<i>Dimension Container</i> .....	88
<i>Search Container</i> .....	88
<i>Dimension Attributes Container</i> .....	88

<i>Dimension Attributes Summary Grid</i> .....	89
Navigation within the Detail Screen.....	91
<i>Attribute Functionality</i> .....	91
<i>Dimension Container</i> .....	91
<i>Attribute Details Container</i> .....	91
<i>Attribute Properties Container</i> .....	92

CHAPTER 6

**OFSAA Dimension Members** ..... 95

Overview .....	95
Summary & Detail Screens .....	95
Navigation within the Summary Screen.....	96
<i>Dimension Container</i> .....	96
<i>Search Container</i> .....	96
<i>Dimension Members Container</i> .....	97
<i>Dimension Members Summary Grid</i> .....	98
Navigation within the Detail Screen.....	99
<i>Dimension Member Functionality</i> .....	99
<i>Dimension Container</i> .....	99
<i>Dimension Member Details Container</i> .....	99
<i>Dimension Member Attributes Container</i> .....	101

CHAPTER 7

**OFSAA Hierarchies** ..... 103

Overview of OFSAA Hierarchies.....	103
Summary & Detail Screens .....	103
Navigation within the Summary Screen.....	103
<i>Dimension Container</i> .....	104
<i>Search Container</i> .....	104
<i>Hierarchies Container</i> .....	104
Hierarchies Summary Grid.....	105
Navigation within the Detail Screen.....	106
<i>Hierarchy Functionality</i> .....	107
<i>Dimension Container</i> .....	107
<i>Hierarchy Properties Container</i> .....	107
<i>Hierarchy Display Grid</i> .....	109
Controlling the Hierarchy Display .....	110
Adding Hierarchies.....	111
Editing Hierarchies.....	113

CHAPTER 8

**OFSAA Filters** ..... 125

Overview of OFSAA Filters.....	125
Summary & Detail Screens .....	125
Navigation within the Summary Screen.....	126
Navigation within the Detail Screen.....	127
Data Element Filters.....	130
Hierarchy Filters.....	135
Group Filters.....	137

	Define Attribute Filter .....	138
<b>CHAPTER 9</b>	<b><i>OFSAA Expressions</i></b> .....	<b>139</b>
	Expressions.....	139
	Summary & Detail Screens .....	139
	Navigation within the Summary Screen.....	140
	Expressions Container .....	140
	Expressions Summary Grid.....	142
	Navigation within the Detail Screen.....	143
	Entity Type & Entity Name .....	145
	Functions and Operators Sub-Containers .....	146
<b>CHAPTER 10</b>	<b><i>Simplified Batches</i></b> .....	<b>157</b>
	Overview of Simplified Batches .....	157
	Summary and Detail Screens.....	158
	Navigation within the Summary Screen.....	158
	Navigation within the Detail Screen.....	160
	Task Selection.....	163
	Running a Simplified batch .....	166
	Viewing the Task Logs .....	169
	Additional Batch Options .....	170
<b>CHAPTER 11</b>	<b><i>Holiday Calendar</i></b> .....	<b>171</b>
	Overview of Holiday Calendars .....	171
	Searching for a Holiday Calendar .....	171
	Creating a Holiday Calendar .....	172
	Executing Holiday Calendar.....	174
	Holiday Exceptions .....	175
	Excel Import/ Export .....	175
<b>CHAPTER 12</b>	<b><i>OFSAA Rate Management</i></b> .....	<b>177</b>
	Overview .....	177
	Interest Rates Summary Screen.....	178
	Interest Rates Detail Screen.....	181
	Currency .....	202
	Currency Rates .....	205
	Currency Exchange Rate Validation.....	209
	<i>Currency Rate Validation – Overview</i> .....	209
	<i>Exchange Rate Validation – Feature Description</i> .....	210
	<i>Currency Rate Validation – Procedure</i> .....	210
	<i>Validating Exchange Rate Relationships</i> .....	210
	Economic Indicators Summary Screen .....	216
	Economic Indicators Detail Screen .....	218

CHAPTER 13	<b>Overview of Profitability Management</b> .....	223
	Introduction .....	223
	Multiple Dimensions of Profitability .....	223
	Overview of Profitability Management Methodologies.....	227
	Initial Loads to the Management Ledger.....	229
CHAPTER 14	<b>Application Preferences</b> .....	233
	Overview of Profitability Management Application Preferences .....	233
	Updating Profitability Management Application Preferences.....	233
CHAPTER 15	<b>Profitability Configuration</b> .....	239
	Setup Information .....	239
	Parallel Execution Configuration .....	240
	Setup Parameters .....	240
	Management Ledger Configuration .....	241
CHAPTER 16	<b>Allocation Specification</b> .....	243
	Introduction .....	243
	Summary & Detail Screens .....	243
	Navigation within the Summary Screen.....	243
	Navigation within the Detail Screen.....	247
	Initial Definition Process Tab .....	248
	<i>Source Process Tab</i> .....	252
	Operator Process Tab.....	260
	Driver Process Tab.....	262
	Outputs Process Tab .....	266
	Review Process Tab.....	272
	Allocation Examples .....	273
	Aggregate Instrument Table .....	285
CHAPTER 17	<b>Allocation Models</b> .....	287
	Overview of Allocation Models .....	287
	Summary and Detail Screens.....	287
	Navigation within the Summary Screen.....	287
	Navigation within the Detail Screen.....	291
CHAPTER 18	<b>Static Table Driver</b> .....	297
	Overview .....	297
	Summary & Detail Screens .....	297
	Navigation within the Summary Screen.....	297
	Navigation within the Detail Screen.....	302
	Creating a New Static Table Driver – Sample Workflow.....	304

	<i>Creating a New Target Leaf</i> .....	306
	Working with Exported Static Table Driver Data.....	314
	<i>Updating Exported Static Table Driver Data</i> .....	315
	<i>Adding New Combinations to Exported Data</i> .....	315
	<i>Validating the Imported Data</i> .....	316
	Large Cross Product Static Table Drivers .....	317
	<i>Limit on Large Cross Products</i> .....	317
	<i>Managing Large Cross Product Static Table Drivers</i> .....	317
	<i>Editing Existing Coefficient Values</i> .....	318
	Using Static Table Drivers.....	318
<b>CHAPTER 19</b>	<b><i>Lookup Table Driver</i> .....</b>	<b>321</b>
	Overview .....	321
	Summary & Detail Screens .....	321
	Navigation within the Summary Screen.....	321
	Navigation within the Detail Screen.....	324
<b>CHAPTER 20</b>	<b><i>Allocation Execution History</i> .....</b>	<b>331</b>
	Allocation Execution History Functionality.....	331
	Allocation Execution History User Interface .....	331
	Allocation Execution Audit Functionality.....	336
<b>CHAPTER 21</b>	<b><i>Viewing PFT Objects in Metadata Browser</i> .....</b>	<b>345</b>
	Metadata Browser (Object and Application View) .....	345
	Viewing PFT Objects in Metadata Browser .....	346
<b>Appendix A</b>	<b><i>Management Ledger</i> .....</b>	<b>349</b>
	Statement of Direction.....	349
	Frequently Asked Questions.....	349
<b>Appendix B</b>	<b><i>Archive-Restore</i> .....</b>	<b>353</b>
	Command Line Migration .....	353
	Command Line Migration Utility.....	353
	Supported Objects .....	353
	Input Values for OBJECTMIGRATION.xml .....	354
	<i>Logging</i> .....	355
<b>Appendix C</b>	<b><i>Multi Language Support</i> .....</b>	<b>357</b>
	Overview .....	357
<b>Appendix D</b>	<b><i>Multicurrency</i> .....</b>	<b>359</b>



OFSAA Rate Management .....	359
Functional Currency .....	360
Currency Codes .....	361
002 code.....	361
Loading the Management Ledger Table .....	361
Multiple Currencies in Allocation Rules .....	361
Using the Currency Dimension as a Filtering Condition .....	362
Allocation Engine Process Processing Flow .....	362
Selecting Rows from Ledger/Stat.....	363
Treatment of Constants and Coefficient Values .....	364
Selecting Rows from Detail Tables .....	364
Inserting Into or Updating Ledger/Stat.....	365
Updating Detail Tables .....	365
Notes on Usage of Statistics and Other Balances.....	365
Monetary Balances and Balance Weighted Objects .....	366
Statistics & Standard Rates .....	366
Examples .....	367

**Appendix E      *Fiscal Year Information* ..... 371**

Configuring a Fiscal Year .....	371
How YTD Calculations Are Affected by a Fiscal Year Configuration.....	372
Examples of Calendar Year and Fiscal Year Configurations.....	372
<i>Using a Calendar Year Configuration</i> .....	373
<i>Using a Fiscal Year Configuration with a 12-Month Duration</i> .....	373
<i>Using a Fiscal Year Configuration with a 6-Month Duration</i> .....	374
Using the Undo Function with a Fiscal Year Configuration .....	375

**Appendix F      *Seeded Financial Elements* ..... 377**

Financial Elements Listed By Number.....	377
--	-----

**Appendix G      *Debit and Credit Conventions* ..... 391**

Conventions within the Management Ledger.....	391
---	-----

**Appendix H      *Financial Element Aggregation Methods* ..... 397**

Overview .....	397
Aggregation Methods .....	397

**Appendix I      *Performance Tuning for Profitability Management* ... 399**

SQL Hints .....	399
<i>Overview</i> .....	399
<i>Types of SQL hints</i> .....	400
<i>How to use SQL Hints</i> .....	400
<i>Sample Queries</i> .....	401
Profitability Management Parallel Execution .....	409
<i>Overview</i> .....	409
<i>Parallelism Options</i> .....	410
<i>Parallel Execution Documentation Resources</i> .....	414

<b>Appendix J</b>	<b>Seeded Batches .....</b>	<b>417</b>
<b>Appendix K</b>	<b>Glossary .....</b>	<b>419</b>

---

# Preface

## **Intended Audience**

Welcome to Release 8 of the *Oracle Financial Services Profitability Management User Guide*.

## **Forward**

This user guide documents OFSAA Profitability Management for all versions of release 8.

See Related Information Sources for more Oracle Applications product information.

## **Documentation Accessibility**

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

Access to Oracle Support

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

## **Structure**

- 1 [Introduction](#)
- 2 [Overview of OFSAA Infrastructure](#)
- 3 [OFSAA Administrative Reporting](#)
- 4 [OFSAA Data Entry Forms and Queries](#)
- 5 [OFSAA Attributes](#)
- 6 [OFSAA Dimension Members](#)
- 7 [OFSAA Hierarchies](#)
- 8 [OFSAA Filters](#)
- 9 [OFSAA Expressions](#)
- 10 [Simplified Batches](#)
- 11 [Holiday Calendar](#)
- 12 [OFSAA Rate Management](#)

This chapter discusses the procedure for creating a Holiday Calendar and generating a list of weekend and holiday dates. Individual applications may consume the holiday calendar events in different ways, so please refer to application specific user guides and/or the Cash Flow Engine Reference Guide for more details.

- 13 [Overview of Profitability Management](#)
- 14 [Application Preferences](#)
- 15 [Allocation Specification](#)
- 16 [Allocation Models](#)
- 17 [Static Table Driver](#)
- 18 [Lookup Table Driver](#)
- 19 [Allocation Execution History](#)

[20 Viewing PFT Objects in Metadata Browser](#)  
[Appendix A, “Management Ledger”](#)  
[Appendix C, “Multi Language Support”](#)  
[Appendix D, “Multicurrency”](#)

This Appendix covers:

Implementation of multiple currencies in OFSAA solutions

How currency translation is used in OFSAA Profitability Management calculations

Usage of multiple-currency output from Profitability Management in relation to the Management Ledger, Instrument, and Transaction Summary tables

[Appendix E, “Fiscal Year Information”](#)

Oracle Financial Services Analytical Applications (OFSAA) support both calendar year (January to December) and fiscal year configurations. If your organization operates on a fiscal year calendar, this appendix documents how to configure your fiscal year within OFSAA and how a fiscal year configuration affects processes that insert into or update the Ledger/Stat table. Note that in the context of this appendix, a fiscal year denotes a non-calendar financial year (that is, a year that does not contain 12 months and/or does not begin in January and end in December).

[Appendix F, “Seeded Financial Elements”](#)

[Appendix G, “Debit and Credit Conventions”](#)

[Appendix H, “Financial Element Aggregation Methods”](#)

[Appendix I, “Performance Tuning for Profitability Management”](#)

[Appendix K, “Glossary”](#)

### ***Related Information Sources***

- Oracle Financial Services Cash Flow Engine Reference Guide
- Oracle Financial Services Analytical Applications Data Model Utilities User Guide
- Oracle Financial Services Analytical Applications Data Model Data Dictionary
- Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration Guide
- Oracle Financial Services Analytical Applications Infrastructure User Guide
- Oracle Financial Services Funds Transfer Pricing User Guide
- Oracle Financial Services Asset Liability Management (OFSALM) User Guide
- Oracle Financial Services Profitability Analytics User Guide

---

**CHAPTER 1*****Introduction******Introduction to Profitability Management***

This chapter presents the following topics:

- Oracle Financial Services Analytical Applications Overview
- Oracle Financial Services Enterprise Performance Management
- Oracle Financial Services Enterprise Risk Management
- Getting Started

***Oracle Financial Services Enterprise Performance Management***

To help financial institutions grow, manage risk, and optimize shareholder value, Oracle delivers a comprehensive, integrated suite of financial services analytical applications.

More than ever, financial institutions, their regulators as well as their shareholders are focused on the need to measure and meet risk-adjusted performance objectives, price products to reflect their true risk, and better understand how their institution is impacted by threats to liquidity, capital adequacy, and exposure to market rate volatility.

***OFSAA Data Model***

OFSAA Data Model (OFSAADM) is a standalone data warehouse with prepackaged data elements for the financial services industry. OFSAADM is also the foundation for the OFS applications. It provides the database structures necessary to support the individual business applications.

***OFSAA Funds Transfer Pricing***

Oracle Financial Services Funds Transfer Pricing is the industry's first and leading matched maturity funds transfer pricing application, enabling financial institutions to determine the spread earned on assets, from funding sources, and the spread earned as a result of interest rate exposure for each and every customer relationship. This enables accurate assessment of profitability along product, channel, and business lines, as well as the centralization of interest rate risk so it can be effectively managed. Oracle Financial Services Funds Transfer Pricing is fully integrated with Oracle's Financial Services Analytical Applications and shares a common customer account level data model.

The transfer pricing process isolates the four major sources of a bank's margin:

- The spread earned on assets
- The spread earned on liabilities
- The spread earned or lost as a result of interest rate exposure
- The spread earned or lost as a result of embedded options such as prepayments

Measuring and managing interest margin offers tremendous profit opportunities. By separating the components of net interest income, Oracle Funds Transfer Pricing isolates rate risk into your funding center where it can be centrally managed. In turn, business units are held accountable for what they can control: pricing and profitability. Armed with this highly accurate information, companies make solid, supportable decisions that lead to increased success in the marketplace.

### ***OFSAA Profitability Management***

Oracle Financial Services Profitability Management enables financial services institutions to calculate profitability by products, channels, segments, and even individual customers. These profitability calculations are adjusted for risk, and they drive Risk-Adjusted Performance Management (RAPM), an imperative for financial services institutions operating in this rapidly evolving and complex industry.

Isolating and analyzing business unit profitability is only the first step in building effective profitability management. To hold operational managers truly accountable for profitability, senior management must give them the power to identify the products that are profitable in each market segment, for each customer.

Together with OFSAA Transfer Pricing, OFSAA Profitability Management is the tool that delivers this power. Profitability Management links general ledger, account-level, and statistical data together to produce detailed financial statements for individual business units, products, and customers. Combining powerful modeling and disciplined accounting, OFSAA Profitability Management delivers complete, accurate, and inclusive profiles of profitability.

### ***OFSAA Pricing Management, Transfer Pricing Component***

Oracle Financial Services Pricing Management, Transfer Pricing Component is an interactive application that provides real-time transfer rates to support pricing loan transactions that reflect immediate market conditions. Risk-based pricing queries are directly integrated with the Oracle Financial Services Funds Transfer Pricing solution and leverage the same business rule logic. Query results incorporate all details of the underlying transfer price and include all elements of a product's profit and loss, including the allocated expense, expected loss, and capital charge.

### ***OFSAA Asset Liability Management***

Volatile market conditions and increasing regulatory pressures are placing greater demands on the risk management function. These stresses are driving financial institutions to review their current risk modeling and measurement capabilities. Yet, these circumstances also provide institutions with the opportunity to update technology solutions to systems fully integrated across the risk and performance spectrum. Departmental, one-off solutions are no longer viable alternatives in an environment where all systems need to work together.

Oracle Financial Services Asset Liability Management (ALM) helps financial services institutions measure and manage interest rate risk and liquidity risk, perform balance sheet valuations, and forecast both net interest income and net income. The Oracle Financial Services ALM solution measures and models every loan, deposit, investment, and off-balance sheet instrument individually, using both deterministic and stochastic methods. This helps institutions gain a better understanding of the risks they have assumed and their sensitivity to changes in economic conditions.

### ***OFSAA Balance Sheet Planning***

Oracle Financial Services Balance Sheet Planning is designed to help financial services institutions budget for a full balance sheet and the associated profit and loss statement.

Banks have a number of very unique needs when looking ahead. They must be sensitive to economic conditions, and create plans that not only forecast future performance, but also the forward risks they are assuming. Most importantly, they require tools that accurately forecast net interest margin. By accurately modeling the detailed and complex events on a bank's balance sheet, for both the current book of business and forecasted new volumes, Oracle Financial Services Balance Sheet Planning enables the delivery of accurate margin forecast and comprehensive, meaningful budgets.

## **Oracle Financial Services Enterprise Risk Management**

Enterprise risk management (ERM) solutions span all areas of risk, including credit, market, operational, liquidity, fixed asset, business and reputation risk, enabling institutions to manage all their risks and provide necessary feeds for compliance-based reporting. The capabilities of Oracle Reveleus for risk management help financial institutions address compliance issues such as Basel II that demand a close operational synergy between finance and other business areas.

### **Governance Risk and Compliance (GRC)**

The Oracle Reveleus GRC (Governance, Risk, and Compliance) framework surrounds both the Oracle Reveleus and Oracle Mantas risk and compliance "point" solutions. It provides an end-to-end GRC solution that leverages a comprehensive data model and robust dashboard functionality. The framework facilitates a transparent view of all GRC-related activities across the enterprise, allowing senior management to identify and mitigate any operational inefficiencies that might potentially increase noncompliance risks or the overall cost of compliance.

### **Customer Insight**

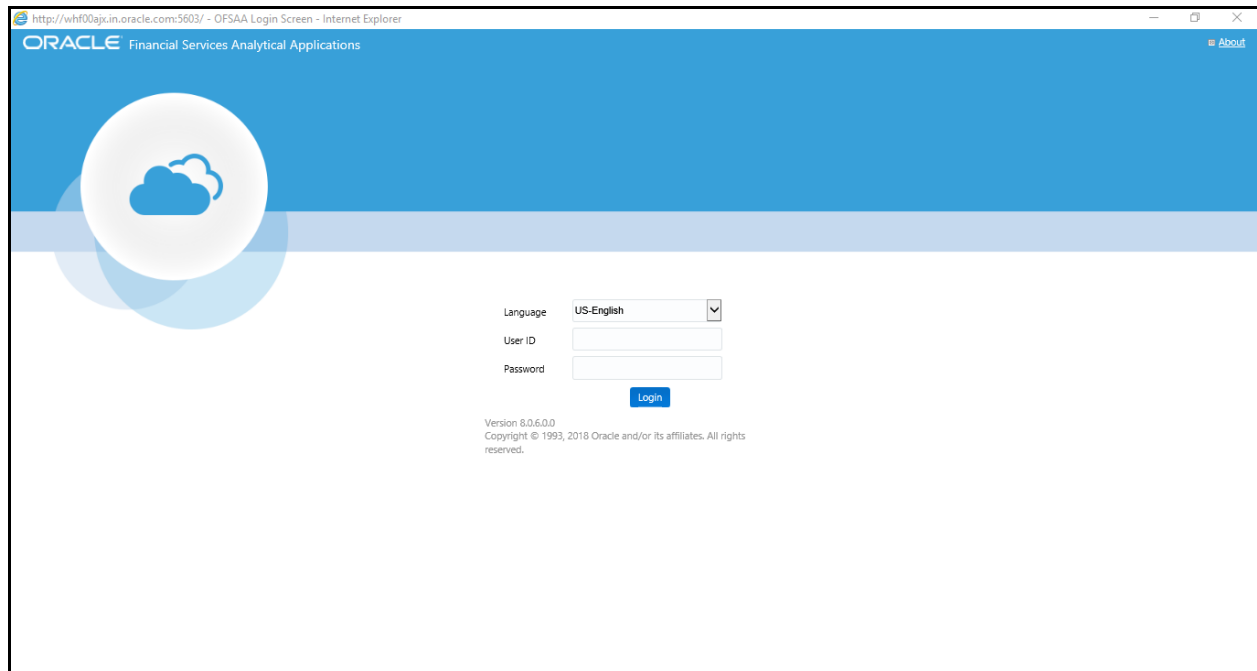
Evaluating customer value is becoming tougher. Customer relationships are becoming more complex, involving several bank personnel, and have extended time periods. Increasingly, banks are customizing and personalizing products and services. While decisions are becoming riskier, time windows available for decision and action are becoming smaller. Such real-world scenarios mandate timely and comprehensive decision support. Oracle's Customer Insight solutions deliver targeted and actionable customer analysis that help financial institutions make quick decisions in a rapidly evolving marketplace.

## **Getting Started**

Your System Administrator will provide you with a link through which you can access OFSAA Infrastructure. OFSAAI security ensures that every user is permitted access to screens based on his or her role. Roles are designed to provide access to standardized sets of application functionality that are needed by defined groups of users. For detailed information on seeded roles and on defining your own Roles, see the *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

## Logging on to OFSAA Infrastructure

On accessing OFSAAI, the login screen is displayed.



Enter your User ID, password, and click Login. The first time you log in, you will be prompted to change your password.

---

**Note:** You can also click the Change Password link in the login page to change your password later.

---





To change your password, enter your old password, enter your new password, enter your new password again for confirmation, and then click OK. Click Cancel to exit the Change Password screen without changing the password.

### ***OFSAA Infrastructure Password Requirements***

OFSAAI passwords must:

- Contain at least 6 characters
- Contain a combination of numbers and characters without spaces.
- Be different from your 5 previous passwords
- Your User ID is locked after three consecutive unsuccessful logon attempts
- Passwords are case sensitive
- Passwords are displayed as asterisks (\*) to ensure that the password is not revealed to other users

Contact your System Administrator if you encounter any of the following problems:

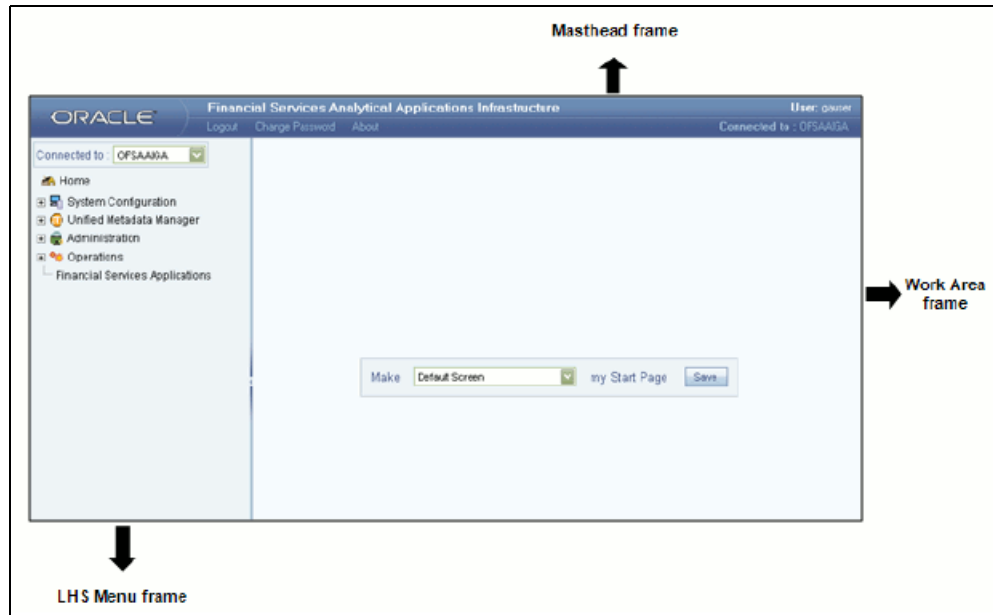
- Your User ID and password are not recognized
- You receive a message while logging in stating that your User ID has been disabled

### ***OFSAA System Administration Home Page***

After logging on, the display will initially default to the OFSAA Infrastructure home page (the Default Screen). Like all main menu pages, the OFSAA Infrastructure home page contains three areas or “frames”:

- Masthead

- Left Hand Side (LHS) Menu
- Work Area



### ***Masthead for the OFSAA Infrastructure Home Page***

The Masthead for the OFSAA Infrastructure home page displays your Username and the name of the Domain to which you are currently connected. It also contains the following links:

- Logout
- Change Password
- About

**Logout:** Clicking on Logout logs you out of the system.

**Change Password :** Clicking on Change Password invokes the Change Password screen.

**About :** Clicking on About displays copyright information.

### ***LHS for the OFSAA Infrastructure Home Page***

On the LHS Menu for the OFSAA Infrastructure home page you may attach your working session to different Information Domains. The LHS Menu also displays submenus that include System Configuration, Unified Metadata Manager, Administration, and Operations.



These topics as well as Information Domains are documented in the *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

### **Work Area for the OFSAA Infrastructure Home Page**

From here, you may reset your default start page. For most users, you will set your start page to Financial Services Applications.

### **Navigating to the Financial Services Applications Home Page**

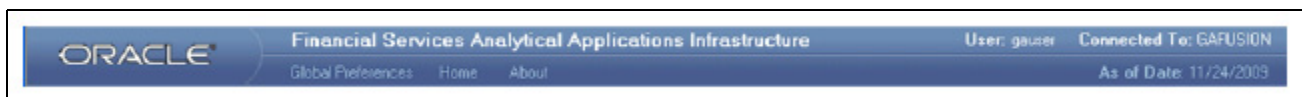
From the OFSAA Infrastructure Home Page, clicking on Financial Services Applications on the LHS Menu opens the Financial Services Applications home page. You may set your start page to default to Financial Services Applications. If you set your start page to default to Financial Services Applications then that is the first page you will see upon logging in.

### **Financial Services Applications Home Page**

#### **Financial Services Applications Masthead**

Similar to the OFSAA Infrastructure masthead, the masthead for the Financial Services Applications home page displays your Username and the name of the Domain to which you are currently connected. In addition, it also displays your current processing As-of-Date and contains links for:

- Global Preferences
- Home
- About



## Global Preferences

Clicking Global Preferences invokes the Global Preferences screen.


### Home

Clicking on Home returns you to the OFSAA Infrastructure home page.

### About

Clicking on About displays copyright information.

### Financial Services Applications LHS Menu

The LHS Menu for Financial Services Applications displays the modules which your firm has installed and to which you have access. The modules that you can access are a function of your Role. You can expand the menu to display submenus with additional options (to view a submenu, click ).

### Work Area

Initially blank, the items displayed within the Work Area for the Financial Services Applications home page is determined by the menu items you select from the LHS.

***OFSAA Infrastructure***

OFSAA Infrastructure includes facilities for creating and maintaining

- Dimensional reference data
- Interest rate & currency exchange rate data
- Process tuning data

OFSAA Infrastructure also includes functionality for building and maintaining rules that may be used by any Oracle Financial Services Analytical Application. These common rule objects include:

- Expressions
- Hierarchies
- Filters

The following chapters document how to navigate the OFSAA user interface to create and maintain dimensional reference data, interest rate & currency exchange rate data, and process tuning data; and to create and maintain OFSAA Infrastructure rules: Expressions, Hierarchies, and Filters:

- [Chapter 3, "OFSAA Administrative Reporting"](#)
- [Chapter 4, "OFSAA Data Entry Forms and Queries"](#)
- [Chapter 5, "OFSAA Attributes"](#)
- [Chapter 6, "OFSAA Dimension Members"](#)
- [Chapter 7, "OFSAA Hierarchies"](#)
- [Chapter 8, "OFSAA Filters"](#)
- [Chapter 9, "OFSAA Expressions"](#)
- [Chapter 12, "OFSAA Rate Management"](#)

The remainder of this chapter documents:

- Overview of Dimensionality in OFSAA
- Overview of OFSAA Rules
  - Hierarchies
  - Filters
  - Expressions

## **Overview of Dimensionality in OFSAA**

Dimensions are used to stratify your business data for processing or reporting purposes. For example, at a minimum General Ledger systems are typically stratified by a General Ledger dimension, an organization or cost center dimension, and a time dimension. Customer account level data will normally have these same dimensions but are often additionally segmented by dimensions such as Product or Customer. You may wish to construct models using a specialized product dimension or other customized dimensions such as geography, cost pool, transaction, activity or channel.

Dimensions are comprised of lists of members. For example, the Organizational Unit dimension might be comprised of Branch #1, Branch #2, Branch #3, etc. Some dimensions may contain tens or even hundreds of thousands of members (e.g., a large organizational or customer dimension) while others may contain as few as two members (e.g., a simple product-type dimension with values of Fixed and Variable).

The member values or “Chart of Accounts” for each dimension may be maintained either internally within the OFSAA Infrastructure or externally.

### **Internally**

OFSAA Infrastructure includes functionality for Adding, Viewing, Editing, or Deleting dimension member values. Which of these functions you can access depends on your logon privilege and on the dimension with which you wish to work. For details, on the creation and maintenance of Dimension Members, see [Chapter 6, "OFSAA Dimension Members"](#).

### **Externally**

You may choose to source and maintain some Charts of Accounts from external systems such as your General Ledger, a Data Warehouse, or from some other external master data management system. See *Oracle Financial Services Analytical Applications Data Model Utilities User Guide* for details on loading dimensional data from external systems.

### **Combination**

You may also choose to maintain some Chart of Account values from an external source while maintaining other Chart of Account values internally.

## **Referential Integrity**

You may have dimension member values defined in your Chart of Accounts that do not exist in your business data, particularly if the dimension member values are used as attribute values (see discussion of attributes below). But you would never want to have an occurrence of a dimension member value in your business data that did not exist in the dimension's Chart of Accounts. When you load business data into OFSAA, you need assurance that there are no dimension member values present in the business data that are not defined in your Chart of Accounts. For details on ensuring referential integrity between your business data and your Chart of Accounts data, see *Installation and Configuration Guide*.

## **Dimension Attributes**

Some OFSAA dimensions support attributes. Dimension attribute values are used to qualify dimension members. For example, a cost center, say Branch #1, might have a Headcount attribute with a value of 32 and a Volume attribute with a value of High. A product dimension member, say 30-year fixed rate jumbo LIBOR mortgage, might

have a Rate attribute with a value of LIBOR, a Size attribute with a value of Jumbo, and a Loan Type attribute of Fixed. A General Ledger account, say Benefit Expense, might have an Account Type attribute with a value of Non-Interest Expense.

Like dimensions, attributes are used to stratify your business data for processing or reporting purposes. Some dimension attributes are seeded within the OFSAA data model and are used extensively in processing. You may define as many user-defined attributes as you need.

OFSAA Infrastructure includes functionality for Adding, Viewing, Editing, or Deleting dimension attributes. Which of these functions you can access depends on your logon privilege and on the dimension with which you wish to work. For details on the creation and maintenance of dimension attributes, see [Chapter 5, "OFSAA Attributes"](#).

Once an attribute has been defined for a dimension, attribute values may be assigned to individual dimension member values. These attribute values may be open-ended (dates, numbers, or strings) or may be restricted to a defined set of members (see further discussion below; also see [Chapter 5, "OFSAA Attributes"](#)).

## **Hierarchies**

Some OFSAA dimensions support hierarchies. Hierarchies may be used to provide sophisticated stratification for either processing or reporting purposes. For example, an organizational hierarchy might start with a Division level containing Western Region, Eastern Region, and Southern Region; the next level down within the hierarchy might be state or county. A product hierarchy might begin with branches for Asset vs. Liability vs. Service products; under the Asset branch, you might define additional branches for Mortgage Lending, Commercial Lending, Consumer Lending, etc. Hierarchies are used extensively in OFSAA models to assign methods to products and to support allocation methodologies.

Like dimension members, dimension hierarchies may be maintained either internally within the OFSAA Infrastructure or externally.

At least one leaf level dimension member is required for hierarchies used in allocations.

### **Internally**

OFSAA Infrastructure includes functionality for Adding, Viewing, Editing, or Deleting hierarchies. Which of these functions you can access depends on your logon privilege and on the hierarchy with which you wish to work. For details, on the creation and maintenance of hierarchies, see [Chapter 7, "OFSAA Hierarchies"](#).

### **Externally**

You may choose to source and maintain some or all of your hierarchies from external systems such as your General Ledger, a Data Warehouse, or from some other external master data management system, such as Oracle Data Relationship Management (DRM). See *Oracle Financial Services Analytical Applications Data Model Utilities User Guide* for details on loading hierarchy data from external systems.

### **Combination**

You may also choose to maintain some hierarchies from an external source while maintaining other hierarchies internally.

## ***Rollup Members for Dimensions Supporting Hierarchies***

For dimensions that support hierarchies, every dimension member value has the property of being either a “leaf” value (child-only or last descendent child value) or a “rollup node” value. For example, if Branch #1, Branch #2, and Branch #3 all belonged to the West rollup point of an organizational hierarchy, then West would also be a dimension member value of the Organizational Unit dimension. Your business data is typically expressed at the “leaf” level.

At least one leaf level dimension member is required for hierarchies used in allocations.

## ***Dimension Types***

OFSAA supports three fundamentally different kinds of dimensions

- Key Processing Dimensions
- Standard Dimensions
- Simple Dimensions

### **Key Processing Dimensions**

- Are accessible as modeling dimensions for all of the OFSAA analytical engines
- Are expressed as columns in nearly all of your business fact tables
- Support both attributes and hierarchies

Metadata for Key Processing Dimensions is stored in four tables:

- A base table (e.g., DIM\_PRODUCTS\_B)
- A translation table (e.g., DIM\_PRODUCTS\_TL)
- An attribute table (e.g., DIM\_PRODUCTS\_ATTR)
- A hierarchy table (e.g., DIM\_PRODUCTS\_HIER)

Base tables store basic information about each dimension member and translation tables store names and descriptions for each dimension member in multiple languages (see *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration Guide* for details on multi-language support). Attribute tables store one or many attribute values for each dimension member. Hierarchy tables store one or more hierarchies for each dimension (you may define as many hierarchies as you wish for any dimension that supports hierarchies).

### **Seeded Key Processing Dimensions**

OFSAA is seeded with 5 Key Processing dimensions:

- Financial Element
- Organizational Unit
- General Ledger Account
- Common Chart of Accounts



- Legal Entity
- Product

Among the 5 seeded Key Processing dimensions, only the Financial Element dimension is pre-populated, but it is only populated in the range between 0 and 10,000 (For more details on the seeded Financial Element dimension see [Appendix F, “Seeded Financial Elements.”](#)). This is a reserved range in which you may not add new members, or edit or delete existing members. You may add user-defined Financial Elements anywhere outside the reserved range.

## User Defined Key Processing Dimensions

See *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration Guide* for details on adding your own user-defined Key Processing dimensions.

## Standard Dimensions

Standard dimensions may support attributes and/or hierarchies depending on how they are configured, but are not used as processing dimensions within any of the analytical application engines.

Like Key Processing Dimensions, metadata for Standard Dimensions is stored in a base table and in a translation table. A Standard Dimension may or may not have an attribute table or a hierarchy table depending on how it is configured. Within the OFSAA modules, Standard dimensions that are defined as having a hierarchy table will support hierarchies and Standard dimensions that are defined as having an attribute table will support attributes. See the *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration Guide* and *Oracle Financial Services Analytical Applications Data Model Data Dictionary* for additional information.

### Seeded Standard Dimensions

OFSAA is seeded with 2 Standard Dimensions

- Country
- Customer

Both dimensions support attributes and hierarchies. Default member values are provided for the Country dimension, but no values are provided for the Customer dimension.

### User Defined Standard Dimensions

See the *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration Guide* for details on adding user-defined Standard dimensions.

## Simple Dimensions

Sometimes referred to as Code dimensions, Simple dimensions are “lists of values” that support neither attributes nor hierarchies. Their 3 key uses are:

- Reserved for use within the analytical application engines
- Stratifying your data for process or report filtering purposes
- Serving as “lists of values” when used as attributes

The OFSAA Data Model comes with more than 150 seeded Simple dimensions. Many of these seeded Simple dimensions are also seeded with dimension members. Unlike Key Processing dimensions and Standard dimensions, Simple dimensions are stored in two tables:

- A code table (e.g., AMRT\_TYPE\_CD)
- An Multi-Language Support or MLS table (e.g., AMRT\_TYPE\_MLS)

Again, see *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration Guide* for details on multi-language support.

### **Reserved Simple Dimensions**

Accrual Basis Code is an example of a seeded Simple dimension that is used extensively by the application modules. It is seeded with a small list of values (such as 30/360, Actual/360, Actual/Actual, etc.) and its value set may not be modified.

Amortization Type Code is another example of a reserved Simple Dimension. Amortization Type Code is populated with reserved values that the Cash Flow engine interprets, and its value set may not be modified. Many other Simple Dimensions are reserved and may not be modified.

### **Simple Dimensions Used for Processing or Reporting Purposes**

This kind of Simple Dimension cannot be used directly by the application engines, but they are accessible to OFSAA Filter rules (for more information on Filter rules, see [Chapter 8, "OFSAA Filters"](#)).

Credit Rating is an example of a seeded Simple Dimension that may be used for processing or reporting purposes. You may, for example, wish to allocate risk equity to individual instruments as a function of Credit Rating. The seeded values for Credit Rating are Moody's credit ratings, but you are free to completely redefine the membership of this dimension. Geographic Location is another example. Geographic Location is pre-populated with US States and Canadian provinces, but you are free to completely redefine the list of geographic locations.

### **Simple Dimensions Used as Attribute Domains**

Both seeded and user-defined attributes often utilize a Simple Dimension to provide a list of values. For example, you might create an Adjustable Type attribute on your Product dimension and you might want this attribute to be limited to the values of Fixed, Variable, or Administered. In this case, you might first build a Simple Dimension called Type whose dimension members were limited to three values: Fixed, Variable, or Administered. Next you could assign the Type dimension to be the domain of your new Adjustable Type attribute.

See *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration Guide* and *Oracle Financial Services Analytical Applications Data Model Data Dictionary* for additional information on which Simple Dimensions you may extend or modify and on how to add your own Simple Dimensions to the data model.

## **Overview of Attributes**

OFSAA Infrastructure includes functionality for Adding, Viewing, Editing, or Deleting dimension attributes. Which of these functions you can access depends on your logon privilege and on the dimension with which you wish to work. For details, on the creation and maintenance of Dimension Attributes, see [Chapter 5, "OFSAA Attributes"](#).

## Attributes

The seeded OFSAA Key Processing Dimensions have a number of seeded or seeded & required attributes. For example, the Common Chart of Accounts dimension includes an Account Type attribute and an Accrual Basis attribute; the Financial Element dimension includes an Aggregation Method attribute. Some of these attributes have important implications in processing within OFSAA.

You may add Attributes to any Key Processing Dimension or to any Standard Dimension that supports Attributes (see sections above describing Key Processing Dimensions and Standard Dimensions).

One of the important properties of an Attribute is its Data Type. OFSAA Infrastructure supports Dimension Attribute Data Types of:

- Date, String, or Number
- Dimension

## Unbounded & Bounded Attribute Data Types

Dimension attribute data types may be either bounded or unbounded. The Date, String, and Number type are unbounded, i.e., there is no predetermined set of attribute values. A Date attribute value may be any valid date, a String attribute value may be any string up to 30 characters in length, and a Number attribute value may be any number. Examples of these kinds of attribute data types might include:

- Cost Center Open Date (a Date attribute on the Organizational Unit dimension)
- Product Memo (a String attribute on the Product dimension)
- Cost Center Headcount (a Number attribute on the Organizational Unit dimension)

Unlike attributes having the Date, String, or Number data type, attributes whose data type is Dimension have defined sets of values. For example, the Product dimension has a seeded attribute called Accrual Basis the data type of which is Dimension. The Accrual Basis dimension is seeded with 8 values. Accrual Basis is a restricted dimension (you may View Accrual Basis dimension members but you may delete them or add new accrual bases), but whenever you assign an Accrual Basis attribute value to one of your Product dimension member values, you must select from the list of 8 available Accrual Bases.

## Overview of OFSAA Rules

OFS Analytical Application models are constructed using combinations of OFSAA rules. For example:

- An Allocation rule might utilize a Hierarchy Filter rule, an Expression rule, and a Static Table Driver rule
- A Transfer Pricing rule might utilize a Hierarchy rule, a Data Element Filter rule, and a Prepayment rule
- A Data Element Filter rule might utilize an Expression rule
- A Group Filter rule always incorporates one or more underlying Data Element Filter rules
- A Hierarchy Filter rules always utilize an underlying Hierarchy rule

The advantage of constructing models from discrete rule components is that individual rules may be re-used dozens or even hundreds of times. For example, if you wish to build a large series of allocation rules which operate on non-interest expense data in the Western region, you may build filters for (1) non-interest expense and (2) the Western region and then utilize these filters in every allocation rule. As another example, if you wanted to assign the same

transfer pricing method and assumptions to all mortgage products, you could build a Hierarchy on the product dimension containing a Mortgage Products rollup point and assign your method and assumptions at the rollup level. You could subsequently utilize the same Hierarchy rule for allocation processing or reporting purposes. Many rule types are specific to individual analytical applications (e.g., Transfer Pricing rules, Asset/Liability Management rules, Allocation rules, etc.) while OFSAA Infrastructure rules are shared across all OFS applications. Rule types that are shared across all applications include:

- Hierarchies
- Hierarchy Filters
- Data Element Filters
- Group Filters
- Expressions

Dimensions, dimension members, attributes, and currency exchange rates are also shared across all applications.

### ***Dependencies***

OFSAA rules often depend on other rules for some or all of their data or usefulness. For example, an Allocation rule may depend on a Data Element Filter rule to select and focus data in a meaningful way. If that Data Element Filter is deleted, then the Allocation rule becomes unusable. OFSAA applications track such dependencies and ensure that users may not delete rules that are consumed by other rules. Additionally, rule summary screens that present lists of rules that you have defined, offer functionality for you to generate reports on rule interdependency.

### ***OFSAA Filters***

OFSAA Infrastructure supports four kinds of filtering:

- Data Element Filters
- Group Filters
- Hierarchy Filters
- Attribute Filters

#### ***Data Element Filters***

Data Element Filters are used to select and focus data. For example, you may transfer price some financial instruments at their source and may therefore wish to ensure that you do not overwrite the transfer rates for any instruments that already have a non-zero transfer rate. In such a case, you could construct a Data Element Filter to isolate only those instruments having a non-zero transfer rate. As another example, you may wish to allocate a charge to demand deposit accounts having small balances. In this case, you could construct a Data Element Filter to focus on just those instrument records having balances less than your chosen threshold.

#### ***Group Filters***

Group Filters are used to join together one or more Data Element Filters. For example, if you built one Data Element Filter that focused on balances less than 10,000 and a second Data Element Filter that focused on accounts

whose origination date was prior to 2009, a Group Filter built on the two Data Element Filters would focus on accounts with balances less than 10,000 AND whose origination dates were prior to 2009.

### **Hierarchy Filters**

Hierarchy Filters allow you to focus on data that rolls up to one or more points within a chosen hierarchy. You may also focus on “leaf” level data or a combination of “leaf” level and rollup levels. For example, if your organization had a Western, Northern, Southern, and Eastern division, you could construct a Hierarchy Filter that focused on data belonging to branch 1, 2, or 7 within the Eastern division plus any data belonging to the Northern division.

### **Attribute Filters**

Attribute Filters allow you to filter on any dimension attribute values including standard, seeded attributes such as Account Type or Accrual Basis as well as on user defined attributes.

## **OFSAA Expressions**

Expressions represent formulas for combining data from your data model (e.g., Column A divided by Column B times Column C). The results of Expressions may be invoked within a number of OFSAA rule types (e.g., Data Element Filters, Allocation rules, etc). For details on building and maintaining Expressions, see [Chapter 9, "OFSAA Expressions"](#).

## **Folders**

One of the fundamental properties of any OFSAA rule is the Folder in which it is stored. Your System Administrator will configure your user ID to have access to one or more folders in which you can store your rules. Your access rights will likely be similar to co-workers who have a similar business role. For details on security management and defining Segment Maintenance in OFSAA, see *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

Storing rules in folders gives you the means to segregate your rules for functional or security considerations. For example, you may wish to:

- Set up folders to store Transfer Pricing rules that are separate from folders you have built for Profitability Management rules and that are also separate from folders you have built for Asset/Liability Management rules

- Establish some folders for rules you are in the process of developing and testing and other folders for fully tested, production ready rules

- Segregate some rules that are fundamental to the definition of other rules (e.g., some Data Element Filters may be used in dozens or hundreds of other rules) in order to ensure that access to those fundamental rules is highly controlled

- Segregate your hierarchies, which are also frequently fundamental to OFSAA models, into a restricted-access folder

### **Visibility of Rules**

Every navigation path to view a rule in OFSAA begins with a Summary screen. Within the Summary screen for each rule type, a user may browse through all of the rules that are stored within Folders to which the user has access. From each Summary screen, users may be granted the ability to Add, Edit, View, or Delete rules.

When building rules in an OFS Analytical Application, you frequently call upon other OFSAA rules (e.g., a Data Element Filter nested within an Allocation rule or a Hierarchy utilized within a Transfer Pricing rule). When nesting a child rule within a parent rule you must specify the child rule by its Name and Folder (where both Name and Folder are presented in drop-down list boxes). In this context, the Folder drop-down list box for every user will include every Folder in the Information Domain regardless of his Folder access rights. This means that a user may invoke a child rule from any Folder even if he cannot access that child rule directly through its Summary screen.

Additionally, whenever a user nests a parent rule within a child rule, the OFSAA user interface provides the user the ability to view a read-only version of the underlying child rule.

## **Access Type**

Another fundamental property of every OFSAA rule is its Access Type.

Access Type may be set to either Read/Write or Read Only. Only the “creator” or “owner” of a rule may modify the rule if it is set to Read Only. The user ID under which a rule is initially built is the owner of a rule. You may modify a rule that you do not own provided that:

- The rule has been stored with the Read/Write access type
- You have been granted Edit privilege for the rule's type
- You have been granted access to the Folder in which the rule is stored

Again, for details on security management in OFSAA, see *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration Guide*.

## **Global Preferences**

Clicking Global Preferences invokes the Global Preferences screen. Global Preferences items are used to configure your user interface. Here, you set your default application language and date format, and your desired Pagination Records. Pagination Records determine how many rows will be displayed on summary screens. If you select Pagination Records to be 25 records, then any screen displaying results in a tabular format will display a maximum of 25 records. To see the next set of 25 records, you would use the Next VCR button

Functional Currency and Signage are installation-time parameters that may not be reset in Global Preferences.

Functional Currency is defined as “the currency of the primary economic environment in which an entity conducts its business”. For details on signage, refer [Appendix G, “Debit and Credit Conventions”](#).

Normally, you will set preferences for yourself, but if you are logged in as an Administrator, you may set Global Preferences for All Users. When setting preferences for All Users, you may restrict the ability of non-Administrators to change a Global Preference item by deselecting the “Is Editable” checkbox. If a preference item is defined as “not editable”, a user who is not an Administrator will inherit the value of the preference item that his Administrator has set for him and he will not have the ability to change it for himself.

Global Preferences

▼ Display

Show Preferences for: PFTUSER ▼

▼ Display

Property Name	Property Value	Is Editable
Default Application Language	US-English ▼	<input checked="" type="checkbox"/>
Date Format	MM/dd/yyyy ▼	<input checked="" type="checkbox"/>
Pagination Records	20	<input checked="" type="checkbox"/>

▼ Ledger

Property Name	Property Value	Is Editable
Functional Currency	US Dollar	<input type="checkbox"/>
Signage	Natural Signage	<input type="checkbox"/>

Apply Cancel Reset to Default

## System Configuration

### Application

Starting with Release 8, once an application pack is installed, you can use only the Production or Sandbox information domain, created during the installation process. Though there is an option to create a new Information Domain, there is no menu to work with the frameworks on this new information domain. This new information domain then created acts only as a Sandbox Infodom. The Create New Application feature allows you (System Administrator) to create a new Application other than the standard OFSAA Applications and associate the standard/default platform framework menu with it, thereby enabling the new application for usage. The standard platform framework menu is seeded and rendered.

For more information, refer to *System Configuration* section in OFSAAI User Guide.





**CHAPTER 3*****OFSAA Administrative Reporting******Introduction***

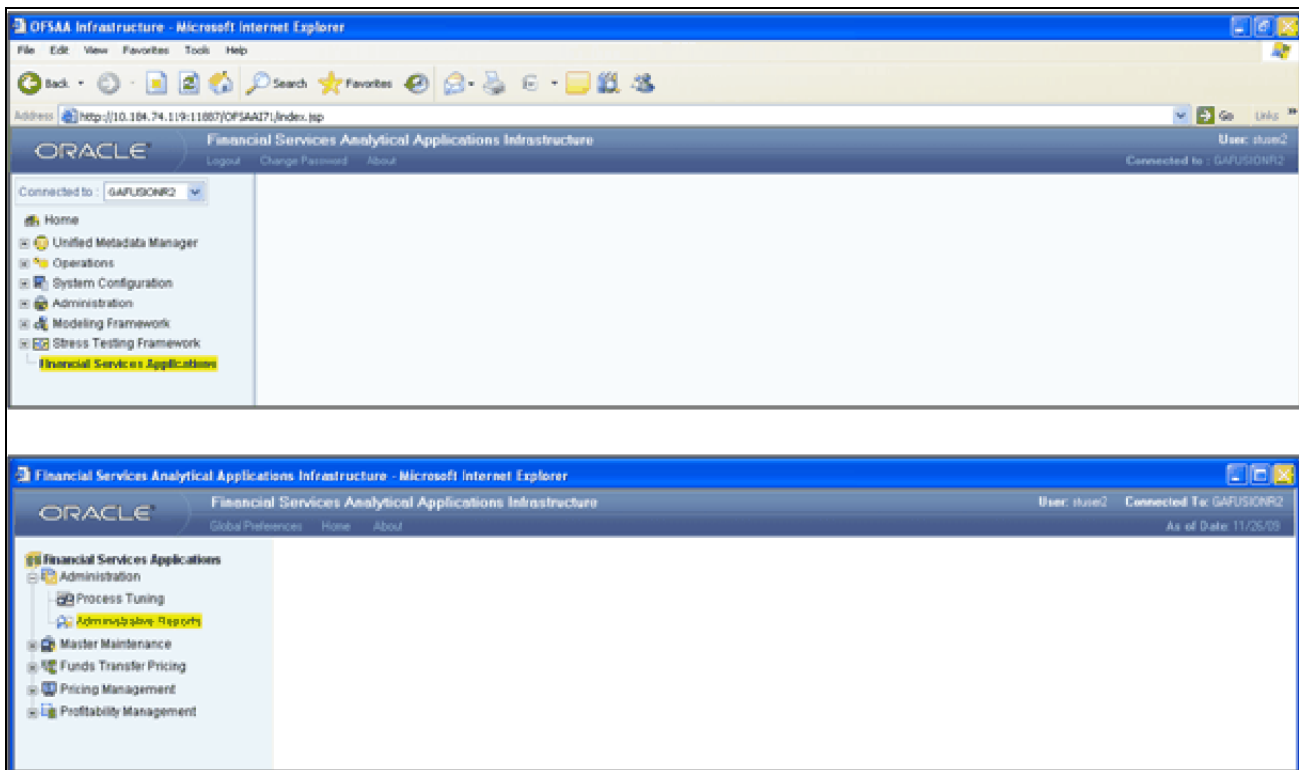
Administrative Reports provide reporting functionality for the Administrators of Oracle Financial Services Analytical Applications (OFSAA) to monitor, analyze, and correct data.

The System Administrator or users with Administrative privileges can view the Administrative Reports.

Logging in to the Administrative Reports

**Procedure**

1. Click, Financial Services Applications >Administration >Administrative Reports. Refer to the following screenshots.



2. The Administrative Reports - Admin Home section is displayed.



**Interactive Dashboards in Administrative Reports**

In the Administrative Reports window, there are four dashboard links:

- Admin Home
- Data Quality
- Object Registration
- Security

**Admin Home**

The Admin Home section is displayed by default when you login. If you are in any other section, click the Admin Home link in the top of the Administrative Reports header, to go to the Admin Home section. Admin Home displays the summary of reports from the other three links.

## Access Permissions

Following are the steps to access permissions to include Defi Forms for the Admin BI Buckets entries.

1. Login to the Oracle Financial Services Analytical Applications Infrastructure
2. Go to Common Object Maintenance > Data Entry Forms and Queries > Forms Designer
3. Click the Assign Rights Radio button
4. Select Admin BI in the Available Applications drop-down list box
5. Select the 4 UI forms (one at a time) in the Available Forms drop-down list (For example, Bucket definition of Book Balance)
6. Click Next
7. Select the users from the Available User List
8. Select the relevant check boxes from View, Add, Edit, Delete, and All above.
9. Click Save Access Rights. You will get the message that user is mapped successfully.
10. Click Back to Forms Designer.
11. Repeat the steps 4 to 11 for other 3 forms (Bucket Definition Maturity Date, Bucket Definition net Rate, Bucket Definition Origination Date)

## Data Quality

To go to the Data Quality reports, click Data Quality link at the top of the Administrative Reports. The following window is displayed.

The screenshot displays the Oracle Data Quality dashboard with the following reports:

**Current Net Rate/Book Balance**  
 Time run: 12/15/2009 2:20:51 PM

Current Net Rate Bucket	Current Book Balance	Record Count
1 - 5 - 9999	270,000,000.00	6
2 - 6 - 9999	22,470,000,000.00	281
3 - 10 - 14 9999	0.00	0
4 - 15 - 4 9999	100,000,000.00	1
<b>Grand Total</b>	<b>22,640,000,000.00</b>	<b>288</b>

**Data Stratification - Current Book Balance**  
 Time run: 12/15/2009 2:20:51 PM

Current Book Balance Bucket	Current Book Balance	Record Count
0	0.00	0
Others	22,640,000,000.00	288
<b>Grand Total</b>	<b>22,640,000,000.00</b>	<b>288</b>

**Origination/Maturity Date**  
 Time run: 12/15/2009 2:20:51 PM

Origination Date Bucket	Current Book Balance	Record Count
Others	22,640,000,000.00	288
<b>Grand Total</b>	<b>22,640,000,000.00</b>	<b>288</b>

**Data Stratification - Maturity Date**  
 Time run: 12/15/2009 2:20:51 PM

Maturity Date Bucket	Current Book Balance	Record Count
1 - 1-Jan-975 - 31-Dec-999	22,700,000,000.00	273
2 - 1-Jan-2000 - 31-Dec-2014	60,000,000.00	2
<b>Grand Total</b>	<b>22,640,000,000.00</b>	<b>288</b>

The Data Quality reports help you to validate the quality of the underlying fact data. By going through the reports, you can identify data discrepancies, missing data, incorrect data, and so on and then undertake the necessary steps to rectify any issues. These data reports can be customized to fit your needs by defining suitable bucket range values as described above.

The four data buckets available are:

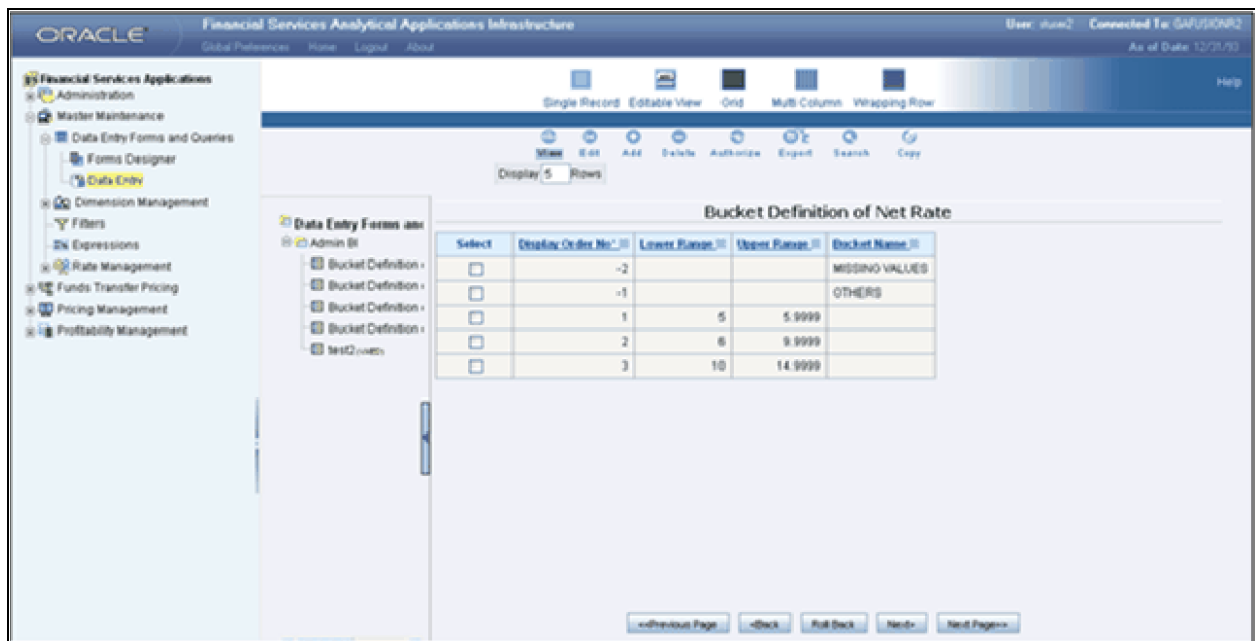
- Book Balance Bucket Definition
- Maturity Date Bucket Definition
- Net Rate Bucket Definition
- Origination Date Bucket Definition

These values can be defined in the following section:

Common Object Maintenance > Data Entry Forms and Queries > Data Entry.

Refer the following screenshot.

- The seeded values are Display Order No => -1 and -2. You should not delete or edit these seeded values.
- Make sure there is no row with 'Display order No.' column with value 0 (zero). In case of AIX environment, display order No. might be generated with '0' (zero value). Delete this row using 'Delete option'.



## Data Quality Dashboard

Data Quality Dashboard page contains eight tabs.

### Data Quality Tab

This section contains a prompt to filter the data based on the value selected from the drop-down list box.

You can select a value from the drop-down list. Based on the selected values, the data is filtered.

There are four reports in this window.

1. Data Stratification - Current Net Rate

The current net rate details for each bucket range value of the Instruments are displayed here.

Current Net Rate/Book Balance		
Data Stratification - Current Net Rate		
Time run: 12/11/2009 4:16:03 PM		
Current Net Rate Bucket	Current Book Balance	Record Count
1 . 5 - 5.9999	270,000,000.00	6
2 . 6 - 9.9999	22,470,000,000.00	251
3 . 10 - 14.9999	0.00	22
5 . 0 - 4.9999	100,000,000.00	1
<b>Grand Total</b>	<b>22,840,000,000.00</b>	<b>280</b>

[Refresh](#) - [Print](#) - [Download](#)

To view a detailed report, click the bucket range for which you need the detailed report. The detailed report for the selected bucket range is displayed.

Data Quality										
Detailed Report - Current Net Rate										
Time run: 12/11/2009 4:20:36 PM										
ID Number	Identity Code	Account Number	Product ID	Org unit ID	GL Account ID	Common COA ID	Origination Date	Maturity Date	Cur Net Rate	Current Book Balance
10033	19990331	L-1000000032	135	1	300120	30012	31-Mar-1996	01-Apr-1996	4.5000	100,000,000.00
<b>Grand Total</b>										<b>100,000,000.00</b>

[Return](#) - [Refresh](#) - [Print](#) - [Download](#)

2. Data Stratification - Current Book Balance

The current book balance details for each bucket range value for the Instruments are displayed here.

Data Stratification - Current Book Balance		
Time run: 12/15/2009 4:05:46 PM		
Current Book Balance Bucket	Current Book Balance	Record Count
1 . 101 - 1000	6,499.00	7
2 . 1001 - 10000	22,146.00	6
3 . 10001 - 100000	4,768,629.00	81
4 . 100001 - 1000000	332,594,117.00	660
Others	88,772,027,473.00	1619
<b>Grand Total</b>	<b>89,110,218,864.00</b>	<b>2372</b>

[Refresh](#) - [Print](#) - [Download](#)

To view a detailed report, click the bucket range for which you need the detailed report. The detailed report for the selected bucket range is displayed.

**Data Quality**  
**Detailed Report - Current Book Balance**  
 Time run: 12/11/2009 4:36:27 PM

ID Number	Identity Code	Account Number	Product ID	Org unit ID	GL Account ID	Common COA ID	Origination Date	Maturity Date	Cur Met Rate	Current Book Balance
10033	19900331	L-1000000032	135	1	300120	30012	31-Mar-1996	01-Apr-1996	4.5000	100,000,000.00
10059	19900331	L-22	135	0	14710	3002	31-Aug-1993	31-Aug-1998	5.0000	50,000,000.00
10302	19900331	L-7	135	0	14710	3001	31-Aug-1993	31-Aug-1998	5.0000	40,000,000.00
10058	19900331	L-21	135	0	14710	3002	31-Jul-1993	31-Jul-1998	5.5000	50,000,000.00
10060	19900331	L-23	135	0	14710	3002	30-Sep-1993	30-Sep-1998	5.5000	50,000,000.00
10301	19900331	L-8	135	0	14710	3001	31-Jul-1993	31-Jul-1998	5.5000	40,000,000.00
10303	19900331	L-5	135	0	14710	3001	30-Sep-1993	30-Sep-1998	5.5000	40,000,000.00
10019	19900331	L-9	135	0	14710	3001	31-Oct-1993	31-Oct-1998	6.0000	40,000,000.00
10057	19900331	L-20	135	0	14710	3002	30-Jun-1993	30-Jun-1998	6.0000	45,000,000.00
10061	19900331	L-24	135	0	14710	3002	31-Oct-1993	31-Oct-1998	6.0000	55,000,000.00
10283	19900331	L-5	135	0	14710	3001	30-Jun-1993	30-Jun-1998	6.0000	40,000,000.00
10017	19900331	L-10	135	0	14710	3001	30-Nov-1993	30-Nov-1998	6.5000	40,000,000.00
10055	19900331	L-19	135	0	14710	3002	30-May-1993	31-May-1998	6.5000	45,000,000.00
10062	19900331	L-25	135	0	14710	3002	30-Nov-1993	30-Nov-1998	6.5000	55,000,000.00
10068	19900331	L-4	135	0	14710	3001	30-May-1993	31-May-1998	6.5000	40,000,000.00

Records 1 - 15

[Return](#) - [Refresh](#) - [Print](#) - [Download](#)

3. Data Stratification - Origination Date

The origination date details for each bucket range value for the Instruments are displayed here.

**Origination/Maturity Date**  
**Data Stratification - Origination Date**  
 Time run: 12/11/2009 4:16:03 PM

Origination Date Bucket	Current Book Balance	Record Count
Others	22,840,000,000.00	280
<b>Grand Total</b>	<b>22,840,000,000.00</b>	<b>280</b>

[Refresh](#) - [Print](#) - [Download](#)

To view a detailed report, click the bucket range for which you need the detailed report. The detailed report for the selected bucket range is displayed.

**Data Quality**  
**Detailed Report - Origination Date**  
 Time run: 12/11/2009 4:39:25 PM

ID Number	Identity Code	Account Number	Product ID	Org unit ID	GL Account ID	Common COA ID	Origination Date	Maturity Date	Current Book Balance
10016	19900331	L-1	135	0	14710	3001	31-Mar-1993	31-Mar-1998	40,000,000.00
10017	19900331	L-10	135	0	14710	3001	30-Nov-1993	30-Nov-1998	40,000,000.00
10018	19900331	L-11	135	0	14710	3001	31-Dec-1993	31-Dec-1998	40,000,000.00
10019	19900331	L-9	135	0	14710	3001	31-Oct-1993	31-Oct-1998	40,000,000.00
10028	19900331	L-3	135	0	14710	3001	30-Apr-1993	30-Apr-2003	40,000,000.00
10029	19900331	L-10000000002	135	1	14710	700000002	31-Jan-1996	31-Jan-1999	100,000,000.00
10030	19900331	L-10000000018	135	1	14710	100000008	31-Jan-1996	31-Jan-1997	100,000,000.00
10031	19900331	L-10000000019	135	1	14710	100000009	31-Mar-1996	31-Jan-1997	100,000,000.00
10032	19900331	L-10000000020	135	1	14710	100000009	01-Apr-1996	31-Jan-1997	100,000,000.00
10033	19900331	L-10000000032	135	1	300120	30012	31-Mar-1996	01-Apr-1996	100,000,000.00
10034	19900331	L-10000000066	135	1	14710	800017	31-Jan-1996	31-Jan-1997	100,000,000.00
10035	19900331	L-10000000067	135	1	14710	800037	31-Jan-1996	31-Jan-1997	100,000,000.00
10036	19900331	L-10000000068	135	1	14710	800038	31-Jan-1996	31-Jan-1997	100,000,000.00
10037	19900331	L-10000000069	135	1	14710	100020	31-Jan-1996	31-Jan-1997	100,000,000.00
10038	19900331	L-10000000070	135	1	14710	100021	31-Jan-1996	31-Jan-1997	100,000,000.00

Records 1 - 15  
[Return](#) - [Refresh](#) - [Print](#) - [Download](#)

4. Data Stratification - Maturity Date

The Maturity Date details for each bucket range value for the Instruments are displayed here.

**Data Stratification - Maturity Date**  
 Time run: 12/11/2009 4:44:38 PM

Maturity Date Bucket	Current Book Balance	Record Count
1 . 1-Jan-1976 - 31-Dec-1999	22,760,000,000.00	276
2 . 1-Jan-2000 - 31-Dec-2015	80,000,000.00	2
<b>Grand Total</b>	<b>22,840,000,000.00</b>	<b>280</b>

[Refresh](#) - [Print](#) - [Download](#)

To view a detailed report, click the bucket range for which you need the detailed report. The detailed report for the selected bucket range is displayed.

**Data Quality**  
**Detailed Report - Maturity Date**  
 Time run: 12/11/2009 4:46:48 PM

ID Number	Identity Code	Account Number	Product ID	Org unit ID	GL Account ID	Common COA ID	Origination Date	Maturity Date	Current Book Balance
10028	19900331	L-3	135	0	14710	3001	30-Apr-1993	30-Apr-2003	40,000,000.00
10054	19900331	L-18	135	0	14710	3002	30-Apr-1993	30-Apr-2003	40,000,000.00
<b>Grand Total</b>									<b>80,000,000.00</b>

[Return](#) - [Refresh](#) - [Print](#) - [Download](#)

**Dimension Wise**

You can select the values to filter the data by selecting the values from the top drop-down list. The following four reports are displayed here:

- Amortization Type / Adjust Type/ Reprice Freq
- Payment Frequency/ Remaining Payments
- Product
- General Ledger Account

The screenshot displays the Oracle Interactive Dashboards interface. At the top, there are navigation tabs: Data Quality, Dimension Wise, FTP Rates, Invalid Leaves, Invalid Codes, Detailed Reports, Ledger Stat, and Product Instrument Play. Below these are filter fields for Instrument Name (Loans), As of Date (30/11/09 12:00:00 AM), Currency Name (All Choices), Line Item Name (All Choices), and Balance Type (All Choices). The main content area is divided into four sections:

- Amortization Type/Adjust Type/Reprice Freq:** A table with columns: Line Item, Asset Type Code, Adj Type Code, Reprice Freq, Current Book Balance, and Record Count. It shows data for Certificates of Deposits.
- Payment Frequency/Remaining Payments:** A table with columns: Payment Multiplier, Payment Term, Remaining Payments, Current Book Balance, and Record Count. It shows various payment terms and their remaining counts.
- Line Items/Natural Account:** A summary table with columns: Line Item, Current Book Balance, and Record Count. It shows a total of 22,846,000,000.00 for 303 records.
- Natural Account:** A table with columns: Natural Account, Current Book Balance, and Record Count. It lists various natural accounts and their balances.

**FTP Rates**

You can filter the data by selecting the values from the drop-down list. The Transfer Pricing Stratification Reports show a summary of the FTP results for all instrument tables.

FTP Rate Detail					
Transfer Pricing Stratification Report for All Products					
Time run: 12/11/2009 4:15:47 PM					
Product Name	Record Count	Avg Balance	WAvg Transfer Rate	WAvg Note Rate	WAvg Gross Spread
Certificate of Deposits	280	1,200,000,000.00	0.0000%	6.9000%	0.0000%

[Refresh](#) - [Print](#) - [Download](#)



You can click the product name to go to the account level - detail report as shown in the following screenshot.

Product Name	ID Number	Identity Code	Account Number	Product ID	Org unit ID	Common COA ID	GL Account ID	Origination Date	Maturity Date	Avg Balance	VI Avg Transfer Rate	VI Avg Note Rate	WI Avg Gross Spread
	10016	19960331	L-1	135	0	3001	14710	31-Mar-1993	31-Mar-1998	40,000,000.00	0.0000%	8.5000%	0.0000%
	10017	19960331	L-10	135	0	3001	14710	30-Nov-1993	30-Nov-1998	40,000,000.00	0.0000%	8.5000%	0.0000%
	10018	19960331	L-11	135	0	3001	14710	31-Dec-1993	31-Dec-1998	40,000,000.00	0.0000%	7.0000%	0.0000%
	10019	19960331	L-8	135	0	3001	14710	31-Oct-1993	31-Oct-1998	40,000,000.00	0.0000%	8.5000%	0.0000%
	10020	19960331	L-3	135	0	3001	14710	30-Apr-1993	30-Apr-2003	40,000,000.00	0.0000%	7.5000%	0.0000%
	10029	19960331	L-1000000002	135	1	100000002	14710	31-Jan-1996	31-Jan-1999	0.00	0.0000%	0.5000%	0.0000%
	10030	19960331	L-1000000018	135	1	100000008	14710	31-Jan-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10031	19960331	L-1000000019	135	1	100000008	14710	31-Mar-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10032	19960331	L-1000000020	135	1	100000008	14710	01-Apr-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10033	19960331	L-1000000032	135	1	30012	300120	31-Mar-1996	01-Apr-1996	0.00	0.0000%	0.5000%	0.0000%
	10034	19960331	L-1000000066	135	1	800017	14710	31-Jan-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10035	19960331	L-1000000067	135	1	800037	14710	31-Jan-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
Certificate of Deposits	10036	19960331	L-1000000068	135	1	800038	14710	31-Jan-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10037	19960331	L-1000000069	135	1	100020	14710	31-Jan-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10038	19960331	L-1000000070	135	1	100021	14710	31-Jan-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10039	19960331	L-1000000071	135	1	820001	14710	31-Jan-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10040	19960331	L-1000000072	135	1	820002	14710	31-Jan-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10041	19960331	L-1000000073	135	1	800043	14710	31-Jan-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10042	19960331	L-1000000074	135	1	800044	14710	31-Jan-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10043	19960331	L-1000000075	135	1	800045	14710	31-Jan-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10044	19960331	L-1000000076	135	1	800046	14710	31-Jan-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10045	19960331	L-1000000077	135	1	100004	14710	31-Jan-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10046	19960331	L-1000000078	135	1	800048	14710	31-Jan-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10047	19960331	L-114803194	135	1	800001	14710	31-Jan-1996	31-Jan-1997	0.00	0.0000%	0.5000%	0.0000%
	10048	19960331	L-12	135	0	3001	14710	31-Jan-1994	31-Jan-1999	40,000,000.00	0.0000%	7.5000%	0.0000%

### Invalid Leaves

Contains reports for invalid leaves based on the list of Dimension Members compared with the Instrument data. You can filter the data selecting the drop-down list provided at the top. The following four reports are displayed here:

- Invalid Products
- Invalid Org Unit
- Invalid Common COA
- Invalid GL Account

### Invalid Codes

Contains the reports for invalid Codes based on the dimensions compared with Instruments. You can filter the data selecting the drop-down list provided at the top. The following seven reports are displayed here:

- Accrual Basis Code
- Amortization Code
- Adjustable Type Code
- Compound Basis Code
- Behaviour Type Code
- Behaviour Sub Type Code
- Net Margin Code

### Detailed Reports

Contains the detailed report for a particular Instrument based on the selection in the drop-down.

Identity Code	ID Number	Account Number	Product ID	Org unit ID	Common COA ID	GL Account ID	Origination Date	Maturity Date	Current Book Balance	Current Par Balance
	10016	L-1		135	0	3001	14710 31-Mar-1993	31-Mar-1998	40,000,000.00	40,000,000.00
	10017	L-10		135	0	3001	14710 30-Nov-1993	30-Nov-1998	40,000,000.00	40,000,000.00
	10018	L-11		135	0	3001	14710 31-Dec-1993	31-Dec-1998	40,000,000.00	40,000,000.00
	10019	L-8		135	0	3001	14710 31-Oct-1993	31-Oct-1998	40,000,000.00	40,000,000.00
	10020	L-3		135	0	3001	14710 30-Apr-1993	30-Apr-2003	40,000,000.00	40,000,000.00
	10029	L-1000000002		135	1	100000002	14710 31-Jan-1998	31-Jan-1999	100,000,000.00	100,000,000.00
	10030	L-1000000018		135	1	100000009	14710 31-Jan-1998	31-Jan-1997	100,000,000.00	100,000,000.00
19980331	10031	L-1000000018		135	1	100000009	14710 31-Mar-1995	31-Jan-1997	100,000,000.00	100,000,000.00
	10032	L-1000000020		135	1	100000009	14710 31-Apr-1995	31-Jan-1997	100,000,000.00	100,000,000.00
	10033	L-1000000032		135	1	30012	300120 31-Mar-1995	31-Apr-1995	100,000,000.00	100,000,000.00
	10034	L-1000000086		135	1	800017	14710 31-Jan-1995	31-Jan-1997	100,000,000.00	100,000,000.00
	10035	L-1000000087		135	1	800037	14710 31-Jan-1995	31-Jan-1997	100,000,000.00	100,000,000.00
	10036	L-1000000088		135	1	800038	14710 31-Jan-1995	31-Jan-1997	100,000,000.00	100,000,000.00
	10037	L-1000000089		135	1	100020	14710 31-Jan-1995	31-Jan-1997	100,000,000.00	100,000,000.00
	10038	L-1000000070		135	1	100021	14710 31-Jan-1995	31-Jan-1997	100,000,000.00	100,000,000.00

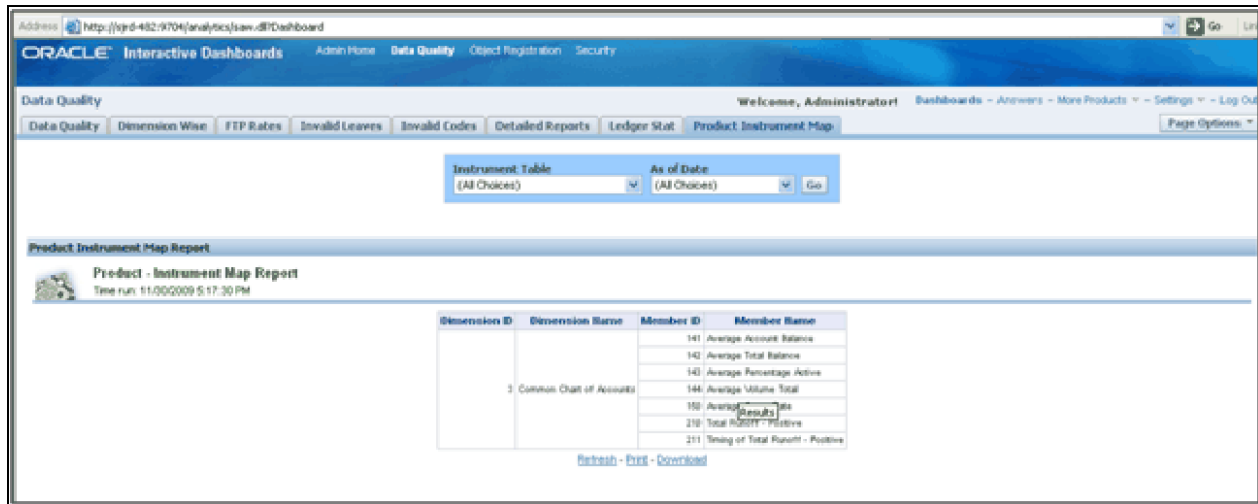
### Ledger Stat

Contains the reports for missing dimension members compared with Ledger Stat data values. You can filter the data by selecting from the drop-down list provided at the top. The following five reports are displayed here:

- Missing Products
- Missing Org Units
- Missing Common COA
- Missing GL Account
- Missing Financial Elements

### Product Instrument Map

Contains the Product Instrument Map report. You can filter the data based the drop-down selection. The Product Instrument Map data is used within both ALM and FTP Process rules. These processes allow users to select data to be included in the process based on a Product Hierarchy. The engines need to reference this data to determine which instrument tables to include in the process.



## Object Registration

To go to the Object Registration reports, click Object Registration at the top of the Administrative Reports page. The details of the registered objects are present in the reports. You can obtain the data type of each object from this report. There are two tabs in this section.

### 1. Object Registration Tab

The details of the classification and number of registered objects are obtained here.

The screenshot shows the 'Registered Objects Summary' report. It features a table with four columns: 'Classification', 'Classification Code', 'Classification Description', and 'No of Registered Objects'. The table lists various classifications such as 'Instrument', 'Ledger Stat', 'Portfolio', 'MLS Descriptions Reserved', etc., along with their respective codes and descriptions. At the bottom of the table, there are links for 'Refresh', 'Print', and 'Download'.

Classification	Classification Code	Classification Description	No of Registered Objects
Instrument	20	Super-type for all instrument tables	21
Ledger Stat	50	Ledger Stat	1
Portfolio	100	Portfolio of instrument tables	8
MLS Descriptions Reserved	150	MLS Description table for Reserved Code values.	8
Codes Reserved (base tbl)	190	Base table for Reserved Code values.	11
TP Cash Flow	200	Classification to identify tables that qualify for TP Cash Flow methods.	21
TP Non-Cash Flow	210	Classification to identify tables that qualify for TP Non-Cash Flow methods.	21
Codes User Defined (base tbl)	290	Base table for User Defined Code columns.	17
MLS Descriptions User Defined	295	MLS Description table for User Defined Code columns.	16
Transaction Profitability	300	Classification to identify transaction summary tables that will be used for profitability.	20
Instrument Profitability	310	Classification to identify instrument tables that will be used for profitability.	20
TP Option Costing	370	Transfer Pricing Option Costing processing	11
PA Lookup Tables	500	Lookup tables used in PA Lookup Table ID	4
Break Funding	530	Break Funding instrument for TP	1
Derivative Instruments	600	Classification to identify derivative instruments	5

### 2. Detailed Report

You can click any classification to navigate to the detailed registered objects report.

The screenshot shows the Oracle Object Registration interface. At the top, there are navigation tabs for 'Object Registration' and 'Detailed Report'. Below this is a search bar with fields for 'Logical Table Name', 'Column Name', 'Logical Data Type', and 'IS NULL'. The main content area displays a 'Registered Columns Report' table. The table has the following columns: Attribute, Column Name, Physical Data Type, Logical Data Type, Data Length, Data Precision, Data Scale, Nullable, Column Order, Processing Key, Balance, Standard Rate, Transfer Price Out, and Related Field. The table contains several rows of data, including entries for 'Account Age', 'Account Age Multiplier', 'Account Officer', 'Accrual Basis Code', 'Accumulation Type Code', and 'Adjustable Type Code'. The table is paginated, showing records 1-25.

Attribute	Column Name	Physical Data Type	Logical Data Type	Data Length	Data Precision	Data Scale	Nullable	Column Order	Processing Key	Balance	Standard Rate	Transfer Price Out	Related Field
Account Age	ACCOUNT_AGE_C	NUMBER	DEFAULT	32	21	0	Y	1	No	No	No	No	ACCOUNT_AGE_MULT_C
Account Age Multiplier	ACCOUNT_AGE_MULT_C	CHAR	MULT	32	8	0	Y	2	No	No	No	No	ACCOUNT_AGE_C
Account Officer	ACCOUNT_OFFICER	VARCHAR2	VARCHAR2	10			N	3	Processing Key	No	No	No	
								4	No	No	No	No	
								5	No	No	No	No	
								10	No	No	No	No	
								24	No	No	No	No	
								31	No	No	No	No	
								100	No	No	No	No	
								118	No	No	No	No	
								121	No	No	No	No	
								122	No	No	No	No	
Accrual Basis Code	ACCRUAL_BASIS_CD	NUMBER	CODE	22	8	0	Y	140	No	No	No	No	
								147	No	No	No	No	
								148	No	No	No	No	
								150	No	No	No	No	
								152	No	No	No	No	
								153	No	No	No	No	
								154	No	No	No	No	
								155	No	No	No	No	
								170	No	No	No	No	
								180	No	No	No	No	
								200	No	No	No	No	
Accumulation Type Code	ACCUM_TYPE_CD	CHAR	CODE	1			N	3	Yes	No	No	No	
Adjustable Type Code	ADJUSTABLE_TYPE_CD	NUMBER	CODE	22	8	0	Y	6	No	No	No	No	
								11	No	No	No	No	

Lookup tables are used within OFSAA Profitability Management application. Lookup tables have to be created and registered within OFSAAI, in order to display them in Lookup Table Driver definition of OFSAA Profitability Management application. For more information on Lookup Table creation, see *Oracle Financial Services Analytical Applications Data Model Utilities User Guide*.

## Security

The details about user privileges, group privileges, functions, and so on for the viewing of the objects are provided here.

The screenshot displays the Oracle Interactive Dashboards Security interface. It features four main sections for viewing privileges:

- User Privileges:** Filtered by User ID: BHARATH, User Name: (All Choices), Segment Name: DEMOSEG. The table lists functions like Data Quality Aid, DI Designer, and Fusion AAI Attributes for user Bhargh.
- Group Privileges:** Filtered by Group Code: POAGRP, Group Name: POAGRP, Segment Name: DEMOSEG. The table lists functions for group Bhargh.
- Role Privileges:** Filtered by Role Code: CHSADMIN, Role Name: (All Choices). The table lists roles like DEMOGRP and CHSADMIN for user Bhargh.
- Function Privileges:** Filtered by Function Code: ALSA00, Function Name: (All Choices). The table lists functions like Action Assessor for user Bhargh.

### User Privileges

The details of a user and various associated privileges can be viewed in this report.

This screenshot shows a detailed view of the User Privileges report. The filters are set to User ID: AA, User Name: (All Choices), and Segment Name: (All Choices). The report title is "User Privileges" and the time is 11/30/2009 5:29:01 PM. The table below shows the details for user "Action Assessor":

User Name	Group Name	Segment Name	Role Name	Function Name
Action Assessor	AAGRP	ROR\$E02	Action Assessor	Access to user options link
				Action Identification
				AI Close
				AI Create
				AI Request for Appro
				AI Save
				Call Remote Web Services
				Close for document
				DOC_MENU
				Document Management Access

You can click the user name to obtain the detailed user report.

**Security**

**Detailed User Report**  
Time run: 11/30/2009 5:33:59 PM

**User Name : Action Assessor**

<b>User ID</b>	AA
<b>User Name</b>	Action Assessor
<b>Created Date</b>	05/25/2009 21:26:08 PM
<b>Created By</b>	RORADMIN
<b>User Start Date</b>	05/25/2009
<b>Last Login Date</b>	09/01/2009 12:33:09 PM
<b>Last Modified By</b>	RORADMIN
<b>Last Modified Date</b>	05/25/2009 21:26:08 PM
<b>Authorize Status</b>	Authorized
<b>E-Mail</b>	snphani@identity.com

[Return](#) - [Refresh](#) - [Print](#) - [Download](#)

### Group Privileges

The details of a user group and various associated privileges can be viewed in this report.

**Group Privileges**

Group Code: AAGRP | Group Name: (All Choices) | Segment Name: (All Choices) | [Go](#)

**Group Privileges**  
Time run: 11/30/2009 5:28:06 PM

Group Name	Role Name	Segment Name	Function Name	User Name
AAGRP	Action Assessor	RORSEG2	Access to user options link	Action Assessor
			Action Identification	Action Assessor
			AI Close	Action Assessor
			AI Create	Action Assessor
			AI Request for Appro	Action Assessor
			AI Save	Action Assessor
			Call Remote Web Services	Action Assessor
			Close for document	Action Assessor
			DOC_MNU	Action Assessor
			Document Management Access	Action Assessor


[Results](#)

Records 1 - 10

[Refresh](#) - [Print](#) - [Download](#)

You can click the group name to get the detailed group report.

**Security**

 **Detailed - Group**  
Time run: 11/30/2009 5:38:41 PM

---

**Group Name : AAGRP**

Group Code	AAGRP
Group Name	AAGRP
Group Desc	Action Assessor Group
Group Type	USER
Created Date	05/25/2009 21:29:38 PM
Created by	RORADMIN
Last Modified Date	05/25/2009 21:29:38 PM
Last Modified By	RORADMIN


[Return](#) - [Refresh](#) - [Print](#) - [Download](#)

### Role Privileges

The details related to role privileges can be viewed in this report.

**Role Privileges**

Role Code: CWSADMIN    Role Name: (All Choices)   

 **Role Privileges**  
Time run: 12/11/2009 6:12:20 PM

Role Name	Function Name	Segment Name	User Name	Group Name
CWS Administrator	Call Remote Web Services	DEMOSEG	authdef	DEMOGRP
			DEMOUSER	DEMOGRP
			test11	DEMOGRP
			varun1	VRNGRP
			varun2	DEMOGRP
		varun3	DEMOGRP	
		vicky	DEMOGRP	
		authdef	DEMOGRP	
		GASEGR2	DEMOUSER	DEMOGRP
		stuser2	STGROUP	

Records 1 - 10

[Refresh](#) - [Print](#) - [Download](#)

You can click the role name to get the detailed role report.

**Security**

**Detailed - Role**  
Time run: 12/11/2009 6:27:09 PM

Role Code	CWSADMIN
Role Name	CWS Administrator
Role Desc	CWS Administrator Role
Role Type	SYSADMIN
Created Date	10/10/1958 00:00:00
Last Modified Date	10/10/1958 00:00:00
Created By	SYSADMIN
Last Modified By	SYSADMIN

[Return](#) - [Refresh](#) - [Print](#) - [Download](#)

### Function Privileges

The details related to function privileges can be viewed in this report.

**Function Privileges**

Function Code:  Function Name:

**Function Privileges**  
Time run: 12/11/2009 6:12:20 PM

Function Name	Role Name	Group Name	Segment Name	User Name
Add Alias	Business Analyst	DEMOGRP	DEMOSEG	authdef
				DEMOUSER
				test11
				varun2
				varun3
			GASEGR2	vicky
				authdef
				DEMOUSER
				test11
				varun2

Records 1 - 10

[Refresh](#) - [Print](#) - [Download](#)

You can click the function name to get the detailed function report.



Security	
 <b>Detailed - Function</b> Time run: 12/11/2009 6:32:37 PM	
<b>Function Name : Add Alias</b>	
Function Code	ALSADD
Function Name	Add Alias
Function Description	The user mapped to this function can add Alias
Function Type	SYS
Created Date	10/10/1958 00:00:00
Created By	SYSADMIN
Last Modified Date	10/10/1958 00:00:00
Last Modified By	SYSADMIN
<a href="#">Return</a> - <a href="#">Refresh</a> - <a href="#">Print</a> - <a href="#">Download</a>	

## Adhoc Reporting – Special Instructions

The following instructions need to be followed whenever you want to carry out the Adhoc Reporting or Adhoc Querying using the OBI Answers tool or any other querying or reporting tools.

In the OBI Answers tool, when you click the Answers link, you will find single subject area, Fusion. Here you can select the tables (folders) and columns related to the Object Registration, Data Quality and Security dashboards. These tables (folders) and columns can also be used for Adhoc Reporting or Adhoc Querying.

Since these tables belong to different functional areas, namely Object Registration, Data Quality and Security, relationships or table joins across functional areas are not supported for Adhoc Reporting. Therefore, during Adhoc Report or request creation you have to make sure that tables and columns selected are within the functional groups mentioned in the following section to get correct results.

### Data Quality

You can use following folders (tables) and columns for Adhoc Querying on Data Quality related objects.

- Dim Stratification
- Dim Leaves
- Dim Invalid Codes
- Dim Ledger Stat
- Dim Product
- Dim Org Unit
- Dim Currency

- Dim General Ledger
- Dim Common COA
- Fact Stratification
- Fact Leaves
- Fact Invalid Codes

**Object Registration**

You can use following folder for Adhoc Querying on Object Registration related queries.

- Dim Classification

**Security**

You can use following folder for Adhoc Querying on Security related queries.

- Dim Security

## CHAPTER 4

# OFSAA Data Entry Forms and Queries

## Introduction to Data Entry Forms and Queries

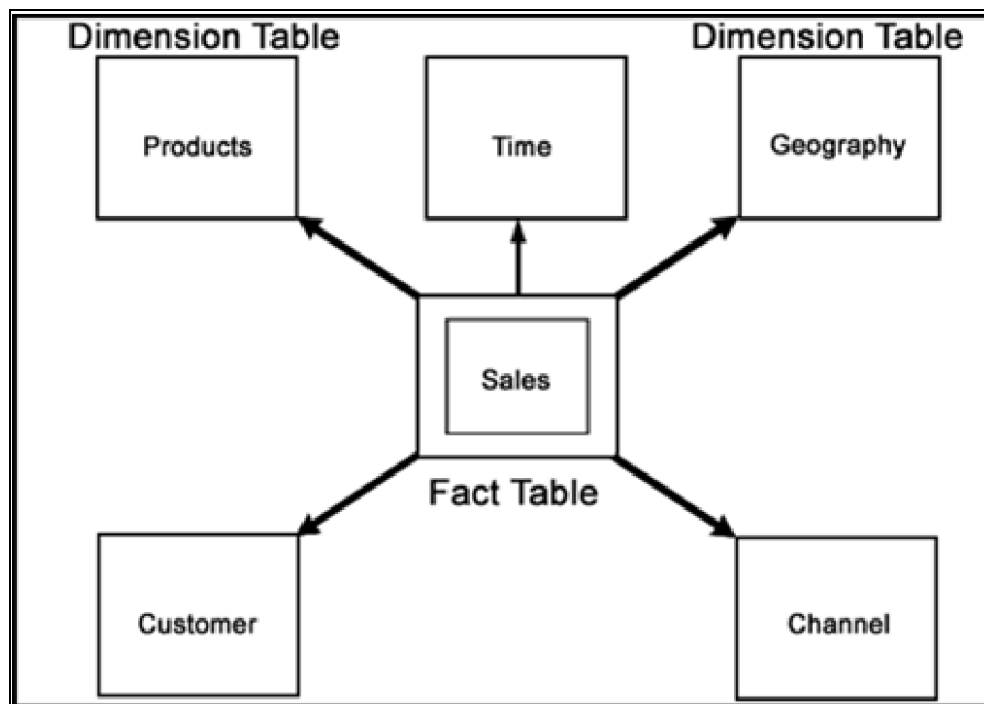
Data Entry Forms and Queries functionality is provided to allow you to create user-friendly Data Entry Screens with a choice of layouts, which aids easy data view and data manipulation. You can select tables based on which you can create forms that can be used in any Application.

### Acronyms

**Defi** Data Entry Forms and Queries.

**Data Model** A logical map that represents the inherent properties of the data independent of software, hardware or machine performance considerations. The model shows data elements grouped into records, as well as the association around those records.

**Star Schemas** It is the simplest of data warehouse schemas. This schema resembles a star diagram. While the center contains one or more fact tables the points (rays) contain the dimension tables



In a star schema, only one join is needed to establish the relationship between the fact table and any one of the dimension tables which optimizes queries as all the information about each level is stored in a row.

## Designing Data Entry Forms

The Data Entry Forms and Queries menu has three options, namely, Form designer, Data Entry and Excel Upload options.

To Design forms, click **Form Designer** to open the **Forms Designer Wizard**. In this screen, you can create, alter, copy or delete forms. **Create a new Form** is the default option that is chosen when you open the wizard. The various steps involved in the creation of the form are displayed in the left pane of the screen. You can also assign User Rights to already defined Forms.

The first screen of the Forms Designer wizard is displayed as Step 2, Step 1 being the login screen.

### Creating a New Form

To design a new form you will have to first choose the Create a New Form option. On entering the screen this option will be selected by default.

An Application refers to a group with a list of forms within it. The Available application field will contain the new application by default.

To create a new form, select New Application from the Available Applications drop-down list, click the field provided for New Application Name and enter an application name.

Enter a form name in the **New Form Name** field.

---

**Note:** To modify an existing form, select Alter Existing Forms option. Make your modifications and move to step 3.

---

Click **Next** to go to step 3. Click **Cancel** to reset the fields.

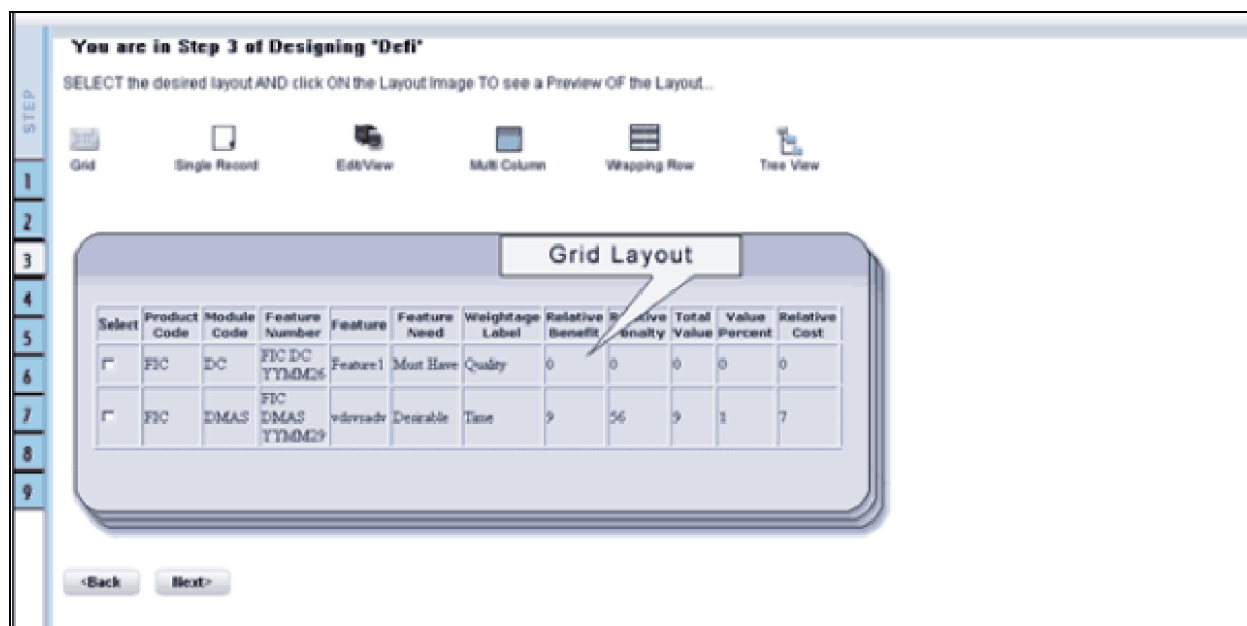


The Layout screen displays a range of pre-defined layouts for your form. The various layouts available are Grid, Single Record, Edit/View, Multi Column, Wrapping Row and Tree View.

To view the display of a layout, click any of the icons on the top of the screen. Click Back to go to the previous screen. Click Next to go to step 4. Click Cancel to refresh the choice of the selected layout.

## The Grid Layout

The Grid Layout is the default layout that is displayed. It will display the records in the form of a table/grid with multiple rows of data.



This layout can be used when you wish to view multiple records from a table at a time.

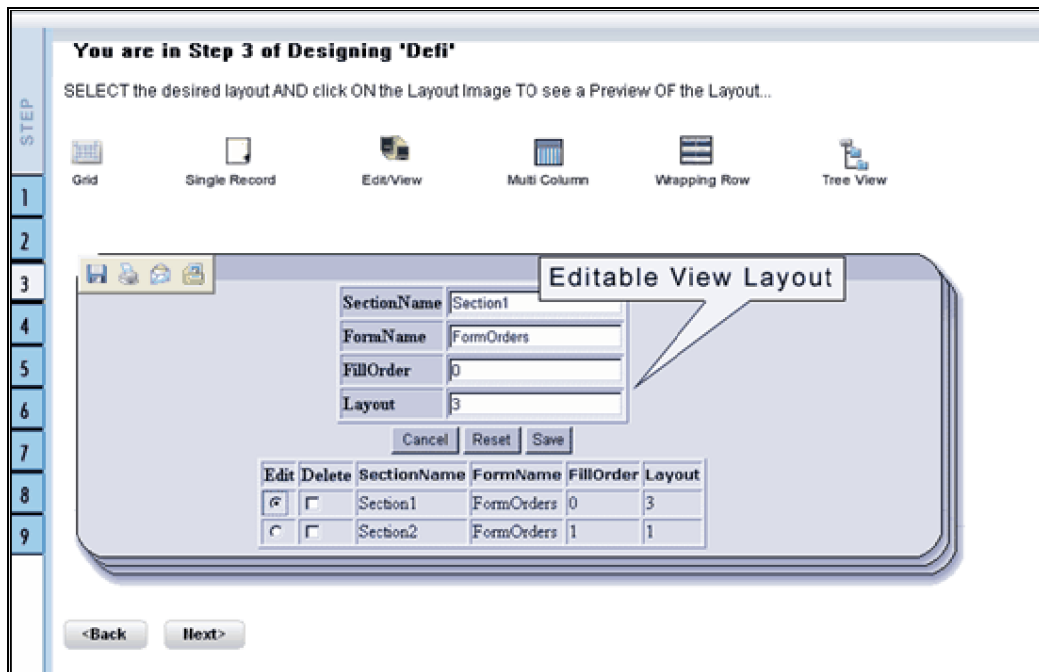
### ***The Single Record Layout***

The Single Record Layout displays a single record at a time.



### ***The Edit/View Layout***

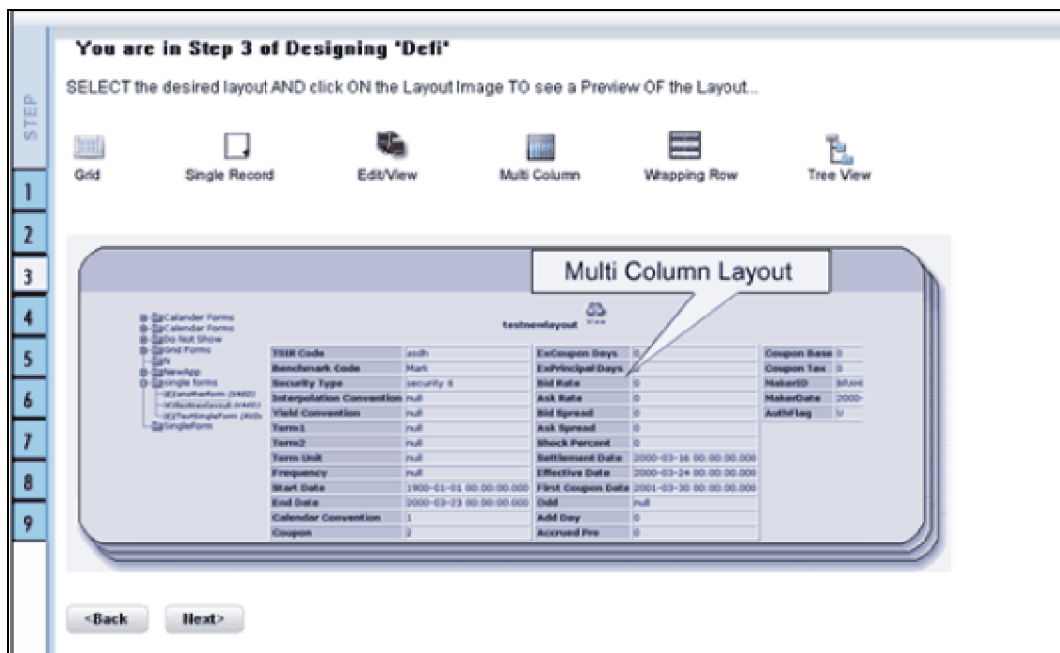
The Edit/View Layout is a combination of the single record and grid layout.



If you choose a record in the grid, that particular record will be displayed in the form, in a single record format, which is editable. By default the first record will be displayed in the editable grid.

## The Multi Column Layout

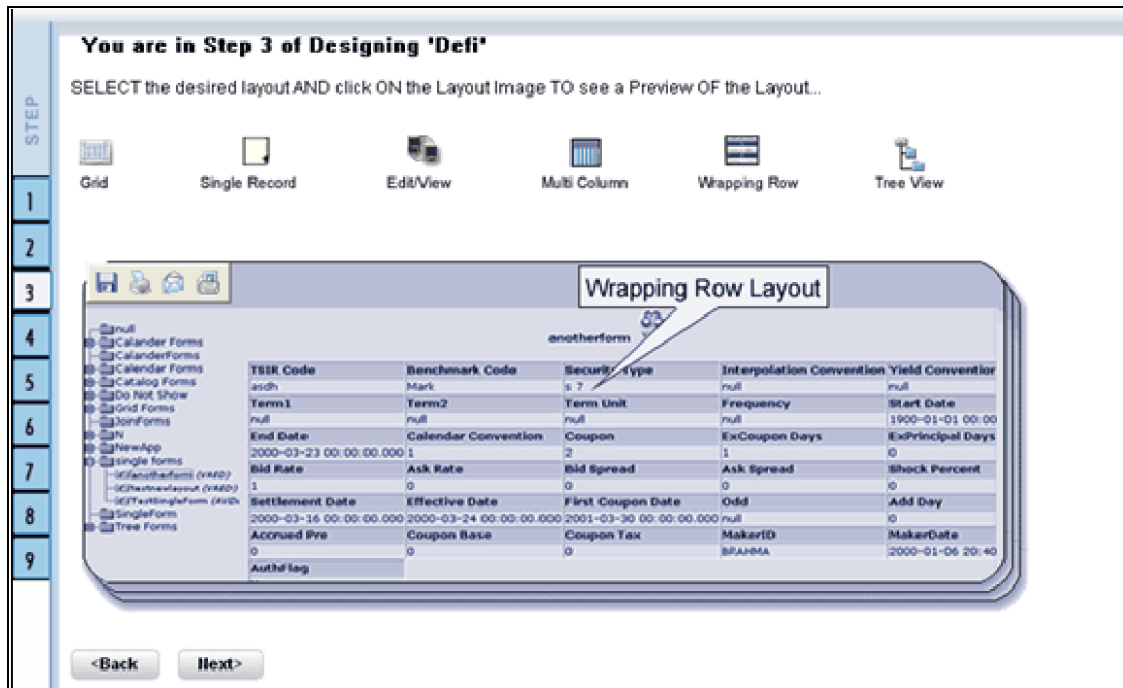
The Multi Column Layout displays a single record with its column in a grid format.



You can view a multi column layout form without having to scroll or with minimum scrolling to view all the columns.

### The Wrapping Row Layout

The Wrapping Row displays rows of a single record in a wrapped manner in a grid format.

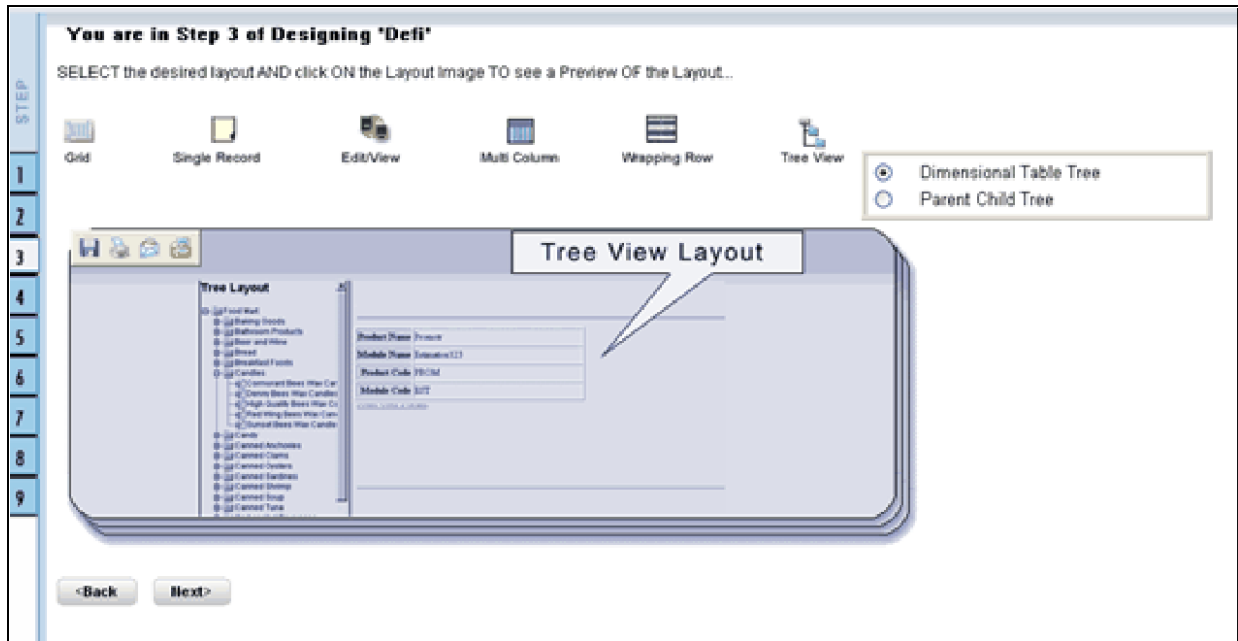


You can view a wrapping row layout form easily without having to scroll horizontally to view all the data.

### The Tree View Layout

The Tree view layout represents the dimensional table. It presents the entire dimension tree with selected dimension details.



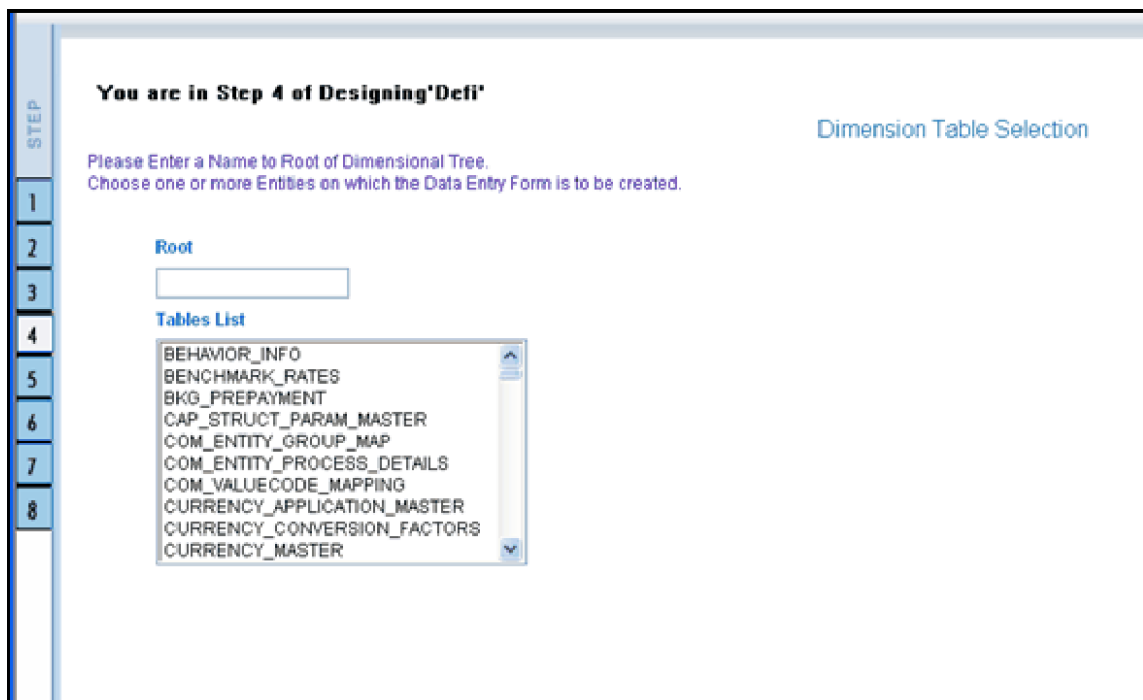


Click Back to go to the previous screen. Click Next to go to step 4. Click Cancel to reset the selected layout option.

For Grid, Single Record, Edit/View, Wrapping Row, Multicolumn and Tree View Layouts

The table screen displays the Tables List box with available tables. You can choose more than one table by pressing the CTRL key for nonadjacent selection or the SHIFT key for multiple adjacent selections.

However, it is necessary that the tables that you select have same data type for the two mapped fields.



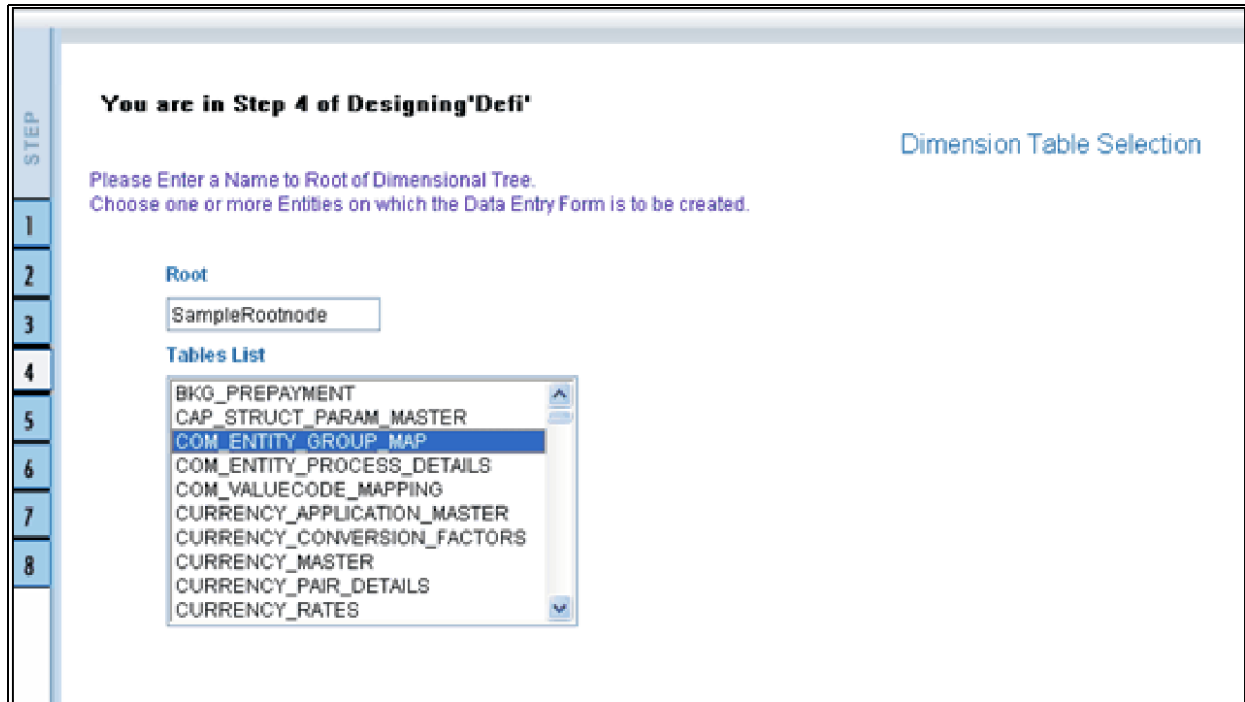
DIM\_ACTION\_GROUP

DIM\_ACTION\_PRIORITY

DIM\_ACTION\_TYPE

Click **Back** to go to the previous screen. Click **Next** to go to step 5. Click **Cancel** to refresh the table list.

If you have selected **Tree View layout** in the previous screen, the Table list screen will be as shown .



Enter a root node name in the Root text box.

Sample Rootnode

Choose a table that you want to be included in the form.

DIM\_ACTION\_GROUP

Click **Back** to go to the previous screen. Click **Next** to go to the join tables screen.

## Joining Tables

In the join table screen, you can select the fields to be joined.



Choose a table name that you wish to add or edit.

DIM\_PRODUCTS

Select the field of the chosen table that you want to add from the Available Fields list box and choose the field that you want to join it with of the latter table. The chosen fields will be displayed in the Join Condition pane.

DIM\_PRODUCTS.V\_D\_PRODUCT\_LEVEL\_1\_CODE=DIM\_PRODUCTS.V\_D\_PRODUCT\_LEVEL\_2\_DESC

Click **Clear** to clear the join condition.

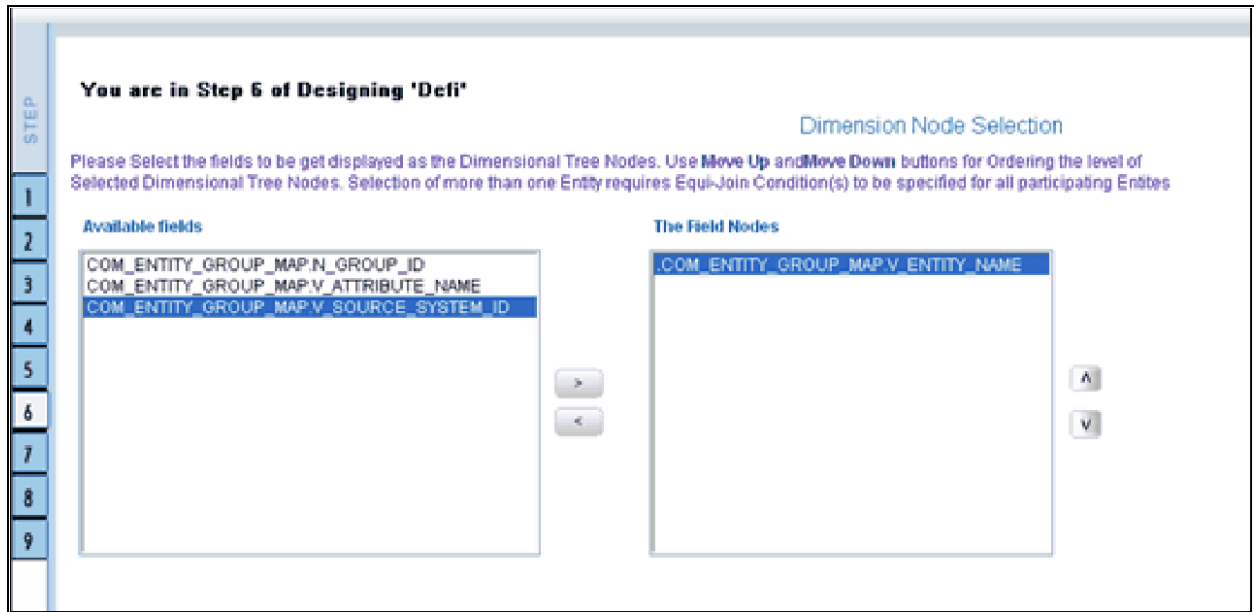
Click **Back** to go to the previous screen. Click **Next**, to join the fields and go to step 5.

Click **Cancel** to refresh the table list.

### For Grid, Single Record, Edit/View, Wrapping Row, Multicolumn and Tree View Layouts.

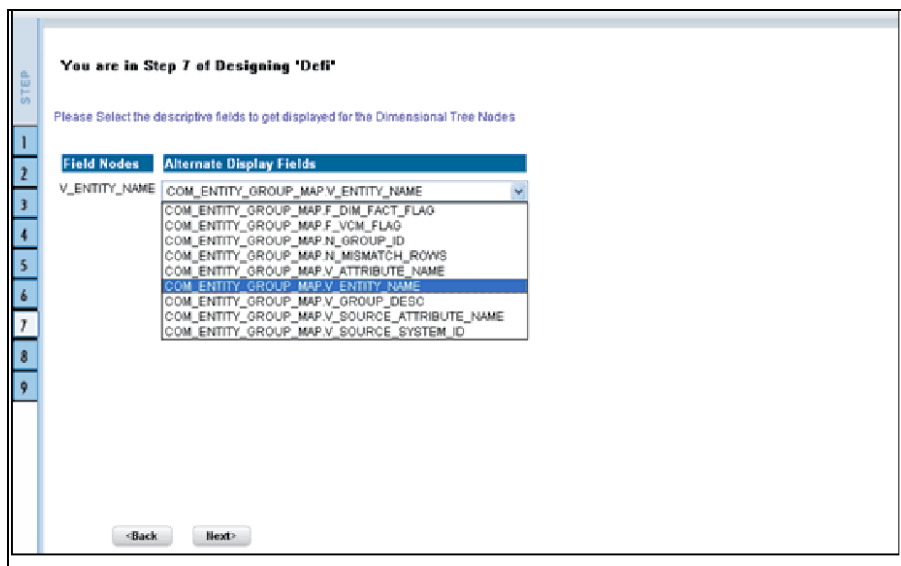
The Fields screen displays the Available Fields pane and Fields for sorting pane. The mandatory fields are highlighted with the '\*' symbol and are automatically displayed in the Selected Fields pane as pre-selected for you and are shown in the Chosen Fields Pane. These mandatory fields are fields (primary key fields and not null fields) that require data to be entered by the user while performing Data Entry on the underlying table.

If you want the fields to be sorted based on the Branch Code, select the field CUSTOMER\_V\_CUST\_BRANCH\_CODE.



Select the fields from the Available Fields pane that you want to be included in the form and click ( > ) to move to the Fields to Display pane. To remove a field(s) from the Fields to Display pane to the Available Fields pane, click ( < ) . To move all the fields to the Fields to Display pane from the Available Fields pane, click ( >> ) . To move all the non mandatory fields to the Available Fields pane, click ( << ) .

To arrange the fields that are displayed in the Fields to Display pane, click up arrow symbol to move a field up and down arrow symbol to move down. This order determines how the Fields are displayed in the Data Entry Form. This order can be different from the order in which the fields are in the underlying table.



Select the fields from the Available Fields pane that you want to be included in the form and click ( > ) to move to Step 8. To remove a field(s) from the Choose Fields pane to the Available Fields pane, click ( < ) . To move all the fields to the Choose Fields pane from the Available Fields pane, click ( >> ) . To move all the non mandatory fields to the Available Fields pane, click ( << ) .

To arrange the fields that are displayed in the Choose Fields pane, click up arrow symbol to move a field up and down arrow symbol to move down. This order determines how the Fields are displayed in the Data Entry Form. This order can be different from the order in which the fields are in the underlying table.

The selected fields will be sorted in ascending order by default. To set the order of the fields in the descending order, check the check-box provided for **Sort By Descending**.

**Note:** The fields displayed in the Sort Fields screen are the list of fields chosen from the Fields Selection screen.

Click **Back** to go to the previous screen. Click **Next** to go one step further.

## For Grid, Single Record, Edit/View, Wrapping Row & Multicolumn Layouts

The index screen displays the form with Field Name, Display Name, In View, In Edit/Add, Allow Add, Store Field as and Rules detail columns. You can specify the details for each of the field names.

Field Name	Display Name	In View	In Edit / Add	Allow Add	Store Field as	Rules	Format Type
N_GROUP_ID*	Group ID	Display	Read Only	<input checked="" type="checkbox"/>	Normal	Rules	NONE
V_ATTRIBUTE_NAME*	Attribute Name	Display	Read Only	<input checked="" type="checkbox"/>	Normal	Rules	NONE
V_ENTITY_NAME*	Entity Name	Display	Read Only	<input checked="" type="checkbox"/>	Normal	Rules	NONE
V_SOURCE_SYSTEM_ID*	Source System Id	Display	Read Only	<input checked="" type="checkbox"/>	Normal	Rules	NONE

You can change the display name of the field name by entering a name in the Display Name field.

### User Expiry Date

You can choose a field to be displayed by selecting the In View option as Display or Do not Display. If the field you choose is a foreign key field or if more than one table is selected, the In View displays Same Field, Alternate Display Field and Do not Display options.

You can specify the edit options for a user by specifying the edit parameters. Click the In **Edit/Add** field to open the drop-down list box. Select the edit parameter that you want to apply to a field. For normal fields you can choose the edit parameters like Text Field, Text Area, Select List, Protected Field, Read Only and Do Not Show. If it is a foreign key field the In Edit/Add parameters displayed will be Read Only, Select List and Do Not Show. For primary key fields the In Edit/Add parameter displayed will be Read Only and Do Not Show. In case of calendar fields, the In Edit/Add parameters displayed are Calendar and Do Not Show.

To give the user permission to add a new record, select the Allow Add option against the field name. If form is saved with add option as true for some fields and false for some fields then it will give an alert saying "Choose Allow Add for the mandatory fields."

To specify the field storage format, click the **Store Field as** field to open the drop-down list. You can choose the store format as Normal, Sequence Generator, Maker Date, Checker Date, Created Date, Modified Date Auth Flag, Maker id, Maker Date, Checker id, Checker Date, Checker Remarks and Maker Remarks.

You can also apply rules to a field. Click Rules to open the screen.

## ***Applying Rules***

You can apply rules only if the In Edit/Add fields are Text Field, Text Area or Protected Field. The rule option is essential for data validation

To specify rules for Text Field, Text Area and Protected Field, click the **Rule** option to open the Specifying Rules and Expressions for Data Validations window.

If a field has a character data type, you can specify here the rule for the field to be displayed without spaces, only characters or alphanumeric. If a field has a numeric data type, you can check the check-box provided for Numeric Only. You can choose whether/not you require Negative values to be displayed by checking/un-checking the field provided for Non Negative, not null will be displayed for the non primary key and/or nullable fields.

Specify the maximum width of the field in the Maximum Width box. You can set the column alignment by selecting the appropriate alignment from the Alignment drop-down list. On specifying the options, click Ok to save.

Click **clear** to Clear the information entered in the screen. Click **Close** to close the screen.

Click **Back** to go to the previous screen. Click **Save** to save the field details setting.

- Only forms with primary key tables can have the authorization option while saving.
- By default, the alignment for character and date data type is taken as Left. And for numbers it is displayed as Right. But the alignment can be changed to Right, Left or Center.

You can specify the edit options for a user by specifying the edit parameters. The In Edit option varies for different fields based on whether the field is foreign, primary or calendar fields.

Click the **In Edit/Add** field to open the drop-down list box. You can choose the edit parameters like Text Field, Text Area, Select List, Read Only, Do Not Show and Expression Generator. Select the edit parameter that you want to apply to a field.

When you select the Select List option in the In Edit/Add column, a pop-up window is displayed.

---

**Note:** The Select List option will not be displayed for foreign key fields.

---

Enter the list of values to be displayed, in the List of Values field. You may choose the field to be Comma Separated Values or Dynamic list of values options.

Comma Separated Values can have only the user specified values while creating a form.

Dynamic Field accepts the fieldname from a table, which it stores in the database, which is used during Data Entry. Specify **Alternate Display Values** in the appropriate field.

Click **OK** to save the specified list of values. Click **Cancel** to clear the List of Values field. Click **Close** to close the pop-up window.

If you select Dynamic List of Values, you will be able to select the table values and then define a filter condition for the selected values, as shown in the following screen-shot.

To give the user permission to add to the field, select the Allow Add option against the field name. The Allow Add option is mandatory for primary key and not null fields.

To specify the field storage format, click the Store Field as field to open the drop-down list box. You can choose the store format as Normal, Sequence Generator, maker ID, Checker ID, Maker Date, Checker Date, Created Date, Modified Date, Maker Remarks, Checker Remarks and AuthFlag.

Click Back to go to the previous screen. Click Save to save the field details setting.

To save the form containing primary key tables, click **Save with Authorization**.

- Only forms with primary key tables can have the authorization option while saving.
- If the user want the value to be displayed with the thousands separator, such a numeric field should be defined under the Display Value Field.

## Committing the Batch

Selecting the check-box provided for Batch Commit virtually groups/baskets the entire set of table forms. With this grouping, the system ensures that if a particular form in the table fails to be executed, the entire set of forms will be returned. Hence, all activities for the selected table is carried out in batches.

## Defining the Form Filter

With the Form Filter screen you can specify expressions for the form filter condition.

**SPECIFYING EXPRESSIONS FOR FORM-LEVEL FILTER CONDITION**

Fields :	Operators :	Functions :
CSSMS_USR_PROFILE.D_CREATED_DATE	(	ABS()
CSSMS_USR_PROFILE.D_DELETED_DTE	)	ACOS()
CSSMS_USR_PROFILE.D_LAST_DISABLED_DTE	*	ADD_MONTHS()
CSSMS_USR_PROFILE.D_LAST_ENABLED_DTE	/	ASCII()
CSSMS_USR_PROFILE.D_LAST_MODIFIED_DATE	+	ATAN()
CSSMS_USR_PROFILE.D_LAST_PWD_CHG_DTE	-	ATAN2()
CSSMS_USR_PROFILE.D_LASTLOGGED_DATE	<	CEIL()
CSSMS_USR_PROFILE.D_LASTLOGIN_DATE	>	CHR()

Expression Viewer :

Click to select the **Field**, the **Operator** and the **Function** from the appropriate panes.

The expression you define will be displayed in the Expression Viewer pane.

Click OK to save the expression. Click Clear to refresh the screen. Click Close to close the screen without saving.

## Data Versioning

Using this screen you can perform data versioning on an authorized form. The table version is tracked based on the date of modification. Check the check-box provided for Enable Data Versioning to ensure that the version is tracked.

**Select Table to hold History Of Changes**

**Enable Data Versioning**

**Table**      CSSMS\_USR\_PROFILE

**Version Identifier**      CSSMS\_USR\_PROFILE.D\_CREATED\_DATE

If you select this option, the modifications made to the particular form are displayed as per date versioning.



## Specifying Message Details

With the messaging screen, you can specify an alert message that can be sent either to the Creator of the form or to an Authorizer. Click the Message Details tab in the index screen. The Message Details screen will be populated on your screen.

**Form Specific Message Details**

Messaging Required

**Available Message Types**

Canceled Request for Creation  
 Canceled Request for Modification  
 Creation and Authorized  
 Creation and Put-On-Hold  
 Creation and Rejected

**Chosen Message Types**

---

**Details for Message Types**

Specific Messages Required

Message Type:

Message Subject:

Message Content:

**Available Fields For Subject** | **Mapped Fields For Subject**

**Available Fields For Content** | **Mapped Fields For Content**

**Available Recipients** | **Mapped Recipients**

Save Cancel

To activate the Messenger feature, you need to check the check-box provided for Messaging Required.

If you do not require this option, leave the Messaging required check box unchecked. Mails will be sent appropriately, only one mail will be sent for the entire batch.

The Message types, Recipients, Subject and Content fields are comes from the metadata.

**Form Specific Message Details**

**Messaging Required**

**Available Message Types**

Canceled Request for Modification

Creation and Authorized

Creation and Put-On-Hold

Creation and Rejected

Creation and Request For Authorizati

>

<

**Chosen Message Types**

Batch Commit

Canceled Request for Creation

**Details for Message Types**

**Message Type** Batch Commit

**Specific Messages Required**

**Message Subject** Infodom - \$INFODOM\$, Application - \$APPLNNAME\$,  
Form -\$FORMNAME\$

**Message Content** Batch Transactions

**Available Fields For Subject**

N\_ACTION\_TYPE\_CODE

N\_ACTION\_TYPE\_KEY

>

<

**Mapped Fields For Subject**

**Available Fields For Content**

N\_ACTION\_TYPE\_CODE

N\_ACTION\_TYPE\_KEY

>

<

**Mapped Fields For Content**

**Available Recipients**

AUTHORIZER

>

<

**Mapped Recipients**

CREATOR

Save
Cancel

Select the Message Type from the Available list and click (>) to move the message type to the Chosen Message Types pane.

Canceled Request for Modification

Form Specific Message Details	
<input checked="" type="checkbox"/> <b>Messaging Required</b>	
<b>Available Message Types</b> Canceled Request for Creation <b>Creation and Authorized</b> Creation and Put-On-Hold Creation and Rejected Creation and Request For Authorizati	<b>Chosen Message Types</b> Canceled Request for Modification
Details for Message Types	
<input type="checkbox"/> <b>Specific Messages Required</b>	
<b>Message Type</b>	Canceled Request for Modification
<b>Message Subject</b>	Infodom - \$INFODOM\$, Application - \$APPLNNAME\$, Form -\$FORMNAME\$ - Cancel Request For Modification
<b>Message Content</b>	Infodom - \$INFODOM\$, Application - \$APPLNNAME\$, Form -\$FORMNAME\$ - Cancel Request For Modification
<b>Available Fields For Subject</b> FIG_MIS_DATE V_CARD_NBR D_ISSUE_DATE D_RECORD_END_DATE D_RECORD_START_DATE	<b>Mapped Fields For Subject</b>
<b>Available Fields For Content</b> FIG_MIS_DATE V_CARD_NBR D_ISSUE_DATE D_RECORD_END_DATE D_RECORD_START_DATE	<b>Mapped Fields For Content</b>
<b>Available Recipients</b> CREATOR	<b>Mapped Recipients</b> AUTHORIZER
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

Select the check-box provided for Specific Messages Required to add a Specific Message.

Select the Message Type from the drop-down list box of the Message Type field. Message line appears automatically based on the selection of the event type.

Canceled Request for Modification

Form Specific Message Details

**Messaging Required**

Available Message Types		Chosen Message Types
<div style="border-bottom: 1px solid black; padding: 2px;">Canceled Request for Creation</div> <div style="background-color: #0056b3; color: white; padding: 2px;">Creation and Authorized</div> <div style="padding: 2px;">Creation and Put-On-Hold</div> <div style="padding: 2px;">Creation and Rejected</div> <div style="border-bottom: 1px solid black; padding: 2px;">Creation and Request For Authorizati</div>	>  <	Canceled Request for Modification

Details for Message Types

**Specific Messages Required**

**Message Type** Canceled Request for Modification

**Message Subject** Infodom - \$INFODOM\$, Application - \$APPLNNAME\$,  
Form -\$FORMNAME\$ - Cancel Request For Modification

**Message Content** Infodom - \$INFODOM\$, Application - \$APPLNNAME\$,  
Form -\$FORMNAME\$ - Cancel Request For Modification

Available Fields For Subject		Mapped Fields For Subject
<div style="border-bottom: 1px solid black; padding: 2px;">FIC_MIS_DATE</div> <div style="padding: 2px;">V_CARD_NBR</div> <div style="padding: 2px;">D_ISSUE_DATE</div> <div style="padding: 2px;">D_RECORD_END_DATE</div> <div style="border-bottom: 1px solid black; padding: 2px;">D_RECORD_START_DATE</div>	>  <	

Available Fields For Content		Mapped Fields For Content
<div style="border-bottom: 1px solid black; padding: 2px;">FIC_MIS_DATE</div> <div style="padding: 2px;">V_CARD_NBR</div> <div style="padding: 2px;">D_ISSUE_DATE</div> <div style="padding: 2px;">D_RECORD_END_DATE</div> <div style="border-bottom: 1px solid black; padding: 2px;">D_RECORD_START_DATE</div>	>  <	

Available Recipients		Mapped Recipients
CREATOR	>  <	AUTHORIZER

Save
Cancel

Subject line will appear automatically based on the selection of the event type.

Infodom - \$INFODOM\$, Application - \$APPLNNAME\$, Form - \$FORMNAME\$ - Cancel Request for Modification

Message content will appear automatically based on the selection of the event type.

Infodom - \$INFODOM\$, Application - \$APPLNNAME\$, Form - \$FORMNAME\$ - Cancel Request for Modification

Double-click the Recipient name in the Available Recipients pane. The selected recipient/s are displayed in the Mapped Recipients pane.

Authorizer

**Form Specific Message Details**

**Messaging Required**

Available Message Types		Chosen Message Types
<div style="border-bottom: 1px solid black; padding: 2px;">Canceled Request for Creation</div> <div style="border-bottom: 1px solid black; padding: 2px; background-color: #0056b3; color: white;">Creation and Authorized</div> <div style="border-bottom: 1px solid black; padding: 2px;">Creation and Put-On-Hold</div> <div style="border-bottom: 1px solid black; padding: 2px;">Creation and Rejected</div> <div style="border-bottom: 1px solid black; padding: 2px;">Creation and Request For Authorizati</div>	>  <	Canceled Request for Modification

**Details for Message Types**

**Specific Messages Required**

**Message Type** Canceled Request for Modification

**Message Subject** Infodom - \$INFODOM\$, Application - \$APPLNNAME\$,  
Form -\$FORMNAME\$ - Cancel Request For Modification

**Message Content** Infodom - \$INFODOM\$, Application - \$APPLNNAME\$,  
Form -\$FORMNAME\$ - Cancel Request For Modification

Available Fields For Subject		Mapped Fields For Subject
<div style="border-bottom: 1px solid black; padding: 2px;">V_CARD_NBR</div> <div style="border-bottom: 1px solid black; padding: 2px;">D_ISSUE_DATE</div> <div style="border-bottom: 1px solid black; padding: 2px; background-color: #0056b3; color: white;">D_RECORD_START_DATE</div> <div style="border-bottom: 1px solid black; padding: 2px;">F_LATEST_RECORD_INDICATOR</div> <div style="border-bottom: 1px solid black; padding: 2px;">V_ACCOUNT_CODE</div>	>  <	FIC_MIS_DATE D_RECORD_END_DATE

Available Fields For Content		Mapped Fields For Content
<div style="border-bottom: 1px solid black; padding: 2px;">FIC_MIS_DATE</div> <div style="border-bottom: 1px solid black; padding: 2px;">V_CARD_NBR</div> <div style="border-bottom: 1px solid black; padding: 2px;">D_RECORD_END_DATE</div> <div style="border-bottom: 1px solid black; padding: 2px; background-color: #0056b3; color: white;">F_LATEST_RECORD_INDICATOR</div> <div style="border-bottom: 1px solid black; padding: 2px;">V_ACCOUNT_CODE</div>	>  <	D_ISSUE_DATE D_RECORD_START_DATE

Available Recipients		Mapped Recipients
CREATOR	>  <	AUTHORIZER

Save
Cancel

- If you select Authorizer, the message will be sent to all the selected authorizer's for authorization.
- You need to click the Save with Authorization tab in the Index screen so that the messages are functional. When you Save with Authorization, the next screen to be populated is the Authorization screen, where you can assign rights.

Similarly, select the Field/s for Subject and Fields for Content. . Available fields for the subject and content can be selected by the user.

Click Save to save the changes. Clicking Cancel pop's-up a message requesting you to confirm the cancellation without saving. Click Ok in the Pop-up.

Click Save with Authorization in the Index screen to save the changes with authorization. Clicking Save opens up the Assigning Rights screen.

## Altering an Existing Form

To alter an existing form, select Alter Existing Forms option. Click the Application name field to open the drop-down list box. Select the name of the application whose form you wish to alter. On choosing the application name, the forms belonging to the selected application will be displayed in the Form Name field. Click the Form Name field to open the drop-down list box and select the form that you want to alter.

Alter forms apply only to fields and not to tables.

Click Next to go to step 3. Click Cancel to discard the form selection that you have made.

**Note:** The forms that are displayed in the Form name list box are dependent on the DSN (Data Source Name) that you have specified.

The alter fields screen displays the list of fields of the selected form in the Chosen Fields pane. In this screen the fields, which are selected while creating the form are displayed in the Chosen Fields list and the fields that are not selected but available are shown under the Available Fields list.

**You are in Step 3 of Alter Form "Defi" - Layout Type : "Tree"**

Choose a set of Fields to Display. Here You can rearrange the fields  
USE the Move Up AND Move Down Buttons TO ORDER the Fields. Field Names are displayed IN the "Entity Name.Field Name" format

**Available Fields**

COM\_ENTITY\_GROUP\_MAP.F\_DIM\_FACT\_FLAG  
COM\_ENTITY\_GROUP\_MAP.F\_VCM\_FLAG  
COM\_ENTITY\_GROUP\_MAP.N\_MISMATCH\_ROWS  
COM\_ENTITY\_GROUP\_MAP.V\_GROUP\_DESC  
COM\_ENTITY\_GROUP\_MAP.V\_SOURCE\_ATTRIBUTE\_NAME

**Fields to Display**

\*COM\_ENTITY\_GROUP\_MAP.N\_GROUP\_ID  
\*COM\_ENTITY\_GROUP\_MAP.V\_ATTRIBUTE\_NAME  
\*COM\_ENTITY\_GROUP\_MAP.V\_ENTITY\_NAME  
\*COM\_ENTITY\_GROUP\_MAP.V\_SOURCE\_SYSTEM\_ID

Buttons: >, <, >>, <<, A, V

Select the fields from the Available Fields pane, which you want to alter and click (>) to move to the Chosen Fields pane. To remove a field/s from the Chosen Fields pane to the Available Fields pane, click (<). To move all the fields to the Chosen Fields pane from the Available Fields pane, click (>>). To remove all the fields to the Available Fields pane, click (<<).

To arrange the fields that are displayed in the Choose Fields pane, click up arrow symbol to move a field up and down arrow symbol to move down. This order determines how the Fields are displayed in the Data Entry Form. This order can be different from the order in which the fields are in the underlying table.

Click **Back** to go to the previous screen. Click **Next** to go further. The alter sort fields screen displays the sort fields to be altered in the Chosen Fields pane with which you want to sort the table.



**You are in Step 4 of Designing 'Defl'**

**Dimension Node Selection**

Please Select the fields to be get displayed as the Dimensional Tree Nodes. Use **Move Up** and **Move Down** buttons for Ordering the level of Selected Dimensional Tree Nodes. Selection of more than one Entity requires Equi-Join Condition(s) to be specified for all participating Entities

Root :

**Available fields**

COM\_ENTITY\_GROUP\_MAP.N\_GROUP\_ID  
COM\_ENTITY\_GROUP\_MAP.V\_ATTRIBUTE\_NAME  
COM\_ENTITY\_GROUP\_MAP.V\_SOURCE\_SYSTEM\_ID

**The Field Nodes**

COM\_ENTITY\_GROUP\_MAP.V\_ENTITY\_NAME

> <

^ v

Select the fields from the Available Fields pane, which you want to alter and click (**>**) to move to the Chosen Fields pane. To remove a field/s from the Chosen Fields pane to the Available Fields pane, click (**<**). To move all the fields to the Chosen Fields pane from the Available Fields pane, click (**>>**). To remove all the fields to the Available Fields pane, click (**<<**).

**Note:** The user can change the root node name specified while designing the form.

Click Back to go to the previous screen. Click Next to go to next step.

The form screen displays the forms to be altered containing Field Name, Display Name, In View, In Edit/Add, Allow Add, Store Field as and Rules detail columns contain the data which are already selected while creating the form. You can alter the details for each of the field names.

**You are in Step 5 of Designing 'Defi'**

Please Select the descriptive fields to get displayed for the Dimensional Tree Nodes

STEP	Field Nodes	Alternate Display Fields
1		
2	V_ENTITY_NAME	COM_ENTITY_GROUP_MAP.V_ENTITY_NAME
3		COM_ENTITY_GROUP_MAP.F_DIM_FACT_FLAG
4		COM_ENTITY_GROUP_MAP.F_VCM_FLAG
5		COM_ENTITY_GROUP_MAP.N_GROUP_ID
6		COM_ENTITY_GROUP_MAP.N_MISMATCH_ROWS
7		COM_ENTITY_GROUP_MAP.V_ATTRIBUTE_NAME
8		COM_ENTITY_GROUP_MAP.V_ENTITY_NAME
9		COM_ENTITY_GROUP_MAP.V_GROUP_DESC
		COM_ENTITY_GROUP_MAP.V_SOURCE_ATTRIBUTE_NAME
		COM_ENTITY_GROUP_MAP.V_SOURCE_SYSTEM_ID

You can change the display name of the field name by entering a name in the Display Name field.

You can choose a field to be displayed by selecting the In View option as Display or Do not Display.

You can specify the edit options for a user by specifying the edit parameters. Click the **In Edit/Add** field to open the drop-down list box. You can choose the edit parameters like Text Field, Text Area, Select List, Protected Field, Read Only, Do Not Show, Check Box and URL. Select the edit parameter that you want to apply to a field.

**Note:** The AlternateDisplayField is displayed only if the selected table contains a foreign key or if the joined tables are selected.

To give the user permission to add to the field, select the Allow Add option against the field name.

To specify the field storage format click the Store Field as field to open the drop-down list box. You can choose the store format as Normal, Encrypted, Auto Generated or Formula.

You can also apply rules to a field. Click Rules to open the screen.

Click Back to go to the previous screen. Click Save to save the field detail settings

## Copying an Existing Form

To copy an existing form, select Copy from Existing Forms option.

Select one of the following options...

- Create a New Form
- Alter Existing Forms
- Copy Forms
- Delete Forms
- Assign Rights
- Message Type Maintenance

From Application: DeFi      To Application: DeFi

Save Form: Defi      As Form: Products

Next      Cancel

Select the application from which you want to copy a form. Click the Application name field to open the drop-down list box. Select the name of the application. On choosing the application name, the forms belonging to the selected application will be displayed in the Form Name field. Click the Form Name field to open the drop-down list box and select the form that you want to copy.

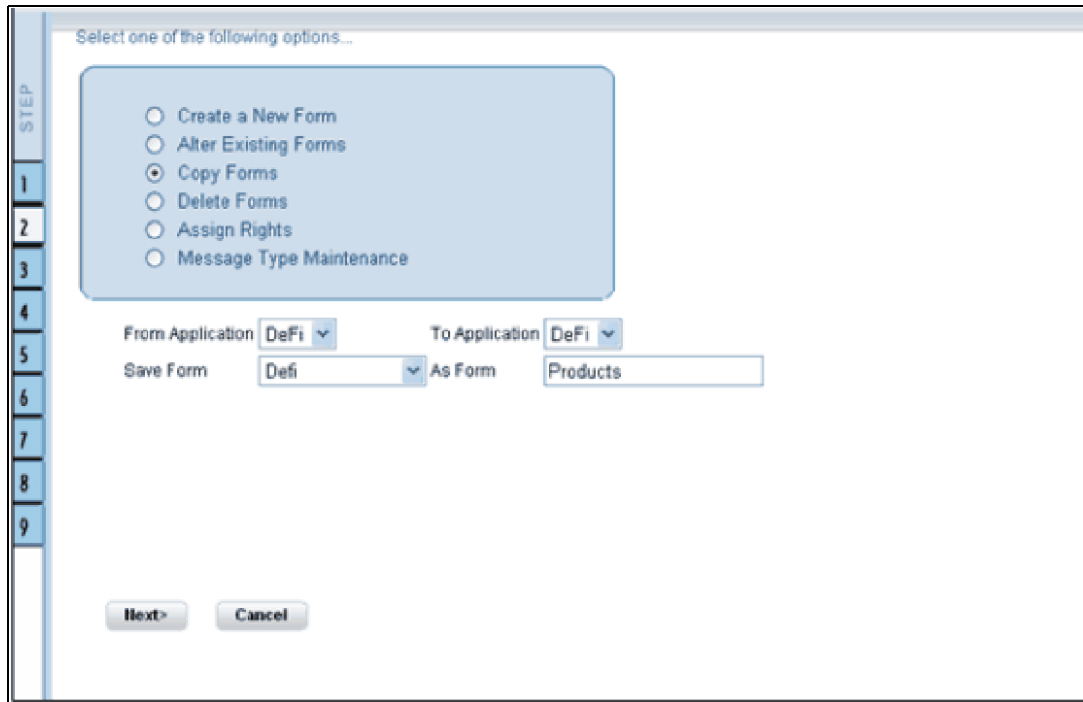
Select the application to which you wish to copy the new form. Click the Application name field to open the drop-down list box and select the application. Enter the name of the new form you are creating in the Save Form as field. This option allows creation of variants for a given form where you might want to change some user rights or some display options and other subtle variations on the form for a given layout and the underlying table.

Click Next to save the details of the newly created form. Click Cancel to discard the form selection that you have made and clear the name entered in the Save Form As field.

## Deleting a Form

To delete an existing form, select **Delete Forms** option.

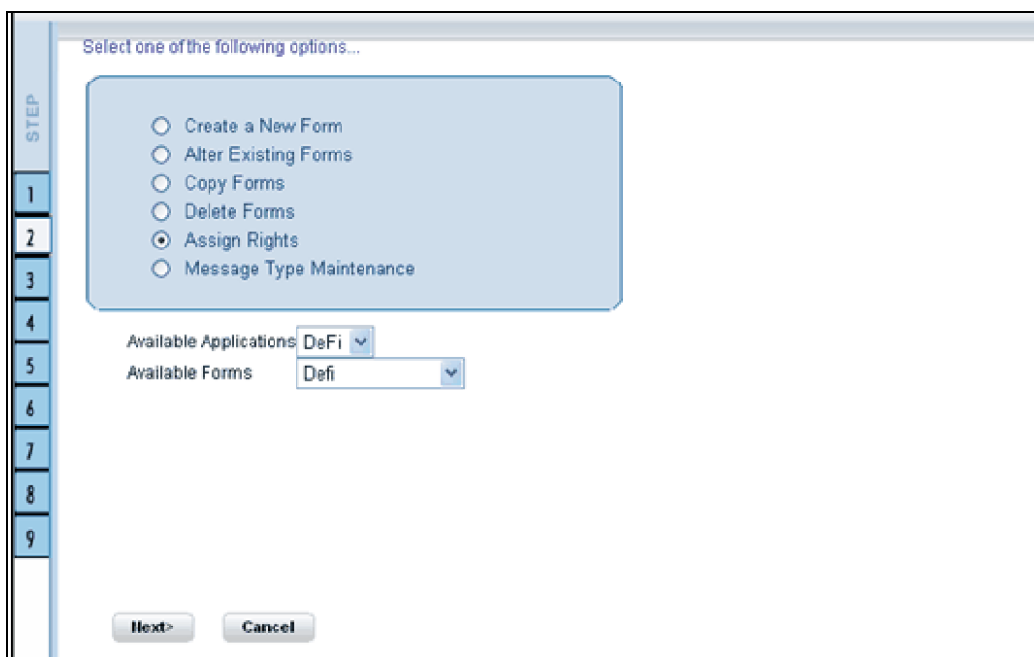
Select the application from which you want to delete a form. Click the Application name field to open the drop-down list box and select the name of the application. On choosing the application name, the forms belonging to the selected application will be displayed in the Form Name field. Click the Form Name field to open the drop-down list box and select the form that you want to delete.



Click **Next** to delete the form. A confirmation message is displayed which reads Do you want to delete the form Loss? Click Ok to confirm deletion. Click **Cancel** to cancel the delete request.

### Assigning Rights to a Form

To assign rights to a form, select Assign Rights option.



Click the Application name field to open the drop-down list box. Select the application from which you wish to choose the form to which you want to assign rights. The chosen application will be displayed in the Application Name field. Click the Form Name field to open the drop-down list box. Select the form you want to assign rights to.

Application name: ST

Form name: TreeForm

Click Next to go to step 3 or Click Cancel to discard the form selection that you have made.

The user list screen displays the list of available and assigned users. Click the < or > buttons to navigate through the list of users in case the number of users are more than what could be accommodated in one page.

Please specify the Access Rights to User for the created form.  
To withdraw all Access Rights from a User, please double click on User Name

AccountsForms

Available User List

MEHAR  
SANTOSH  
SHANKAR  
RORUSER  
SATISH

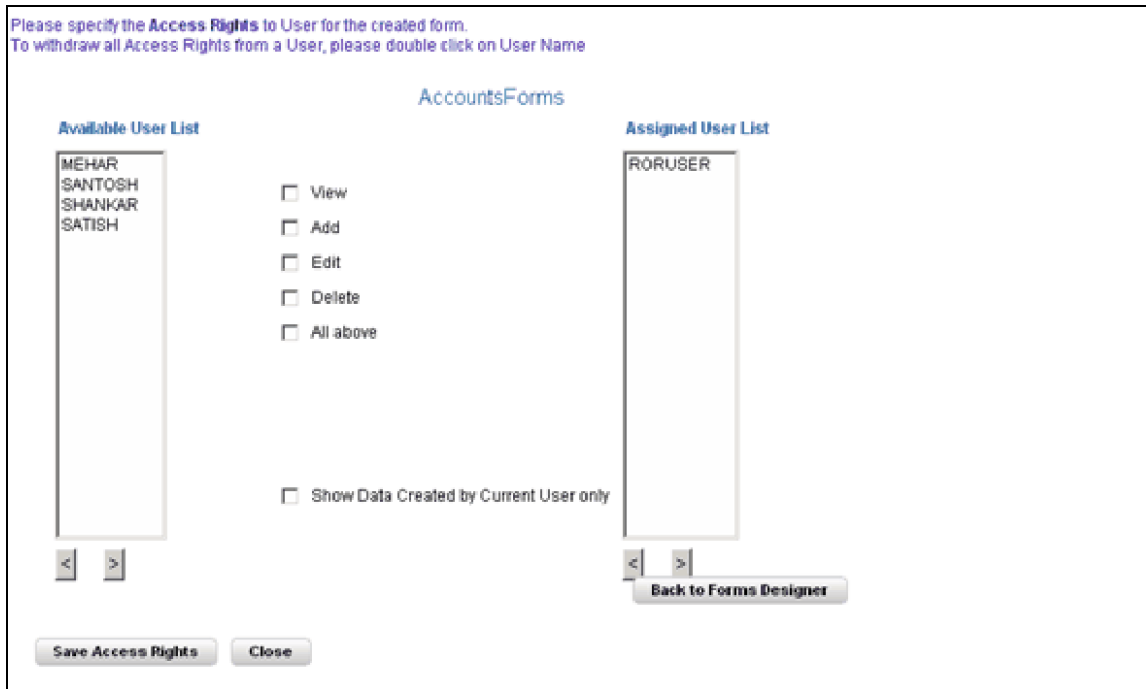
Assigned User List

View  
 Add  
 Edit  
 Delete  
 All above

Show Data Created by Current User only

Save Access Rights Close Back to Forms Designer

Select the user to whom you want assign the rights to from the Available User List pane and select the rights you want to assign to the user. You can assign View, Add, Edit, Delete or All above options. The user name will move to the Assigned User List once it is saved.



The Authorize and Auto-Authorize options are applicable for all the forms that have been saved with the Authorize option.

Click Save Access Rights to save the rights you have defined for the selected Users. Click Close to close the Assign Rights screen.

To go back to the forms designer, click the Back to Forms Designer button. When you have finished assigning users, click Finish to return to the splash screen.

To open help for the Forms Designer screen click Help. To return to the DI splash screen, click Home.

## Message Type Maintenance

For the maintenance of the Message Types, select the Message Type Maintenance option in the first screen of the Forms Designer Wizard.

Select one of the following options...

STEP

1

2

3

4

5

6

7

8

9

Create a New Form

Alter Existing Forms

Copy Forms

Delete Forms

Assign Rights

Message Type Maintenance

Next> Cancel

Click Next to continue. The Message Type Maintenance screen is displayed.

**Message Type Maintenance**

**Message Type** Batch Commit

**Message Subject** Infodom - \$INFODOM\$, Application - \$APPLNAME\$, Form -\$FORMNAME\$

**Message Content** Batch Transactions created and modified. Testing

**Available Recipients**

**Mapped Recipients** CREATOR  
AUTHORIZER

Save Cancel

Select the message category from the Message Type drop-down list. The message types are the same as those available in the Message Details screen as explained earlier.

Canceled Request for Creation

The Message Subject field displays the subject line that will be displayed for the particular Message Type selected. You can make the required modifications in this field.

Infodom - \$INFODOM\$, Application - \$APPLNNAME\$, Form -\$FORMNAME\$ - Cancel Request For Creation

The Message Content field displays the content of the message for the particular Message Type selected. You can make the required modifications in this field.

Infodom - \$INFODOM\$, Application - \$APPLNNAME\$, Form -\$FORMNAME\$ - Cancel Request For Creation

Double-click the Recipient name in the Available Recipients pane. The selected recipient/s moves to the Mapped Recipients pane.

Authorizer

---

**Note:** If you select Authorizer, the message will be sent to all the selected authorizers for authorization.

---

Click Save to save the changes. On clicking Cancel, a message requesting you to confirm the cancellation without saving is displayed. Click OK to discard the changes.

## Entering Data

Click Data Entry to open the screen. In this screen you can view, edit, add, delete or authorize data. You can view, edit, add, delete or authorize based on the rights that are assigned to you during form creation.

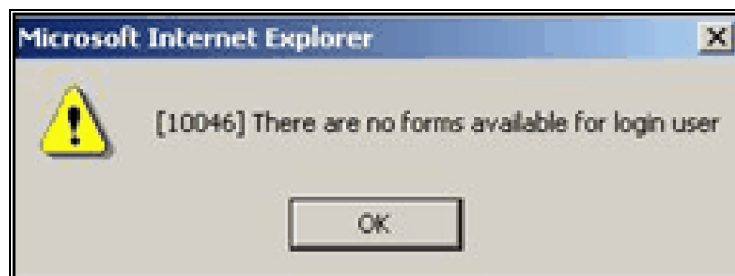
You can query the record to get specific data using search option. You are also provided with the option to export the result set.

---

**Note:** To open multiple Data Entry windows, specify the URL of the Data Entry window as the value in the URL box in the Design Oracle Menu screen. It enables the user to perform the search operation in more than one screen.

---

For more details on designing the Oracle Menu section, see *Oracle Financial Services Analytical Applications Infrastructure User Guide*.



If the logged-in-user is not mapped with the forms and applications it will display an alert saying “There are no forms available for the login user”.

The Forms frame displays the list of existing forms. The forms that are displayed are dependent on the rights assigned to user.

---

**Note:** The roll back option can be used only for authorized record to undo the modifications made to it.

---



## Viewing Form Details

Forms are displayed in View format by default. The forms are displayed based on the application names in the left pane. Based on the rights that are assigned to the user, he can access View, Edit, Add, Delete options. Click a form to view the details. The form details will be displayed in the layout that it was defined.

### Application Name: CAT Test

Form Name: Auth

In case of grid layout, the number of rows displayed is based on the number of rows that you specify in the Display ---Rows field.

Display 5 Rows

To view all the records in the table click Next to go to the next set of records. To return to the previous set of records, click Back.

It is recommended that you assign an easily viewable number of rows to be displayed. The user can view records as a Single Record, Editable View, Grid layouts, multi column or wrapped rows layout.

---

**Note:** The roll back option can be used only for authorized record to undo the modifications made to it.

---

## Single Record layout

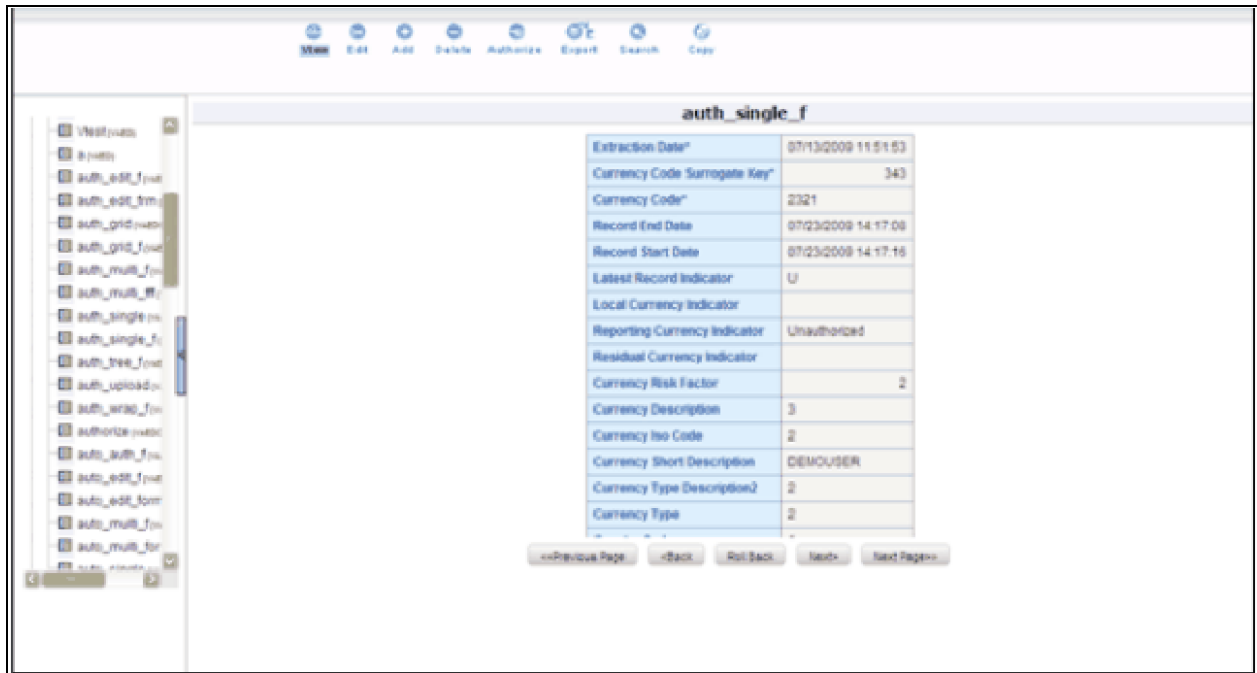
To view a single record at a time, click Single Record.

To view all the records in the table click Next to go to the next record. To return to the previous record, click Back.

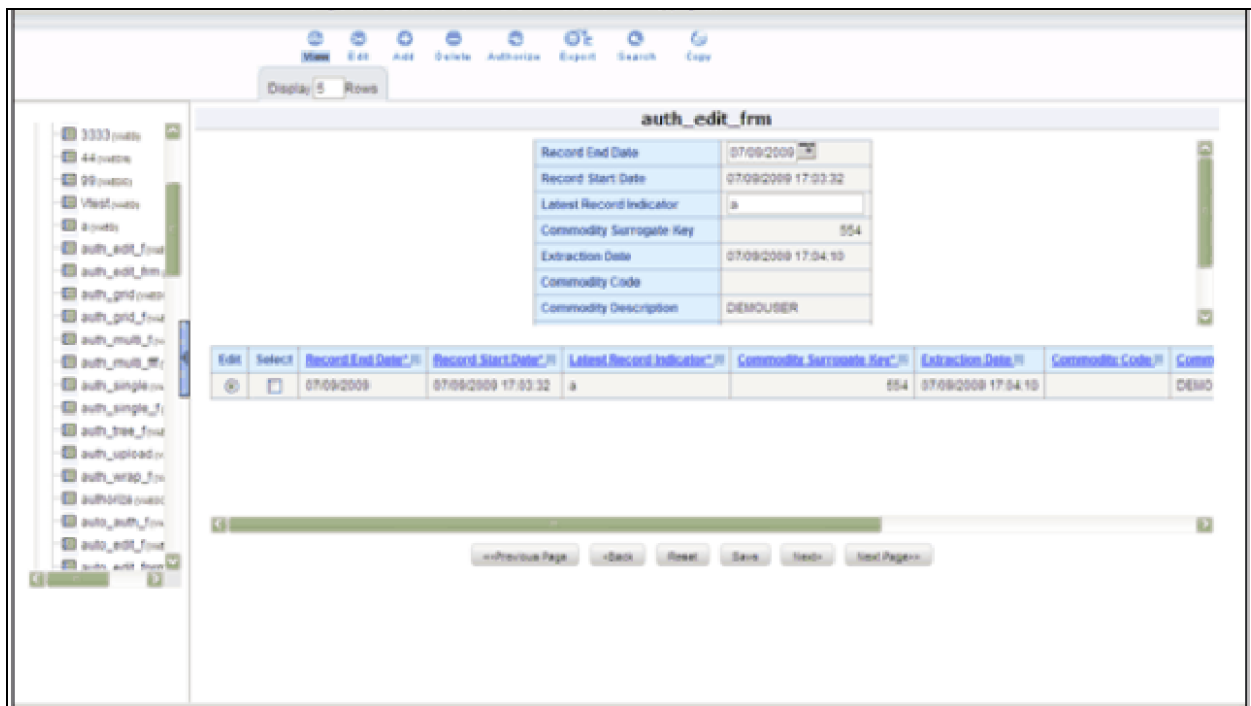
---

**Note:** The roll back option can be used only for authorized record to undo the modifications made to it.

---



To view a record in the editable view, click Editable View. In the editable view you can view and edit records, which can be selected from the grid of records.



The editable view format displays a single record in the top frame, which can be edited. The grid layout is displayed in the frame below. To edit a record, select the Edit option against the record in the grid layout. The selected record will be displayed as a single record with editable fields. The main advantage is you can easily compare the changes you have made in the data with the data shown in grid format

The number of records to be displayed in the form can be specified in the Display Rows text box.

To save the changes that you have made to the field/s, click Save. Click Reset to return the record settings to its original field settings.

## Grid Layout

To view a record in a grid layout, click Grid. This is the default layout that is displayed while viewing a form.

Select	Extraction Date	Currency Code	Record End Date	Record Start Date	Latest Record Indicator	Local Currency
<input type="checkbox"/>	07/13/2009 11:51:53	343	2321	07/13/2009 11:59:15	U	
<input type="checkbox"/>	12/31/1999	2	EUR	07/10/2009 16:30:24	07/10/2009 16:30:05	N
<input type="checkbox"/>	12/31/1999	3	JPY	12/31/1999	11/30/1999	Y
<input type="checkbox"/>	12/31/1999	4	GBP	12/31/1999	11/30/1999	N
<input type="checkbox"/>	12/31/1999	5	EUR	07/10/2009 17:05:43	07/10/2009 17:05:43	N

To sort a record based on a field, click the column title hyperlink.

Click FIC\_MIS\_DATE column title hyperlink to sort fields alphabetically based on the MIS Date.

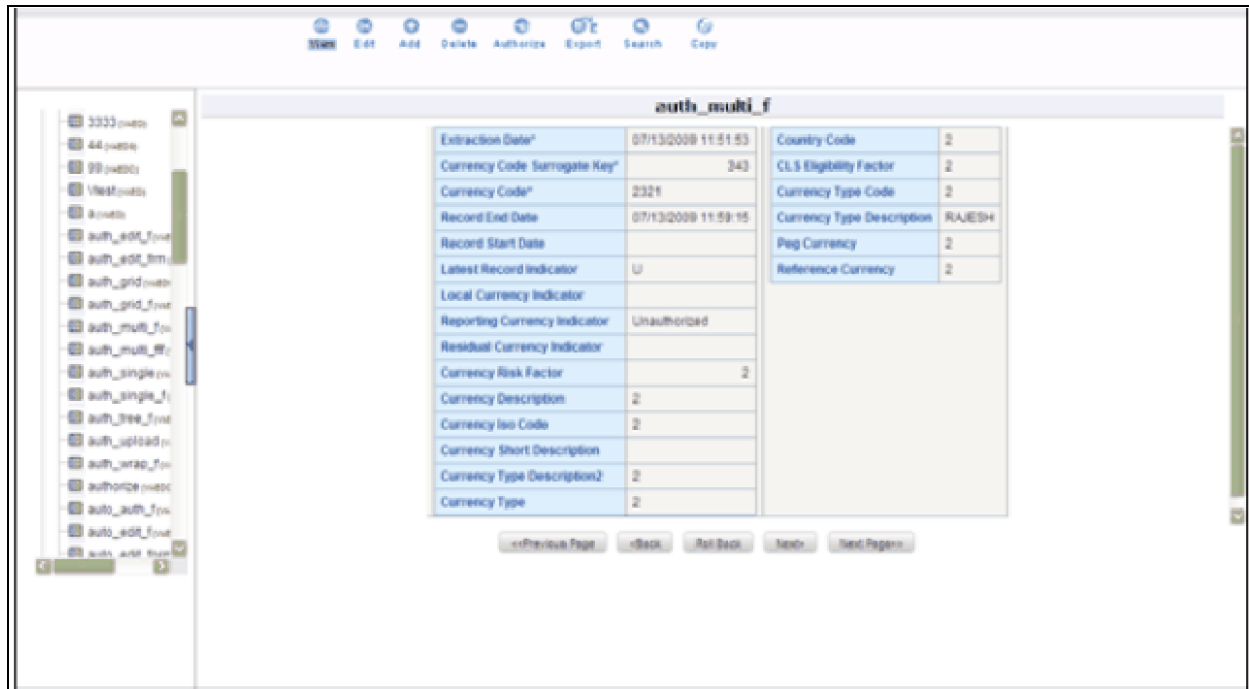
The number of records to be displayed in the form can be specified in the Display Rows text box.

To view all the records in the table click Next. To return to the previous set of records, click Back.

**Note:** The roll back option can be used only for authorized record to undo the modifications made to it.

## Multi Column Layout

To view all the columns of a selected record, click Multi Column. This layout enables you to view a record without having to scroll or with minimum scrolling to view all the columns.



To view all the records in the table, click Next. To return to the previous record, click Back.

**Note:** The roll back option can be used only for authorized record to undo the modifications made to it.

### Wrapping Row Layout

To view all the rows of a selected record, click Wrapping Row. This layout enables you to view a wrapping row easily without having to scroll horizontally to view all the data.

Extraction Date*	Currency Code Surrogate Key*	Currency Code*	Record End Date	Record Start Date
07/13/2009 11:51:53	343	2321	07/13/2009 11:59:15	
Latest Record Indicator	Local Currency Indicator	Reporting Currency Indicator	Residual Currency Indicator	Currency Risk Factor
U		Unauthorized		2
Currency Description	Currency Iso Code	Currency Short Description	Currency Type Description2	Currency Type
2	2		2	2
Country Code	CLS Eligibility Factor	Currency Type Code	Currency Type Description	Peg Currency
2	2	2	RAJESH	2
Reference Currency	2			

To view all the records in the table click Next to go to the next record. To return to the previous record, click Back.

**Note:** The roll back option can be used only for authorized record to undo the modifications made to it.

## Editing a Form

To edit a form, check the select option in the Select column and click **(edit)** button. The form fields are displayed with editable fields, which you can modify if required.

To save the changes that you have made to the field/s, click Save. Click Reset to return the record/table settings to its original field settings.

The records also can be edited in the layouts mentioned earlier (in View mode). All the fields will be displayed in the editable grid, but the primary key fields cannot be edited so will be displayed in un-editable format.

Click Next to edit the next set of records. To return to the previous records, click Back.

## Roll Back

The Roll Back option can be used only for authorized records. After the records are edited and saved, the roll back operation can be performed in view mode by selecting the particular record to undo the modifications made to it.

Once the record is edited using Edit option, its modified flag is set and the record with its modified flag set alone can be rolled back.

If the record is not an authorized record it will display a alert saying roll back functionality is available only for records which has its authorization flag set to “Modified”.

## Adding Rows to a Form

To add data to the selected table in a form, specify the number of rows to be added in Rows field in the options frame and click **(add)** button.

Enter data in these new fields and save them by clicking Save. Click Reset to return the table settings to its original field settings.

**Note:** If the user want to see the display with the thousands separator for any numeric columns, the input should be given in the same manner.

## Deleting a Record in a Form

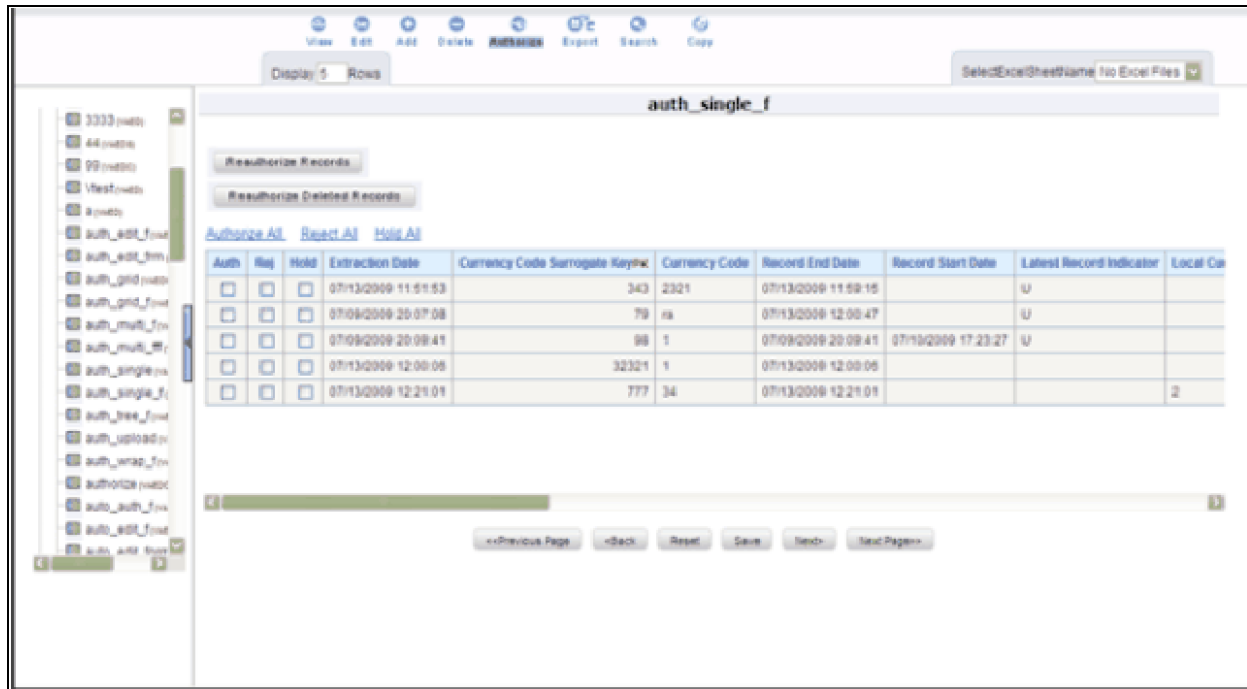
To delete a record, check the select option in the Select column against the record you wish to delete and click delete.

A warning message will be displayed prompting you to confirm deletion. Click Ok to delete the record or Cancel to discard deletion.

## Authorizing Data

The Data Entry module allows you to authorize data entered during form creation provided the table contains a primary key field.

To authorize data, select the form and click Authorize. The records that are required to be authorized will be displayed with Auth, Rej, au\_fname, au\_idPK (Primary Key identifier), au\_lname, AuthFlag (representing the status of the record), contract, MakerDate, MakerID and phone details. If there are no records to authorize, the screen will display the message No records to be authorized.



**Note:** The primary key fields in the record are displayed in red as PK.

You can authorize only those records that are created by another user if you have authorization rights. If you do have authorization rights, the record created by another user, the AuthFlag will display U, which suggests that it is an unauthorized record requiring authorization.

If the record is already authorized, the AuthFlag will display A. If any record is rejected, if the form is authorized, then the AuthFlag will be displayed as R. If the Authorized user put the records on Hold, then the AuthFlag will be displayed as UH. After Authorizing, the Auth Flag will be set to A.

Any changes (modification / deletion) made to the records after authorization will be marked for authorization and can only be authorized by the users who have the privilege. The new records created before authorization can be edited or deleted by any user who has the privilege to do the same.

To authorize a record, select the Auth option against the record you wish to authorize and click Save. To reject/delete a record, select the check box in the Reject column against the record and click Save. A warning message is displayed. The records selected for deletion will be deleted from the table. Proceed? Click OK to confirm deletion or click Cancel to delete the request.

**Note:** You can authorize forms only if the authorize rights have been assigned to you.

To authorize the next set of records, click Next. To authorize all the records in the form, select the Authorize all option and click Save.

To reject all records, select the Reject all and click Save. If the authorizer is not sure whether to authorize the records the Hold all option can be used, which can be authorized or rejected at the later point.

## Re-authorizing Records

The Records to be reauthorized link will be displayed only if some other user has modified the authorized records. An authorized record that has been modified will display the AuthFlag M, which suggests that the record have been modified. To reauthorize modified records, click the Records to be re-authorized link to open a new screen.

**Modified Record Authorization :**

Authorizer : NEWUSER2 Authorization Date : 07/13/2009 12:19:20

Auth	Rej	Hold	Extraction Date	Currency Code	Surrogate KeyPK	Currency Code	Record End Date	Record Start Date	Latest Record Indicator	Local Currency Indicator	Reporting Currency
Edited Data :											
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	07/23/2009 14:20:25		666	3	07/13/2009	07/23/2009 14:20:25	0		Modified
Original Data :											
			07/13/2009 12:14:49		666	3	07/13/2009 12:19:20	07/13/2009 12:20:10	0		Modified

Navigation buttons: <<Previous Page, <Back, Reset, Close, Save, Next, Next Page>>

The record that needs re-authorization is displayed with new details. The original authorized record is also displayed. To re-authorize/reject/put-on-Hold the records select the Auth, Rej or On-Hold option against it and click Save.

Click Next to authorize the next set of records,. To close the screen, click Close. Reset Button will reset the check-boxes to null.

A mail will be sent to the creator that records are authorized/rejected/put-on-hold.



## CHAPTER 5

## OFSAA Attributes

## Overview

Attributes refers to the distinguished properties or qualifiers that describes a dimension member. Attributes may or may not exist for a simple dimension. Attributes section is available within the Dimension Management section of Financial Services Applications module.

You (Business Analysts) need to have ETL Analyst function role mapped to access Attributes.

This chapter describes the creation and maintenance of OFSAA Attributes. Specifically, this chapter will cover:

- Attribute Summary & Detail Screens
- Navigation within the Attribute Summary Screen
- Navigation within the Attribute Detail Screen

See *OFSAA Infrastructure* for details on how Attributes are used in OFS Analytical Applications. See *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration Guide* for details on bulk loading dimension attributes

## Summary &amp; Detail Screens

Upon initially navigating to Common Object Maintenance > Dimension Management > Attributes, a summary screen is displayed showing all of the defined Attributes for one dimension (your first dimension alphabetically that supports attributes). By selecting a dimension and by using search criteria, you can control the set of Attributes that are displayed. When you Add, Edit, Copy, or View an Attribute, a detail screen is displayed.

Numeric Code	Name	Data Type	Required
1	Band Lower Bound Value	Number	No
3	Band Type	VARCHAR	No
2	Band Upper Bound Value	Number	No

## Navigation within the Summary Screen

When you first navigate to the Attributes summary screen, the Attributes associated with your first dimension are presented in a summary grid. The Attributes summary screen has three containers:

- Dimension
- Search
- Dimension Attributes

### Dimension Container

Attributes are properties of dimensions. To add or maintain an attribute, you must first select the dimension to which an attribute belongs. The Dimension drop-down list in the Dimension container displays all of your Key Processing dimensions (both seeded and user-defined) and any Standard Dimensions that are configured to support attributes.

### Search Container

Among other properties, each attribute possesses a Code value, a Name, and a Data Type. You may search on any of these properties in the Search container.

The screenshot shows a search interface with a search bar on the left containing the text "Search". To the right of the search bar are three input fields: "Numeric Code" (a text box), "Name" (a text box), and "Data Type" (a dropdown menu).

### Search Control

Enter your desired search criteria and click the Search control.

### Reset Control

Clicking the Reset control clears any search criteria you may be using and refreshes the screen.

### Dimension Attributes Container

The Dimension Attributes container presents a grid containing all of the Attributes that meet your search criteria. The Dimension Attributes summary grid offers several controls that allow you to perform different functions when an Attribute is selected.

The screenshot shows a table with columns: Numeric Code, Name, Data Type, and Required. There are three rows of data. Above the table is a toolbar with icons for Add, View, Edit, Copy, Check Dependencies, and Delete. Below the table is a pagination control showing "Page 1 of 1 (1-3 of 3 items)" and navigation arrows.

<input type="checkbox"/>	Numeric Code	Name	Data Type	Required
<input type="checkbox"/>	1	Band Lower Bound Value	Number	No
<input type="checkbox"/>	3	Band Type	VARCHAR	No
<input type="checkbox"/>	2	Band Upper Bound Value	Number	No

To select an Attribute, click a check box in the first column of the grid. More than one Attribute can be selected at a time but this will cause some of the controls to become disabled. Clicking a check box a second time de-selects an Attribute.

You may select or deselect all of the Attributes in the summary grid by clicking the check box in the upper left hand corner of the summary grid directly to the left of the Code column header.

### **Add**

Clicking the Add control begins the process of building a new Attribute. The Add control is disabled if any rows in the grid have been selected.

### **View**

Selecting a single row out of the grid enables the View control. Clicking the View control allows you to view the detailed definition of an Attribute on a read-only basis. The View control is only enabled when a single Attribute has been selected.

### **Edit**

Selecting a single row out of the grid enables the Edit control. Clicking the Edit control allows you to modify an existing Attribute. The Edit control is only enabled when a single Attribute has been selected.

### **Copy**

Selecting a single row out of the grid enables the Copy control. Clicking the Copy control allows you to create a copy of an existing Attribute. The Copy control is only enabled when a single Attribute has been selected.

### **Check Dependencies**

Select an attribute and then click the Check Dependencies control to generate a report on all dimension members that utilize your selected attribute. For example, if you have created an attribute named Color and have assigned values of Red, White, and Blue to dimension members #1, #2, and #3, Checking Dependencies for the Color attribute will result in a report indicating that dimension members #1, #2, and #3 utilize the Color attribute.

The Check Dependencies control is only enabled when a single attribute has been selected.

### **Delete**

Selecting one or more rows out of the grid enables the Delete control. Clicking the Delete control deletes the Attribute(s) you have selected. OFSAAI will not allow you to delete Attributes which have any dependencies (see Check Dependencies above).

## **Dimension Attributes Summary Grid**

The following columns categorize each Attribute in the summary grid:

- Numeric Code
- Name
- Data Type
- Required
- Seeded

### **Numeric Code**

Displays the Attribute's Code value. You can also manually enter the code in the Numeric Code field.

**Name**

Enter the Attribute's short name.

---

**Note:** The characters '&' ' ' are restricted in the name field.

---

**Description**

Enter the description.

---

**Note:** The characters '~&+ ' "@ are restricted in the description field.

---

**Data Type**

Select the Data Type as DATE, DIMENSION, NUMBER, or STRING from the drop-down list.

If NUMBER is selected as the Data Type: The Scale field is enabled with "0" as default value. Enter a Scale value  $\geq 0$ . If it is left as 0, values for this attribute will be limited to Integers. If you wish to enable decimal entries for this attribute, the maximum Scale value must be  $> 0$  and  $\leq$  the scale defined for NUMBER\_ASSIGN\_VALUE in the dimension's underlying attribute table. Please refer to the *Data Model Utilities Guide* for further details on the attribute table.

**Required**

Select Yes or No. If this is set to No, an attribute value is optional for the associated dimension members.

---

**Note:** This field is disabled in Add and Edit modes if any members already exist for the Dimension upon which this attribute is defined.

---

**Default Value**

If Required Attribute is set to Yes, a Default Value must be entered, otherwise it is optional.

If **DIMENSION** is selected as the Data Type:

- Select the required Dimension from the drop-down list in the Dimension field.
- Select the Default Value from the drop-down list of members mapped to the selected Dimension.

If the required Member is not listed in the drop-down then select **--More--** and the Member Search window is displayed. For more information refer search.

If **NUMBER** is selected as the Data Type:

- Enter a numeric value in the Default Value field, and it must be consistent with the Scale you have defined.

If **DATE** is selected as the Data Type:

- Click Calendar button to select a valid date as the Default Value from the calendar.

If **STRING** is selected as the Data Type:

- Enter alphanumeric value in the Default Value field.
- The Maximum characters allowed in Default value field for String Data Type is 1000.

## Navigation within the Detail Screen

When you Add, Edit, Copy, or View an Attribute, the Attributes Detail screen is displayed.

The screenshot shows the 'Attributes' detail screen. At the top, there is a 'Dimension' dropdown menu set to 'Bands'. Below it is a search section with fields for 'Numeric Code', 'Name', and 'Data Type'. A table of attributes is displayed below the search fields. The table has columns for 'Numeric Code', 'Name', 'Data Type', 'Required', and 'Seeded'. The table contains three rows of data.

Numeric Code	Name	Data Type	Required	Seeded
1	Band Lower Bound Value	Number	No	Yes
3	Band Type	VARCHAR	No	Yes
2	Band Upper Bound Value	Number	No	Yes

Page 1 of 1 (1-3 of 3 items) Records Per Page 3

## Attribute Functionality

See Overview of OFSAA Infrastructure for details on how Attributes are used in OFS Analytical Applications.

## Dimension Container

Displays the name of the dimension to which an Attribute belongs. When Editing or Copying an existing Attribute, this is a read-only value. When Adding a new Attribute, you must select the dimension to which you want your new attribute to be assigned.

The screenshot shows a 'Dimension' container with a dropdown menu. The dropdown is currently set to 'Bands'.

## Attribute Details Container

### Numeric Code

When Adding a new Attribute, you must select a Code value to which you want your new attribute to be assigned. Code values for each attribute belonging to a dimension must be unique, but you are otherwise free to pick any numeric value you wish. You may also click the Generate Code shortcut key to automatically generate a new Code value.

The screenshot shows the 'Attribute Details' container. It contains three input fields: 'Numeric Code' (value: 5556), 'Name' (value: Customer Type), and 'Description' (value: Customer Type).

You must supply a Code value before you can save a new Attribute definition and you may not change a Code value once an Attribute definition has been saved.

**Note:** When Editing an existing Attribute, Code is a read-only value; and when Copying an existing Attribute, you must supply a new Code value

### Name

Used to name or re-name your Attribute. You must supply a Name (or a new name if you wish to re-name your Attribute) before you can save an Attribute definition.

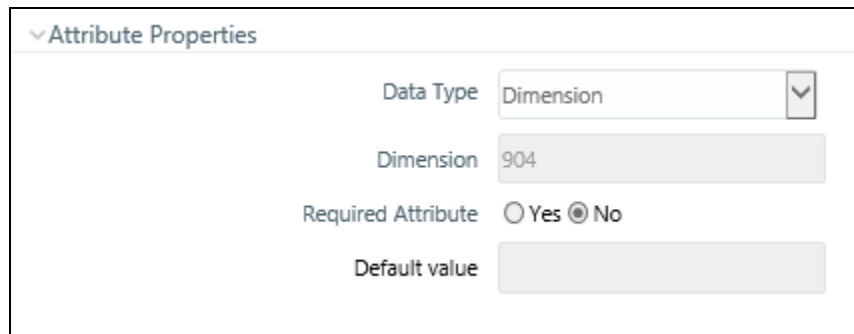
### Description

Used to store a longer description of your Attribute.

## Attribute Properties Container

### Data Type

Select a Data Type for your Attribute. For Attributes that have already been saved, you may not modify Data Type. If you have selected the Date, Number, or String Data Type, the Dimension drop-down list is disabled.



Attribute Properties

Data Type: Dimension

Dimension: 904

Required Attribute:  Yes  No

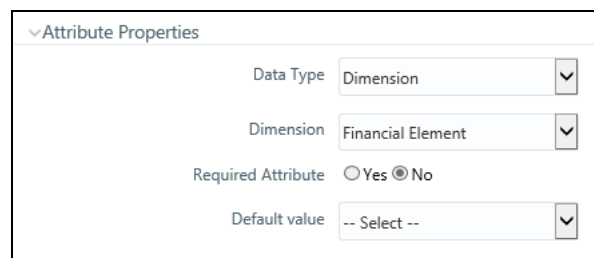
Default value:

- If **DATE** is selected as the Data Type:

Click on following button to select a valid date as the Default Value from the calendar.



- If **DIMENSION** is selected as the Data Type:
  - If you have selected the Dimension Data Type, the Dimension drop-down list is enabled. Select the required Dimension from the drop-down list in the Dimension field.



Attribute Properties

Data Type: Dimension

Dimension: Financial Element

Required Attribute:  Yes  No

Default value: -- Select --

- Select the Default Value from the drop-down list of members mapped to the selected Dimension. If the required Member is not listed in the drop-down list then select --More—and the Member Search screen is displayed.

For more information refer search.

- If **NUMBER** is selected as the Data Type:  
The Scale field is enabled with “0” as default value.

Enter the scale value between 0 and 5.

The Scale can be edited to a value between 0 and 5 thereby making it a decimal. Scale value less than scale defined is permitted. Scale value greater than scale value is automatically trimmed.

Key in valid numeric in the Default value field on which the Attribute is to be defined.

- If **STRING** is selected as the Data Type:  
Enter alphanumeric value as the Default Value in the Default value field.

### Required Attribute

Various OFSAA engines utilize attributes within their processing. For example, both OFSAA Transfer Pricing and OFSAA Profitability Management utilize the Account Type attribute in order to understand whether a balance is a debit balance or a credit balance, on balance sheet or off balance sheet.

You may not create user-defined Required attributes.

### Default Value

Use to set a default value for your attribute. When you create a new dimension member, each attribute that has a Default Value is set to its default value for your convenience.





## CHAPTER 6

## OFSAA Dimension Members

## Overview

Members refers to the individual items that constitutes a dimension when data is categorized into a single object.

For example, Product, Organization, Time, and so on. Members is available within Dimension Management section of the Infrastructure system.

You (Business Analysts) need to have ETL Analyst function role mapped to access Members.

This chapter describes the creation and maintenance of OFSAA Hierarchies. Specifically, this chapter will cover:

- Member Summary & Detail Screens
- Navigation within the Member Summary Screen
- Navigation within the Member Detail Screen

See Overview of OFSAA Infrastructure, page 2-1 for details on how Members are used in OFS Analytical Applications. See the *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration Guide* for details on bulk loading members.

## Summary &amp; Detail Screens

Upon initially navigating to Common Object Maintenance > Dimension Management > Members, a summary screen is displayed showing all of the defined Dimension Members for one dimension (your first dimension alphabetically). By selecting a dimension and by using search criteria, you can control the set of Dimension Members that are displayed. When you Add, Edit, Copy, or View a Dimension Member, a detail screen is displayed.

Members

Members

Dimension: Accrual Basis

Search: Search Reset

Alphanumeric Code: [ ]

Numeric Code: [ ]

Name: [ ]

Description: [ ]

Enabled:  Yes  No

Is Leaf:  Yes  No

Attribute Name: [ ]

Attribute Value: [ ]

+ Add View Edit Copy Check Dependencies Delete

Alphanumeric Code	Numeric Code	Name	Is Leaf
<input type="checkbox"/> 30/360	1	30/360	No
<input type="checkbox"/> 30/365	4	30/365	No
<input type="checkbox"/> 30/A	5	30/Actual	No
<input type="checkbox"/> A/360	2	Actual/360	No
<input type="checkbox"/> A/365	6	Actual/365	No
<input type="checkbox"/> A/A	3	Actual/Actual	No
<input type="checkbox"/> B/252	7	Business/252	No
<input type="checkbox"/> NONE	-1	None	No

Page 1 of 1 (1-8 of 8 items) K < > X

Records Per Page: 8

## Navigation within the Summary Screen

When you first navigate to the Dimension Members summary screen, the Dimension Members associated with your first dimension are presented in a summary grid. The Dimension Members summary screen has three containers:

- Dimension
- Search
- Dimension Members

### Dimension Container


To add or maintain a Dimension Member, you must first select the dimension to which a Dimension Member belongs. The Dimension drop-down list box in the Dimension container displays all of your dimensions that are configured to be viewable (not every dimension is viewable through the application user interface; for details on which dimensions are viewable, see the *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration Guide*).



The screenshot shows a container titled "Dimension" with a dropdown menu. The dropdown is currently set to "Accrual Basis".

### Search Container

Each Dimension Member possesses a Code value, a Name, and a Description. Dimension Members also have the properties of being enabled or disabled and of being a leaf member (last-descendent-child value) or a rollup member (representing a rollup point in a hierarchy). Finally, each Dimension Member value is assigned an attribute value for each its attributes. You may search on any of these properties in the Search container.



The screenshot shows a search container with the following fields and controls:

- Alphanumeric Code:
- Numeric Code:
- Name:
- Description:
- Enabled:  Yes  No
- Is Leaf:  Yes  No
- Attribute Name:
- Attribute Value:

Search and Reset buttons are located in the top right corner.

### Search Control

Enter your desired search criteria and click the Search control.

### Reset Control

Clicking the Reset control clears any search criteria you may be using and refreshes the screen.

## Dimension Members Container

Dimensions are comprised of finite lists of Dimension Members. The Dimension Members container presents a grid containing all of the Dimension Members that meet your search criteria. The Dimension Members summary grid offers several controls that allow you to perform different functions when a Dimension Member is selected.

To select a Dimension Member, click a check box in the first column of the grid. More than one Dimension Member can be selected at a time but this will cause some of the controls to become disabled. Clicking a check box a second time deselects a Dimension Member.



You may select or deselect all of the Dimension Members in the summary grid by clicking on the check box in the upper left hand corner of the summary grid directly to the left of the Code column header.

### Add

Clicking the Add control begins the process of building a new Dimension Member. The Add control is disabled if any rows in the grid have been selected.

### View

Selecting a single row out of the grid enables the View control. Clicking the View control allows you to view the detailed definition of a Dimension Member on a read-only basis. The View control is only enabled when a single Dimension Member has been selected.

### Edit

Selecting a single row out of the grid enables the Edit control. Clicking the Edit control allows you to modify an existing Dimension Member. The Edit control is only enabled when a single Dimension Member has been selected.

### Copy

Selecting a single row out of the grid enables the Copy control. Clicking on the Copy control allows you to create a copy of an existing Dimension Member. The Copy control is only enabled when a single Dimension Member has been selected.

### Check Dependencies

To generate a report on the hierarchies that utilize a selected Dimension Member, select a single Dimension Member and click the Check Dependencies control. The Check Dependencies control is only enabled when a single Dimension Member has been selected.

### Delete

Selecting one or more rows out of the grid enables the Delete control. Clicking the Delete control deletes the Dimension Member(s) you have selected.

---

**Note:** OFSAA Infrastructure will not allow you to delete any Dimension Member that is utilized within a hierarchy (see Check Dependencies above).

---

## Dimension Members Summary Grid

The following columns categorize each Dimension Member in the summary grid:

- Alphanumeric Code
- Numeric Code
- Name
- Is Leaf

<input type="checkbox"/> Alphanumeric Code	Numeric Code	Name	Is Leaf
<input type="checkbox"/> 30/360	1	30/360	No
<input type="checkbox"/> 30/365	4	30/365	No
<input type="checkbox"/> 30/A	5	30/Actual	No
<input type="checkbox"/> A/360	2	Actual/360	No
<input type="checkbox"/> A/365	6	Actual/365	No
<input type="checkbox"/> A/A	3	Actual/Actual	No
<input type="checkbox"/> B/252	7	Business/252	No
<input type="checkbox"/> NONE	-1	None	No

Page 1 of 1 (1-8 of 8 items) Records Per Page 8

### Alphanumeric Code

Displays the Dimension Member's code value.

### Numeric Code

Displays the Dimension Member's numeric code value.

### Name

Displays the Dimension Member's short name. Performing a “mouse-over” on a Dimension Member Name displays the Dimension Member's description.

### Is Leaf

Displays the Dimension Member's “Is Leaf?” property. Members may be declared to be either “leaf members” or “rollup members”. When building a hierarchy in OFSAAI, “leaf members” are those dimension members that have no descendent dimension members. In building a dimension member set, you must have at least some rollup members (Is Leaf = No) before you can construct a multilevel hierarchy.

Your business data is normally expressed at the leaf level. An individual mortgage instrument record, for example, would belong to a (leaf level) cost center for a (leaf level) principal balance General Ledger Account and would carry a (leaf level) designation of Product.

At least one leaf level dimension member is required for hierarchies used in allocations.

## Navigation within the Detail Screen

When you Add, Edit, Copy, or View a Dimension Member, the Dimension Members Detail screen is displayed.

## Dimension Member Functionality

See Overview of OFSAA Infrastructure for details on how Dimension Members are used in OFS Analytical Applications.

## Dimension Container

Displays the name of the dimension to which a Dimension Member belongs. When Editing or Copying an existing Dimension Member, this is a read-only value. When Adding a new Dimension Member, you must select the dimension to which you want your new Dimension Member to be assigned.

## Dimension Member Details Container

### Alphanumeric Code

The **Alphanumeric Code** field is editable only if the selected Dimension accepts Alphanumeric Code. For example, **Billing Method** Dimension. Else, the field is Read Only and the value is fetched from the **Numeric Code** field entered. Enter the required Alphanumeric Code. Ensure that the code has a maximum of 14 characters and there are no special characters like & ' ~ " @ + included.

### Numeric Code

Enter the Numeric Code by doing any of the following:

- To auto-generate a Numeric Code, click the following button.



A system generated code is displayed.

- Manually enter the required code which is auto validated for uniqueness. A maximum of 14 numeric characters can be specified

---

**Note:** If the selected Dimension accepts only Numeric Code, then the specified Numeric Code is auto populated to the **Alphanumeric Code** field also.

---

### **Name**

Enter the **Name** of the Member.

---

**Note:** The characters &' ” are restricted

---

### **Description**

Enter the required Description for the Member.

---

**Note:** The characters ~&+ ' "@ are restricted.

---

### **Enabled**

This field is set to **Yes** by default and is editable only in **Edit** screen.

---

**Note:** You can change the option to No only when the particular member is not used in any hierarchy. The disabled members will not be displayed in Hierarchy rules, or UIs which are based on Hierarchies, such as Hierarchy Filters and hierarchical assumption browsers used in applications.

---

### **Is Leaf**

This field is set to **Yes** by default.

If Yes, the particular member can be used as a leaf node in any hierarchy and child cannot be added to this node.

If No, the node becomes a non leaf and can have child nodes.

---

**Note:** A member created as Non Leaf having child nodes to it in any hierarchy cannot be made Leaf.

---

### **Copy Attribute Assignment From**

---

**Note:** If the Dimension is selected as “Common Chart of Accounts”, then **Copy Attribute Assignment From** option is enabled.

---

Instead of manually completing all the attribute value assignments for a new Dimension Member, you may copy attribute assignments from a pre-existing Dimension Member. To do so, click the ellipses adjacent to **Copy Attribute Assignment From** field to open the Member Browser window. You may select a member directly or you may search for the member whose attributes you wish to copy. Once you have selected a member and clicked OK, the attribute values of the member you selected are copied into the definition of your new Dimension Member.

### Members

Dimension: Common Chart of Accounts

Search Q ↻

Alphanumeric Code       Numeric Code

Name

Description

Enabled  Yes  No      Is Leaf  Yes  No

Attribute Name        Attribute Value

Dimension Members 1 - 20 / 21 K < > >










Alphanumeric Code	Numeric Code	Name
1000295	1000295	2ND-1000295
100000001	100000001	Adj type cd=0
821	821	Alloc Employee Benefits Expense
19	19	Annuity
98765432820043	98765432820043	AUTOD_CCOA1
2	2	COA MEMBER 2
3	3	COA MEMBER 3
101	101	Commercial Loans Fixed
50006	50006	Conv Adjustable: actual/360
10001	10001	Conventional Fixed
820	820	Employee Benefits Expense
EARNINGASSETS	100	fiscal year common
131	131	Installment Loan-Indirect
20	20	Leases

Q

## Dimension Member Attributes Container

If you have not copied attribute values from a pre-existing dimension member using the Copy Attribute Assignment From feature described above, assign them in the Member Attributes container.

- Dimension Member Attributes have defined data types including String, Number, Date, and Dimension. If an attribute is a Date data type, use the calendar control to select a date. If an attribute is a Number or String data type, enter your free-form value directly. If an attribute is a Dimension data type, select your value from the drop-down list.
- Attributes marked with an asterisk indicate required attributes.

Member Attributes			
Attribute		Value	
Account Type *		Off Balance Sheet	
Rollup Signage *		+	
Accrual Basis		-- Select --	
Common Chart of Account			
COMMON COA TYPE			
COMMON COA TYPE DESC			

**Default Values**

If a Member Attribute is defined as having a default value, the default value will be displayed when you create a new Dimension Member.



## CHAPTER 7

## OFSAA Hierarchies

**Overview of OFSAA Hierarchies**

Hierarchies refers to dimension members that are arranged in levels, with each level representing the aggregated total of the data from the level below. One dimension type can have multiple hierarchies associated with it. Hierarchies is available within the Dimension Management section of Infrastructure system. You (Business Analysts) need to have ETL Analyst function role mapped to access Hierarchies.

This chapter describes the creation and maintenance of OFSAA Hierarchies. Specifically, this chapter will cover:

- Hierarchy Summary & Detail Screens
- Navigation within the Hierarchy Summary Screen
- Navigation within the Hierarchy Detail Screen

See the *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration Guide* for details on bulk loading hierarchies.

**Summary & Detail Screens**

Upon initially navigating to Common Object Maintenance > Dimension Management > Hierarchies, a summary screen is displayed showing all of the defined Hierarchies for one dimension (your first dimension alphabetically that supports hierarchies). By selecting a dimension and by using search criteria, you can control the set of Hierarchies that are displayed. When you Add, Edit, Copy, or View a Hierarchy, a detail screen is displayed.

Name	Display Level	Created By	Creation Date	Last Modification Date
AUTOD_CCA_01	1	AUTOUSER	01/25/2016 15:03:36	05/26/2016 15:29:55

**Navigation within the Summary Screen**

When you first navigate to the Hierarchies summary screen, the Hierarchies that are stored within your current default folder and that are associated with your first dimension are presented in a summary grid. The Hierarchies Summary screen has three containers:

- Dimension

- Search
- Hierarchies

## Dimension Container

Select the Dimension of the hierarchies you wish to View. When you change your dimension selection in the summary screen, the screen refreshes and displays the Hierarchies that are stored within your current default folder.

**Note:** The dimensions drop-down list will only display dimensions that are configured to support hierarchies.

## Search Container

Your default Folder functions as a search constraint. The value of your default Folder is set in Application Preferences. You may select a different Folder or you may remove the Folder constraint entirely by selecting the “blank” Folder, i.e., no Folder. You may also search by Hierarchy Name, Dimension Member Code, and/or Dimension Member Name.

### Search Control

Searches for Hierarchy Name, Dimension Member Code, and Dimension Member Name are wildcard searches, i.e., searching for Hierarchy Names like “standard” will find any hierarchies that include “standard” within their names. When searching by Dimension Member Code or by Dimension Member Name, the application will find any or all hierarchies that utilize that dimension member but only if the member is not found in the hierarchy's “orphan” branch (see discussion of Orphan branches below). Enter your desired search criteria and click on the Search control.

### Reset Control

Clicking on the Reset control clears any search criteria you may be using, resets Folder to your default Folder, and refreshes the screen.

## Hierarchies Container

The Hierarchies container presents a grid containing all of the Hierarchies that meet your search criteria. The Hierarchies summary grid offers several controls that allow you to perform different functions when a Hierarchy is selected.

+ Add View Edit Copy Check Dependencies Delete					
<input type="checkbox"/>	Name	Display Level	Created By	Creation Date	Last Modification Date
<input type="checkbox"/>	AUTOD_CCA_01	1	AUTOUSER	01/25/2016 15:03:36	05/26/2016 15:29:55

Page 1 of 1 (1-1 of 1 items) K < > X Records Per Page 1

To select a Hierarchy, click on a check box in the first column of the grid. More than one Hierarchy can be selected at a time but this will cause some of the controls to become disabled. Clicking on a check box a second time deselects a Hierarchy.

You may select or deselect all of the Hierarchies in the summary grid by clicking on the check box in the upper left hand corner of the summary grid directly to the left of the Name column header.

### Add

Clicking on the Add control begins the process of building a new Hierarchy. The Add control is disabled if any rows in the grid have been selected.

### View

Selecting a single row out of the grid enables the View control. Clicking on the View control allows you to view the detailed definition of a Hierarchy on a read-only basis. The View control is only enabled when a single Hierarchy has been selected.

### Edit

Selecting a single row out of the grid enables the Edit control. Clicking on the Edit control allows you to modify an existing Hierarchy. The Edit control is only enabled when a single Hierarchy has been selected.

### Copy

Selecting a single row out of the grid enables the Copy control. Clicking on the Copy control allows you to create a copy of an existing Hierarchy. The Copy control is only enabled when a single Hierarchy has been selected.

### Check Dependencies

To generate a report on the OFSAA rules that utilize a selected Hierarchy, select a single Hierarchy and click on the Check Dependencies control. The Check Dependencies control is only enabled when a single Hierarchy has been selected.

### Delete

Selecting one or more rows out of the grid enables the Delete control. Clicking on the Delete control deletes the Hierarchy or Hierarchies you have selected.

**Note:** OFSAA Infrastructure will not allow you to delete any Hierarchy that is utilized by another OFSAA rule (e.g., a Transfer Pricing Processing rule, an Allocation rule, etc). See Check Dependencies above.

## Hierarchies Summary Grid

The following columns categorize each Dimension Member in the summary grid:

- Name
- Display Level
- Created By

- Creation Date
- Last Modification Date

Name	Display Level	Created By	Creation Date	Last Modification Date
<input type="checkbox"/> AUTOD_CCA_01	1	AUTOUSER	01/25/2016 15:03:36	05/26/2016 15:29:55

Page 1 of 1 (1-1 of 1 Items)    Records Per Page 1

**Name**

Displays the Hierarchy's Name. Performing a “mouse-over” on a Hierarchy's Name displays the Hierarchy's Description.

**Display Level**

Display Level is the default level at which a hierarchy is displayed by the application when you open it.

**Created By**

Created By displays the name of the user who created a Hierarchy. If a Hierarchy is saved as Read Only, only the user who created the rule, i.e., the owner, may modify the rule.

**Creation Date**

Displays the date and time at which a Hierarchy was created.

**Last Modification Date**

Displays the date and time at which a Hierarchy was last modified.

**Navigation within the Detail Screen**

When you Add, Edit, Copy, or View a Hierarchy, the Hierarchies Detail screen is displayed.

The screenshot shows the 'Hierarchies' detail screen. At the top, there are 'Save' and 'Cancel' buttons. The main area is divided into several sections:

- Dimension:** A dropdown menu set to 'Common Chart of Accounts'.
- Hierarchy Properties:**
  - Name:** A text input field.
  - Description:** A larger text input area.
  - Folder:** A dropdown menu set to 'AUTOSEG' with a plus icon.
  - Access Type:** Radio buttons for 'Read Only' (selected) and 'Read/Write'.
  - Automatic Inheritance:** Radio buttons for 'Yes' and 'No' (selected).
  - Display Signage:** Radio buttons for 'Yes' (selected) and 'No'.
  - Show Member Code:** A dropdown menu set to 'Only Name - No Code'.
  - Initial Display Level:** A dropdown menu set to '1 - Level 1'.
  - Orphan Branch:** Radio buttons for 'Yes' and 'No' (selected).
- Member Properties:** A panel on the right side with input fields for:
  - Alphanumeric Code
  - Numeric Code
  - Name
  - Description
  - Enabled (radio buttons for Yes/No)
  - Is Leaf (radio buttons for Yes/No)
  - Created By
  - Creation Date

At the bottom left, there are buttons for 'Show Hierarchy' and 'Show Results', and a note: 'Right click here to build tree' with an 'Orphan Branch' icon.

The Audit Trail container is a standard footer container for every OFSAA rule type. It displays Created By, Creation Date, Last Modified By, and Modification Date on the Audit Trail tab. The User Comments tab may be used to add comments to any rule, subject to a maximum of 4000 characters.

Audit Trail   User Comments	
System ID :300068	
Created By: AUTOUSER	Created Date: 01/25/2016 15:03:36
Last Modified by: PFTUSER	Last Modification Date: 05/26/2016 15:29:55

## Hierarchy Functionality

See Overview of OFSAA Infrastructure for details on how Hierarchies are used in OFS Analytical Applications.

## Dimension Container

For a new Hierarchy, select the dimension upon which you want to build a hierarchy. For an existing Hierarchy, you may not change the dimension.

Dimension

Dimension: General Ledger Account

## Hierarchy Properties Container

Specify a Hierarchy Name and Description, select a Folder in which the Hierarchy is to be stored, and specify whether you want the Hierarchy to be “Read/Write” or “Read Only” (Access Type). Naming your Hierarchy is required before it can be saved. Default values for Folder and Access Type are stored in Application Preferences. The Hierarchy Properties container also stores values for Automatic Inheritance, Display Signage, Show Member Code, Initial Display Level, and Orphan Branch.

Hierarchy Properties

Name:

Description:

Folder:  +

Access Type:  Read Only  Read/Write

Automatic Inheritance:  Yes  No

Display Signage:  Yes  No

Show Member Code:

Initial Display Level:

Orphan Branch:  Yes  No

### Name and Description

Displays the hierarchy's name and description. Name is a required property; you cannot save a hierarchy until you supply a name.

### Folder

Select the folder in which you want the hierarchy to be stored.

Automatic Inheritance

- Click Yes to inherit the hierarchy properties of the parent to the child.
- Click No to define a new hierarchy.

### Show Member Code

“Show Member Code” governs the default display behavior of a hierarchy. This code may be set to one of three different values:

- **Alphanumeric Code to Left of Name:** Displays Alphanumeric Code on the Left side of Member name.
- **Alphanumeric Code to Right of Name:** Displays Alphanumeric Code on the Right side of Member name..
- **Only Name:** No Code: Displays only the Member Name.
- **Numeric Code to Left of Name:** Displays the Numeric Code on the Left side of Member name.
- **Numeric Code to Right of Name:** Displays the Numeric Code on the Right side of Member name.

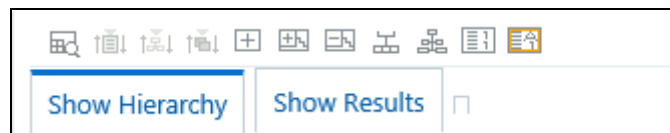
While viewing or editing a hierarchy, you may override whatever default behavior has been set by clicking on the “Show Code Values (Right)”



or on the “Show Code Values (Left)”



controls that are located on the title bar of the hierarchy display grid.



### Orphan Branch

When you initially enter the Hierarchies Detail screen in Add mode (i.e., for a brand new hierarchy), Orphan Branch is set to Yes. When set to Yes, unassigned leaf members and node members are displayed within a special “Orphan” branch. This can be convenient because the user interface includes functionality that allows you to “cut and paste” members of the orphan branch into desired locations within your hierarchy. The Orphan Branch can also be convenient when you are maintaining an existing hierarchy because when it contains members, it provides a visual cue that dimension members have been defined that have yet to be assigned locations within a hierarchy. You may suppress the display of the Orphan Branch by setting Orphan Branch to No.

### Access Type

You may set Access Type to Read-Only or Read/Write. When set to Read-Only, only the owner – the creator of a rule is its owner – may modify or delete the hierarchy.

## Display Signage

Display signage is not used within any of the OFSAA engines, but is used within the Oracle Financial Services Profitability Analytics OBI application. Its purpose is to inform reporting applications on how to subtotal information on financial reports, most typically for income statement reporting. The default value for all new dimension members is “+”.

## Initial Display Level

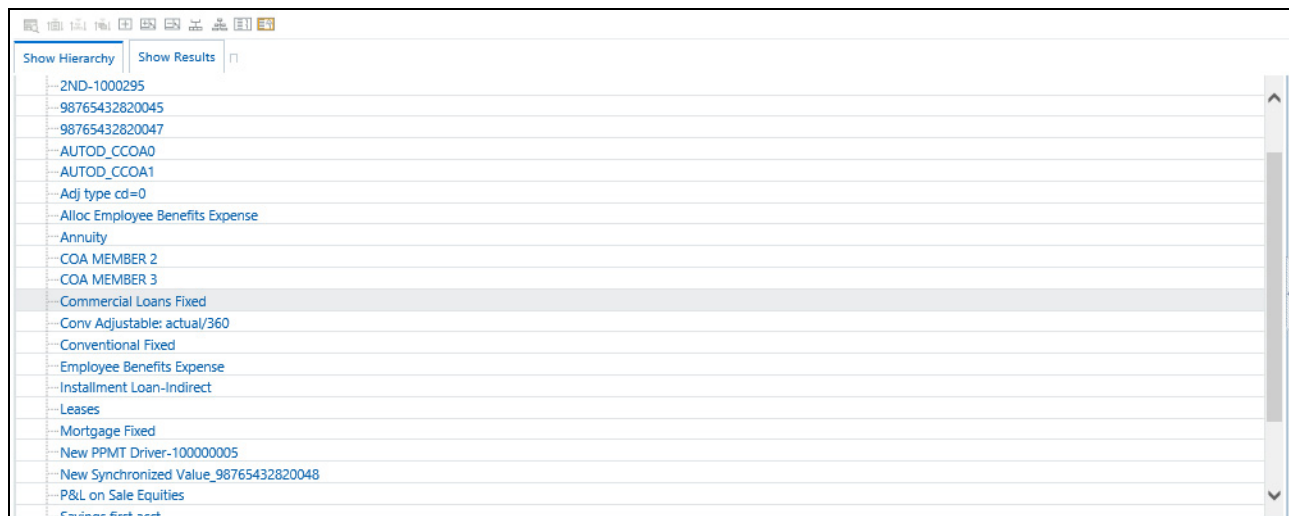
This setting controls the level at which a hierarchy is displayed when you initially enter the Hierarchies Detail screen. This setting is particularly useful when you are working with a hierarchy that has many levels. In a deep, many-leveled hierarchy, only one or two high level rollup points would be displayed if the hierarchy were displayed in its entirety.

## Hierarchy Display Grid

The main body of the Hierarchies Detail screen displays a visual representation of your hierarchy. When building a hierarchy within a given dimension, you have access to every dimension member defined for that dimension. For dimensions that support hierarchies, every dimension member has the property of either being a “leaf” member or a “rollup node” member. You construct your hierarchy by attaching leaf members to roll up node members and by attaching rollup node members to other rollup node members depending on how “deep” or multi-leveled you want your hierarchy to be.

At least one leaf level dimension member is required to be assigned under a rollup node member as a non orphan leaf in order for the hierarchy to be used in PFT.

Hierarchies may be either “balanced” or “unbalanced”. In a balanced hierarchy, all leaf members reside at the same “level” or depth. The following example shows an unbalanced hierarchy in which you see leaf members (shown in blue) three levels beneath to “root” or top of the hierarchy within the Bank 1 branch, but in which you also see leaf members two levels beneath the root level in the Company C, Mortgage Origination, and Statement Processing branches.



Generally, there is no limit to the number of levels of depth you may build into a hierarchy for Standard Dimensions that support hierarchies. For Key Processing Dimensions, which always support hierarchies, you may construct hierarchies with up to 14 levels of depth. For details on depth limitations and configuring maximum depth levels for hierarchies, see *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration Guide*.

## Controlling the Hierarchy Display

There are a number of controls on the title bar of the hierarchy that allow you to modify how the hierarchy is displayed.

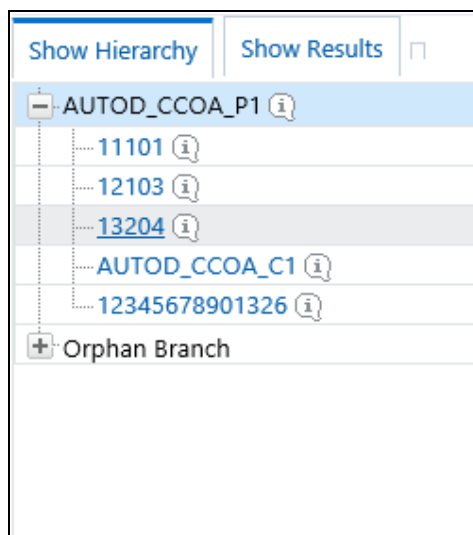
### Search

Clicking on the Search control invokes a pop-up dialog in which you can perform wildcard searches for dimension members by Member Code or by Member Name.

The screenshot shows a search dialog box with the following elements:

- Title bar: Search (with a dropdown arrow), search icon, and refresh icon.
- Radio buttons for search criteria:
  - Alphanumeric Code
  - Numeric Code
  - Member Name
- Input fields for each criterion.
- Search Tree Nodes Limit: 100
- Close button at the bottom.

When you click on the Search button within the pop-up dialog, the dialog closes, your search is executed, and results are returned within the Hierarchy Display Grid under the heading of Show Results.



When viewing search results under the Show Results heading, you may locate a dimension member within the hierarchy by selecting it and clicking on the Focus control. After selecting a dimension member and clicking on Focus, the application responds by focusing on your selected member under the Show Hierarchy heading (that is, under the normal hierarchy display).



This method of searching for dimension members will only find assigned members, that is, members that are not found in the Orphans Branch.

### Expand All / Collapse All

When you initially enter the Hierarchies Detail screen, clicking on this control expands every branch of your hierarchy to display its entire depth. Click on this control a second time to shrink the hierarchy display down to (1) the root node of the hierarchy plus (2) the “Orphans” branch.

- The Orphan branch is only displayed when a hierarchy's Orphan Branch property is set to Yes.
- For very large hierarchies, the Expand All / Collapse All control is disabled. You may define what constitutes a very large hierarchy. For more information, see *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration Guide*.

### Expand Branch

When you click on a node member and then click on Expand Branch, your selected rollup node is expanded to its maximum depth.

### Collapse Branch

Select a dimension member and then click on the Collapse Branch control to shrink the display of any members beneath the dimension member you have chosen.

### Focus and Unfocus

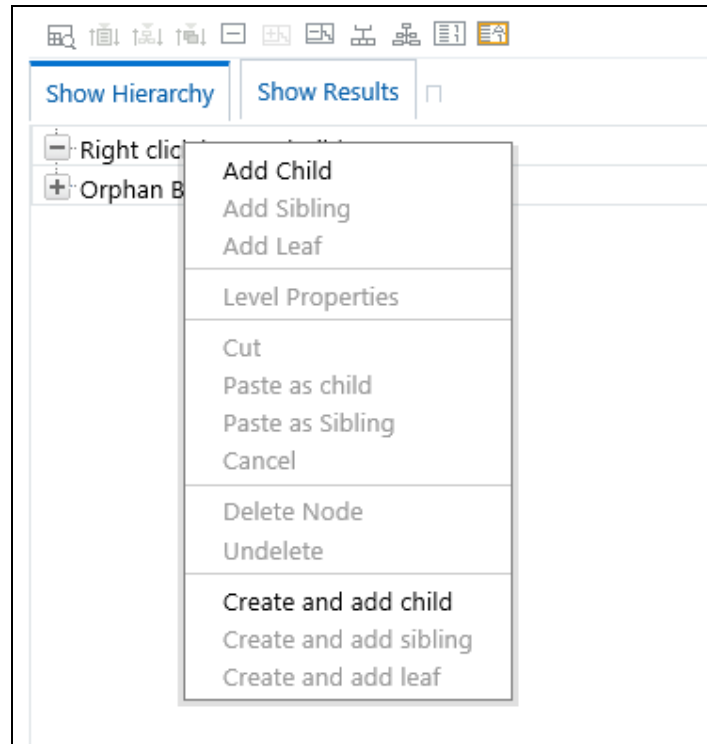
When you click on a node member and then click on the Focus control, only your selected node and its descendents are displayed. Clicking on Unfocus removes the focus that you established using the Focus control described above.

Show Code Values (Left) and (Right)

Click on Show Code Values (Left) to show code values to the left of each dimension member name. Click on Show Code Values (Right) to show code values to the right of each dimension member name. The default behavior for your hierarchy is controlled by the “Show Member Code” property discussed above.

## Adding Hierarchies

When creating a new hierarchy, you must initially provide a name and, optionally, a description. You must also select the Folder in which you want the hierarchy to be stored, the hierarchy's Read Only vs. Read/Write (Access Type) property and the other Hierarchy Properties described above. Select **Read-Only** option to give other users access to only view the hierarchy definition. Select **Read/Write** option to give all users the access to view, modify (including Access Type) and delete the hierarchy definition. Default values for Folder and Access Type are stored in Application Preferences.



To begin the process of building a new hierarchy, right mouse click at the top of the hierarchy (see example above) to invoke a pop-up dialog in which you may select dimension members. Every hierarchy must begin with a top level rollup node. Since you cannot have a leaf member as a top level rollup node, your only choice at this point is Add Child. Clicking on Add Child invokes pop-up dialog in which you may select a rollup node.

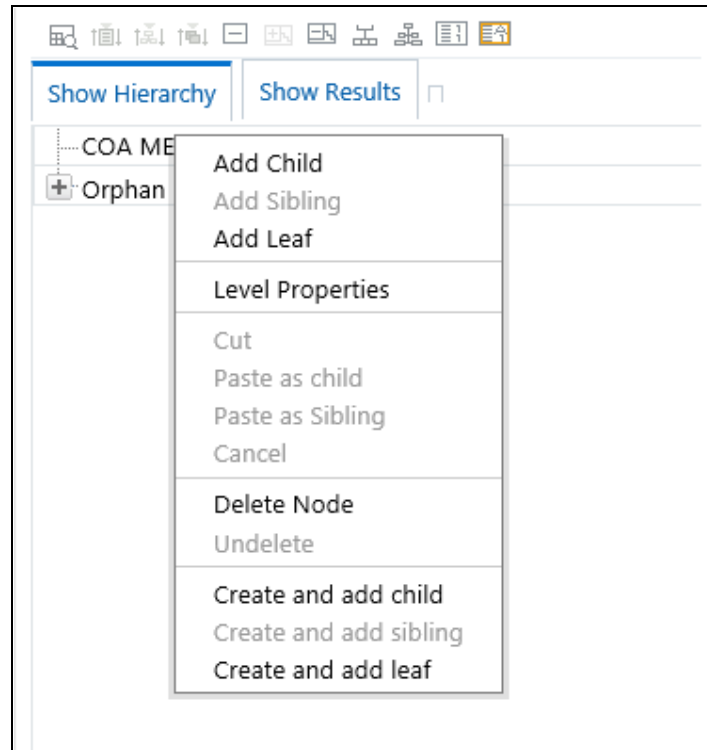
Within the Add Member pop-up dialog, use the shuttle box controls to select the member from the left hand side that you want to serve as the root member for your hierarchy. Note that in this context, the Show Members column only lists rollup members (i.e., a leaf member may not serve as the root of a hierarchy). When you click OK, the Add Member dialog closes and the original hierarchy display is modified to show your selected root member.

At this point, the process of adding additional dimension members (leaves and rollup nodes) is the same for existing hierarchies as it is for new hierarchies.

## Editing Hierarchies

### Inserting New Members into a Rollup Node

Select a rollup node member and then right mouse click. This action will display a number of choices including Add Child, Add Sibling, and Add leaf. The following example continues from the above description of creating a new organizational unit hierarchy.



In this current example, we have selected a rollup node member which serves as the root member of the hierarchy. Because OFSAA Infrastructure does not support hierarchies with multiple “tops”, you may not add a sibling member (i.e., a member to be stored at the root level of the hierarchy) so the Add Sibling option is disabled.

Whenever you select the “Add Child” or “Add Sibling” or “Add Leaf” option, the Add Member dialog is displayed.

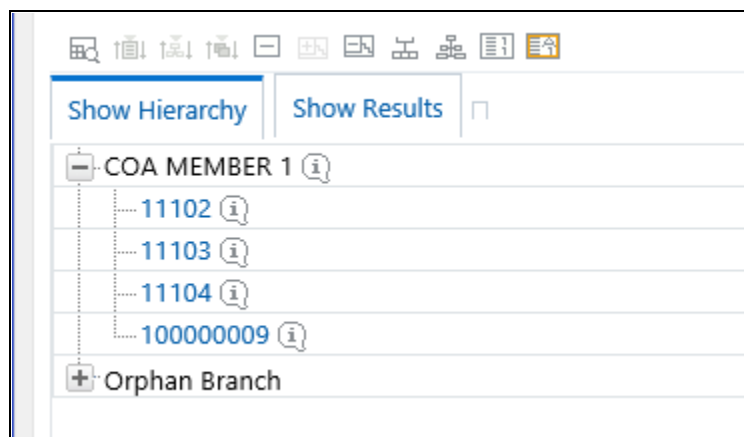
### **Add Child**

For the “Add Child” option, the Add Member dialog presents all dimension members that have not already been assigned elsewhere (i.e., all members that are not found in the orphan branch). The value or values you select will be placed in the hierarchy one level beneath the node you originally selected.

In the following example, Company A, Company B, and Company C have been selected to act as the children of the All Orgs root node.

Select the required Member and click **Select** button. The Member is displayed in the Selected Members panel. Click **Select All** button to select all Members which are shown in the Show Members pane. Click **Select All From Server** button to select all nodes/ members in the server. You can click **Deselect** button to deselect a Member or click **Deselect All** button to deselect all the Members. You can click **Search** button to search for the required member using Alphanumeric code, Numeric Code, Name, Description, Attribute Name, or Attribute Value.

When you click OK on the Add Member dialog, the dialog box closes and the underlying the Hierarchy Display is updated to reflect your choices.

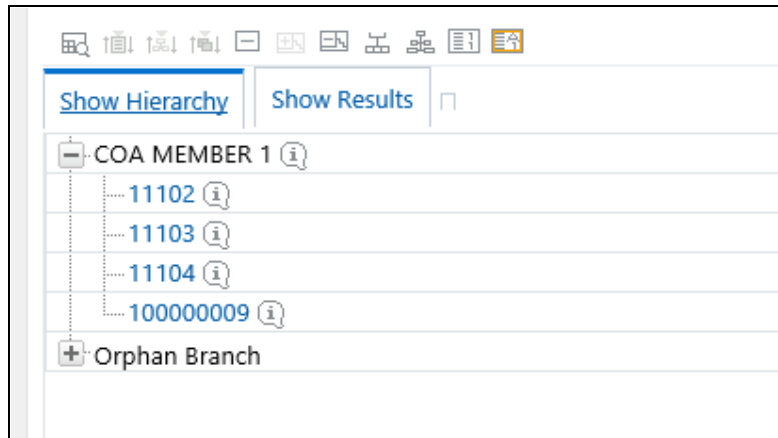


## Add Sibling

For the “Add Sibling” option, the Add Member dialog presents all dimension members that have not already been assigned elsewhere (i.e., all members that are not found in the orphan branch). The value or values you select will be placed in the hierarchy at the same level as the node you originally selected. In the following example, “Corporate – Parent” has been selected to be a sibling to Company C, i.e., to be inserted at the same level within the hierarchy as Company C.

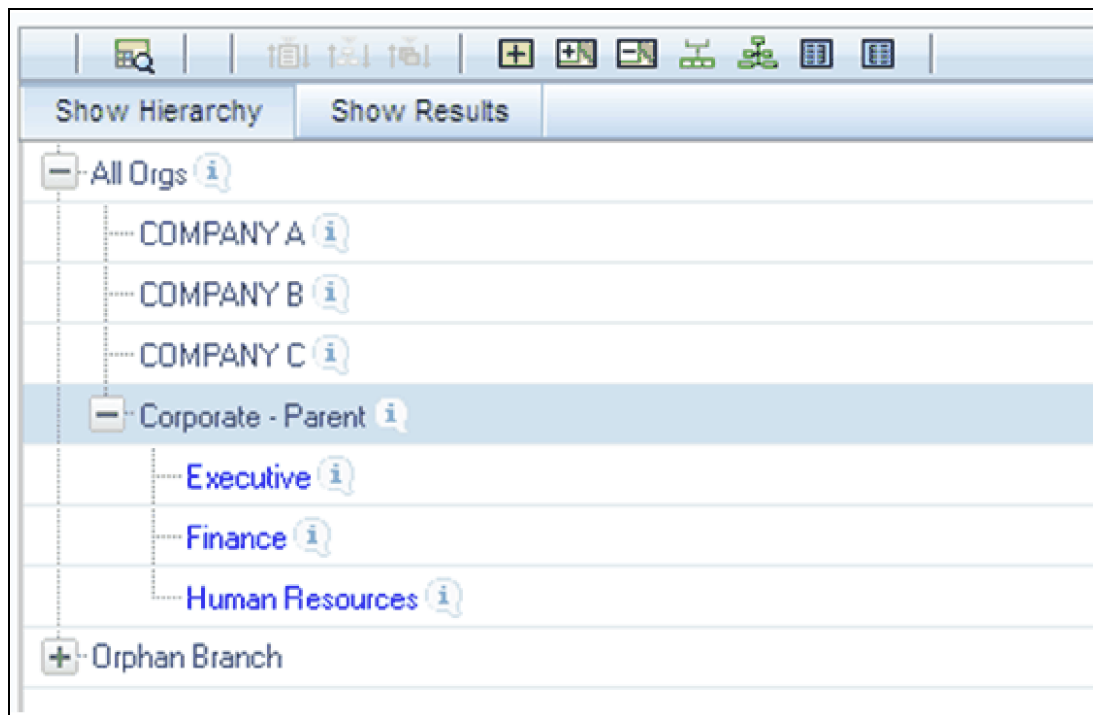
The screenshot shows the 'Add Member' dialog box. At the top right are 'Ok' and 'Close' buttons. Below is a 'Members' section with a search bar containing 'Search' and 'Reset' buttons. The search bar has a magnifying glass icon on the left and a circular arrow icon on the right. Below the search bar are input fields for 'Alphanumeric Code', 'Numeric Code', 'Name', 'Description', 'Attribute Name' (a dropdown menu), and 'Attribute Value'. Below these fields are two panels: 'Show Members' and 'Selected Members'. The 'Show Members' panel contains a list of members with a vertical scrollbar. The 'Selected Members' panel contains a table with columns 'Alphanumeric Code', 'Numeric Code', and 'Name'. Between the two panels are navigation buttons: '>', '>>', a refresh icon, '<', and '<<'.

Select the required Member and click **Select** button. The Member is displayed in the Selected Members panel. You can click **Select All** button to select all Members which are shown in the Show Members pane. Click **Select All From Server** button to select all nodes/ members in the server. You can click **Deselect** button to deselect a Member or click **Deselect All** button to deselect all the Members. You can also Click **Search** button to search for the required member.

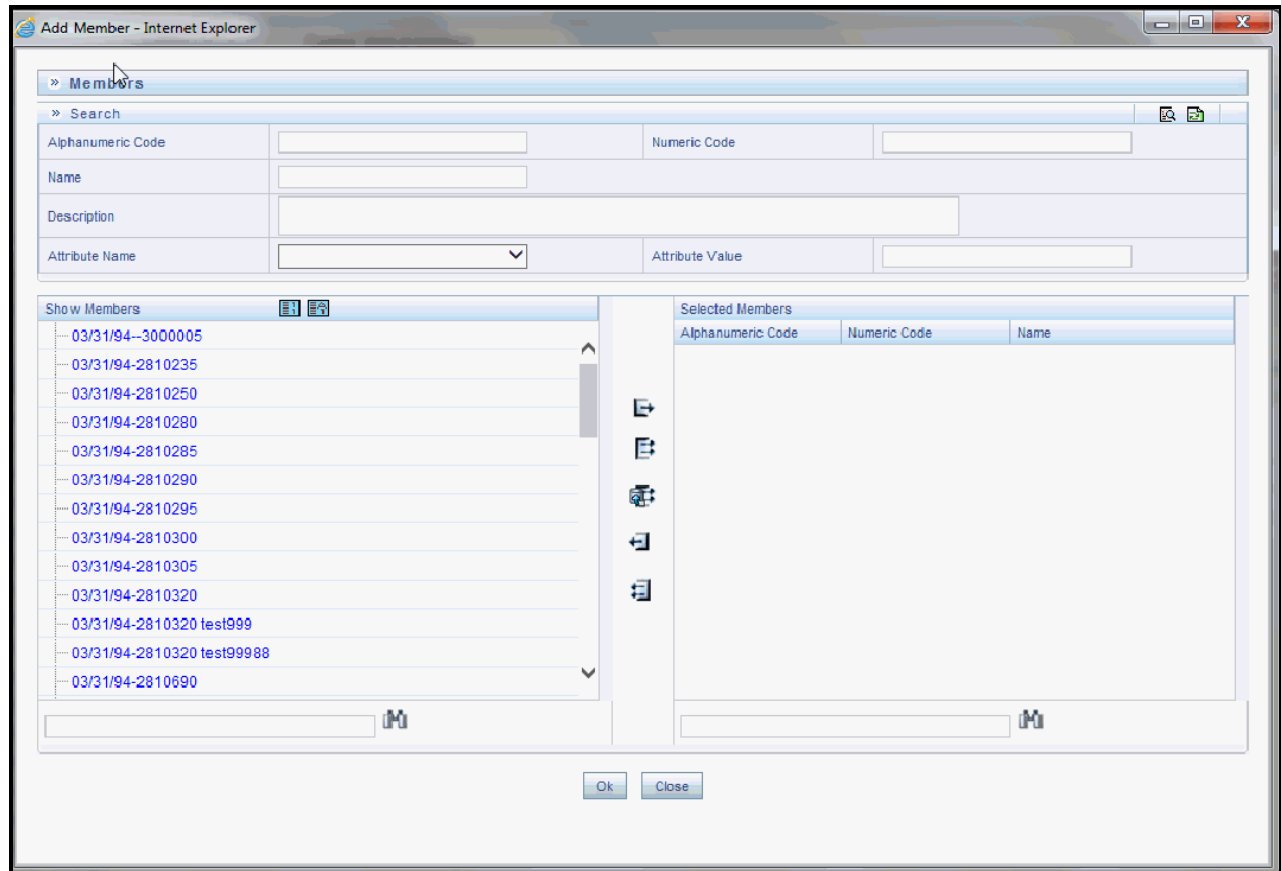


### Add Leaf

For the “Add Leaf” option, the Add Member dialog presents all leaf members that have not already been assigned elsewhere. The leaf value or values you select will be placed in the hierarchy one level beneath the node you originally selected.



In the example above, Executive, Finance, and Human Resources have been selected to be children of the Corporate – Parent rollup node.



Select the required Member and click **Select** button. The Member is displayed in the Selected Members panel. You can click **Select All** button to select all Members which are shown in the Show Members pane. Click **Select All From Server** button to select all nodes/ members in the server. You can click **Deselect** button to deselect a Member or click **Deselect All** button to deselect all the Members. You can also Click **Search** button to search for the required member.

### **Level Properties**

Selecting the Level Properties option invokes the Level Properties dialog. Within this dialog, you may provide names and descriptions for each of the levels within your hierarchy.



Level Number	Name	Description
Level1	<input type="text"/>	<input type="text"/>
Level2	<input type="text"/>	<input type="text"/>
Level3	<input type="text"/>	<input type="text"/>
Level4	<input type="text"/>	<input type="text"/>

The Level Properties dialog will display as many levels as you have currently built out. In the above example, names and descriptions are being added at a time when only 3 levels have been built within the underlying hierarchy. If a fourth level is built into the hierarchy, you could return to Level Properties to add a name and description to level 4.

Assigning names or descriptions to levels is completely optional. Within the Hierarchy Properties Container, the drop-down list for “Initial Display Level” will reflect your level name choices. Your choices may also be utilized within downstream reporting applications, but level names and descriptions are otherwise not utilized within OFSAA.

Initial Display Level

- 1 - Level 1
- 2 - Level 2
- 3 - Level 3
- 4 - Level 4
- 5 - Level 5

## **Cut**

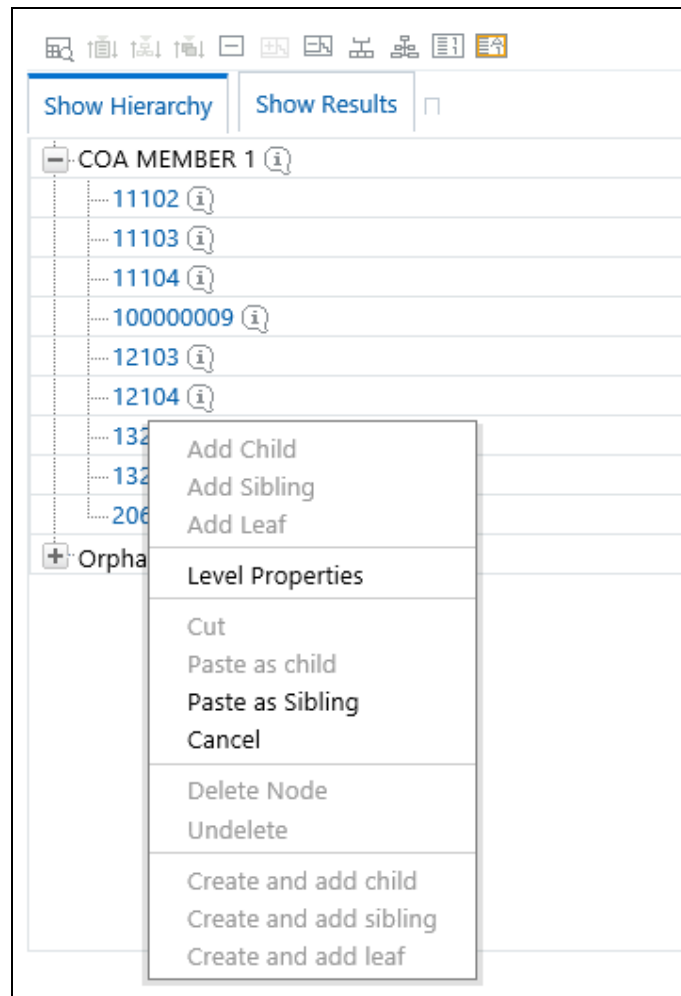
If you wish to move a dimension member, click on a leaf member or a rollup member and select Cut. You may wish to move a dimension member because it is in the wrong location within the hierarchy. You may also wish to Cut a dimension member from the Orphan Branch in order to subsequently paste it within the body of your hierarchy. Any Cut operation must be followed by a Paste as Child, Paste as Sibling, or Cancel operation. Until you have executed a Cut operation, Paste as Child, Paste as Sibling, and Cancel are disabled; conversely, once you have executed a Cut operation, Paste as Child, Paste as Sibling, and Cancel are enabled.

## **Paste as Child or Paste as Sibling**

After performing a Cut operation, you may perform a Paste as Child or Paste as Sibling operation. If you have Cut a dimension member and then subsequently selected leaf member, the Paste as Child option is disabled, i.e., no dimension member may be a child of a leaf member. Leaf members, by definition, can have no children. Since

OFSAA supports unbalanced hierarchies, any dimension member, regardless of whether it is a leaf member or a rollup member, may be pasted as a sibling to any other dimension member.

In the following example, Human Resources has been Cut from the “Corporate – Parent” rollup node so that it may be pasted as a child into Company C.

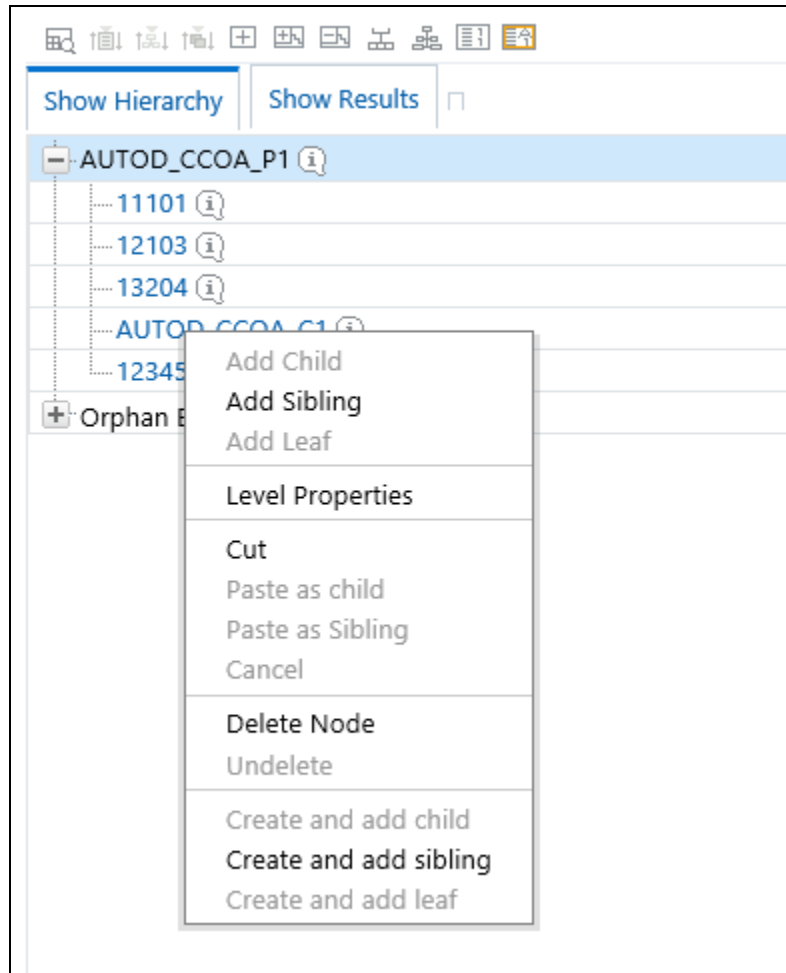


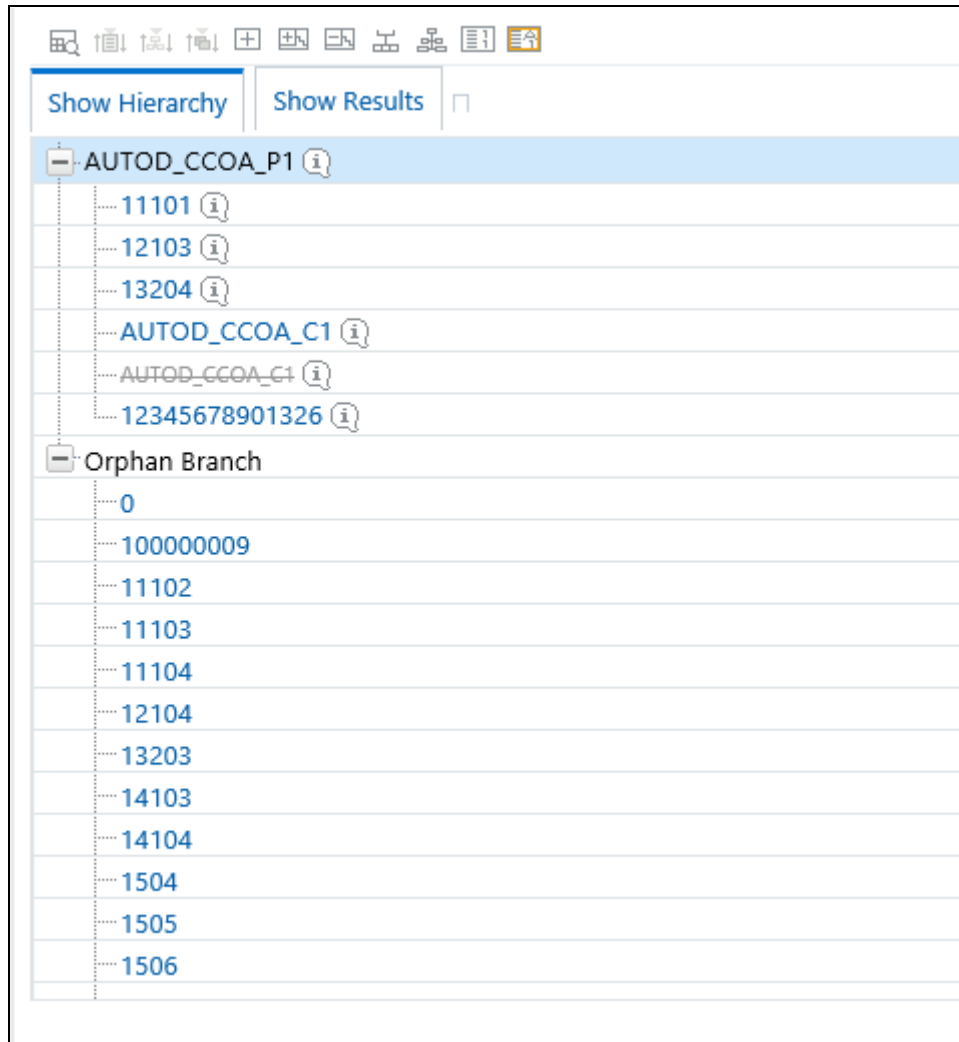
### **Cancel**

The Cancel option is only enabled after you have executed a Cut operation. Once you have Cut a dimension member, you may not perform any other operation except for Paste as Child, Paste as Sibling, or Cancel (you may, however, still invoke Level Properties; see discussion above on Level Properties). Utilizing the Cancel option allows you to select and subsequently Cut another dimension member (i.e., after you have performed a Cancel operation, Paste as Child, Paste as Sibling, and Cancel are disabled).

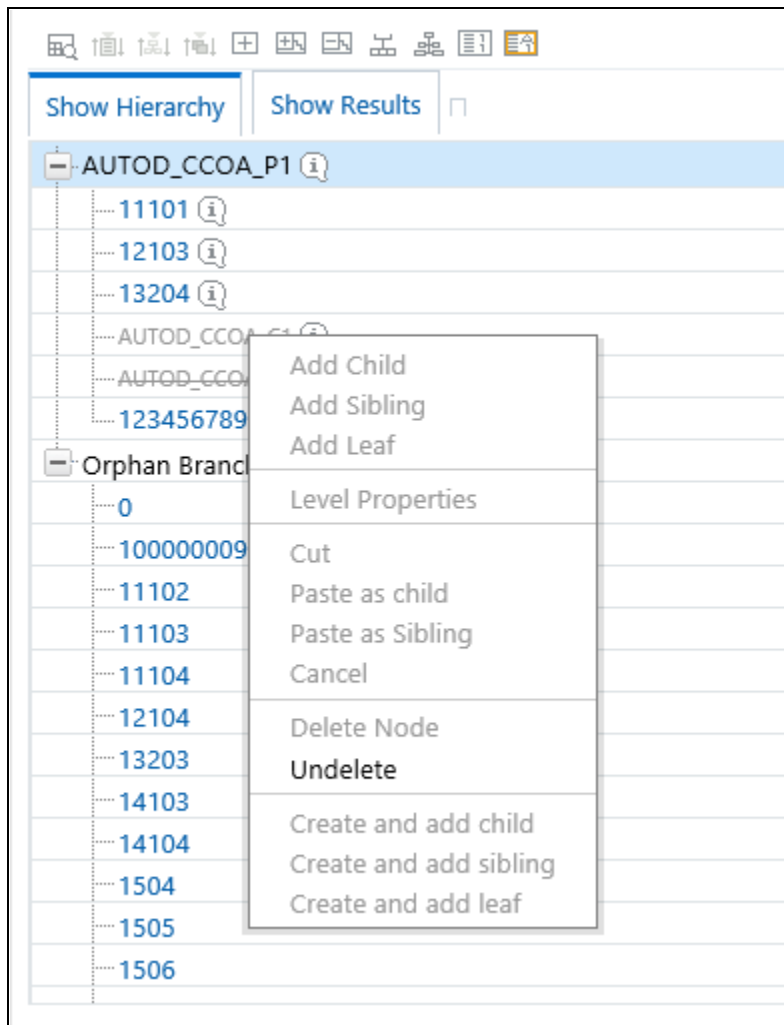
### **Delete Node & UnDelete**

To move a dimension member back into the Orphans Branch, click on a member and then select the Delete option. Once you have performed a Delete operation, the member selected for deletion appears with a strikethrough font. The following example shows the deletion of the Human Resources dimension member.



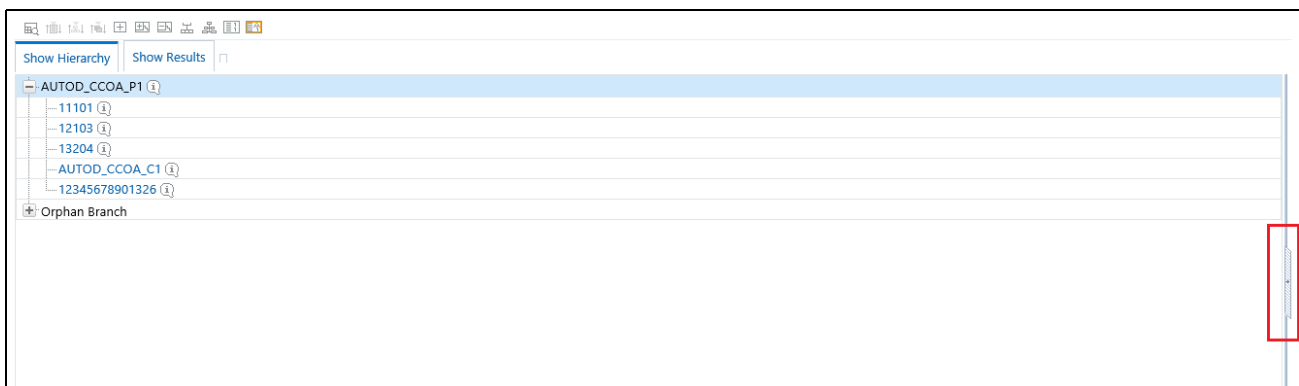


Note that after having deleted the Human Resources member, it may be “restored” to its original condition by executing an UnDelete operation.

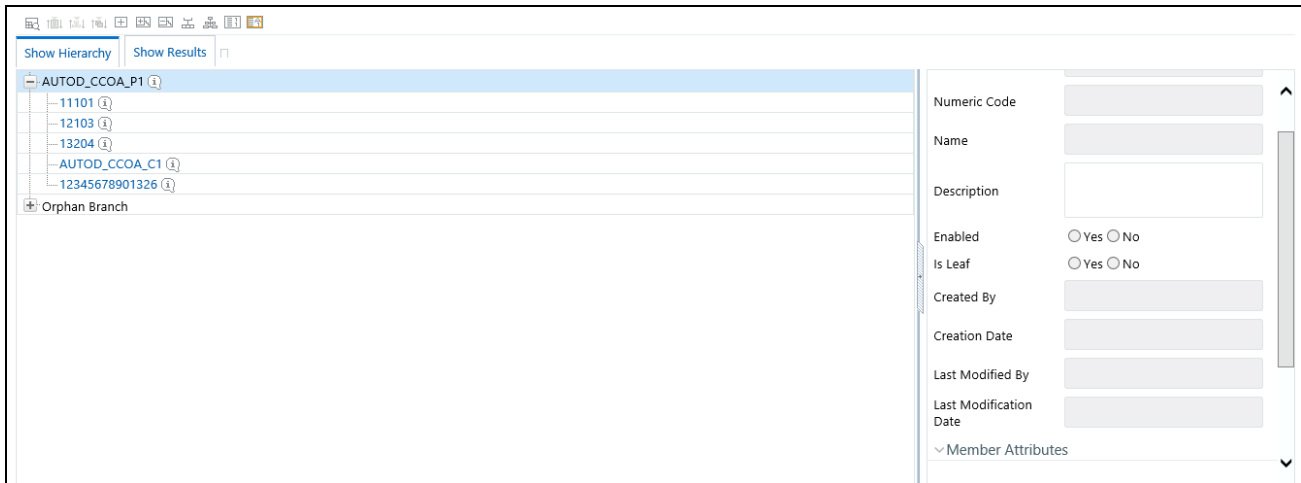


### Member Properties Display

Click on the separator bar at the extreme right hand side of the Hierarchy Display Grid to invoke the Member Property display.



The Member Property display shows all of the properties of the currently selected dimension member (the member selected within the hierarchy). While the Member Property display is open, you may use your Up arrow, Down arrow, Left arrow, or Right arrow keys to move from dimension member to dimension member.



## CHAPTER 8

## OFSAA Filters

## Overview of OFSAA Filters

Filters in the Infrastructure system facilitates you to filter metadata using the defined expressions. Filters is a licensed module and is accessible to the users who are mapped with the SYSFILTERS role in the Oracle Financial Services Analytical Application Infrastructure system.

This chapter describes the creation and maintenance of OFSAA Filters. Specifically, this chapter will cover:

- Filters Summary & Detail Screens
- Navigation within the Filters Summary Screen
- Navigation within the Filters Detail Screen

See [Chapter 2, "Overview of OFSAA Infrastructure"](#) for details on how Filters are used in OFS Analytical Applications.

## Summary &amp; Detail Screens

Upon initially navigating to Common Object Maintenance > Filters, a summary screen is displayed showing a set of Filters. Using search criteria, you can control the set of Filter rules that are displayed. When you Add, Edit, or View a rule, a detail screen is displayed.

Name	Type	Modification Date	Modified by
AUTOD_AF_LTD	Attribute	12/11/2017 16:08:26	AUTOUSER
AUTOD_DF_AC_01	Data Element	10/11/2017 16:11:03	PFTUSER
AUTOD_DF_LS_01	Data Element	10/11/2017 16:07:46	PFTUSER
AUTOD_DF_LS_01	Data Element	12/11/2017 16:04:19	AUTOUSER
AUTOD_DF_LTD	Data Element	11/08/2017 11:07:20	PFTUSER
AUTOD_DF_LTD	Data Element	04/14/2017 14:36:28	PFTUSER
AUTOD_DF_ML_01	Data Element	10/11/2017 16:09:55	PFTUSER
AUTOD_GF_AC_01	Group	10/11/2017 16:11:24	PFTUSER
AUTOD_GF_LS_01	Group	10/11/2017 16:08:09	PFTUSER
AUTOD_GF_LTD	Group	02/15/2018 15:17:15	PFTUSER
AUTOD_GF_LTD	Group	04/14/2017 14:38:01	PFTUSER
AUTOD_GF_ML_01	Group	10/11/2017 16:10:25	PFTUSER
AUTOD_HierFilter	Hierarchy	01/27/2016 16:05:19	AUTOUSER
Data_Filter_01	Data Element	01/02/2017 14:39:31	PFTUSER
Data_Filter_01	Data Element	03/23/2017 10:40:32	PFTUSER
LTD_GRPfilter_02	Group	01/04/2017 10:40:06	PFTUSER
LYD_datafilter_02	Data Element	01/04/2017 10:39:35	PFTUSER
mgmt_led_data_filter	Data Element	06/07/2016 19:30:24	PFTUSER
mgmt_led_group_filter	Group	06/07/2016 19:32:36	PFTUSER
prod_attrib_filter_1	Attribute	08/19/2016 13:59:37	PFTUSER

## Navigation within the Summary Screen

When you first navigate to the Filters summary screen, the filters stored within your current default Folder are presented in a summary grid. The Filters summary screen has two containers: Search and Filters.

### Search Container

Your default Folder functions as a search constraint. The value of your default Folder is set in Application Preferences. You may select a different Folder or you may remove the Folder constraint entirely by selecting the “blank” Folder, i.e., no Folder. You may also search by Filter Name or by Filter Type. Filter Types, shown in a drop-down list, include:

- Data Element Filter
- Hierarchy Filter
- Group Filter
- Attribute Filter

### Search Control

Enter your desired search criteria and click on the Search control.

### Reset Control

Restores the default Folder, removes any Filter Name or Filter Type constraint you may have specified, and refreshes the screen.

The screenshot shows the 'Filters' summary screen. At the top, there is a search bar with a 'Search' button and a 'Reset' button. Below the search bar, there are two dropdown menus: 'Folder Name' (set to 'AUTOSEG') and 'Type'. Below these are several action icons: '+ Add', 'View', 'Edit', 'Copy', 'Check Dependencies', 'Delete', and 'View SQL'. The main part of the screen is a table with the following columns: Name, Type, Modification Date, and Modified by. The table contains 10 rows of filter rules.

Name	Type	Modification Date	Modified by
<input type="checkbox"/> AUTOD_AF_LTD	Attribute	12/11/2017 16:08:26	AUTOUSER
<input type="checkbox"/> AUTOD_DF_AC_01	Data Element	10/11/2017 16:11:03	PFTUSER
<input type="checkbox"/> AUTOD_DF_LS_01	Data Element	10/11/2017 16:07:46	PFTUSER
<input type="checkbox"/> AUTOD_DF_LS_01	Data Element	12/11/2017 16:04:19	AUTOUSER
<input type="checkbox"/> AUTOD_DF_LTD	Data Element	11/08/2017 11:07:20	PFTUSER
<input type="checkbox"/> AUTOD_DF_LTD	Data Element	04/14/2017 14:36:28	PFTUSER
<input type="checkbox"/> AUTOD_DF_ML_01	Data Element	10/11/2017 16:09:55	PFTUSER
<input type="checkbox"/> AUTOD_GF_AC_01	Group	10/11/2017 16:11:24	PFTUSER
<input type="checkbox"/> AUTOD_GF_LS_01	Group	10/11/2017 16:08:09	PFTUSER

### Filters Container

The Filter container presents a grid containing all of the Filter rules that meet your search criteria. The Filter summary grid offers several controls that allow you to perform different functions when a Filter rule is selected.

To select a Filter rule, click on a check box in the first column of the grid. More than one Filter can be selected at a time but this will cause some of the controls to become disabled. Clicking on a checkbox a second time de-selects the Filter.

You may select or deselect all of the Filters in the summary grid by clicking on the check box in the upper left hand corner of the summary grid directly to the left of the Name column header.

### Add

Clicking on the Add control begins the process of building a new Filter. The Add control is disabled if any rows in the grid have been selected.



**View**

Selecting a single row out of the grid enables the View control. Clicking on the View control allows you to view the contents of a Filter on a read-only basis. The View control is only enabled when a single Filter has been selected.

**Edit**

Selecting a single row out of the grid enables the Edit control. Clicking on the Edit control allows you to modify a previously saved Filter. The Edit control is only enabled when a single Filter has been selected.

**Copy**

Selecting a single row out of the grid enables the Copy control. Clicking on the Copy control allows you to create a copy of an existing Filter. The Copy control is only enabled when a single Filter has been selected.

**Check Dependencies**

To generate a report on the OFSAA rules that utilize a selected Filter, select a single Filter and click on the Check Dependencies control. The Check Dependencies control is only enabled when a single Filter has been selected.

**Delete**

Selecting one or more rows out of the grid enables the Delete control. Clicking on the Delete control deletes the Filter(s) you have selected. OFSAAI will not allow you to delete Filters which have any dependencies (see Check Dependencies above).

**Filters Summary Grid**

The following columns categorize each Filter in the summary grid:

- Name
- Type
- Modification Date
- Modified By

**Name**

Displays the Filter's short name. Performing a “mouse-over” on a Filter Name displays the Filter's description.

**Type**

Displays the Filter's type: Data Element, Hierarchy, Group, or Attribute.

**Modification Date**

Displays the date and time at which a Filter was last modified.

**Modified By**

Displays the name of the user who last modified a Filter.

***Navigation within the Detail Screen***

When you Add, Edit, Copy, or View a Filter, the Filter Detail Screen is displayed.

In addition to Name, Description, Folder, and Access Type, the definition of a new Filter begins with declaring a “Filter Type”. Supported Filter Types include Data Element, Hierarchy, Group and Attribute.

The Audit Trail container is a standard footer container for every OFSAA rule type. It displays Created By, Creation Date, Last Modified By, and Modification Date on the Audit Trail tab. The User Comments tab may be used to add comments to any rule, subject to a maximum of 4000 characters.

**Note:** See Overview of OFSAA Infrastructure chapter for details on how Filters are used in OFS Analytical Applications.

### Folder Name

Select the Folder Name where the Filter is to be stored from the drop-down list. For more information, refer to OFSAAI User Guide.

### Access Type

Select the Access Type as Read Only or Read/Write.

- Read-Only: Select this option to give other users the access to only view the filter definition.

**Note:** A user with Phantom and Write role can modify or delete the filter even though the access type is selected as Read-only.

- Read/Write: Select this option to give all users the access to view, modify (including Access Type) and delete the filter definition.

### Filter Name

Enter the filter name in the Filter Name field.

**Note:** The characters '&' ' ' are restricted.

### Description

Enter the description of the filter in the Description field.

---

**Note:** The characters ~&+' " @ are restricted.

---

### Filter Type Selection Container

Select a filter type from the Filter Type drop-down list.

### Data Element

Data Element Filter is a stored rule that expresses a set of constraints. Only columns that match the data type of your Data Element selection are offered in the Data Element drop down list box. For example:

Balances between 10,000 and 20,000 Accounts opened in the current month Loans with amortization terms greater than 20 years

Data Element Filters can access most instrument columns and most columns in the Management Ledger. Data Element Filters are used within other OFSAA rule types (For example, Allocation rules, Transfer Pricing rules, Asset | Liability Management rules, and so on).

### Hierarchy

Hierarchy Filter allows you to utilize rollup nodes within a Hierarchy to help you exclude (filter out) or include data within an OFSAA rule.

For example, You might want to process data for a specific set of divisions or lines of business where you have a Hierarchy rule that expresses those divisions or lines of business as rollup nodes. A Hierarchy Filter could be constructed to "enable" the Commercial and Retail lines of business while NOT enabling the Wealth Management line of business. Each of these lines of business might include a handful or even thousands of cost centers. When incorporated into an OFSAA processing rule, this Hierarchy Filter would include every cost center in the Commercial and Retail lines of business.

### Group Filters

Group Filters may be used to combine multiple Data Element Filters with a logical "AND".

For example, if Data Element Filter #1 filtered on mortgage balances greater than 100,000 and Data Element Filter #2 filtered on current mortgage interest rates greater than 6%, you could construct a Group Filter to utilize both Data Filters. In this case, the resulting Group Filter would constrain your data selection to mortgage balances greater than 100,000 AND current mortgage interest.

### Attribute Filters

**Attribute Filters** are created using defined Attributes. Attribute filters facilitates you to filter on one or more Dimension Type Attributes. For each attribute, you can select one or more values.

Consider a filter that selects all records where the dimension Common Chart of Account member represents an attribute value Expense account, i.e., the attribute "Account Type" = Expense.

Now, using Attribute Filters, you can specify complex criteria as given below:

Common Chart of Accounts where the Account Type attribute is Earning Assets or Interest-bearing Liabilities, and the Accrual Basis attribute is Actual/Actual Also, You could further refine the filter by adding another condition for: Organizational Unit where the Offset Org ID is a specific Org member.

The Filter then saves these criteria rather than the member codes which meet the criteria at the time the Filter is saved. During execution, the engine dynamically selects all records from your processing table (for example, Mortgages, Ledger, and so on), which meet the specified member attribute criteria.

## Data Element Filters

### Data Element Selection Container

In this container, specify the columns that you want to include in your Data Filter and, for each column you choose, a Filter Method.

### Data Element Selection

To select the columns you want to include in your Data Filter, click on the ellipses to invoke a Data Element Selection pop up window.

3. In the Data Element Selection section, click the below button.



The **Data Element Selection** screen is opened.

- Select any of the following Filter Classification Type from the drop-down list:
    - **Classified** - This is the default selection and displays all the classified EPM specific entities. If you are an EPM user, you need to select this option while defining Data Element Filter to list all the related entities.
    - **Unclassified** - This option displays all the non-classified i.e. non EPM specific entities. If you are a non EPM user, you need to select this option while defining Data Element Filter to list all the related entities.
    - **All** - This option will select all the tables available in the selected Information Domain irrespective of whether an entity has its table is classified or not.
  - In the **Entity Name** drop-down list, select a database table. The associated members are displayed in the **Show Members** section.
  - Select the required member and click **Select**. The member is listed in the **Selected Members** panel. Click **Select All** to move all Members. You can click **Deselect** to deselect a Member or click **Deselect All** to deselect all Members.
  - Click **OK**. The Data Element is selected. The selected Data Elements are displayed in the Data Element Selection field.
4. Select the **Filter Method** from the drop-down list.
- Click **Add** to list the completed filter conditions in the Filter Conditions grid.
  - Click **Update** after modifying a filter condition to update in the Filter Conditions grid.

- Click the below buttons to move a selected Filter Condition up or down.
  - Click **Delete** button to delete selected individual Filter Conditions records.
5. Click **Add/Update** in the **Filter Definition** screen if you are creating a new or updating an existing Filter definition.

### Filter Method

For each column you wish to include in your Data Filter definition, you must specify a Filter Method:

- Specific Values
- Ranges
- Another Data Element
- Expression

### Specific Values

Specific Values are used to match a selected database column to a specific value or values that you provide.

You may add additional values by clicking on the Add control, and you may reduce the number of Specific Values by clicking on the check box to the left of a value and then clicking on the Delete control.

- When comparing Specific Values for a character type column, you must provide Specific Values that are character strings.
- When comparing Specific Values for a date type column, you must provide Specific Values that are dates (the application displays a Calendar control).
- When comparing Specific Values for a numeric column, you must provide Specific Values that are numbers.

**Note:** You may either include or exclude Specific Values.

## Ranges

Ranges are used to match a selected database column to a range of values or to ranges of values that you provide. You may either include or exclude Range values. Range Type is available for OFSA Datatype Term, Frequency, Leaf, Code, and Identity and Column Datatype Date, Numeric and Varchar.

You may add additional values by clicking on the Add control, and you may reduce the number of Ranges by clicking on the check box to the left of a value and then clicking on the Delete control.

- If the Column Datatype is VARCHAR, provide Specific Values (alphanumeric) that are character strings.
- If the Column Datatype is DATE, provide Specific Values that are dates (the application displays a Calendar control).
- If the Column Datatype is Numeric, provide Specific Values that are numbers.
- If OFSA Datatype is LEAF, provide either numeric values or click Search button to select the numeric member ids. ♣
- If OFSA Datatype is CODE, provide either numeric values or click Search button to select the numeric member ids.
- If OFSA Datatype is IDENTITY, provide specific numeric values. However, no validation is done during save to validate the input value for a valid identity code.

Select Include Values or Exclude Values to include or exclude the selected values

- You may use any of the following operators when choosing the Another Data Element Filter Method : = , <> ( meaning “ not equal to ” ) , < , > , <= , or >=
- You may either include or exclude Range values.

* Data Element Selection	
Data Element Selection	CASA - Average Net Book Balance
Filter Method	Ranges
* Ranges	
<input type="checkbox"/> Values	
<input type="checkbox"/> >=	0.00 <=
<input type="checkbox"/> >=	10000.00 <=
<input checked="" type="radio"/> Include Values <input type="radio"/> Exclude Values	

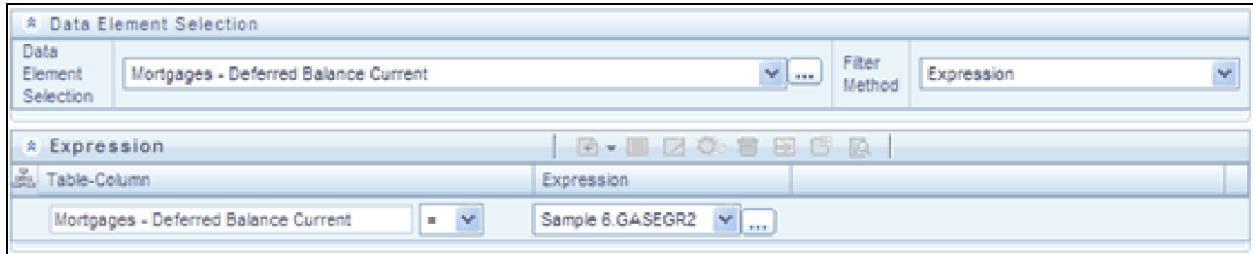
## Another Data Element

Another Data Element is used to match a selected database column to another database column. When constructing an Another Data Element Filter Method, you may only compare a column to other columns that you have already selected (the Data Element drop down list box will only contain columns that you have already selected).

**Note:** You may use any of the following operators when choosing the Another Data Element Filter Method: = , <> ( meaning “not equal to” ) , < , > , <= , or >=

## Expression

Expression is used to match a selected database column to the results of an OFSAA Expression rule (see Chapter 9: OFSAA Expressions).

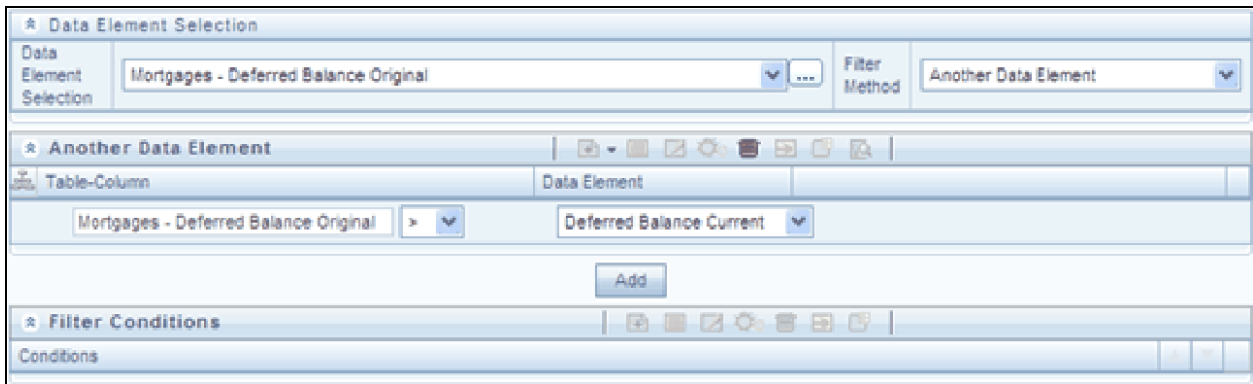


**Note:** You may use any of the following operators when choosing the Expression Filter Method: =, <> (meaning “not equal to”), <, >, <=, or >=

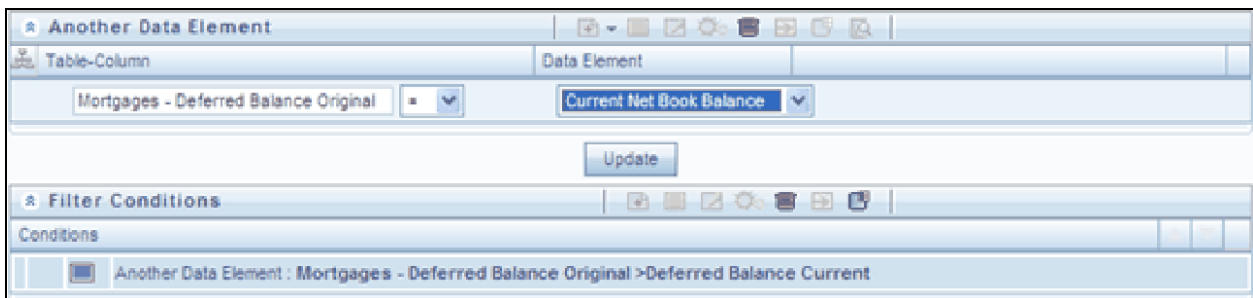
### Building Filter Conditions

Each time you complete a Filter Method specification, you must Add it to the Filter Conditions container. The reason you must do so is because you may compare the same columns multiple times. For example, you may compare Deferred Current Balances to a Specific Value and to Another Data Element and to a Range of Data Elements and to an Expression all in the same Data Element Filter rule.

### Filter Conditions Container



Once you have created one or more Filter Condition records, you may click on a record in the Filter Conditions container in order to select it. Once you have selected a Filter Conditions record, you may modify your original definition of that Filter Condition and then Update it.





After you click on Update, your Filter Condition is updated.

The screenshot shows a software interface with two main sections. The top section, titled 'Another Data Element', contains a 'Table-Column' area with two dropdown menus: 'Mortgages - Deferred Balance Original' and 'Current Net Book Balance'. Below these is an 'Add' button. The bottom section, titled 'Filter Conditions', contains a 'Conditions' area with a single entry: 'Another Data Element : Mortgages - Deferred Balance Original =Current Net Book Balance'.

### Delete

You may delete individual Filter Conditions records by clicking on the Delete control .

### View SQL

Click on the View the SQL control to generate a window displaying the SQL equivalent of your Data Element Filter. You may generally copy and paste this SQL into queries that you write. View SQL, however, will not accurately reflect Expression Filter Methods.

## Hierarchy Filters

### Hierarchy Selection Container

When you have selected the Filter Type as Hierarchy, define the Filter conditions by doing the following in the Hierarchy Selection section:

1. Select the required **Dimension** from the drop-down list.

The screenshot shows a 'Filter Type Selection' window. The 'Filter Type' dropdown is set to 'Hierarchy'. Below it is the 'Hierarchy Selection' section, which contains two dropdown menus: 'Dimension' set to 'COUNTRY' and 'Hierarchy' set to 'Sample Country Hierarchy'.

2. Select the associated **Hierarchy** from the drop-down list. You can select **More** to search for a specific Hierarchy in the *Hierarchy more* dialog.
3. Select any combination of rollup points and leaf (last descendent child) values.
4. In the New – Filter Details screen, you can:
  - Click the below button to search for a hierarchy member using Dimension Member Alphanumeric Code, Dimension Member Code, Dimension Member Name, or Attribute and by keying in Matching Values in the Search dialog.



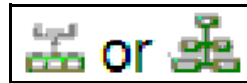
- Click the below buttons to expand or collapse the members under a node.



- Click the below buttons to expand a branch or collapse a branch



- Click the below buttons to focus or unfocus a selected node except the root node.



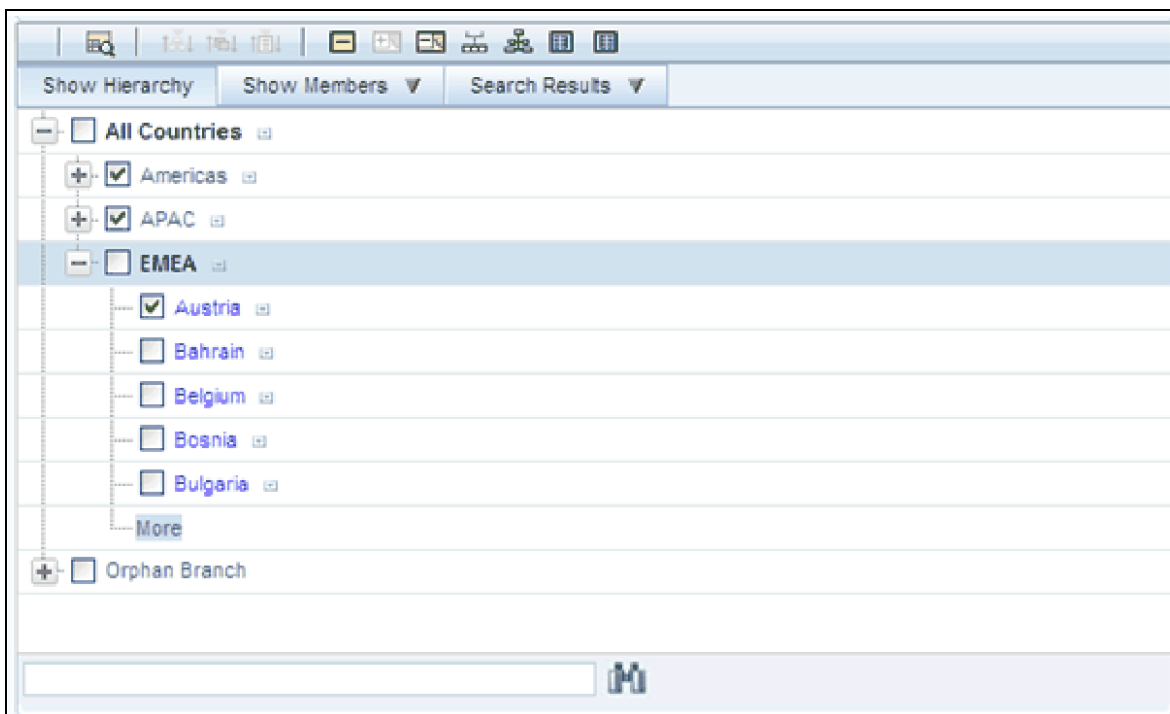
- Click the below buttons to toggle the display of Numeric Code or Alphanumeric code at left of the nodes, right of the nodes or to hide.



- You can also click the below button to find a member present in the nodes list using key words. For large tree (nodes>5000), this search will not return any value if the tree is not expanded.



5. Click **Save** to save the Filter definition.
6. Once you have selected a dimension and a hierarchy in the Hierarchy Selection container, a representation of the hierarchy you selected is displayed.
7. Select any combination of rollup points and leaf (last descendent child) values. In the example above, the Americas and APAC rollup points have been selected; and the leaf value for Austria has been selected. This Hierarchy Filter includes any country that rolls up to either the Americas or to APAC; it also includes Austria. This Hierarchy Filter excludes all countries that roll up to EMEA except for Austria.



### View SQL

Once you have selected one or more leaves and/or rollup nodes, click on the View the SQL control to generate a window that displays the SQL equivalent of your Hierarchy Filter.

## Group Filters

### Advanced Container

When you have selected the Filter Type as Group, define the Filter conditions by doing the following in the **Data Element Filters** grid:

1. Select the check boxes adjacent to the required member names in the **Available Filters** section and click **Select**. The selected member is displayed in the **Selected Member** section. Click **Select All** to select all the Members. You can also click **Deselect** to deselect a Member or click **Deselect All** to deselect all the Members. You can also click **Search** button to search for a member in the *Data Element Filter Search* dialog using **Folder Name** and **Filter Name**.

2. Click **Save** to validate the entries and save the filter details.

## Define Attribute Filter

When you have selected the Filter Type as Attribute, define the Filter conditions by doing the following in the Attribute Selection section:

1. Select the required **Dimension** from the drop down list.
2. Select the associated **Attribute** from the drop down list. The list consists of only Dimension Type attributes for selection.
3. Click Attribute Values button in the Attribute Values grid.
4. The Attribute Values window is displayed. In the Attribute Values window, the Dimension field under Dimension grid is auto populated with the Dimension name with which the selected Attribute is defined and is non-editable. In the Search grid you can search for Attribute Values depending on Alphanumeric Code, Numeric Code, or Name.
5. Select the checkbox(s) adjacent to the Alphanumeric Codes of the required Attribute Values in the Attribute Values grid and click OK. The Attribute Values grid displays the selected attribute values. Select Attribute Value(s) in the Attribute Values grid and click **Delete** button to delete it. You can use the Attribute Values present in the Attribute Values grid to generate conditions.
6. Click **Add** button in the Attribute Values grid. The Filter Conditions grid is populated with the filter condition using all the Attribute values.

**Note:** You cannot define two conditions using the same attributes. Because conditions are joined with a logical 'AND' and this will make the query invalid.

7. In the Filter Conditions grid, you can select a condition to view the Attribute Values used to generate it and can update the condition. You can also click **View SQL** button to view the SQL statement in View SQL window. Click **Expand** button to view a long filter condition in View Condition dialog
8. Click **Save**. The Attribute Filter definition is saved.

## CHAPTER 9

*OFSAA Expressions***Expressions**

An Expression is a user-defined tool that supplements other IDs and enables to manipulate data flexibly. Expression has three different uses:

- To specify a calculated column that the Oracle Financial Services Analytical Application derivatives from other columns in the database.
- To calculate assignments in data correction.
- To create calculated conditions in data and relationship filters.

Calculations like average daily balances, current net book balance, average current net book balance, and weighted average current net rate can be created through Expressions.

This chapter describes the creation and maintenance of OFSAA Expressions. Specifically, this chapter will cover:

- Expressions Summary & Detail Screens
- Navigation within the Expressions Summary Screen
- Navigation within the Expressions Detail Screen

See [Chapter 2, "Overview of OFSAA Infrastructure"](#) for details on how Expressions are used in OFS Analytical Applications.

**Summary & Detail Screens**

Upon initially navigating to Common Object Maintenance > Expressions, a summary screen is displayed showing a set of Expressions. Using search criteria, you can control the set of Expression rules that are displayed. When you Add, Edit, or View a rule, a detail screen is displayed.

The screenshot shows the 'Expression Summary' interface. At the top, there are search filters for 'Folder Name' (set to PFTSEG) and 'Return Type' (set to Numeric). Below the filters is a table with the following data:

Expression Name	Folder Name	Return Type	Created By	Creation Date
<input type="checkbox"/> AUTOD_Expression_01	AUTOSEG	Numeric	PFTUSER	10/12/2017 11:07:51

At the bottom of the table, it indicates 'Page 1 of 1 (1-1 of 1 items)' and 'Records Per Page | 1'.

## Navigation within the Summary Screen

When you first navigate to the Expressions summary screen, the expressions stored within your current default Folder are presented in a summary grid. The Expressions summary screen has two containers: Search and Expressions.

### Search Container

Your default Folder functions as a search constraint. The value of your default Folder is set in Application Preferences. You may select a different Folder or you may remove the Folder constraint entirely by selecting the “blank” Folder, i.e., no Folder. You may also search by Expression Name or by Return Type. Return Types, shown in a drop-down list, include:

- Date
- String
- Numeric

### Search Control

Enter your desired search criteria and click on the Search control.

### Reset Control

Restores the default Folder, removes any Expression Name or by Return Type constraint you may have specified, and refreshes the screen.

## Expressions Container

The Expressions container presents a grid containing all of the Expression rules that meet your search criteria. The Expressions summary grid offers several controls that allow you to perform different functions when an Expression rule is selected.

Expression Name	Folder Name	Return Type	Created By	Creation Date
<input type="checkbox"/> AUTOD_Expression_01	AUTOSEG	Numeric	PFTUSER	10/12/2017 11:07:51

To select an Expression, click on a check box in the first column of the grid. More than one Expression can be selected at a time but this will cause some of the controls to become disabled. Clicking on a checkbox a second time de-selects the Expression.

You may select or deselect all of the Expressions in the summary grid by clicking on the check box in the upper left hand corner of the summary grid directly to the left of the Expression Name column header.

### Add

Clicking on the Add control begins the process of building a new Expression. The Add control is disabled if any rows in the grid have been selected.

### View

Selecting a single row out of the grid enables the View control. Clicking on the View control allows you to view the contents of an Expression on a read-only basis. The View control is only enabled when a single Expression has been selected.

**Edit**

Selecting a single row out of the grid enables the Edit control. Clicking on the Edit control allows you to modify a previously saved Expression. The Edit control is only enabled when a single Expression has been selected.

**Copy**

Selecting a single row out of the grid enables the Copy control. Clicking on the Copy control allows you to create a copy of an existing Expression. The Copy control is only enabled when a single Expression has been selected.

**Delete**

Selecting one or more rows out of the grid enables the Delete control. Clicking on the Delete control deletes the Expression(s) you have selected. OFSAAI will not allow you to delete Expressions which are utilized within other OFSAA rules, e.g., within a Data Element Filter, an Allocation rule, etc.

## ***Expressions Summary Grid***

The following columns categorize each Expression in the summary grid:

- Expression Name
- Folder Name
- Return Type
- Created By
- Creation Date

### **Expression Name**

Displays the Expression's short name. Performing a “mouse-over” on an Expression Name displays the Expression's description.

### **Folder Name**

Displays the name of the Folder in which a rule is stored.

### **Return Type**

Expression rules operate like SQL function calls in the sense that they return a single data value. Return Type indicates the type of data that is returned by an Express rule: String, Date, or Numeric. See additional discussion below under Expression Functionality.

### **Created By**

Created By displays the name of the user who created the Expression rule. If an Expression is saved as Read Only, only the user who created the rule, i.e., the owner, may modify the rule.

### **Creation Date**

Displays date and time at which the rule was initially created.



## Navigation within the Detail Screen

When you Add, Edit, Copy, or View an Expression, the Expressions Detail Screen is displayed.

The Audit Trail container is a standard footer container for every OFSAA rule type. It displays Created By, Creation Date, Last Modified By, and Modification Date on the Audit Trail tab. The User Comments tab may be used to add comments to any rule, subject to a maximum of 4000 characters.

## Expression Functionality

OFSAA Expressions operate similarly to SQL function calls in that they apply a formula to one or more database columns in order to return a single value. For example, you might apply a YEAR formula to a date column to yield a numeric return value of a year. For a single instrument row in your data having an origination date of 01/01/2010, this Expression would return the number 2010. As another example, you might construct an Expression that evaluates a balance times a rate times an accrual basis factor to return a current month revenue or expense number.

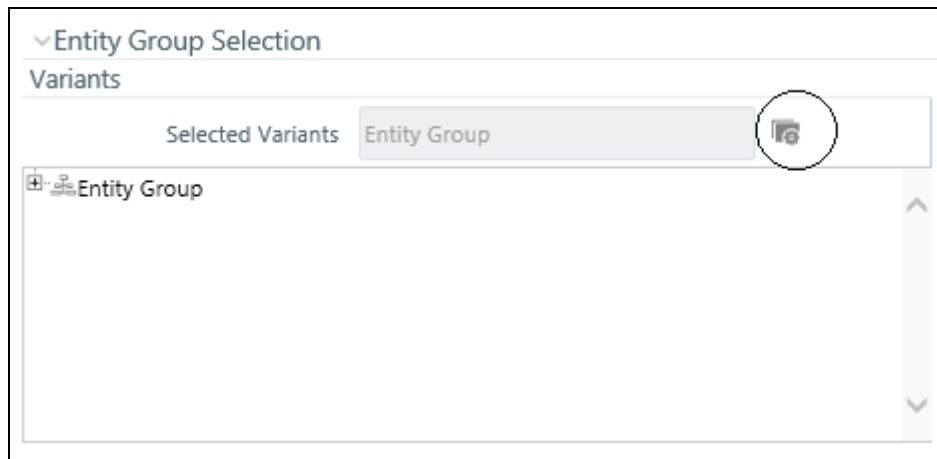
## Expression Details Container

As with any OFSAA rule type, you must supply a Name, Folder, and Access Type; you may optionally supply a Description.

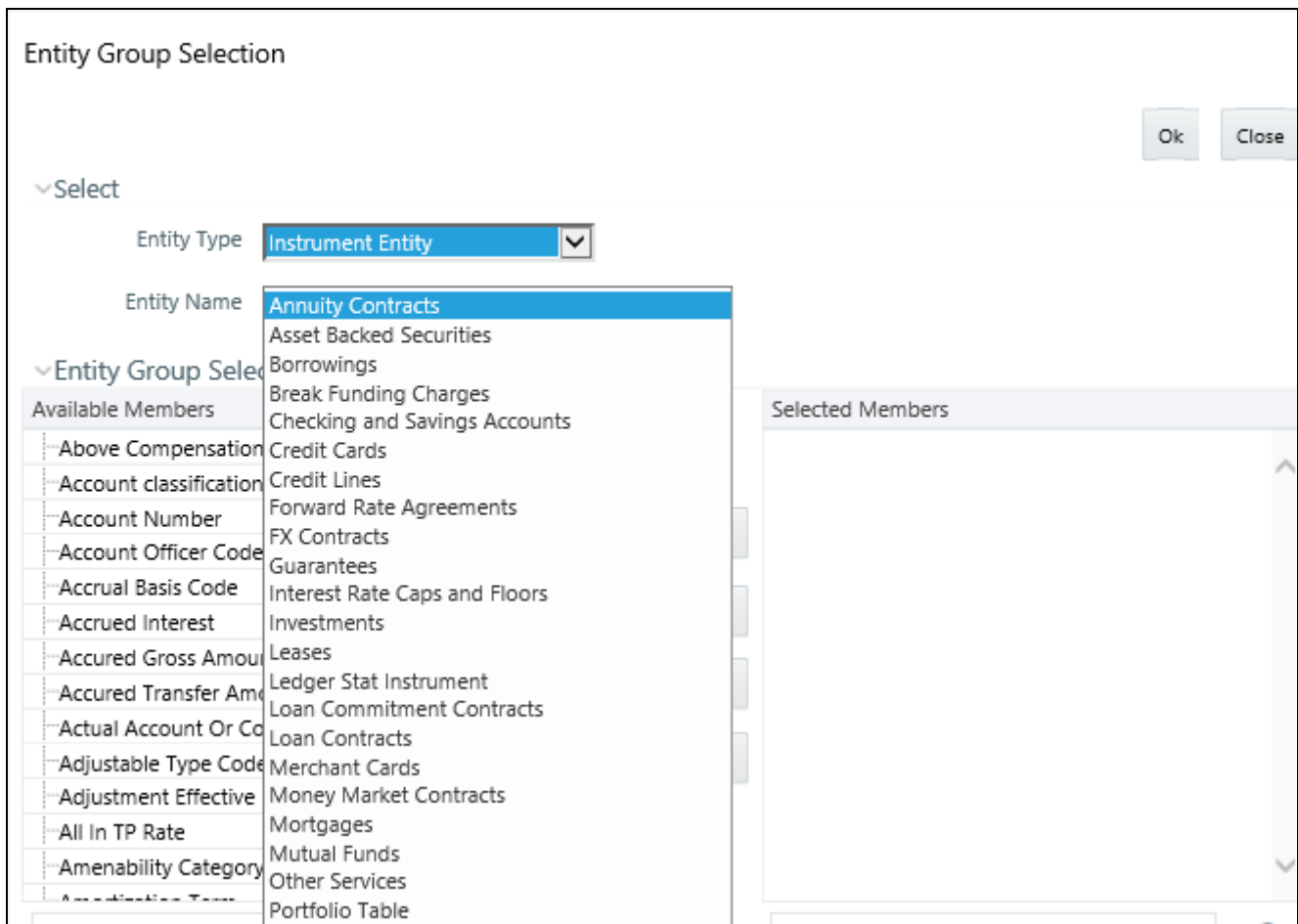
## Entity Group Selection Container

### Variants Sub-Container

To begin the process of creating a new Expression rule, you must specify the tables and columns (or even other expressions) that you want to use in the definition of your new Expression. Click on the ellipses (...) to the right of “Selected Variants” in the Variants sub-container of the Entity Group Selection container.



Clicking on the Selected Variants ellipses invokes an Entity Group Selection pop-up dialog. Here you may select an Entity Type and an Entity Name.



## Entity Type & Entity Name

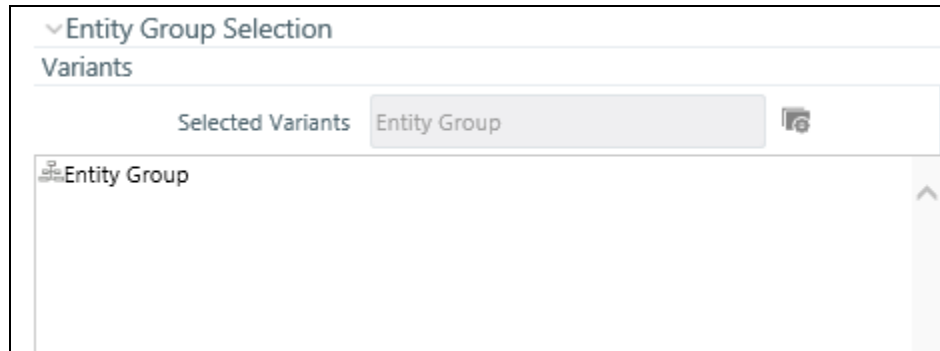
Entity Types include Instrument Entity, Non- Instrument Entity, and Expression.

### Instrument Entity Type

When you select the Instrument Entity type (the most commonly chosen type), the Entity Name drop-down list is populated with all registered Instrument and Transaction Summary type tables; the management ledger table (Ledger/Stat) is also available.

Select a table from the Entity Name drop-down list and then use the shuttle box controls (>, >>, <, <<) to select the column name or names you want to use within your Expression rule.

When you click OK, the Variant Selection dialog closes and Variants sub-container of the underlying Expression rule is updated. Click on the “+” control to expand the Entity Group hierarchy in order to see the tables and columns you have selected (see example below).



### ***Non-Instrument Entity Type***

Selecting the Non-Instrument Entity Type also populates the Entity Name drop-down list with Instrument and Transaction Summary type tables, but it additionally populates the Entity Name drop-down list with a number of metadata tables (tables that store non-business data).

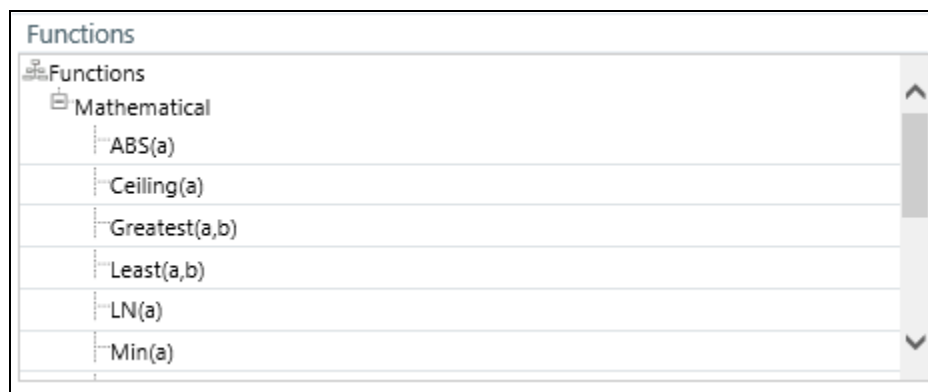
### ***Expression Entity Type***

Selecting the Expression Entity Type populates the Entity Name drop-down list with all of the currently defined Expression rules. The steps required to select an Expression rule are identical to those described above for selecting tables and columns.

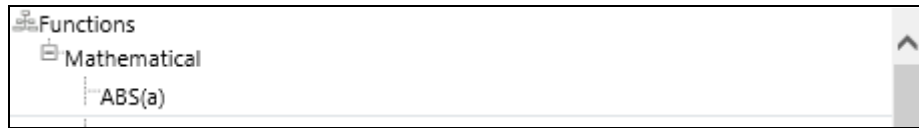
## ***Functions and Operators Sub-Containers***

Once you have specified the tables and columns and/or Expressions that you want to utilize within your new Expression rule, you work with Functions and Operators to construct your desired formula.

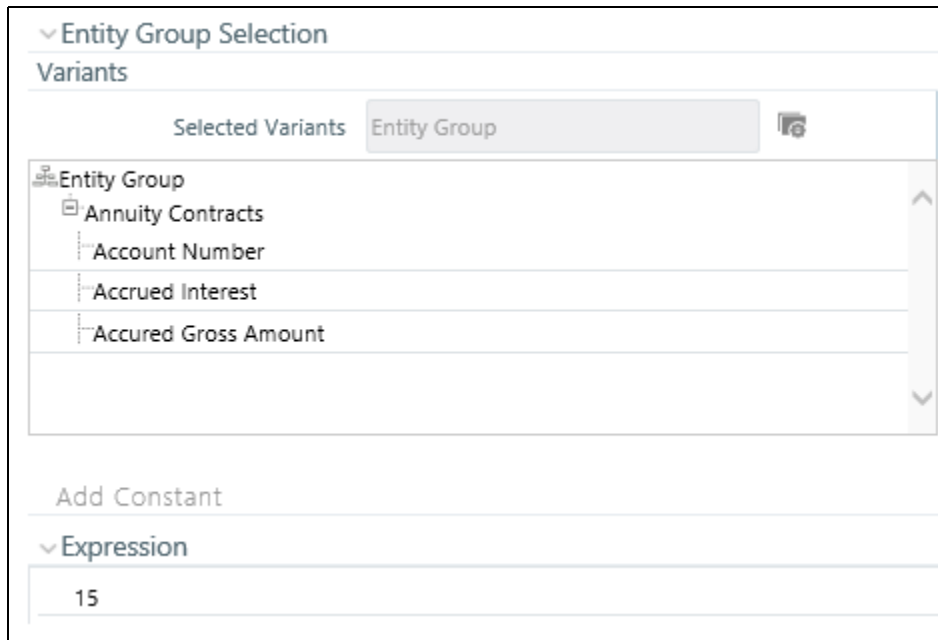
Double click on Variants (tables and columns and/or Expressions), Functions, and Operators to add them into the Expressions container.



For example, to construct an Expression that returns the absolute value of the Matched Spread column from the Mortgages table, begin by double clicking on the absolute value operator (ABS) in order to add it into the Expression container.



Next, double click on the Matched Spread column from the Mortgages table in order to add the column to the Expression container.



Finally, double click on the Right Parenthesis operator in order to close the expression.



Again, the result of your double clicking operation is to update the formula within the Expression container.

▼ Expression

15

+

LN(

63363

**Constants**

You may also add constant coefficient values to your formulas. For example, rates are generally stored in the form of percentages where 5.12 in the database means 5.12%.

You may wish to modify your absolute value formula to divide by 100. To do so you would continue by double clicking on the division operator (/). After the formula is updated to include the division operator, click on the Add Constant control located in the first row beneath the Entity Group Selection container.

▼ Entity Group Selection

Selected Variants Entity Group

Entity Group

- Entity Group
- └ Mortgages
  - Actual Account Or Contract Number
  - Matched Spread

Functions

- Mathematical
  - ABS(a)
  - Celling(a)
  - Greatest(a,b)
  - Least(a,b)
  - LN(a)
  - Min(a)

Operators

- Arithmetic
  - +
  - 
  - \*
  - /
- Comparison
- Other

Add Constant

▼ Expression

15

+

LN(

63363

Enter a constant value and click on the right arrow control, thus completes the expression.

▼ Add Constant

Constant Value  ✕

▼ Expression

15

+

LN(

63363

After the Variants, Functions and Operators are entered, click on Save in the New - Expression screen to validate the entries and capture the defined Expression.

The records are inserted successfully and are listed in Expressions list in the Expression Summary screen. Click Cancel if you want to ignore the entries and return to the Expression screen.

#### Operator types

The operators available are of 3 types:

- Arithmetic
- Comparison
- Other

Operatpr Type	Notation	Example Depicting Usage
Arithmetic	+	CUR_BOOK_BAL = CUR_PAR_BAL + DEFERRED_CUR_BAL
	-	AS_OF_DATE = MATURITY_DATE - REMAIN_TERM_C
	*	Remaining Balance after Offset = Opening balance - (Expected balance on every payment date * Mortgage offset %)
	/	CUR_PAYMENT = ORG_BOOK_BAL / (ORG_TERM / PMT_FREQ [in months])
Comparison	=	CUR_PAYMENT = principal + interest
	<>	If ADJUSTABLE_TYPE_CD <> 0, INTEREST_RATE_CD = 001 to 99999.
	>	If ORIGINATION_DATE > AS_OF_DATE, LAST_PAYMENT_DATE = ORIGINATION_DATE.
	>=	AS_OF_DATE >= ORIGINATION_DATE
	<	AS_OF_DATE < NEXT_REPRICE_DATE
	<=	If ORIGINATION_DATE <= AS_OF_DATE, LAST_PAYMENT_DATE >= ORIGINATION_DATE
Other	(	Parentheses group segments of an expression to make logical sense.
	)	MATURITY_DATE <= NEXT_PAYMENT_DATE + (REMAIN_NO_PMTS_C * PMT_FREQ)
	,	The comma separates statements of a function.

### **Function Types and Functions**

You select the type of function for your expression from the Type list. The choices are:

- Mathematical Functions
- Date Functions
- String Functions
- Other Functions

The type of function you select determines the choices available in the Function box. These unique functions in the Functions Sub-container enable you to perform various operations on the data. The following table lists each available function and Detail on the operations of each function in which it appears.

Function Type	Function Name	Notation	Description	Syntax	Example Depicting Usage
Mathematical	Absolute	ABS(a)	Returns the positive value of the database column	{ABS( } followed by {EXPR1 without any embedded or outermost left-right parentheses pair} followed by { )}	ABS (-3.5) = 3.5. ABS(F), ABS(F + C), ABS(F + C * R + F) are possible. However, ABS((F + C + R)), ABS((F + (MAX * CEILING))) are not possible.
	Ceiling	Ceiling (a)	Rounds a value to the next highest integer	Ceiling(column or expression)	3.1 becomes 4.0, 3.0 stays the same
	Greatest	Greatest(a,b ) GREATEST(column or expression, column or expression)	Returns the greater of 2 numbers, formulas, or columns	Greatest(column or expression, column, or expression)	Greatest(1.9,2.1) = 2.1
	Least	LEAST(column or expression, column or expression) Least (a,b)	Returns the lesser of 2 numbers, formulas, or columns	Least(column or expression, column or expression)	Least(1.9,2.1) = 1.9
	Natural Log	LN(number) LN(a)	Returns the natural logarithm of a number Natural logarithms are based on the constant e (2.71828182845904).	LN(number) where number is the positive real number for which you want the natural logarithm	LN(86) equals 4.454347 LN(2.7182818) equals 1
	Minimum	Min(a)	Returns the minimum value of a database column	Min(Column)	
	Maximum	Max(a)	Returns the maximum value of a database column	Max(Column)	



Function Type	Function Name	Notation	Description	Syntax	Example Depicting Usage
	Power	POWER(coefficient, exponent) Power(a,b)	Raises one value to the power of a second	{POWER()} followed by {EXPR1 without any embedded or outermost left-right parentheses pair followed by {,} followed by {EXPR1 without any embedded or outermost left-right parentheses pair} followed by {)}	<b>Valid examples:</b> POWER(F, R) POWER(F + C * R, F / R) <b>Invalid examples:</b> POWER(F + POWER, R) POWER( MAX, C)
	Round	ROUND (number, precision) Round(a,b)	Rounds a value to a number of decimal places	Round(x, n) returns x rounded to n decimal places	Round(10.52354,2)=10.52
	Sum	Sum(a)	Sums the total value of a database column. Sum is a multi-row function, in contrast to +, which adds 2 or more values in a given row (not column)	Sum(Column)	
	Weighted Average	WAvg (column being averaged, weight column) WAvg(a,b)	Takes a weighted average of one database column by a second Column. WAvg cannot appear in any expression. If you have two formulas called F1 and F2, both of which are WAvg functions, then you can form a third formula F3 as F1 + F2. If F3 is chosen as a calculated column, then an error message appears and the SQL code is not generated for that column. This is similar for nested WAvg functions if F3 is WAvg and it has F1 or F2 or both as its parameters.	WAvg(Column A, Column B)	WAvg(DEPOSITS.CUR_NET_RATE, DEPOSITS.CUR_BOOK_BAL)
<p><b>Note:</b> You cannot use the Maximum and Minimum functions as calculated columns or in Data Correction Rules. The Maximum, Minimum, Sum, and Weighted Average functions are multi-row formulas. They use multiple rows in calculating the results.</p>					

Function Type	Function Name	Notation	Description	Syntax	Example Depicting Usage																				
Date	Build Date	BuildDate(year,month,days)	Requires three parameters, (CCYY,MM,DD) (century and year, month, day). It returns a valid data and enables you to build a date from components.  Caution: If the parameters are entered incorrectly, the date is invalid.	BUILDDATE(CCY Y,MM,DD)	BuildDate(95,11,30) is invalid (invalid century).  BuildDate(1995,11,30) is valid.																				
	Go Month		Advances a date by x number of months. Go Month does not know the calendar. For example, it cannot predict the last day of a month. Typical functionality is illustrated in the following table:	GOMONTH(Date column, Number of months to advance)	GOMONTH(DEPOSITS.ORI GINATION_DATE,DEPOSITS .ORG_TERM)  <b>Valid examples:</b> GOMONTH(F, F + R + C) GOMONTH(F, R)  <b>Invalid examples:</b> GOMONTH(F + (R + C), MAX) GOMONTH((F * C), F)																				
		<p><b>Example:</b></p> <table border="1"> <thead> <tr> <th>Date Column</th> <th># of Months</th> <th>GOMONTH</th> <th>Comments</th> </tr> </thead> <tbody> <tr> <td>1/31/94</td> <td>1</td> <td>2/28/94</td> <td>Because 2/31/94 does not exist</td> </tr> <tr> <td>1/15/94</td> <td>2</td> <td>3/15/94</td> <td>Exactly two months: 15<sup>th</sup> to 15<sup>th</sup></td> </tr> <tr> <td>2/28/94</td> <td>3</td> <td>5/28/94</td> <td>Goes 28<sup>th</sup> to 28<sup>th</sup>: does not know that 31<sup>st</sup> is the end of May</td> </tr> <tr> <td>6/30/94</td> <td>-1</td> <td>5/30/94</td> <td>Goes back 30<sup>th</sup> to 30<sup>th</sup>: does not know that 31<sup>st</sup> is the end of May</td> </tr> </tbody> </table>				Date Column	# of Months	GOMONTH	Comments	1/31/94	1	2/28/94	Because 2/31/94 does not exist	1/15/94	2	3/15/94	Exactly two months: 15 <sup>th</sup> to 15 <sup>th</sup>	2/28/94	3	5/28/94	Goes 28 <sup>th</sup> to 28 <sup>th</sup> : does not know that 31 <sup>st</sup> is the end of May	6/30/94	-1	5/30/94	Goes back 30 <sup>th</sup> to 30 <sup>th</sup> : does not know that 31 <sup>st</sup> is the end of May
Date Column	# of Months	GOMONTH	Comments																						
1/31/94	1	2/28/94	Because 2/31/94 does not exist																						
1/15/94	2	3/15/94	Exactly two months: 15 <sup>th</sup> to 15 <sup>th</sup>																						
2/28/94	3	5/28/94	Goes 28 <sup>th</sup> to 28 <sup>th</sup> : does not know that 31 <sup>st</sup> is the end of May																						
6/30/94	-1	5/30/94	Goes back 30 <sup>th</sup> to 30 <sup>th</sup> : does not know that 31 <sup>st</sup> is the end of May																						
	Year	Year(date)	Year(x) returns the data for year x.	Year(Column) returns the year in the column, where the column is a date column.	Year(Origination Date) returns the year of the origination date.																				

Function Type	Function Name	Notation	Description	Syntax	Example Depicting Usage
	Month	Month(date )	Month(x) returns the month in x, where x is a numbered month.	Month(Column) returns the month in the column, where the column is a date column.	Month(9) returns September.  Month(Origination Date) returns the month of the origination date.
String	Trim All	AllTrim(a)		Trims leading and following spaces, enabling the software to recognize numbers (entered in All Trim) as a numeric value, which can then be used in calculating.	
Other	If statement	If(a=b,c,d)	The IF function should always have odd number of parameters separated by commas. The first parameter is an expression followed by a relational operator, which is in turn followed by an expression.  <b>Note:</b> Avoid embedding multiple individual formulas in subsequent formulas. This can create an invalid formula.	If(Condition, Value if True, Value if False).  {IF( } followed by EXPR2 followed by {>   <   <>   =   >=   <= } followed by EXPR2 followed by {,{ } followed by EXPR followed by },) followed by EXPR;n followed by { } where n = 1, 2, 3, .....	If(LEDGER_STAT.Financial=110, LEDGER_STAT.Month 1 Entry,0)  IF(((MAX + SUM) >= 30), F, POWER) are valid.
	Lookup	Lookup(Orig Col,LookupCol,...,ReturnedCol)	Enables you to assign values equal to values in another table for data correction. LOOKUP function should always have an odd number of parameters separated by commas and with a minimum of 3 parameters.  <b>Note:</b> Lookup is used exclusively for data correction.	Lookup(O1,L1,O 2,L2,...On,Ln,R) where O=Column from Original table L=Column from Lookup table R=Column to be Returned. So the previous statement would read: where O1=L1 and O2=L2... Returned value R	Valid examples: LOOKUP(F, R, R) LOOKUP(F, R, F, F, F) Invalid examples: LOOKUP(F) LOOKUP(F, R) LOOKUP(F + R, (F + R), MAX)

## Expression Container

The Expression container displays your formula step-by-step as you build it. When your formula is complete you may save the Expression rule. Saving an Expression rule invokes validation logic to ensure that you have constructed a legitimate formula, i.e., one that may be legally applied to your database.

The screenshot displays the 'Expression' configuration window. At the top right are 'Save' and 'Cancel' buttons. The 'Expression Details' section includes 'Expression Name' (AUTOD\_Expression\_01), 'Description', 'Folder Name' (PFTSEG), and 'Access Type' (Read Only / Read/Write). The 'Entity Group Selection' section shows a tree view with 'Entity Group' selected, containing 'Mortgages', 'Actual Account Or Contract Number', and 'Matched Spread'. The 'Functions' list includes ABS(a), Ceiling(a), Greatest(a,b), Least(a,b), LN(a), and Min(a). The 'Operators' list includes Arithmetic (+, -, \*, /) and Comparison. The 'Add Constant' section shows a 'Constant Value' of 63363. The 'Expression' grid contains the formula: 15 + LN(63363). At the bottom, an 'Audit Trail' section shows 'System ID :200335', 'Created By PFTUSER', and 'Created Date 10/12/2017 11:07:51'.

In the Expression grid, you can right-click the expression and do the following:

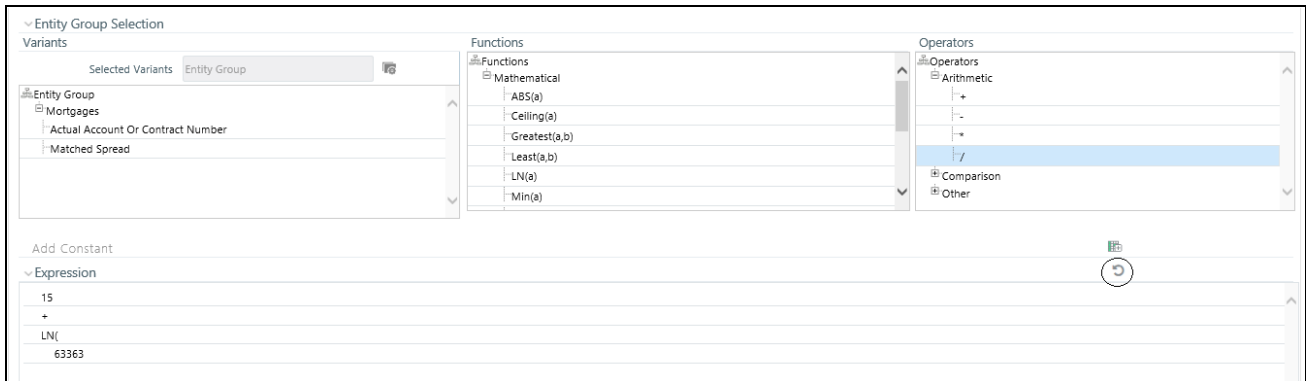
Click **Replace Expression** to replace the expression with a new one.

Click **Insert Expression After** to insert a new expression after the selected expression.

Click **Delete** to delete a selected expression.

You can also click **clear(-)** button in the Expression grid to clear the Expression.

If you encounter errors or if for any reason you want to reconstruct a formula, click on the Clear control at the top of the Expression container.



After clicking on the Clear control, you will be prompted to ensure you wish to clear the formula. If you answer OK, the formula is erased and you may start over in specifying your formula.



***Overview of Simplified Batches***

Simplified Batches provide a streamlined version of the functionality available in the broader standard Oracle Financial Services Analytical Applications (OFSAA) Infrastructure batch module (Common Object Maintenance > Operations). The Simplified Batch feature is designed to facilitate creation, maintenance, and execution of batches for the EPM application engines and various other OFSAA components. Users will find its drag-and-drop approach to task selection especially helpful in the management of large batches.

***Standard OFSAA Infrastructure Batching Functionality***

You may view, execute, schedule, and monitor Simplified Batches within the broader OFSAA Infrastructure Operations batching framework. However, Simplified Batches may be built and maintained only through summary and detail screens with Common Object Maintenance > Operations > Simplified Batch user interfaces.

***Batch Maintenance***

You may view Simplified Batches here, but they will be non-editable. You can create and modify them within the Simplified Batch screens, described later in this chapter.

***Batch Execution***

Simplified Batches may be executed either from the Simplified Batch summary screen (see details as follows) or from OFSAA Infrastructure > Operations > Batch Execution.

Other Operations Batch Functionality

Other Operations batch functionality including Batch Scheduler, Batch Monitor, Batch Processing Report, Batch Cancellation, and View Log apply equally to Simplified Batches and to batches that have been built under Common Object Maintenance > Operations > Batch Maintenance.

The remainder of this chapter describes the creation, maintenance and execution of Simplified Batches. Specifically, this chapter will cover:

- Simplified Batch Summary and Detail Screens
- Navigation within the Simplified Batch Summary Screen
- Navigation within the Simplified Batch Detail Screen
- Task Selection
- Running a Simplified Batch
- Viewing the Task Logs of a Simplified Batch
- Additional Batch Options

## Summary and Detail Screens

When you initially navigate to Common Object Maintenance > Operations> Simplified Batch, it displays a summary screen listing any existing Simplified Batch rules. Using search criteria, you can control the set of batches that are displayed.

When you Add, Edit, or View a rule, a detail screen is displayed.

### Navigation within the Summary Screen

The Simplified Batch summary screen presents two containers: a Search container and Simplified Batch container (the summary grid). When you first navigate to the Simplified Batch summary screen, the batches stored within your preferred Folder are presented in a summary grid. Your **preferred** Folder is determined by the set of Application Preferences that were last saved in which you selected the **Active for Master Maintenance** checkbox.

#### Search Container

The Search container allows you to restrict the batches visible within the summary grid by any combination of Name, Folder, or Status. Searches by Name are wildcard searches, that is, searching for Name like *pricing* will find all batches that include *pricing* within their names.

#### Search Control

Enter your desired search criteria and click the **Search** control.

#### Reset Control

Click the **Reset** control to clear any search criteria you may be using and refresh the Simplified Batch summary grid.

#### Simplified Batch Container

The Simplified Batch container presents a grid containing all of the batches that meet your search criteria. The Simplified Batch summary grid offers several controls that allow you to create a new batch or perform different functions when a batch is selected.

To select a batch rule, click a checkbox in the first column of the grid. More than one batch can be selected at a time but this will cause some of the controls to become disabled. Clicking on a checkbox a second time deselects the batch. You may select or deselect all of the batches in the summary grid by clicking on the checkbox in the upper left hand corner of the summary grid directly to the left of the Name column header.

You can sort on any column in the summary grid by clicking on the column header (the column name).

The screenshot shows the 'Simplified Batch' interface. At the top right, it says 'As of Date: 06/25/2014'. Below this is a search section with a 'Search' dropdown, a 'Name' text input field, a 'Status' dropdown menu set to 'Ongoing', and a 'Folder' dropdown menu set to 'AUTOSEG'. To the right of the search fields are search and refresh icons. Below the search section is a table with the following columns: Name, Creation Date, Created By, Last Modification Date, Last Modified By, Folder, Last Execution Date, and Status. The table is currently empty, displaying 'No Results Found' at the bottom. Above the table, there are several icons for actions like '+', 'edit', 'delete', 'refresh', and 'print', along with a '0 - 0 / 0' indicator and a 'Jump to page' input field.



## Add

Clicking the **Add** control begins the process of building a new Simplified Batch. The Add control is disabled if any rows in the grid have been selected.

## View

Selecting a single row in the grid enables the **View** control. Clicking the View control allows you to view the detailed definition of a Simplified Batch on a read-only basis. The View control is only enabled when a single batch has been selected.

## Edit

Selecting a single row in the grid enables the Edit control. Clicking the **Edit** control allows you to modify an existing Simplified Batch. The Edit control is only enabled when a single batch has been selected.

## Delete

Selecting one or more rows in the grid enables the Delete control. Clicking the **Delete** control deletes the batch or batches you have selected.

## Copy

Selecting a single row in the grid enables the Copy control. Clicking the **Copy** control allows you to create a copy of an existing Simplified Batch. The Copy control is only enabled when a single Simplified Batch has been selected. You can also duplicate a batch using the Save As option from the detail page, as described later in this chapter.

## Run

Selecting a single row in the grid enables the Run control. Clicking on the **Run** control executes the selected Simplified Batch rule. The Run control is only enabled when a single Simplified Batch rule has been selected. The As-of-Date and other parameters applicable to each task within a Simplified Batch will vary by task type (for details, see Specifying Task Parameters chapter in *Oracle Financial Services Analytical Applications Infrastructure User Guide*).

## Refresh

Click the **Refresh** control to refresh the Simplified Batch summary grid, including updates to the Last Execution Date and Status.

## ***Simplified Batch Summary Grid***

The following columns categorize each Simplified Batch in the summary grid:

- Name
- Creation Date
- Created By
- Last Modification Date
- Last Modified By
- Folder
- Last Execution Date
- Status

## Name

Displays the Simplified Batch rule's short name. Performing a **mouse-over** on a batch name displays the rule's system identifier, long name, and description. Note that the rule's long name consists of a concatenation of the Information Domain (InfoDom) and the rule's short name. This long name is the name under which the batch is submitted internally through Oracle Financial Services Analytical Applications (OFSAA) Infrastructure.

**Creation Date**

Displays the date and time at which a Simplified Batch rule was created.

**Created By**

Displays the name of the user who created a Simplified Batch rule.

**Last Modification Date**

Displays the date and time at which a Simplified Batch rule was last modified.

**Last Modified By**

Displays the name of the user who last modified a Simplified Batch rule.

**Folder**

Displays the name of the Folder in which a Simplified Batch rule is stored.

**Last Execution Date**

Displays the date and time at which a Simplified Batch was last executed.

**Status**

Displays the latest status for a batch rule and includes the following possible values:

- Complete
- Failed
- New
- Ongoing
- Cancelled

## ***Navigation within the Detail Screen***

When you Add, Edit, or View a batch, the Simplified Batch Detail Screen is displayed. The detail screen presents five containers for defining and maintaining your batch:

- Simplified Batch - header level information such as Rule Name, and so on
- Batch Execution Type - Sequential vs. Parallel
- Search Task Details - Search container
- Task Details - Task Details grid
- Audit Trail

The Audit Trail container is a standard footer container for every OFSAA rule type. It displays Created By, Creation Date, Last Modified By, and Modification Date on the Audit Trail tab. The User Comments tab may be used to add comments to any rule, subject to a maximum of 4000 characters.

The detail page also provides action buttons specific to the mode you are using (Add, Edit, or View). In View mode, you can duplicate the rule using the Save As button. This will allow you to specify the basic identifiers for the new rule, for example, Name, Description, Folder, and the Access Type selection.

### **Simplified Batch Container**

Specify the Simplified Batch Name and Description, select a Folder in which the batch is to be stored, and specify whether you want the batch rule to be **Read/Write** or **Read Only** (Access Type). Naming your batch is required before it can be saved.

**Note:** Unlike other OFSAA rules, you cannot edit the Name after you have created the rule.

The maximum length for the Name is based on len (name+infodomain), subject to the total length not exceeding 42 characters. Underscore (\_) is the only special character allowed for simplified batch name.

The maximum length for the Description is 300 characters. Allowed special characters for Simplified Batch description are underscore (\_), hyphen (-), Colon (:), Dot (.), and Space ( ).

Simplified Batch  
 Name\* Simple Batch Rule  
 Description This rule creates a batch from a Standard PFT Process followed by 7 allocation rules  
 Folder Name\* PFTSEG  
 Access Type\*  Read Only  Read/Write

### **Batch Execution Type Container**

Select whether you want the tasks in your batch to run in Parallel or to run Sequentially. Should you wish to run some tasks sequentially and some in parallel, you must utilize the OFSAA Infrastructure Operations batching framework to construct your batch. The Simplified batching framework provides a simpler and faster pathway to specify many of the batches that you will require, but the Infrastructure Operations framework offers you more flexibility.

Batch Execution Type  
 Batch Execution Type\*  Parallel  Sequential

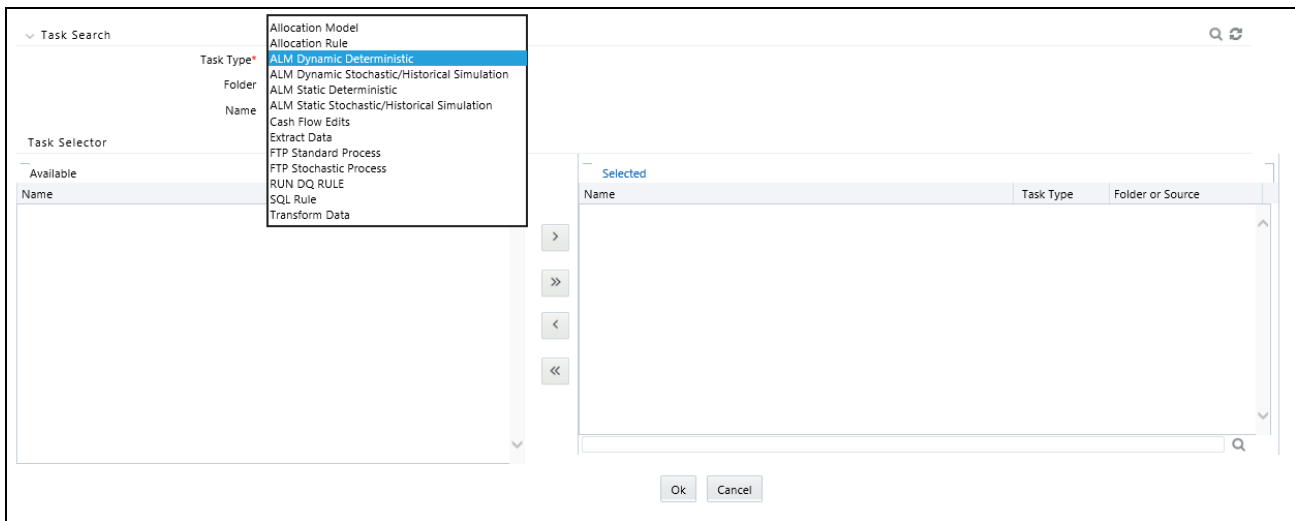
### **Search Container**

You may sometimes construct batches containing dozens or even hundreds of tasks. In such cases it may become difficult to isolate a specific task. The Search container allows you to **focus** the Task Details display to a specific task name or task type. The Rule Name search is a wildcard search. Click the **Search** control to execute a search. Click the **Refresh** control to remove your search criteria and refresh the Task Details Container to display all tasks within the batch.

Search Task Details  
 Rule Name  Task Type   
 Search Refresh

## Task Details Container

For a new batch rule, the Task Details container will initially be empty. Through the use of the Task Selector described below, you will populate the Task Details container with the processes that you wish to run in your batch. For a Sequential batch, the Task Details container presents the tasks in the order in which they will be run. For a Parallel batch, all tasks will be initiated in parallel.



## Task Number

Each task is itemized serially by task number.

## Rule Name

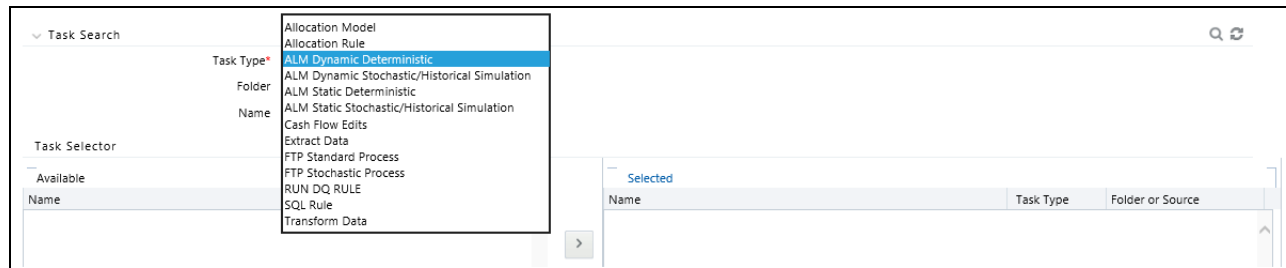
Displays the task's short name, such as an Allocation rule name. The complete or **long name** under which the task is submitted for execution (becoming the task's Batch Run ID) is a concatenation of several identifiers, as noted below in the View Log section.

## Task Type

Using Simplified Batches you may submit different tasks for each of the application engines and various other components. For ALM, the task types you may submit within a Simplified Batch include:

- ALM Dynamic Deterministic
- ALM Dynamic Stochastic/Historical Simulation
- ALM Static Deterministic
- ALM Static Stochastic/Historical Simulation
- Cash Flow Edits
- Extract Data
- RUN DQ RULE
- SQL Rule
- Transform Data

This list will vary based on installed Application pack(s).



### As of Dates and Additional Parameters

The As-of-Date applicable to each task within a Simplified Batch will vary by task type. Some task types also accept runtime parameters. For additional information on task parameters, see *Specifying Task Parameters* chapter in *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

### Folder or Source

For Extract Data tasks, values in this column represent Sources. For Transform Data tasks, this column is not applicable and is left blank. For all other task types, values in this column represent the Folder in which a rule is stored.

## Task Selection

### Controls on the Task Details Container

The Task Details container provides a Task Selector control that invokes a pop-up screen to support the maintenance of tasks in your batch. The Task Details container also provides a Delete control for removing tasks from your batch.



### Delete

Selecting one or more rows out of the Task Details grid enables the Delete control. Clicking on the **Delete** control deletes the task or tasks that you have selected. You can also remove tasks through the Task Selection pop-up screen.

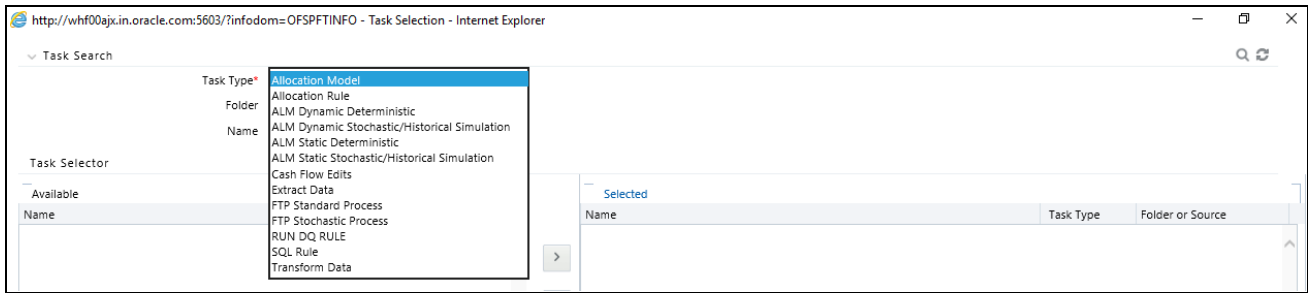
### Task Selector

The Task Selector control is always enabled. Clicking on the **Task Selector** invokes the Task Selection pop-up screen.

### Navigation within the Task Selection screen

Use the Task Selection screen to:

- Choose tasks to include in your batch
- Remove tasks from your batch
- Re-order tasks within your batch



**Task Search Container**

When you initially invoke the Task Selection screen, the list of **Available** tasks on the left-hand side of the Task Selector container is blank. To populate the list of available tasks, you must execute a search using the Search control. Prior to clicking on the Search control, you must select a specific type of task from the Task Type list box, and optionally specify search criteria for Folder (or Source) and Name, where applicable to the task type.

**Selecting Task Types**

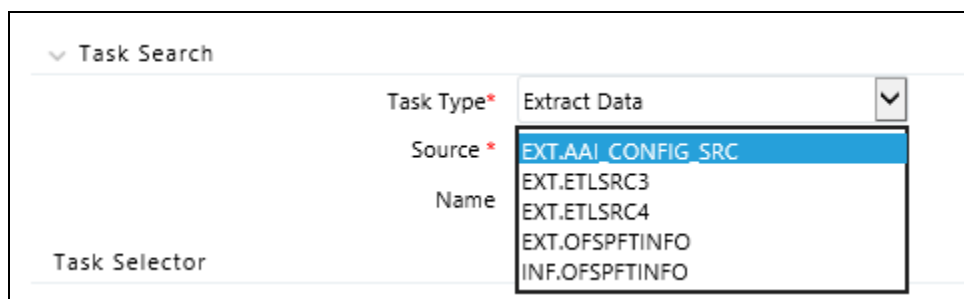
For ALM, the available task types include:

- ALM Dynamic Deterministic
- ALM Dynamic Stochastic/Historical Simulation
- ALM Static Deterministic
- ALM Static Stochastic/Historical Simulation
- Cash Flow Edits
- Extract Data
- RUN DQ RULE
- SQL Rule
- Transform Data

This list will vary based on installed Application pack(s).

**Source or Folder**

- **Extract Data Task**
- When you choose the Extract Data task, you must select a data source prior to executing your search.



- **Transform Data Task**
- When you choose the Transform Data task, it is not necessary for you to select a Folder prior to executing your search.
- **All Other Task Types**
- When you select any other Task Type, you may constrain your search to a particular folder. Alternatively, you may search for your tasks in all Folders by selecting **blank** (all folders).

The screenshot shows a 'Task Search' interface. On the left, there is a 'Task Search' label with a dropdown arrow. To the right, there are three input fields: 'Task Type\*', 'Source\*', and 'Name'. A dropdown menu is open, showing a list of task sources: 'EXT.AAI\_CONFIG\_SRC', 'EXT.ETLSRC3', 'EXT.ETLSRC4', 'EXT.OFSPFTINFO', and 'INF.OFSPFTINFO'. The 'INF.OFSPFTINFO' option is highlighted in blue.

Executing a search for tasks always refreshes the Available list of tasks, but never has any impact on tasks that you have already selected (tasks on the right-hand side of the screen entitled **Selected**).

### Name

You may further refine your search by entering a task Name. Searches by Name are wildcard searches, that is , searching for Name like *rate* will find all tasks that include *rate* within their Name.

## Task Selector Container

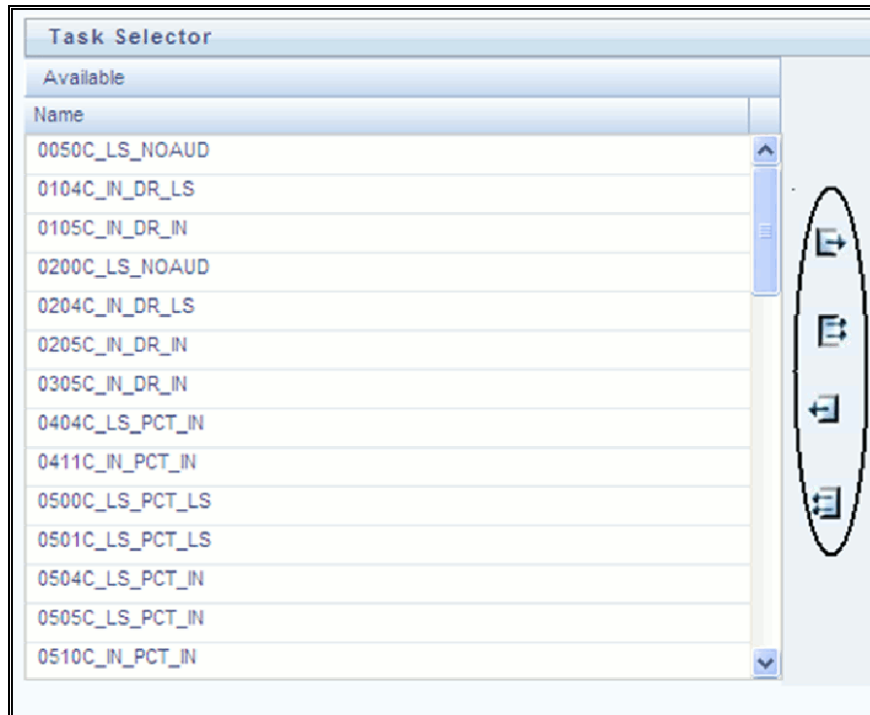
### Navigating the list of Available tasks

For task types with potentially large volumes of tasks, the Available list is divided into pages. Use the vertical scrollbar to navigate to the bottom of each page, then click **more** to display the names in the next page. To navigate back a page, scroll to the top of the page and click the hyperlink to return to the Previous page. You can control the number of tasks listed per page by changing the setting in `SIMPLE_BATCH_PAGE_SIZE` parameter in `FusionApps.properties` file in `WEB-INF/props` under context root. The recommended default value is 50 records/page. As you increase the value over a certain amount, you will start to see a slower response in the UI.

### Moving tasks between the Available and Selected lists

You may use the Select or Select All shuttle controls to move rules from the displayed page of the Available list to the Selected list or vice versa. The Select control moves rules from the left to the right, adding rules from the Available list to the bottom of the Selected list. Moving rules from the right to the left removes rules from the Selected list. You can also use the drag-and-drop feature (described as follows) to place the rules in a specific position in the Selected list to specify execution order for a Sequential type batch.

On both the Available side and the Selected side, you may highlight multiple tasks. To select a range of tasks, click on the first task, hold down the Shift key, and then click on the last task. To select a non-sequential list of tasks, you may select them individually while holding down the Control key.



Tasks within the Available list of tasks are disabled (greyed-out) once they have been selected (that is, included within the Selected list of tasks).

### Drag and Drop Operations

The Task Selector also supports drag and drop operations for choosing Available tasks, removing Selected tasks, or rearranging the sequence of your Selected tasks.

- You may click on an Available task and then drag and drop it anywhere within the Selected list, where the targeted placement is displayed in red line.
- Dragging a task from the Selected list and dropping it into the Available list removes that task from the Selected list.
- To select or deselect a range of tasks, click first task in the range to be selected, hold the Shift key down, click last task in the range to be selected, and then drag and drop your set of tasks.
- To select or deselect a set of tasks which are not in sequence, click each task individually while holding down the Control key; after clicking on the last task you may drag and drop the set of tasks.

### Finding a Selected Task

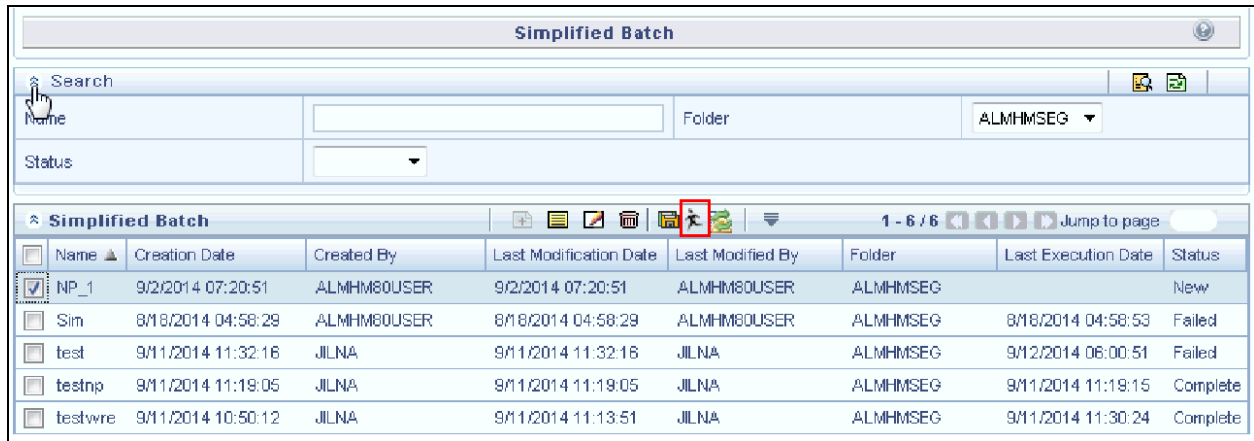
To search for any of your selected tasks, enter the task name or partial name in the field under the Selected list, then click the **adjacent** button to find each successive match.

## Running a Simplified batch

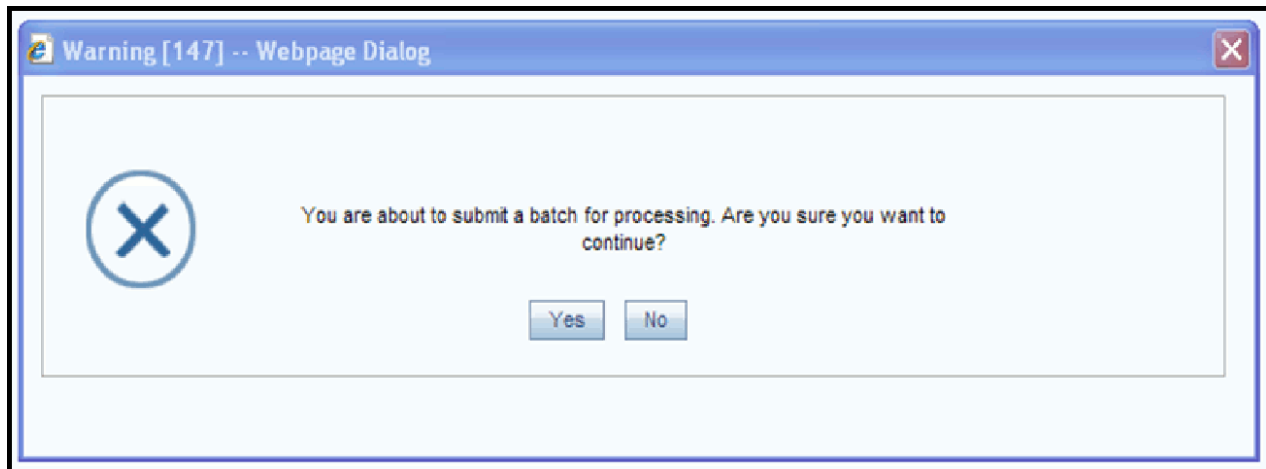
To run a Simplified Batch from the Simplified Batch summary page, complete the following steps:

1. Select a Simplified Batch from the summary grid.

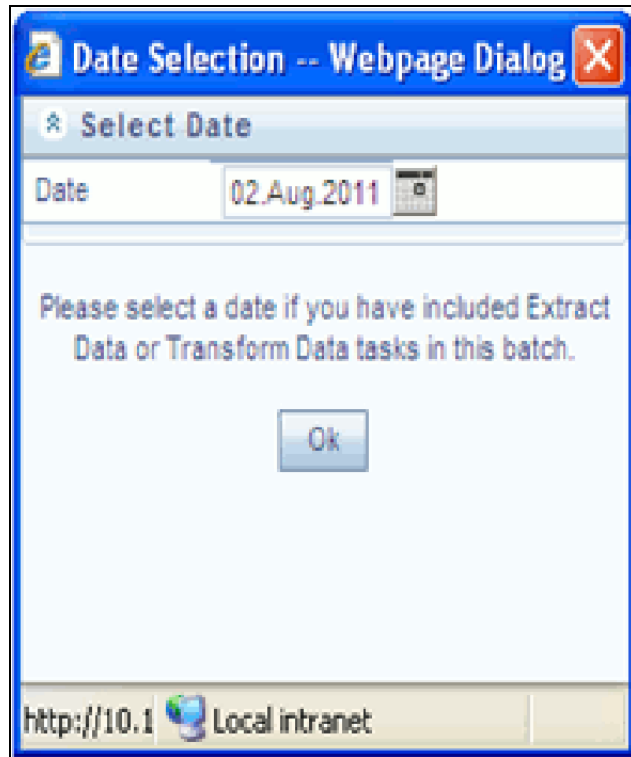




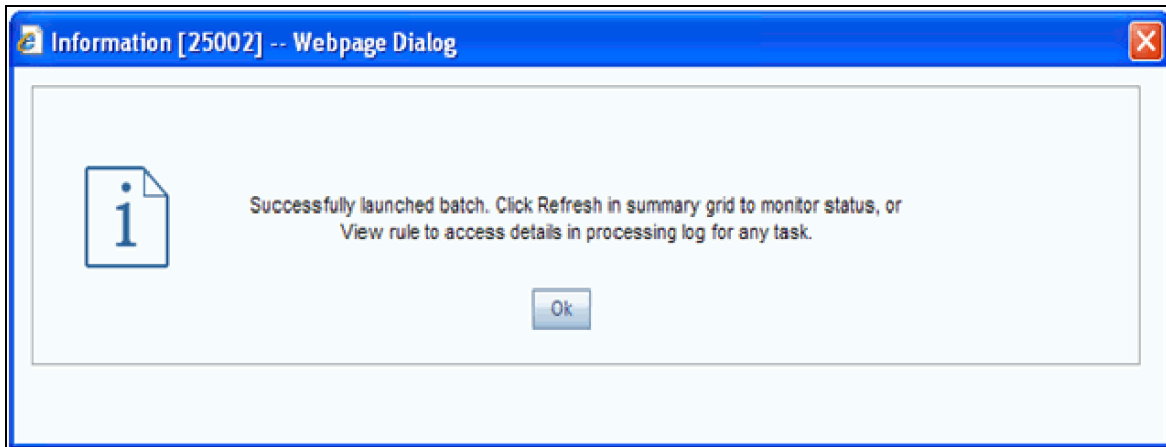
2. Click **Run** to execute the batch.
3. A popup window will appear asking for confirmation before proceeding. Click **Yes** to proceed, or **No** to return to the summary page.



4. If you clicked **Yes** to proceed, you will see a Date Selection popup. If you have included any Extract Data or Transformation Data tasks in your batch, select the appropriate date for use in processing (it will ignore this date for all other task types). The date defaults to the system date. If you wish to change the date, you can either directly enter a date in the format displayed, or you can use the Calendar tool to select a date.



5. Click **Ok** to proceed.
6. You will now see a message indicating the batch has been launched. Click **Ok** to proceed.



7. You can monitor the status from the summary grid or detail page.
8. To monitor from the summary grid, click **Refresh**. The summary grid will update the Last Execution Date with the current date and the Status at completion (for example, **Complete**, or **Failed** if there was a critical problem).
9. To monitor from the detail page, please refer to the section on Viewing the Task Logs.

---

**Note:** This currently applies to all task types except Extract Data and Transform Data.

---

## Viewing the Task Logs

**Note:** This currently applies to all task types except Extract Data and Transform Data.

To view the processing log for each task in your batch, you can navigate to the log from the Simplified Batch detail page:

1. Select your batch from the summary grid.
2. Click **View** to navigate to the detail page.
3. For a specific task in the Task Details container, navigate to the processing log through View Log.

The screenshot displays the 'Simplified Batch' detail page. The main content area includes several sections:

- Simplified Batch:** Fields for Name (smb a loc2), Description (another quick test), Folder Name (RTSEG), and Access Type (Read/Write, Read Only).
- Batch Execution Type:** Radio buttons for Parallel and Sequential (selected).
- Search Task Details:** Fields for Rule Name and Task Type.
- Task Details:** A table with the following data:
 

Task No.	Rule Name	Task Type	Optional Parameters	Folder or Source	View Log
01	SMB FTP Charge Credit Cards 1	Allocation Rule		RTSEG	VIEW LOG

At the bottom of the page, there are 'Save As' and 'Close' buttons, and a metadata section with 'Audit Trail' and 'User Comments' tabs. The metadata section includes:

- System ID: 40016278
- Created By: QTUSER
- Creation Date: 08/06/2011 05:38:34
- Last Modified By: QTUSER
- Last Modification Date: 08/06/2011 05:38:34

4. In the View Log page, you can drill down to the log details through the Task ID hyperlink.

As of Date	Component	Folder	Task Name	Task ID	Process Type	Status	Process Step	Records Processed	Number of Errors	Start Date	End Date	Elapsed Time	User	Batch Run ID
13.FEB.2013	Funds Transfer Pricing	EPM615EG	123	1000064538	FTP Standard Process	Completed		6	0	13.FEB.2013 16:45:06	13.FEB.2013 16:45:40	00:00:34	OFSAAUSER	EPM61INFO_test5_20130213_1

The Task ID represents the System ID of the task (for example, in Profitability Management, an Allocation Rule's System ID).

The task's Batch Run ID is a concatenation of several identifiers. For example, Batch Run ID 'OELPMINFO\_Task1\_smb alloc2\_20110805\_2' represents the following:

- Information Domain name (for example, OELPMINFO)
- Task sequence within the Simplified Batch (for example, Task1)
- Simplified Batch rule name (for example, smb alloc2)
- The run date in YYYYMMDD format and the sequential run number for that date (for example, 20110805\_2, for the second execution on August 5, 2011).

## ***Additional Batch Options***

As noted in the Overview for this chapter, you may also view, execute, schedule, monitor, cancel, and view task logs for Simplified Batches within the broader OFSAA Infrastructure Operations batching framework.

---

**CHAPTER 11**

# *Holiday Calendar*

This chapter discusses the procedure for creating a Holiday Calendar and generating a list of weekend and holiday dates. Individual applications may consume the holiday calendar events in different ways, so please refer to application specific user guides and/or the Cash Flow Engine Reference Guide for more details.

This chapter covers the following topics:

- [Overview of Holiday Calendars](#)
- [Searching for a Holiday Calendar](#)
- [Creating a Holiday Calendar](#)
- [Executing Holiday Calendar](#)
- [Holiday Exceptions](#)
- [Excel Import/ Export](#)

## ***Overview of Holiday Calendars***

A Holiday is a day designated as having special significance for which individuals, a government, or some religious groups have deemed that observance is warranted and thus no business is carried on this day. The Holiday Calendar code can range from 1 to 99999.

The procedure for working with and managing a Holiday Calendar is similar to that of other OFSAA business rules. It includes the following steps:

- Searching for a Holiday Calendar.
- Viewing and Updating a Holiday Calendar.
- Copying a Holiday Calendar.
- Deleting a Holiday Calendar.
- Check Dependencies in the Holiday Calendar definitions.
- Refresh the Holiday Calendar summary page.

## ***Searching for a Holiday Calendar***

Search for a Holiday Calendar to perform any of the following tasks:

- View
- Edit
- Copy
- Delete
- Check Dependencies

- Refresh

Name	Created By	Creation Date	Last Modified By	Last Modification Date	Status
Bug 16751123	CHANNAMMA	9/4/2014 17:34:44	CHANNAMMA	9/4/2014 17:49:25	Processed
copyadS1sdgfhgjhkjhkjkkjkkj	ALMHM80USER	8/18/2014 10:42:24	ALMHM80USER	8/18/2014 10:42:57	Defined
HM_Holiday_Calender1	ALMHM80USER	8/18/2014 18:42:41	ALMHM80USER	8/18/2014 18:42:41	Defined
Holiday	ALMHM80USER	8/18/2014 10:39:18	JILNA	8/20/2014 10:31:36	Defined
Holi	ALMHM80USER	8/18/2014 10:38:58	ALMHM80USER	8/18/2014 10:40:32	Defined

### Prerequisites

Predefined Holiday Calendar

### Procedure:

1. Navigate to the Holiday Calendar summary page. This page is the gateway to all Holiday Calendars and related functionality. You can navigate to other pages relating to Holiday Calendar from this page.
2. Enter the Search criteria.
  - Enter the name of the Holiday Calendar.
  - Click the Search icon.

Only holiday calendars that match the search criteria are displayed.

**Note:** You can control the number of rows to display on screen by selecting the "Pagination Options" icon from the action bar.

## Creating a Holiday Calendar

You create holiday calendars to capture holidays for a given date range for any organization. It is possible to create and use multiple holiday calendars.

### Procedure:

1. Navigate to the Holiday Calendar summary page.
2. Click Add Holiday Calendar.
3. The Holiday Calendar details page is displayed.
4. Enter a code value for the new holiday calendar.

**Note:** The code, is a numeric identifier for the holiday calendar. The code value must be a number between 1 and 99999. The code value you assign to the new holiday calendar must be unique.

- 
5. Enter the name and a brief description for the holiday calendar.

---

**Note:** The name you assign to the holiday calendar must be unique. Name can hold a maximum of 30 characters.

---

6. In the Holiday Properties grid, select not more than two weekend days. Then choose the Holiday Period. The Holiday Period can be defined for a range of up to 40 years less than the current date and 40 years greater than the current date, totally spanning a maximum of 80 years.
7. In the Holiday Details grid, define the Holiday details for the any period within the holiday range defined in step 6. There are two types of holidays that can be defined: **Fixed** and **Moving**.

A **fixed holiday** is one which is deemed as a holiday for every year in the holiday period, for that particular day. 25th December – Christmas, is a fixed holiday.

---

**Note:** To define a fixed holiday, input the holiday date for the first occurrence in the date range. For example, if your Date Range runs from 01-JAN-2000 to 31-DEC-2050, you should input the fixed holiday, Christmas, as 25-DEC-2000. The holiday calendar procedure will populate all subsequent 25-DEC entries in the holiday list table (FSI\_HOLIDAY\_LIST). A HOLIDAY\_TYPE code = 0 is a Fixed type holiday, code = 1 is a Moving type holiday, and code = 2 is a weekend.

---

The holiday calendar procedure will also ensure that holiday and weekend entries are not duplicated. For example, if weekends are defined as Saturday/Sunday and Christmas falls on a weekend day, there will be only one entry in the FSI\_HOLIDAY\_LIST table. The PREVIOUS\_WORKINGDAY and NEXT\_WORKINGDAY fields designate the valid prior and following working days, respectively.

A **moving holiday** is one which is deemed as a holiday only for that particular date and year, and not for every year in the holiday period. All occurrences of a moving holiday must be input manually.

20th August 2012 is a moving holiday on account of the muslim festival, Ramzan.

8. Once the holiday calendar definition is saved, its status in the summary page is marked as **defined**.
9. A holiday calendar created can also be deleted. Select one or more rows of holiday calendar definitions and click the Delete control.

### **Excel Import / Export**

Excel import/export functionality is used for adding/editing holiday calendar definitions.

## Executing Holiday Calendar

You execute a holiday calendar definition to generate calendar dates listing the various types of holidays for a given holiday period.

### Prerequisites

Predefined Rules

### Procedure:

1. Navigate to the Holiday Calendar summary page.
2. Search for a rule.
3. Select a Holiday Calendar and Click the **Generate Calendar Dates** icon to execute the selected holiday calendar. **Holiday list for holiday ID #1 generated successfully** message appears (**where #1 is the holiday calendar code**). The holiday list can be confirmed by querying the FSI\_HOLIDAY\_LIST table.
4. The status of a holiday calendar where holiday dates have been generated displays as “processed” in the status column in the summary page.

**Tip:** In case you do not want to Generate Calendar dates immediately, you can select that particular holiday calendar anytime later from the summary page with its status defined, and then click the **Generate Calendar Dates** icon to execute the selected holiday calendar.

5. The generated holiday list is no longer valid if
  - There is a change in the definition of the holiday calendar.
  - There is any update or modification to the Holiday Exceptions defined for that holiday calendar.

In such a case, the user will get a message “This holiday calendar has been modified, Please generate the holiday list again” and the holiday calendar state will be changed to “Defined” until the holiday list is regenerated with new definition.



## Holiday Exceptions

**Note:** This section is applicable to Release Version 6.1 and higher.

1. You can specify exceptions to holidays. As a prerequisite, a holiday calendar should have been properly defined and the status of the holiday calendar in the summary page should be **Processed**. Generating the holiday list will populate the holidays (weekends, fixed and moving) along with the working days. Then the **Show Exceptions** button is enabled in the detail page. Any changes in the holiday definition will disable the “Show Exceptions” button. The user must generate the holiday list again to define or view the exceptions.
2. Click **Show Exceptions** in the Holiday Exceptions grid. The Holiday Exceptions window opens.
3. The search block in the Exceptions page has 6 fields: From(Year) , To(year), Fixed Holidays, Moving Holidays, Holiday Date and All Exceptions.

**From and To** - Denotes the range of years which is a subset out of the holiday list generated, for which exceptions are required to be defined.

**Fixed Holidays** – You can filter the list of holidays by the type of Fixed Holidays.

**Moving Holidays** – You can filter the list of holidays by the type of Moving Holidays.

**Holiday Date** – For a particular known holiday date, exceptions can be defined.

**All Exceptions** - This checkbox when selected lists all the exceptions, if already defined, for the holidays within the From, To Date range.

The search result gives the list of all holidays based on the selection of the above search criteria fields.

4. In the Holiday Exceptions block, there are two types of exceptions that can be defined: **Not a holiday** and **Shift to**.

Any holiday can be marked as not a holiday, in which case that day is removed from the Holiday List. If the drop-down list in the exception type is selected as “Not a Holiday”, then the “shift to” date picker field is disabled. Spring earlier considered as a holiday in the holiday calendar can be marked as Not a Holiday in the Exceptions Window. Further the user can write his comments or remarks in the Notes Text Box next to the Exception Type drop-down list.

Any holiday can be shifted to another day, in which case the earlier declared holiday is removed from the Holiday List, while the shifted to day is included as a holiday.

## Excel Import/ Export

Excel import/export functionality is used for adding/editing holiday exceptions.

Holiday Exceptions - Edit - Windows Internet Explorer

**Holiday Exceptions**

Holiday Calendar > Holiday Calendar Definition > Holiday Exceptions

Search

From Year: 2008 To Year: 2012

Fixed Holidays: Moving Holidays:

Holiday Date: All Exceptions:

Holiday Name	Holiday Type	Holiday Date	Exception Type	Shift To	Notes
<input type="checkbox"/> Australia Day	Fixed	26-Jan-2008			
<input type="checkbox"/> Good Friday	Fixed	06-Apr-2008			
<input type="checkbox"/> ANZAC Day	Fixed	25-Apr-2009			
<input type="checkbox"/> ANZAC Day	Fixed	25-Apr-2010			
<input type="checkbox"/> New Years Day	Fixed	01-Jan-2011			
<input type="checkbox"/> Easter Monday	Fixed	09-Apr-2011			
<input type="checkbox"/> Queens Birthday	Fixed	11-Jun-2011			
<input type="checkbox"/> New Years Day	Fixed	01-Jan-2012			

Export Import

Apply Save Cancel

---

**CHAPTER 12*****OFSAA Rate Management*****Overview**

OFSAA Rate Management is a comprehensive utility enabling you to manage currencies, yield curves, and interest rate & currency exchange rate data with a high degree of security and control. OFSAA Rate Management also allows you to maintain economic forecasts such as GDP growth, inflation rates, or unemployment projections that may be linked to your models for interest rates, exchange rates, or new business growth.

Historical rate data obtained from OFSAA Rate Management is utilized within all of the Enterprise Performance Management (EPM) applications (OFSAA Funds Transfer Pricing, OFSAA Profitability Management, OFSAA Asset Liability Management, and OFSAA Balance Sheet Planning). Details regarding how each application draws upon Rate Management data are provided in the body of this chapter.

**Rate Management Subject Areas**

There are four primary subject areas or modules within Common Object Maintenance > Rate Management:

- Interest Rates
- Currency
- Currency Rates
- Economic Indicators

---

**Note:** Economic Indicators option is not applicable for Hedge Management.

---

**Interest Rates**

The quality and availability of interest rate information varies throughout the world. In many markets, gathering comprehensive rate information is a challenge because of insufficient security types, inconsistent quoting conventions, and lack of liquidity. The Interest Rates module within OFSAA Rate Management allows you to define and manage complex yield curve definitions using multiple rate formats and other rate attributes to give you data storage capabilities appropriate to your market. The Interest Rates module also supports the entry and maintenance of historical rate data for each yield curve you define.

Historical interest rate data from OFSAA Rate Management is utilized within OFSAA Transfer Pricing to generate transfer rates and option costs. Historical interest rate data is also utilized within OFSAA Asset Liability Management and within OFSAA Balance Sheet Planning in the generation of forecasted interest rate scenarios.

**Currency**

Financial institutions commonly transact business in more than one currency. Transacting business in multiple currencies demands functional capabilities for multi-currency accounting and currency rate management.

OFSAA Rate Management's Currency module supports the definitions and maintenance of currencies. Currency definitions are fundamental to the definition of both interest rate yield curves and currency exchange rates. A key attribute of every yield curve is the currency with which it is associated; and currency exchange rates can only be

established between defined currencies. OFSAA Rate Management provides a comprehensive list of ISO-defined currencies; you may also define and add your own user-defined currencies.

## Currency Rates

OFSAA Rate Management's Currency Rates module draws upon the currencies you have defined and activated in the Currency module to support the entry and maintenance of historical exchange rates. Currency exchange rates are utilized within:

- OFSAA Funds Transfer Pricing “Ledger Migration” processes (see *Oracle Financial Services Analytical Applications Funds Transfer Pricing User Guide*)
- OFSAA Asset Liability Management currency consolidation process (see *Oracle Financial Services Analytical Applications Asset Liability Management (OFSALM) User Guide*)
- OFSAA Profitability Management multi-currency allocations (see *Oracle Financial Services Analytical Applications Profitability Management (OFSPM) User Guide*)
- OFSAA Hedge Management and IFRS Valuations (see *Oracle Hedge Management and IFRS User Guide*)

## Economic Indicators

An economic indicator is any economic statistic such as the Consumer Price Index (CPI), the growth rate of the Gross Domestic Product (GDP), the unemployment rate, the Purchasing Managers Index, indices of consumer confidence, etc. Such macroeconomic statistics tell us how well the economy has behaved in the past. Some economic indicators are referred to as “lagging” indicators while others are classified as “leading” indicators. Leading indicators may provide insights into the future direction of the economy.

OFSAA Rate Management's Economic Indicators module allows you to define and store such historical indicators. Economic Indicators provide baselines from which OFSAA Asset Liability Management can generate forecasts of future values of economic statistics that can affect new business or other modeling assumptions.

## Interest Rates Summary Screen

When you first navigate to Common Object Maintenance > Rate Management > Interest Rates, an empty screen will be displayed. After you have constructed one or more interest rate curves, navigating to Common Object Maintenance > Rate Management > Interest Rates will display a summary screen showing each of the interest rate curves that you have previously built.

Code	Name	Rate Format	Compound Basis	Accrual Basis	Currency	Curve Type	Creation Date	Created By	Last Modification Date	Last Modified By
1	IRC Mortgage Valuation	Yield to Maturity	Annual	Actual/Actual	US Dollar	Interest Rate Curve	06/22/2015 15:57:29	PMUSER	06/26/2015 02:18:08	PMUSER
111	thomas_test	Zero Coupon Yield	Annual	Actual/Actual	US Dollar	Interest Rate Curve	06/26/2015 02:46:54	PMUSER	06/26/2015 02:48:04	PMUSER
801	CS FTP Funding Curve	Zero Coupon Yield	Annual	Actual/Actual	US Dollar	Interest Rate Curve	07/16/2015 21:09:33	CHRIS	07/17/2015 13:31:59	CHRIS
1860	jv_LIBOR SWAP USD	Zero Coupon Yield	Annual	Actual/Actual	US Dollar	Interest Rate Curve	07/15/2015 01:35:09	PMUSER	07/15/2015 01:37:47	PMUSER
7896	smbcopiedas7896	Zero Coupon Yield	Annual	Actual/Actual	US Dollar	Interest Rate Curve	07/30/2015 03:31:35	PMUSER	07/30/2015 03:33:07	PMUSER
7897	SMBtemp7897Hybrid	Zero Coupon Yield	Annual	Actual/Actual	US Dollar	Interest Rate Curve	07/29/2015 18:00:04	PMUSER	07/29/2015 18:00:25	PMUSER
7898	SMBtemp7898	Zero Coupon Yield	Annual	Actual/Actual	US Dollar	Interest Rate Curve	07/29/2015 18:05:39	PMUSER	08/03/2015 16:41:51	PMUSER
7899	SMBtemp7899	Zero Coupon Yield	Annual	Actual/Actual	US Dollar	Interest Rate Curve	07/29/2015 18:05:58	PMUSER	07/30/2015 03:30:54	PMUSER

---

## **Search Container**

A Search container is provided in which you may search for interest rate curves by Code, Name, Currency, Rate Format or Curve Type.

### **Search Control**

Enter your desired search criteria and click on the Search control.

### **Reset Control**

Clicking on the Reset control removes any Code, Name, Currency, Rate Format or Curve Type constraint you may have specified and refreshes the screen.

## **Interest Rates Container**

The Interest Rates container presents a grid containing all of the interest rate curves that meet your search criteria. The Interest Rates summary grid offers several controls that allow you to perform different functions when an interest rate curve is selected.

To select an interest rate curve, click on a check box in the first column of the grid. More than one interest rate curve can be selected at a time but this will cause some of the controls to become disabled. Clicking on a checkbox a second time de-selects the interest rate curve.

You may select or deselect all of the interest rate curves in the summary grid by clicking on the check box in the upper left hand corner of the summary grid directly to the left of the Name column header.

### **Add**

Clicking on the Add control begins the process of building a new interest rate curve. The Add control is disabled if any rows in the grid have been selected.

### **View**

Selecting a single row out of the grid enables the View control. Clicking on the View control allows you to view the contents of an interest rate curve on a read-only basis. The View control is only enabled when a single interest rate curve has been selected.

### **Edit**

Selecting a single row out of the grid enables the Edit control. Clicking on the Edit control allows you to modify a previously saved interest rate curve. The Edit control is only enabled when a single interest rate curve has been selected.

---

**Note:** You can control the number of rows to display on screen by selecting the "Pagination Options" icon from the action bar.

---

### **Copy**

Selecting a single row out of the grid enables the Copy control. Clicking the Copy control allows you to create a copy of an existing interest rate curve. The Copy control is only enabled when a single interest rate curve has been selected.

### **Check Dependencies**

Select an interest rate curve and then click the Check Dependencies control to generate a report on all rules that utilize your selected interest rate curve.

The Check Dependencies control is only enabled when a single interest rate curve has been selected.

### **Delete**

Selecting one or more rows out of the grid enables the Delete control. Clicking on the Delete control deletes the interest rate curves you have selected.

### **Execute the Historical Interest Rates Data Loader**

This Icon executes a function to import historical rates and parameters for all defined Interest rate Curves. For more information on setting up the automated process, see *Oracle Financial Services Analytical Applications Data Model Utilities User Guide*.

To execute a historical Interest Rate data load, click on the **Execute the Historical Interest Rates Data Loader** icon. A warning message will appear “Upload all available Interest Rates and Parameters?” Click ok, and all historical rates and parameters will be loaded.

### **Interest Rates Summary Grid**

The following columns categorize each interest rate curve in the summary grid:

- Code
- Name
- Rate Format
- Compound Basis
- Accrual Basis
- Currency
- Curve Type
- Creation Date
- Created By
- Last Modification Date
- Last Modified By

#### **Code**

Displays the interest rate curve's code. Code is a number in the range 1 to 99999. It is unique. Performing a “mouse-over” on a row within the grid displays the interest rate curve's detailed description.

#### **Name**

Displays the interest rate curve's short name.

#### **Rate Format**

Displays the interest rate curve's rate format (zero coupon or yield-to-maturity).

#### **Compound Basis**

Displays the interest rate curve's compounding basis (Annual, Semiannual, Monthly, or Simple).

#### **Accrual Basis**

Displays the interest rate curve's Accrual Basis (e.g., 30/360, Actual/Actual, etc).

**Currency**

Displays the currency (the Reference Currency) to which an interest rate curve is applicable.

**Curve Type**

Displays the curve type - Interest Rate Curve or Volatility Curve.

**Creation Date**

Displays the date and time at which an interest rate curve was created.

**Created By**

Displays the name of the user who created an interest rate curve.

**Last Modification Date**

Displays the date and time at which an interest rate curve was last modified.

**Modified By**

Displays the name of the user who last modified an interest rate curve.

## ***Interest Rates Detail Screen***

When you Add, Edit, or View an interest rate curve, the Interest Rate Code Detail Screen is displayed. The Interest Rate Code detail screen is comprised of an Interest Rate Code Details container, 5 Interest Rate Code tabs, and an Audit Trail container.

The Audit Trail container is a standard footer container for every OFSAA rule type. It displays Created By, Creation Date, Last Modified By, and Modification Date on the Audit Trail tab. The User Comments tab may be used to add comments to any rule, subject to a maximum of 4000 characters.

### ***Interest Rate Code Details Container***

This is explained in detail in the following sections.

**Interest Rate Code**

When constructing a new yield curve, you must specify an Interest Rate Code between 1 and 9999999. Interest Rate Codes are used internally to uniquely identify yield curves. When working with Rate Manager or other OFS Analytical Applications, you generally reference yield curves by Name, not by Interest Rate Codes. Interest Rate Codes, however, are embedded within your instrument data (for example, the INTEREST\_RATE\_CD and T\_RATE\_INT\_RATE\_CD columns within your instrument data are populated with Interest Rate Codes). Once you have saved a yield curve, you may not renumber its Interest Rate Code.



Interest Rate Code Details

Interest Rate Code \*   
(Hint: 1 - 9999999)

Name \*

Description

Volatility Curve

Risk Free

Display for All Currencies

Reference Currency \*

Structure Type \*

### Name & Reference Currency

You must also provide a Name and Reference Currency for your yield curve. Unlike Interest Rate Codes, you may rename or change the Reference Currency for previously saved yield curves. While you may choose to rename a yield curve, however, it is very unlikely that you will choose to modify a yield curve's Reference Currency. A yield curve's Reference Currency is the currency for which your market rates are valid. For example, the Reference Currency for a Prime Rate yield curve would be US Dollars. LIBOR or other internationally quoted rates are always quoted with respect to an underlying Reference Currency (e.g., US Dollar LIBOR, Euro LIBOR, etc). The drop-down list box for Reference Currencies displays only “Active” currencies. See the section below entitled Currency for a discussion of Active and Inactive currencies.

### Description

You may optionally provide a description of your yield curve; you may modify a yield curve's description at any time.

### Structure Type

Another required attribute for each yield curve is its Structure Type. Structure Type supports both Standard and Hybrid yield curve definitions. Hybrid yield curves are re-expressions of one or more pre-existing Standard yield curves. For additional information, see Hybrid Term Structure Tab under Interest Rate Code Tabs below. Once you have saved the yield curve, you may not change the Structure Type selection.

### Volatility Curve

You can select the Volatility Curve option to indicate the curve selected will contain volatility rates. If you select this option, all other curve attributes become disabled and the curve is used exclusively for managing volatility details. Both the Terms tab and Historical Rates tab are available with this option. Once you have saved the yield curve, you may not change the Volatility Curve selection.

---

**Note:** Volatility curves are used in Funds Transfer Pricing for calculating Rate Lock Option Costs.

---

### Risk Free

This is for tagging IRC's as risk free. This field is an optional field. This flag is editable both in new and edit modes. It is available for non-hybrid curves and hybrid curves and not available for Volatility Curves.

### Display for all currencies

This option allows you to designate certain Interest Rate Curves that you would like to make available for assumption mapping to any currency. Typically, assumption rules will filter the list of Interest Rate Codes based on the currency, when defining assumptions for a specific product/currency combination. When this option is enabled the Interest Rate Code will appear in assumption rules for all currencies.



## Interest Rate Code Tabs

Interest Rate Code tabs are used to define your yield curve and to add, edit, or delete historical interest rate data. The Interest Rate Code tabs are:

- Attributes
- Terms
- Historical Rates
- Parameters
- Hybrid Term Structure



### Navigating Between Interest Rate Code Tabs

For new yield curves, you must begin with the Attributes tab. Once you have selected attributes for a yield curve, you will typically not change them. When you have finished assigning attributes, navigate to the Terms tab in order to establish a term structure for your yield curve or volatility curve, i.e., an overnight rate, a one month rate, a three month rate, etc. To navigate to the Terms tab, you may either select the Apply button on the Attribute tab or you may simply click the Terms tab.

---

**Note:** You must have specified an Interest Rate Code, a Name, and a Reference Currency in the Interest Rate Code Details container before you can navigate to the Terms tab.

---

As with your yield curve attributes, once you have established a term structure for your yield curve you will only rarely come back to change it. The first time you navigate to the Terms tab, an initial 1 month term point is provided, but even if this is the one and only term point you want for your curve, you must select the Apply button to tell the system that you have finished your term structure specification. In future revisions to your curve's definition, you may navigate directly to the Historical Rates tab simply by clicking on it, but if you modify your term structure in any way, you must always select the Apply button on the Term tab before you can navigate to the Historical Rates tab.

The Historical Rates tab is used to input interest rate or volatility data. This is the tab you will most often return to for maintaining your interest rates database over time. To navigate to the Historical Rates tab, you may either select the Apply button on the Terms tab or you may simply click the Historical Rates tab if you have already established your term structure.

---

**Note:** You must have specified (1) an Interest Rate Code, a Name, and a Reference Currency in the Interest Rate Code Details container and (2) a term structure in the Terms tab before you can navigate to the Historical Rates tab.

---

### Attributes Tab

Yield curve attributes include Rate Format, Compounding Basis, Accrual Basis, and Curve Identifier. Once you have initially saved a yield curve, you typically will not change these attributes, but Rate Management will allow you to do so.

**Note:** All Attribute fields will be disabled, if Volatility Curve is checked.

### Rate Format

As part of your yield curve definition, you must select either the Zero Coupon or Yield-to-Maturity rate format. Regardless of which format you select, rates entered into Rate Management (in the Historical Rates tab) are always entered in nominal form, e.g., 5.125% or 6.875%, not as discount factors. For details on how the two rate formats affect internal cash flow engine calculations, see the *Oracle Financial Services Analytical Applications Cash Flow Engine Reference Guide*.

### Compounding Basis

You must also select a compounding basis for your yield curve: Annual, Semiannual, Monthly, or Simple. Annual is the most common method. The Monthly option is enabled based on Rate Format selection (if Rate Format is selected as "Yield to Maturity"). For details on Compounding Basis and how different compounding bases affect cash flow calculations in OFSAA, see the *Oracle Financial Services Analytical Applications Cash Flow Engine Reference Guide*.

### Accrual Basis

You must also select an accrual basis for your yield curve. The Accrual Basis list depends on Compounding Basis selection. If the Compounding Basis is selected as **Annual**, **Semiannual**, or **Monthly**, then the following Accrual Basis types are available:

#### Actual/Actual

- 30/365
- Actual/365

If the Compounding Basis is selected as **Simple**, then the following Accrual Basis types are available:

- 30/360
- Actual/360
- Actual/Actual
- 30/365
- 30/Actual
- Actual/365

For details on Accrual Basis and how different accrual bases affect cash flow calculations in OFSAA, see *Oracle Financial Services Analytical Applications Cash Flow Engine Reference Guide*.

### Curve Identifier

This is a drop-down list that contains the 6 curve identifier types, that is, Par Treasury, Zero-coupon Treasury, Par LIBOR/Swap, Zero-coupon LIBOR/Swap, Prime Lending Rate, and Cost of Funds Index. These curve identifiers can be mapped to the Adjustable Rate Mortgage (ARM) indices used for ADCO prepayment model processing. The Reference Currency for these indices is always US Dollar.

**Note:** This identifier is optional when the IRC is not being used for ADCO prepayment model processing. This is applicable only in ALM and HM applications.

## Terms Tab

Use the Terms tab to construct your yield curve's term structure. You may specify as many yield curve terms as you wish ranging from 1 day to 100 years.

Term	Multiplier
5	Months
10	Months
15	Months

Apply

Note, when constructing a volatility curve, there are two term types available, e.g. Contract Term and Expiration Term. You must add the relevant terms for each of these dimensions. Select the Apply button after defining terms for each dimension to save the assumptions before proceeding.

## Adding New Term Points

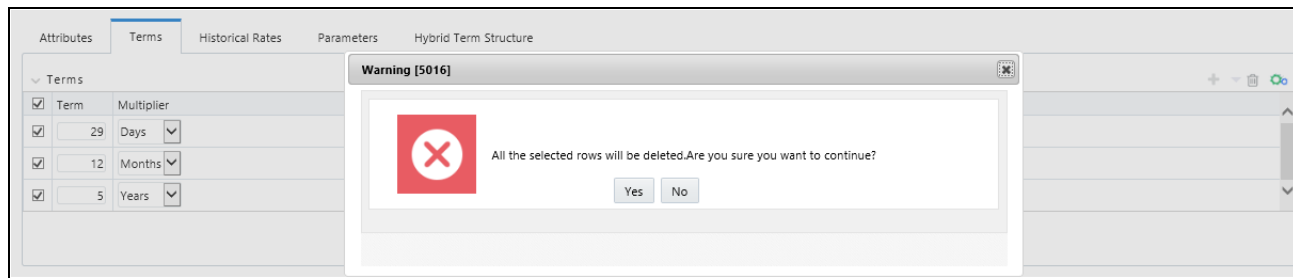
Click the Add control to add new term points. After clicking the Add control, select a Term value and a Multiplier (e.g., 7 days, 2 months, 5 years, etc). You can also add term points using the Data Input Helper.

Term	Multiplier
29	Days
12	Months
5	Years

Apply

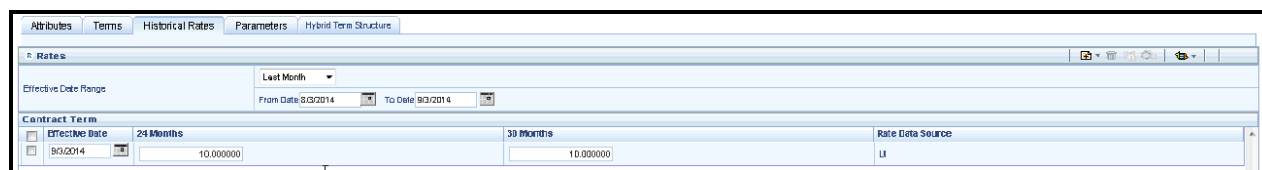
## Deleting Existing Term Points

If you wish to delete an existing term, select the term point (or terms) you wish to delete and click the Delete control, and confirm that you want to delete the term (or terms) you have selected.



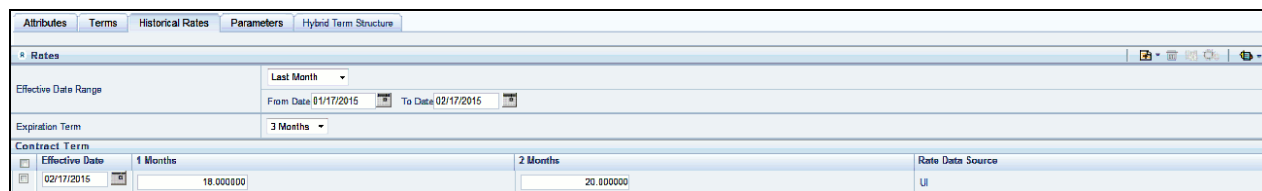
### Historical Rates Tab

Use the Historical Rates tab to enter, modify, or view interest rate data. Data should be entered as simple percentages (e.g., 5.125, 4.875, etc). If you are entering historical rates for a volatility curve, you will have to enter volatilities for the combination of contract term and expiration term. Select the desired expiration term from the drop list to complete the volatility matrix. Effective dates should be entered separately for each expiration term in the list.



By default, the Historical Rates tab will display interest rate data for the past month (i.e., for the 30 days leading up to the current date). Click the Effective Date Range drop-down list box to expand your view to the last 3 months, 6 months, one year, 3 years, 6 years or all rate data.

**Note:** Volatility curves are only applicable to FTP Rate lock option cost calculations.



If you are entering historical rates for a volatility curve, you will have to enter volatilities for the contract term. Select the desired expiration term from the droplist to complete the volatility matrix. Effective dates should be entered for each expiration term.

The Rate Data Source is a parameter that depicts from where the rates have been taken from, either they are entered through the UI or loaded through the data loader or generated using the Generate Rates of Hybrid IRC.

### Generate Graph

Generate Graph option allows you to generate the graph for selected Historical rates.

To generate a graph, follow the steps given below:

1. Select the Effective Date Range for which you want to generate the graph.

The From Date and To Date fields will be automatically updated after selecting the Effective Date Range.

2. Select the Effective Date using corresponding calendar options.

The screenshot shows the 'Rates' configuration window with the 'Contract Term' table. The table has columns for 'Effective Date', '12 Months', '24 Months', '36 Months', and 'Rate Data Source'. The 'Effective Date' column contains three dates: 3/4/2015, 2/4/2015, and 4/15/2015. The other columns are empty, and the 'Rate Data Source' column contains 'UI' for all rows. An 'Apply' button is visible at the bottom.

Effective Date	12 Months	24 Months	36 Months	Rate Data Source
3/4/2015				UI
2/4/2015				UI
4/15/2015				UI

3. Enter the term points in respective fields and click Apply button.

4. Select the Terms using the corresponding check-boxes.

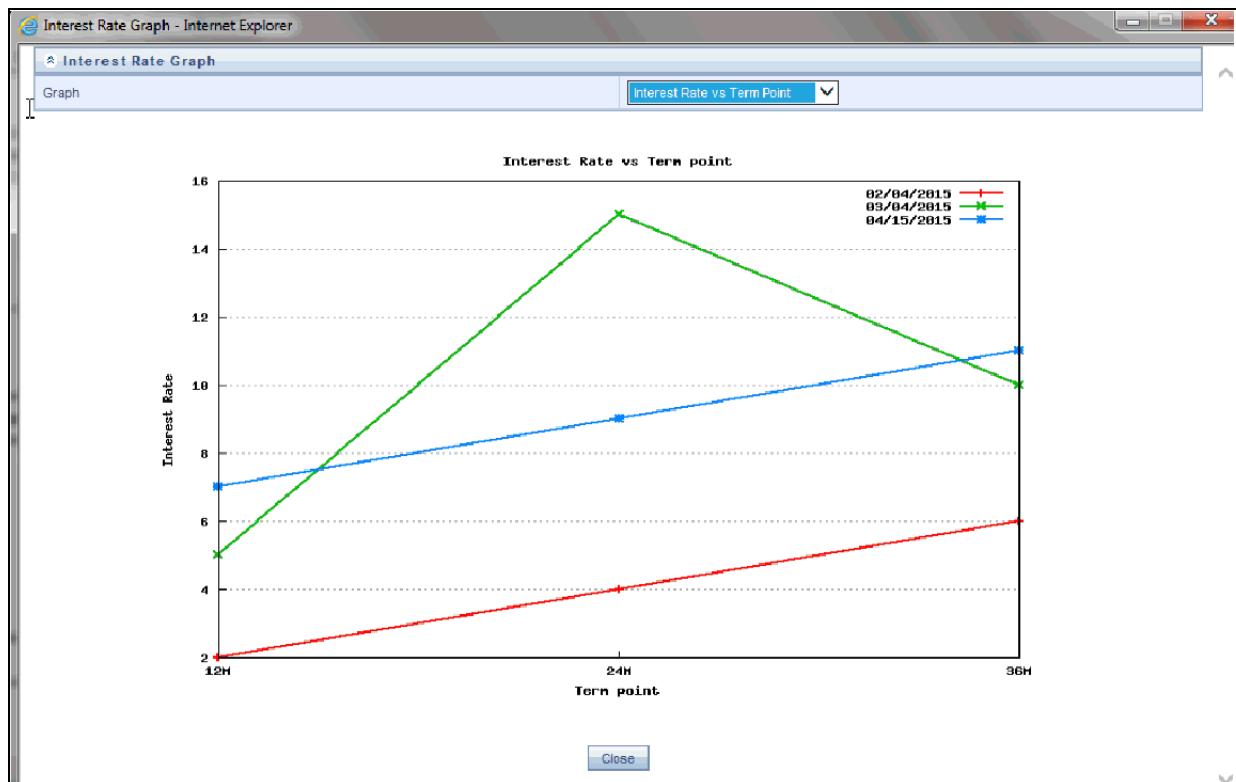
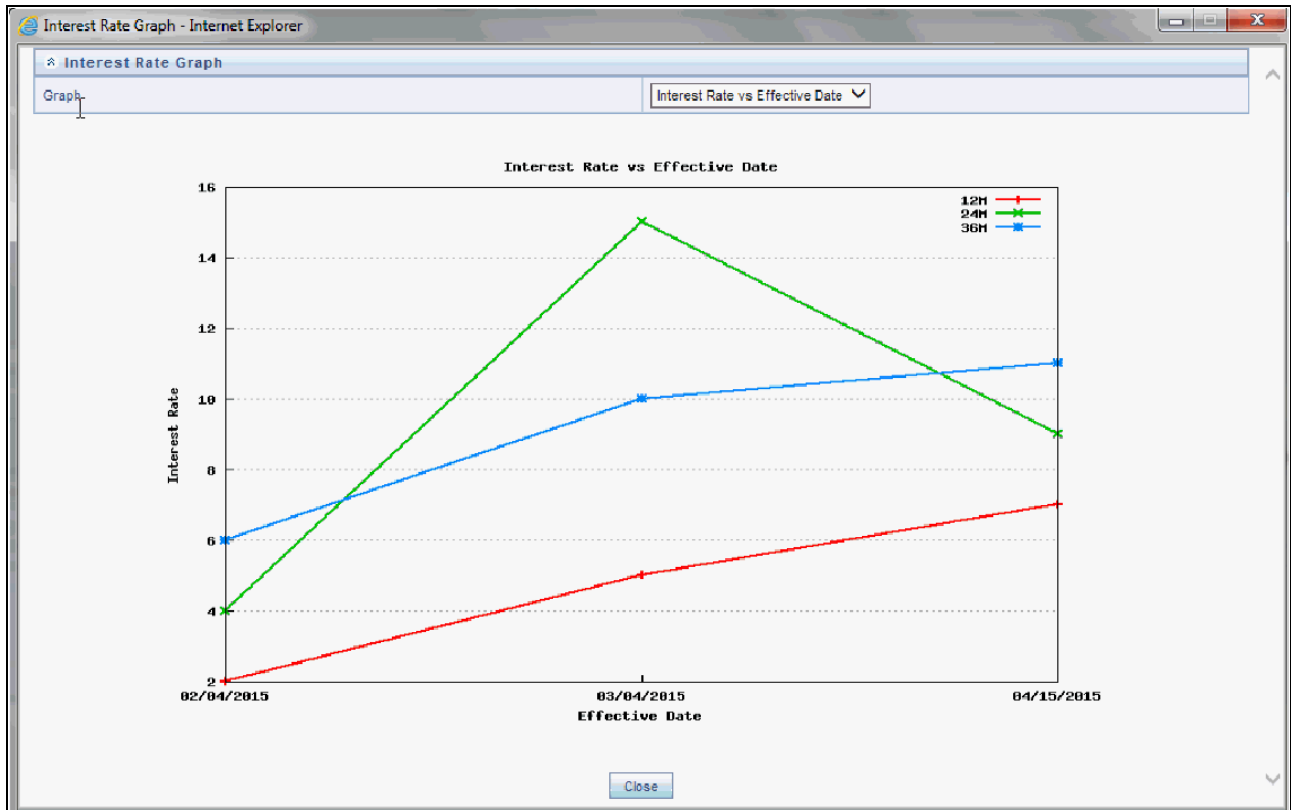
The screenshot shows the 'Rates' configuration window with the 'Contract Term' table. The table has columns for 'Effective Date', '12 Months', '24 Months', '36 Months', and 'Rate Data Source'. The 'Effective Date' column contains three dates: 3/4/2015, 2/4/2015, and 4/15/2015. The '12 Months' column contains values 5.000000, 2.000000, and 7.000000. The '24 Months' column contains values 15.000000, 4.000000, and 9.000000. The '36 Months' column contains values 10.000000, 6.000000, and 11.000000. The 'Rate Data Source' column contains 'UI' for all rows. Checkmarks are present in the 'Effective Date' column for all rows. An 'Apply' button is visible at the bottom.

Effective Date	12 Months	24 Months	36 Months	Rate Data Source
3/4/2015	5.000000	15.000000	10.000000	UI
2/4/2015	2.000000	4.000000	6.000000	UI
4/15/2015	7.000000	9.000000	11.000000	UI

5. Click on Generate Graph icon.

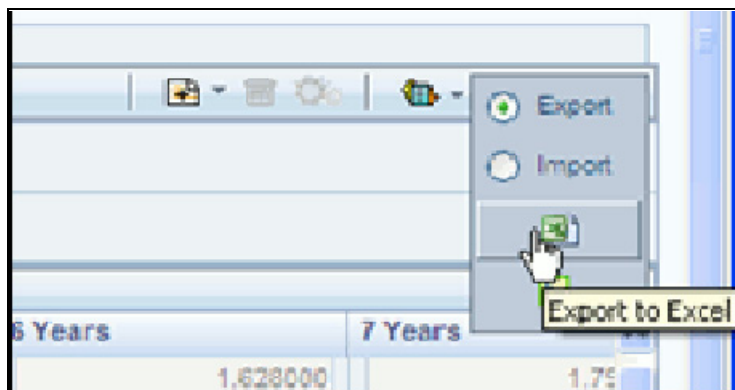


6. The graph will be generated. Here, you can view the graph for Interest Rate vs Effective Date or Interest Rate vs Term Point.

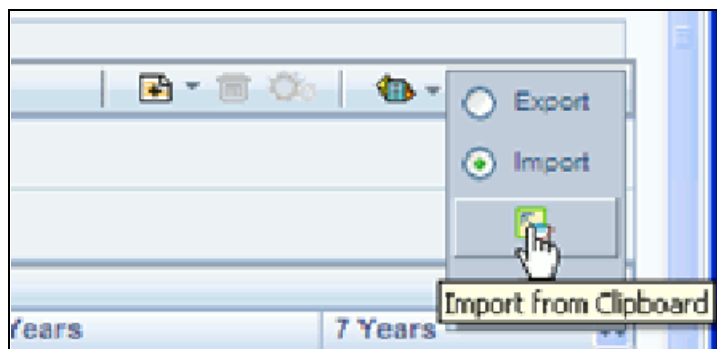


### Excel Import/Export

To aid in data entry, use the excel import/export functionality to add or edit rate data to historical rates. On the rates tool bar, click the excel icon. Click the 'export' toggle box to export data for the chosen selected effective date range. Within the same block, choose to export to excel. Doing so will launch the excel application and output the data grid, including headers.



You can add rows to the grid in the same format. After all data is input (or existing data is edited), copy the range of the grid you want to append, and go back to the rates screen. In the same block, click the 'import' toggle box, and directly below click the import from clipboard icon. The data copied from excel will appear in the rate history grid.



1. If appending data that preexisted for the same effective date, the import it will overwrite existing data.
2. In some cases, there are fields that will be output to the grid that are not editable, such as bucket start and end dates (when defining forecast assumptions). Be sure to only add or edit data in the columns that would be editable in the UI itself.

### Deletion of Historical Rates

You can delete historical rates entered by selecting one or more rows and then clicking the delete control.

### Data Input Helper

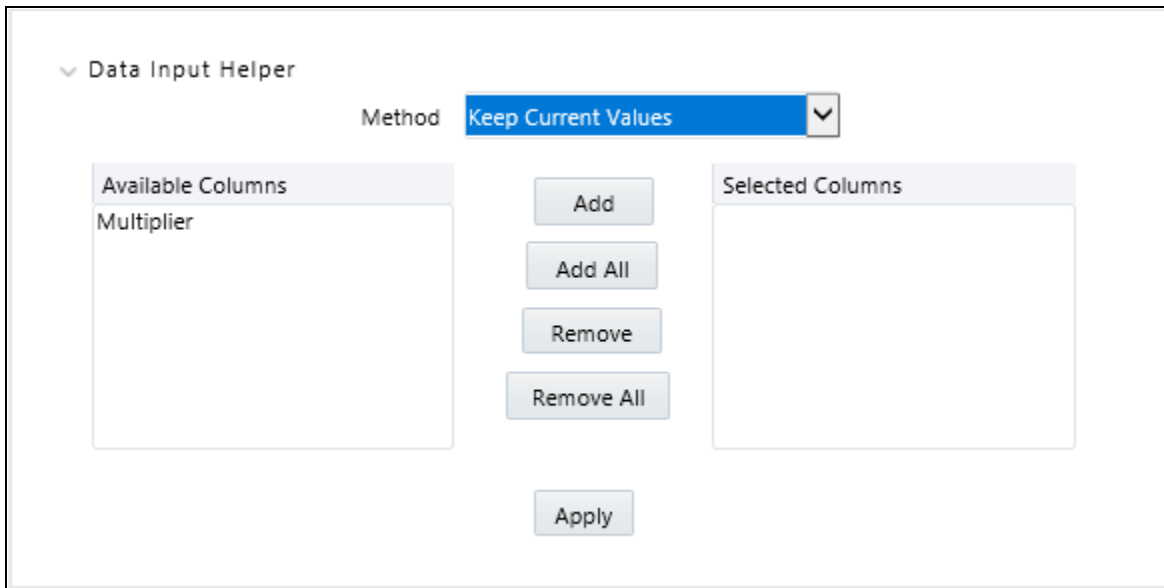
You may want to utilize the Data Input Helper to copy from a row where you have already defined the interest rate definition or apply a fixed value down the page. The following optional steps describe how to use this feature.

1. Select the checkbox next to the rows that you want to work with or use the “Select ALL” option by selecting the checkbox on the header row.

2. Select the Data Input Helper icon.



3. From the Data Input Helper – popup screen, select **Method** – “Grow by Amount” or some other appropriate method.



▼ Data Input Helper

Method **Keep Current Values** ▼

Available Columns

Multiplier

Add

Add All

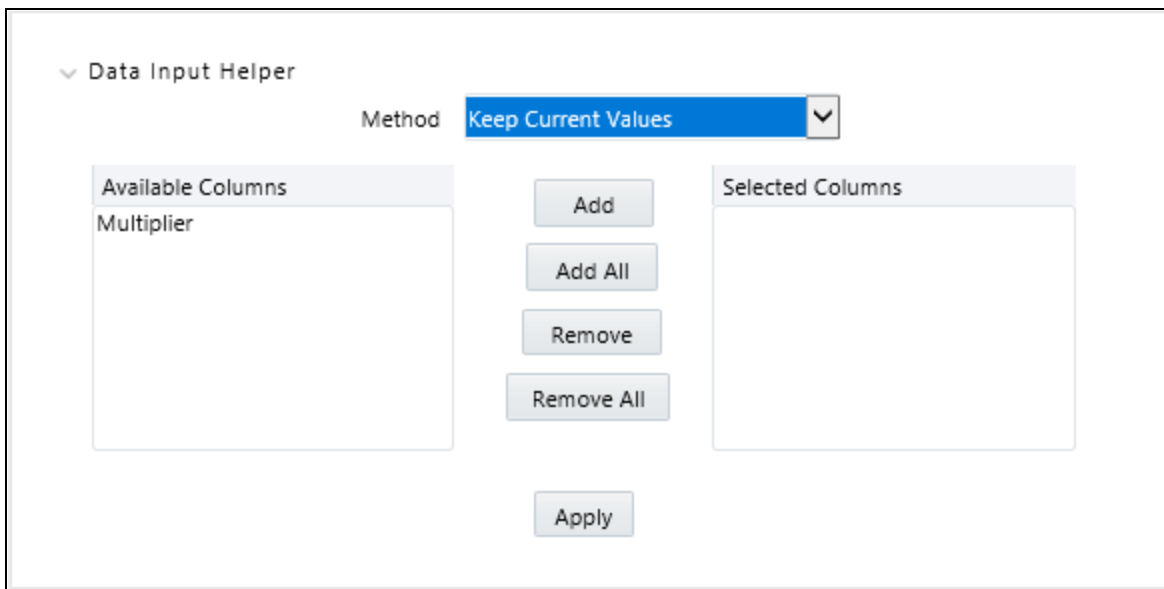
Remove

Remove All

Selected Columns

Apply

4. Select the term points from the left side (Available Columns) of the shuttle box.



▼ Data Input Helper

Method **Keep Current Values** ▼

Available Columns

Multiplier

Add

Add All

Remove

Remove All

Selected Columns

Apply

Based on selected **Method**, the list of **Available Columns** will vary. If **Method** is selected as "Increment By Months", "Increment By Days", or "Increment By Years", then only Effective Date option will be displayed in **Available Columns** list.



▼ Data Input Helper

Method **Keep Current Values** ▼

Available Columns

Multiplier

Add

Add All

Remove

Remove All

Selected Columns

Apply

After clicking on **Apply** button, you can modify the Start date and enter the Incremental Value.

▼ Data Input Helper

Method **Keep Current Values** ▼

Available Columns

Selected Columns

Multiplier

Add

Add All

Remove

Remove All

Apply

5. Select **APPLY**.

### Rate Lookup Behavior Between Term Points

The OFSAA cash flow engine is common to OFSAA Funds Transfer Pricing, OFSAA Pricing Management (Transfer Pricing Component), OFSAA Asset Liability Management, and OFSAA Balance Sheet Planning. In looking up rates from OFSAA Rate Management, the cash flow engine will, where necessary, perform an interpolation between yield curve term points. For example, in determining a straight term transfer rate (common for products such as time deposits), the engine may need to determine a three month rate from a yield curve that contains only a one month rate and a six month rate. In such a case, the cash flow engine will perform an

interpolation to determine the implied three month rate. While each of the above applications supports simple linear interpolation, OFSAA Transfer Pricing and OFSAA Asset Liability Management also support cubic and quartic spline interpolation methods. These more advanced methods will be supported for all OFS Analytical Applications in a future release.

### **Rate Lookup Behavior Beyond Term Points**

In cases where the cash flow engine needs to determine a rate from a yield curve for a term point smaller than the first term point of the yield curve, the engine will utilize the first term point of the yield curve. For example, if the engine needs to determine an overnight rate from a yield curve whose first term point is one month, the engine will utilize the one month rate. Similarly, in cases where the cash flow engine needs to determine a rate from a yield curve for a term point greater than the longest term point on the yield curve, the engine will utilize the last term point of the yield curve. For example, if the engine needs to determine a 30 year rate from a yield curve whose last term point is 10 years, the engine will utilize the 10 year rate.

### **Rate Lookup Behavior Between Effective Dates**

In looking up rates from OFSAA Rate Management for a business date, the cash flow engine may find that there is no rate data for that specific business date. For example, in generating an original term transfer rate for an instrument with an origination date of June 14, 2010, the cash flow engine may find rate data for May 31, 2010 and for June 30, 2010 but no rate data for any dates between May 31, 2010 and for June 30, 2010. In such cases, the cash flow engine always falls back to the latest available rate data prior to the business date of interest (May 31, 2010 in this case).

### **Parameters Tab**

Fixed income instruments are used for forecasting and simulating the Cash Flows. The Cash Flow Engine needs interest rate models to simulate the evolution of interest rates. The Cash Flow Engine uses these models as part of the stochastic engine. You can enter the parameters for these models in several ways:

- System-generated calculations via Parameter Estimation
- Direct input into the UI
- Excel Import
- UI entry via Data Input Helper methods
- Data Loader

Following interest rate models are available:

- Extended Vasicek
- Ho and Lee
- Merton
- Vasicek

### **Parameter Estimation**

This section explains the procedure to calculate estimated parameters.

### **Conditions for Parameter Estimation**

If the following conditions are met, you can calculate parameters for any Term Structure Model for a given Effective Date, based on your relative lookback term and a sufficient number of observations (available historical rates) for the

IRC. If you rerun with a different lookback term, it will overwrite the existing parameters for the selected Term Structure Model on that IRC's Effective Date.

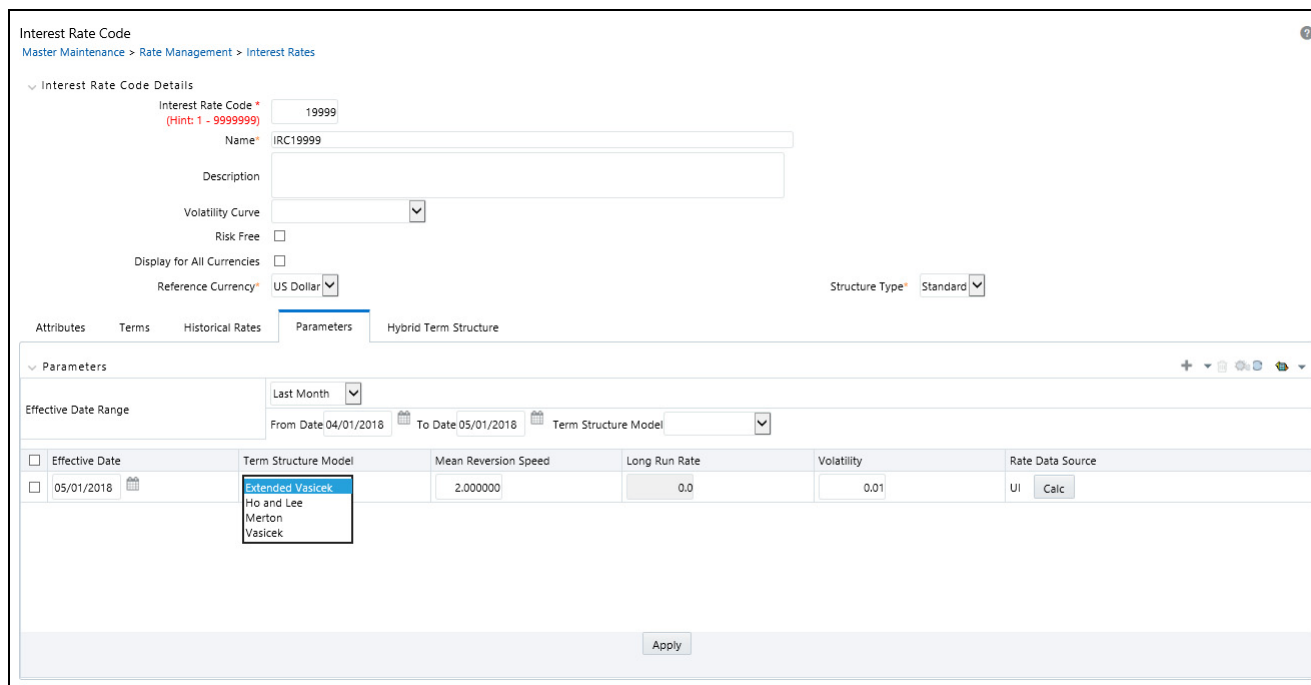
Conditions:

- **Term point:** Underlying historical rates must be available for a 30-Day or 1-Month term point.
- **Minimum number of historical rates:** A total of at least 10 historical rates ("observations") are required, on appropriate lookback dates.
- **Lookback dates:** Historical rates must be available on dates looking back from the Parameter's Effective Date (the End Date), in roughly 30-day intervals moving backwards from End Date to Start Date, for a minimum of 10 intervals.

For example: If the first rate's Effective Date is 1 Jan. 2013, then the second rate's date should be 2 Dec. 2012 (1 Jan. 2013 - 30 days = 2 Dec. 2012), and so on. If a rate is not found for the required date, the engine looks for a rate within the neighborhood of 5 days up or down (therefore a total range of 10 days), searching iteratively starting with Date -1, then Date +1, through Date +5, then Date -5. The next rate lookup would be 60 days prior to the End Date, and so on.

The minimum relative term for all lookbacks must be at least 300 Days (i.e., to accommodate a minimum of ten 30-day intervals). Using the above logic, if a rate is not found for the lookup date (or date within the neighboring range), an error will be logged in FSI\_PROCESS\_ERRORS with ID\_Number there identifying the Interest Rate Code, then the parameter estimation engine will exit.

**Procedure**



1. If you wish to edit or recalculate existing Parameters, you can enter the Effective Date Range filter.

Following Standard toolbar buttons are available here:

- Add

- Delete
  - Data Input Helper
  - Refresh
  - Excel Export/Import
2. Initially, default parameters for the Extended Vasicek Model will be displayed for one Effective Date (the system date on which the Interest Rate Code was created). You can edit these parameters. You can also add new parameters using the Add button. Use the Refresh button to return to existing Parameters.

**Note:** Steps 3 and 4 are applicable only if you are Adding or Editing Parameters.

If you are adding or editing parameters, you can use the Data Input Helper or Excel Import/Export. For more details, refer Excel Import/Export and Data Input Helper.

3. Enter the Effective Date. Here, Effective Date cannot be greater than Current System Date.
4. Select the Model from Term Structure Model drop-down list. Effective Date + Term Structure Model combination must be unique within this IRC.

The following term structure models are utilized in stochastic modeling of interest rates in OFSAA Transfer Pricing and OFSAA Asset Liability Management:

- Extended Vasicek
- Ho and Lee
- Merton
- Vasicek

5. The parameters needed by the models are as follows:

Model	Parameter 1	Parameter 2	Parameter 3
Vasicek	Volatility	Mean Reversion Speed	Long Run Rate
Ho and Lee	Volatility		
Merton	Volatility		
Extended Vasicek	Volatility	Mean Reversion Speed	

Values for Long Run Rate and Volatility are assumed to be percentages.

For example, a long run rate of 5% is displayed as 5.000. To maintain the integrity of your data, Rate Management restricts the input values it accepts. Note the valid range and default setting for each parameter.

Parameter	Valid Range	Default Value
Mean revision speed	0.00 to 500	0.0
Long run rate	0.00% to 500%	0.0
Volatility	0% to 500%	0.01

6. Rate Data Source will indicate if you have directly entered Parameters through the UI. It will also allow you to Calculate the Parameters to display system-generated values.

- When you select **Calculate** option, the following **Term Structures Parameter Estimation** window is displayed:

If you are directly entering the Parameters, then enter values appropriate to the Model, as noted above, then click **Apply** else click **Calculate**.

Interest Rate Model	Extended Vasicek	
End Date	01/06/2016	
Relative Term	Term	Multiplier
	<input type="text"/>	Days <input type="button" value="v"/>
Start Date	<input type="text"/>	
Number of Observations	<input type="text"/>	

- End Date will be auto-populated with Effective Date. Enter the Relative Term with Multiplier to define the rate lookback period. The Term/Multiplier value should be at least 300 Days.
- Start Date is automatically updated after entering Relative Term.

Interest Rate Model	Extended Vasicek	
End Date	01/06/2016	
Relative Term	Term	Multiplier
	300	Days <input type="button" value="v"/>
Start Date	03/12/2015	
Number of Observations	10	

- Click Calculate Number of Observations to confirm number of rates found for appropriate dates within the Relative Term. If there are at least 10 observations, the Estimate button will become active. For more information, refer to Conditions for Parameter Estimation section.
- Click Estimate to Calculate the Parameters and store them in the historical parameters table (FSI\_IRC\_TS\_PARAM\_HIST). You will then see a confirmation box. Click Ok. The calculation will complete and you will be directed back to the Parameters screen.

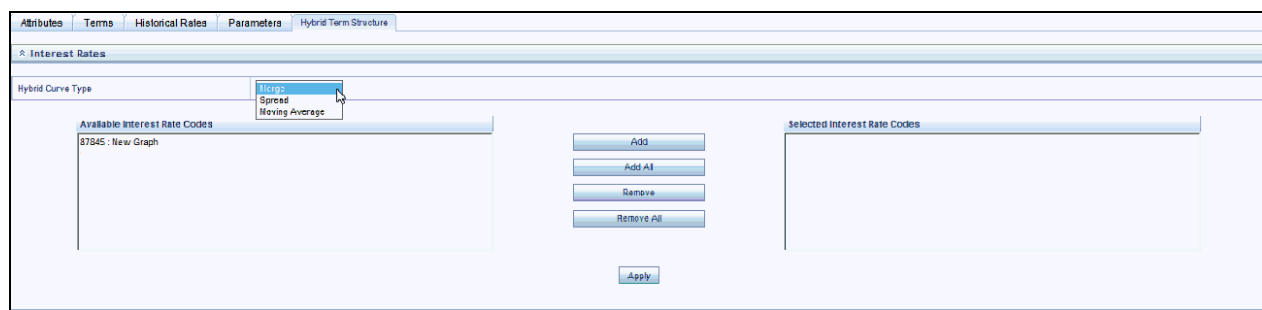
12. Similar to the validation used for direct input from the UI, if any of the calculated parameters are outside of the valid range, the engine will write an error message.

You can delete parameters by selecting one or more rows and then clicking Delete. For details on term structure models and stochastic processing, refer to *Oracle Financial Services Funds Transfer Pricing User Guide*, *Oracle Financial Services Asset Liability Management (OFSALM) User Guide*, and *Oracle Financial Services Analytical Applications Cash Flow Engine Reference Guide*.

### Hybrid Term Structure Tab

Hybrid term structures will allow you to specify three types of hybrid yield curves:

- Spread
- Moving Average
- Merge



**Note:** The Parameters and Hybrid Term Structure tabs will be disabled, if Volatility Curve is checked.

Hybrid yield curves are built up from either one or more standard yield curves. When you add, modify, or delete any historical rate data from a standard yield curve, the data associated with any related hybrid yield curve should also be updated (see Generate Historical Rates below). Once defined, Hybrid Yield Curves are used like any other interest rate curve in the system. You can reference these curves within OFSAA application business rules that allow selection of an Interest Rate Code.

#### Spread

A Spread hybrid yield curve is defined as the difference between two standard yield curves. The “spread” type of hybrid yield curve may be useful in establishing liquidity risk or basis risk yield curves.

#### Moving Average

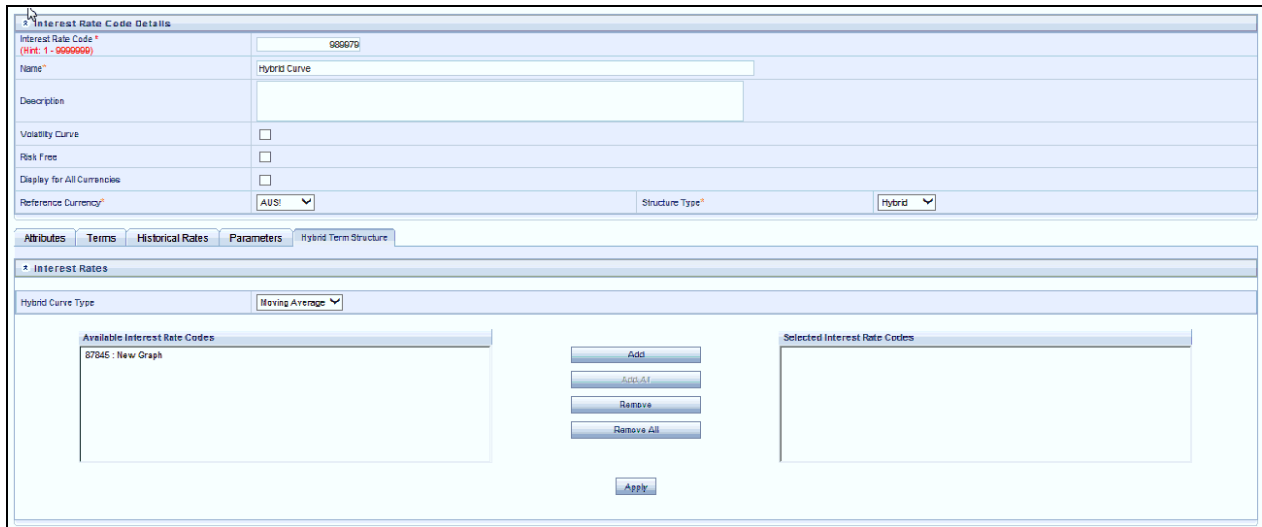
Moving average hybrid yield curves represent moving average data of a single underlying standard yield curve. These curves are typically used in Funds Transfer Pricing.

#### Merge

Merge hybrid yield curves represent a blending of two or more underlying yield curves. In constructing a “merge” type of hybrid yield curve, you specify the percentage weighting applicable to each of the underlying standard hybrid yield curves.

#### Defining a Hybrid Curve

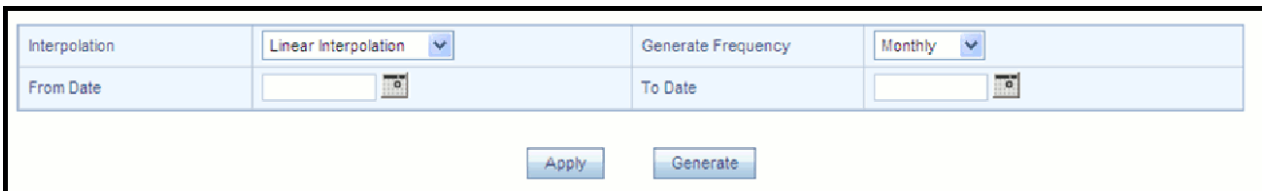
To define a hybrid curve select the Source Type: Hybrid, in the Hybrid Term Structure tab, select the Hybrid Curve Type (Spread, Moving Avg., or Merge), and select the Interest Rate Code(s) for the hybrid type. Click “Apply.”



**Generate Historical Rates**

Once a hybrid curve is defined, you can generate historical rates as far back as the rate source curves allow. The Generate Frequency determines the frequency of the historical rates populated with the Generate function. If you choose the generate frequency of monthly, it will generate month end values only. If you choose daily, it will generate the maximum number of historical values.

To generate the rates, select the interpolation method (Linear, Cubic or Quartic), select the Generate Frequency (Daily, Weekly, Bi-Weekly, or Monthly) and enter the specific date range (From Date / To Date), Click the “Generate” button. The rates will be populated and you will be directed to the Historical Rates tab to view the results.



If Hybrid Curve Type selected as Moving Average, then you can select the Moving Average Term in Days, Months, or Years.

The screenshot displays the 'Interest Rates' application window with the 'Hybrid Term Structure' tab selected. The interface includes a 'Hybrid Curve Type' dropdown set to 'Moving Average'. Below this are two panes: 'Available Interest Rate Codes' (empty) and 'Selected Interest Rate Codes' (containing 'STB45 : New Graph'). Between these panes are buttons for 'Add', 'Add All', 'Remove', and 'Remove All', along with an 'Apply' button. The 'Hybrid Term Structure Weights' section contains a table with columns for 'Selected Term Structures', 'From Date', 'To Date', and three 'Months' columns. The table has one row with 'STB45 : New Graph', '2/4/2015', '4/15/2015', and '12 Months', '24 Months', and '36 Months' respectively. At the bottom, there are fields for 'Moving Average Term' (checked), 'Interpolation' (set to 'Linear Interpolation'), 'Generate Frequency' (set to 'Monthly'), 'From Date', and 'To Date', with 'Apply' and 'Generate' buttons.

### Automate (Schedule) the Generate Rates Action in Hybrid IRCs

The generation of the Hybrid IRC rates which is currently available within the Rate Management user interface > Hybrid IRC, as a manual process, can additionally be scheduled to run through an ICC Batch process or through a Simplified Batch process.

When running Generate Rates through the user interface, there are 3 parameters provided by the user.

- Interest Rate Code: For example, the user is in edit mode for a particular Hybrid IRC when they execute the Generate Rates option, so the IRC is implied.
- FROM DATE: this is the start date of the generate rates process
- TO DATE: this is the date to which hybrid rates will be computed

When generating Hybrid Rates through a batch (ICC or Simplified), specify 'ALL' in the Parameter list to generate rates for all Hybrid IRCs. On the other hand, a list of IRCs can be specified by means comma separated values within quotes.

'IRC\_Code1, IRC\_Code2, IRC\_Code3' and so on

**Note:** If you want to use all IRCs, then use 'ALL' in the Parameter list.

Additionally, the "TO DATE" is determined based on the Effective Date specified for the Batch and the "FROM DATE" for each IRC is determined by referring to the last (maximum) effective date in the current historical rates table +1 day.

There is one exception to the calculated "FROM DATE". In cases where a new Hybrid IRC is generating rates for the first time, then the minimum Effective Date from the Parent IRC is used as the "FROM DATE".

**Note:** Hybrid Rates generated by the Generate Rates procedure are written to the FSI\_IRC\_RATE\_HIST table. Outputs from the procedure can be verified within the Interest Rates UI or by querying this table directly.

The following topics are covered in this section:



- Dependent Tables
- Prerequisites to running - Hybrid IRC batches
- Batch Creation and Running Batches
- Viewing Log Messages

### Dependent Tables

FSI\_IRCS: Holds header information of all Interest Rate Codes.

FSI\_IRC\_HYBRID\_STRUCT\_WEIGHT: Holds the Hybrid IRCs' - interest rate term, interest rate term mult and rate weightage. It is the child Table of FSI\_IRCS.

FSI\_IRC\_RATE\_HIST: This Output table holds calculated rates for the Hybrid IRCs, as well as all historical rates for non-hybrid IRCs input through the Interest Rates UI or loaded directly using the interest rates data loader program.

FSI\_MESSAGE\_LOG: Holds error log information.

DATE\_TASK\_MASTER: Located in the Config Schema, this table should have a seeded date for IRC\_HYBRID\_SCHEDULER DT.

DATE\_TASK\_STEP\_PRECEDENCE : Similar to DATE\_TASK\_MASTER, this table should also have an entry for IRC\_HYBRID\_SCHEDULER DT

### Prerequisites to Running Hybrid IRC Batches

- The Function FN\_FSI\_IRC\_HYBRID\_SCHEDULER should be in “compiled” status in the ATOMIC schema. Sometimes, the functions created in Database can be in invalid state due to some issues (by the installer), so you can use compile option in DB to get them in valid state.
- Users can create the ICC batch using the "IRC\_HYBRID\_SCHEDULER" rule name, which is seeded with the installation.
- Hybrid IRC rates are calculated based on their underlying standard IRC rates, which are also stored in the FSI\_IRC\_RATE\_HIST table. Historical Rates should already exist for all parent and dependent IRCs for the relevant effective date range.

---

**Note:** Seeded data related to Batch information should be present in the DATE\_TASK\_MASTER and DATE\_TASK\_STEP\_PRECEDENCE tables in the Config schema.

---

### Batch Creation

As mentioned earlier, there are two methods for creating and running batch processes, ICC Batch and Simplified Batch. The following section describes how to setup and run Hybrid Rate Generation using both approaches.

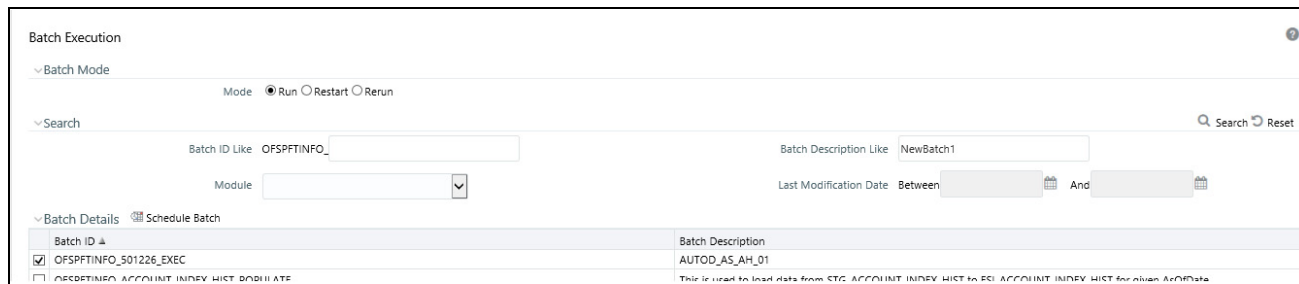
#### Running Hybrid Rate Generation using ICC Batch:

1. Navigate to: *Common Object Maintenance > Operations > Batch Maintenance > Create Batch*
  - Component = “Transform Data”
  - Rule Name = “IRC\_HYBRID\_SCHEDULER”
  - Parameter List = 'IRC\_Code'

---

**Note:** If you want to use all IRCs, then use 'ALL' in the Parameter list.

---



2. Save the batch
3. Navigate to: *Applications > Common Object Maintenance > Operations > Batch Execution*
4. Search for and Select the Batch created under Batch Maintenance.
5. Input the “Information Date” for the batch.
6. Select “Execute Batch”

To schedule a future Hybrid Rate Generation Batch Process, or to schedule Hybrid Rate Generation Batch Process on a recurring basis, do the following:

1. Navigate to *Applications > Common Object Maintenance > Operations > Batch Scheduler*.
2. Search for and select the Batch to schedule.
3. Select New Schedule or Existing Schedule:

**Note:** An Existing Schedule can be selected only if there are existing scheduled batches to view.

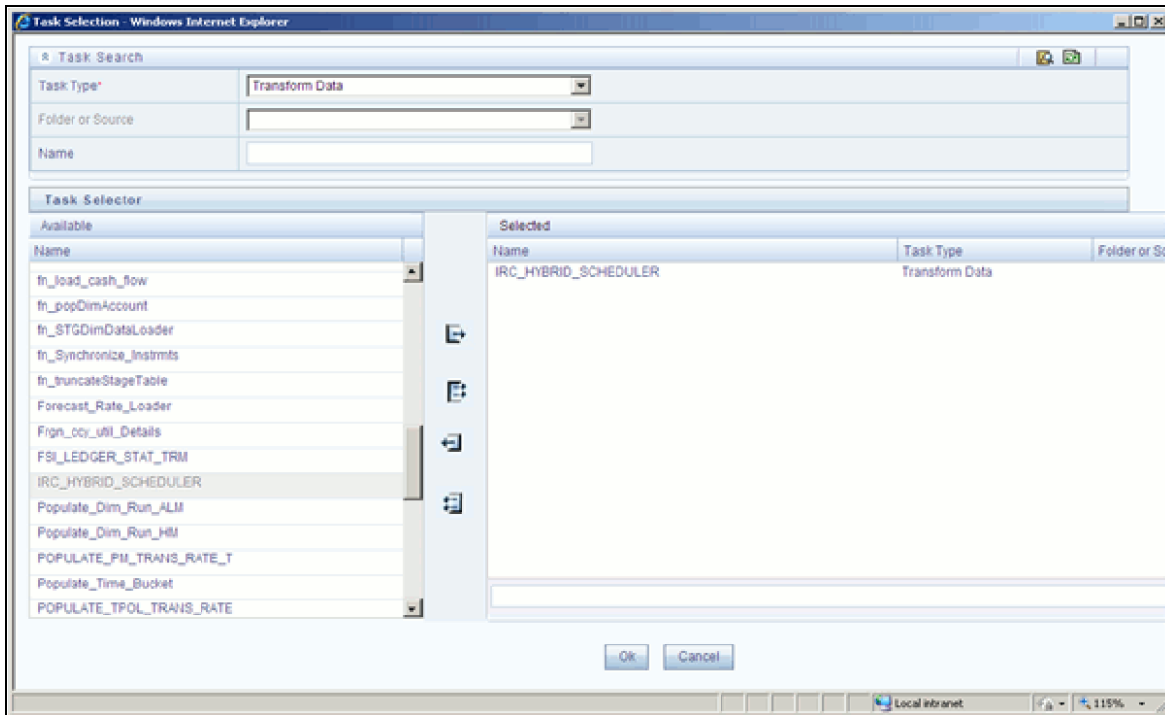
4. If New Schedule is selected, the New Schedule grid appears. Enter the Scheduled task name.  
 \_ To run the Hybrid Rate Generation process once, select once. Go to [Step 6](#).  
 \_ To schedule the Hybrid Rate Generation on a recurring basis, select Daily, Weekly, Monthly or Adhoc.
5. In the Schedule Task grid, in the Date field, enter the start date and end date.
6. In the Run Time field, input the time for the next validation to be run.
7. Click **Save** to set the schedule as specified or **Cancel** to drop your changes.

Any error messages or warnings generated during the Hybrid Rate Generation process are displayed in the View Log.

### Running Hybrid Rate Generation using Simplified Batch

To run Hybrid Rate Generation using Simplified Batch, use the following steps:

1. Under Financial Service Applications, click *Applications > Common Object Maintenance > Operations > Simplified Batch > Add*.
2. From the Task Details block, click Select Task. In the Task Selection window, choose Task Type as Transform Data. Then, click Search. In the Task Selector, Scroll down the left shuttle box (Available tasks) and select IRC\_HYBRID\_SCHEDULER (move to right shuttle box), then click **Ok**.



3. Enter the parameters and click **Save**.
4. In the Simplified Batch summary page, search for and select your Hybrid Rate Generation batch, then click the Run icon.
5. Click YES to confirm you want to continue, enter a date (this will be your “TO DATE”) and click **Ok** to continue.
6. You will receive a message that the batch has been successfully launched.
7. Click **Ok**. When the batch is complete, optionally navigate to Operations > View Log to view the processing log.
8. Any error messages or warnings are accessible from the View Log screen.

---

**Note:** Simplified Batch does not yet provide access to logs for Transform Data tasks.

---

### Viewing Log Messages

Any error messages or warnings generated during the Rate Generation Batch are displayed in the Log Information window. To access this window:

1. Click *Operations > View Log*.
2. Enter search criteria: Data Transformation and related Batch ID
3. Select the Task ID hyperlink to view the log information

Users can additionally query the FSI\_MESSAGE\_LOG table directly to view the error log details.

### Data Loader

The Data Loader Icon executes a function to import historical interest rates and parameters for all defined IRCs. For more information on setting up the automated process, see *Oracle Financial Services Analytical Applications Data Model Utilities User Guide*.

**Note:** `FusionApps.properties` file should present under `Web Server <deployed path>/WEB-INF/props` and `$_FIC_WEB_HOME/webroot/WEB-INF/props` directories.

Update the entry details of source name. It should be "infodom\_SOURCE\_NAME=Actual\_Source\_Name". For example, if Infodom Name is "DEV6INFO" and source Name is "DEVETLSRC", then entry will be "DEV6INFO\_SOURCE\_NAME=DEVETLSRC". Launch the Historical Interest Rates Data Loader from Interest rates screen after making this change.

To execute a data load, click the Data Loader icon. A warning message will appear *Update all available interest rates and parameters?*

**Note:** Upon clicking the data loader icon, the loader will execute all interest rates and parameters.

Code	Name	Rate Format	Compound Basis	Accrual Basis	Currency	Curve Type	Creation Date	Created By	Last Modification Date	Last Modified By
1	IRConc	Zero Coupon Yield	Annual	Actual/Actual	US Dollar	Interest Rate Curve	8/19/2014 10:31:34	JILNA	8/28/2014 11:38:14	JILNA
2	irc2	Zero Coupon Yield	Annual	Actual/Actual	Dollar	Interest Rate Curve	8/28/2014 06:51:37	JILNA	8/28/2014 06:51:50	JILNA
55	fgfgg	Zero Coupon Yield	Annual	Actual/Actual	Dollar	Interest Rate Curve	8/28/2014 06:16:44	ALMHM80USER		
78	sdt	Zero Coupon Yield	Annual	Actual/Actual	Dollar	Interest Rate Curve	8/28/2014 07:48:46	ALMHM80USER	8/28/2014 07:48:55	ALMHM80USER
131	PAR_TSY_1YR	Zero Coupon Yield	Annual	Actual/Actual	US Dollar	Interest Rate Curve	11/7/2012 17:35:57	JOBUSER28	11/7/2012 17:36:02	JOBUSER28

## Currency

Upon initially navigating to Common Object Maintenance > Rate Management > Currencies, a summary screen displays a comprehensive list of more than 170 seeded ISO currency codes.

Code	Currency Name	Reference Interest Rate Code	Reporting Currency	Status
			No	Inactive
USD	US Dollar		Yes	Active

## Search Container

A Search container is provided in which you may search for currencies by Name, by Currency (by ISO currency code), by status or by reporting currency.

### Search Control

Enter your desired search criteria and click on the Search control.

### Reset Control

Clicking on the Reset control removes search constraint you may have specified and refreshes the screen.

### **Currencies Container**

The Currencies container presents a grid containing all of the currencies that meet your search criteria. The Currencies summary grid offers several controls that allow you to perform different functions when a currency is selected.

To select a currency, click on a check box in the first column of the grid. More than one currency can be selected at a time but this will cause some of the controls to become disabled. Clicking on a checkbox a second time de-selects the currency.

You may select or deselect all of the currencies in the summary grid by clicking on the check box in the upper left hand corner of the summary grid directly to the left of the Code column header.

### Add

Clicking on the Add control begins the process of adding a new currency. The Add control is disabled if any rows in the grid have been selected.

### Edit

Selecting a single row out of the grid enables the Edit control. Clicking on the Edit control allows you to modify an existing currency. The Edit control is only enabled when a single currency has been selected.

---

**Note:** You can control the number of rows to display on screen by selecting the "Pagination Options" icon from the action bar.

---

### Delete

Selecting one or more rows out of the grid enables the Delete control. Clicking on the Delete control deletes the currency or currencies you have selected.

### Check Dependencies

Select a currency and then click the Check Dependencies control to generate a report on all rules that utilize your selected currency.

The Check Dependencies control is only enabled when a single currency has been selected.

### **Currencies Summary Grid**

The following columns categorize each currency shown in the summary grid:

- Code
- Currency Name
- Reference Interest Rate Code
- Reporting Currency
- Status

**Code**

For seeded currencies, these are ISO Currency Codes. For user-defined currencies, these may be any pure character string (no numbers) up to a length of 3 characters.

**Currency Name**

For seeded currencies, these are ISO Currency Codes. For user-defined currencies, these may be any string up to a length of 40 characters.

**Reference Interest Rate Code**

Reference Interest Rate Code is the Interest Rate Code with which a currency is associated for forecasting purposes. You may define multiple yield curves each of which has the same Reference Currency, but a currency may only have one Reference Interest Rate Code.

**Reporting Currency**

A reporting currency is an active currency to which balances in other currencies may be consolidated in order to facilitate reporting. Balances in reporting currencies may be, in turn, consolidated to the functional currency. For example, an American multinational bank might consolidate its holdings in Asian currencies to the Japanese yen (reporting currency) and its balances in European currencies to the Euro (reporting currency) after which it might consolidate these reporting currencies to the U.S. dollar (functional currency).

**Status**

The status of any currency may be either Active or Inactive. You must “activate” a currency before you can:

- Define that currency as a Reference Currency for a yield curve (see Reference Currency above under the section entitled Interest Rates Detail Screen)
- Enter exchange rate data for a currency (see the section below entitled Currency Rates)
- Define Forecast Rates for that currency within OFSAA Asset Liability Management (see Forecast Rates in the *Oracle Financial Services Asset Liability Management (OFSALM) User Guide*)

**Editing Currencies**

To edit a currency, select a currency and then click on the Edit control. You may not modify any currency Code value. You may, however, modify Currency Name, the Reference Interest Rate Code, the Reporting Currency value (limited to Yes or No), or the currency's Status (limited to Active or Inactive).

You may not inactivate any currency that is:

- Defined as the Reference Currency for any yield curve
- Associated with any exchange rate data
- Utilized within any Forecast Rates rule within OFSAA Asset/Liability Management

Once you have completed your edits, click **Save**.

**Adding Currencies**

The first row in the Currencies container is an empty row that is ready for you to edit as a new currency. You may select this empty row, click on the Edit control, and generate a new currency. You may also generate a new currency by clicking on the Add control. This will generate another empty row at the top of the Currencies container that is ready for you to edit.

## Deleting Currencies

Click on one or more currencies and then select the Delete control. You may not delete any currencies that are utilized elsewhere in the system (see exclusions above under Editing Currencies).

## Currency Rates

Upon navigating to Common Object Maintenance > Rate Management > Currency Rates, the Currency Rates screen is displayed. Within this screen, you may manage historical exchange rates between currencies.

Currency Rates

▼ Currency Selection

From Currency\* US Dollar ▼ To Currency\* ▼

Rate Type\* Floating Rate ▼ Effective Date Range Last Month ▼

From Date 04/01/2018 📅 To Date 05/01/2018 📅

For ALM, TP and PFT, **From Currency** is to be defaulted to the Initial Currency selection from the Assumption Management Defaults in the active Application Preferences UI. You may select another From Currency. The From Currency drop-down list displays all Active currencies.

## Preparing to Work with Exchange Rate Data

In order to begin the process of viewing, entering, modifying, or deleting exchange rate data, you must supply a “To Currency” value. As with the From Currency, the To Currency drop-down list box displays only Active currencies.

Currency Rates As of Date: 25/06/2014

↕ Currency Selection

From Currency\* US Dollar ▼ To Currency\* Albanian Lek ▼

Rate Type\* Floating Rate ▼ Effective Date Range Last Month ▼

From Date 16/06/2015 📅 To Date 16/07/2015 📅

↕ Floating Currency Rates Exchange Rate per 1 unit of Albanian Lek

Effective Date	Albanian Lek	Status	Data Source Code
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/> 13/07/2015	4.000000000	Valid	UI

Save Cancel

After having selected a To Currency value, select a value for Rate Type: Floating Rate or Fixed Rate (the default selection is Floating Rate). Once you have selected a To Currency value, a second container is displayed: Floating Currency Rates (if you have chosen a Rate Type of Floating Rate) or Fixed Currency Rates (if you have chosen a Rate Type of Fixed Rate).

### Floating Rates

Floating exchange rates, such as those between the US Dollar (USD), the Pound Sterling (GBP), the Japanese Yen (JPY), and the Euro (EUR), are market driven and may change from day-to-day, hour-to-hour, or minute-to-minute.

### Fixed Rates

Some countries, especially smaller countries or countries that have experienced significant inflation in the recent past, may wish to “peg” their currency to a larger, more stable currency such as the US Dollar, Japanese Yen, or Euro.

### Adding Exchange Rate Data

Once you have specified a value for To Currency, the Floating Currency Rates container appears. If you wish to define a Fixed Rate relationship, select the “Fixed Rate” Rate Type and the Floating Currency Rates container will be replaced by the Fixed Currency Rates container.

Both Currency Rates containers initially display a single blank row followed by the most recent month's exchange rate data (if any such exchange rate data already exists). To enter a single new exchange rate data point, enter your data into the blank row and click **Save**.

Currency Selection		Effective Date Range	
From Currency*	US Dollar	To Currency *	Albanian Lek
Rate Type*	Floating Rate	Effective Date Range	Last Month
From Date	16/06/2015	To Date	16/07/2015

Floating Currency Rates			
Exchange Rate per 1 unit of Albanian Lek			
Effective Date	Albanian Lek	Status	Data Source Code
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>	13/07/2015	4.000000000	Valid UI

### Effective Date

Directly enter a date or select the calendar control to choose an effective date for your new exchange rate data point.

Rate Management stores historical exchange rate data. You may not enter exchange rate data for dates greater than the current date. For more information regarding rate forecasts and the relationship between historical exchange rates for forecasted exchange rates, see *Forecast Rate Scenarios* chapter. If you have gaps in your historical exchange rate data, any OFS Analytical Application that needs to perform a rate translation function will fall back to the most recent date for which exchange rate data exists. For example, if an OFS Analytical Application needs to translate a rate from USD to EUR for February 22, 2010 and the latest available USD to EUR rate data in the Rate Management database is February 11, 2010, the application will utilize the exchange rate for February 11, 2010.

### Status

Status is a read-only display which is updated after the Currency Rates Validation has been run.

### Data Source Code

The Data Source Code is displayed read only and indicates whether the rates were input through the UI or via the data loader.



## Currency Exchange Rate

For both Floating Rates and Fixed Rates, units of the From Currency are converted to one unit of the To Currency. For example:

From Currency	To Currency	Approximate Rate
USD – US Dollar	GBP—Pound Sterling	1.50
USD – US Dollar	EUR – Euro	1.36
USD – US Dollar	JPY – Japanese Yen	0.01105

## Adding Multiple Exchange Rates

Click **Add** to add additional blank rows into which you may enter additional Effective Dates and Exchange Rates. When you have finished adding multiple new exchange rates, click **Save**.

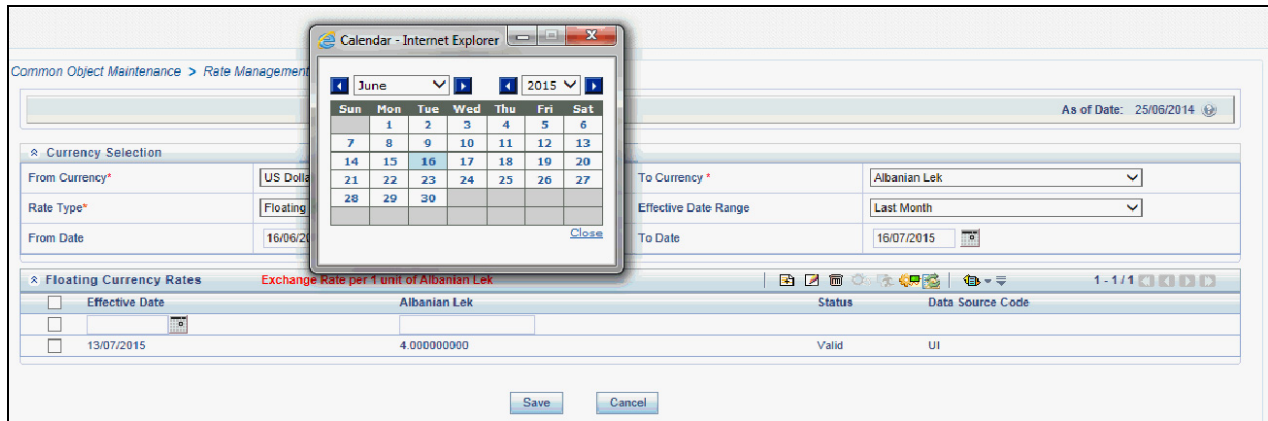
## Editing Exchange Rate Data

Clicking on the check box on the left hand side of any row of data enables the **Edit**. After clicking **Edit**, the row becomes active and you may edit Effective Date and/or the Exchange Rate after which you may Save your changes.

## Viewing Exchange Rate Data

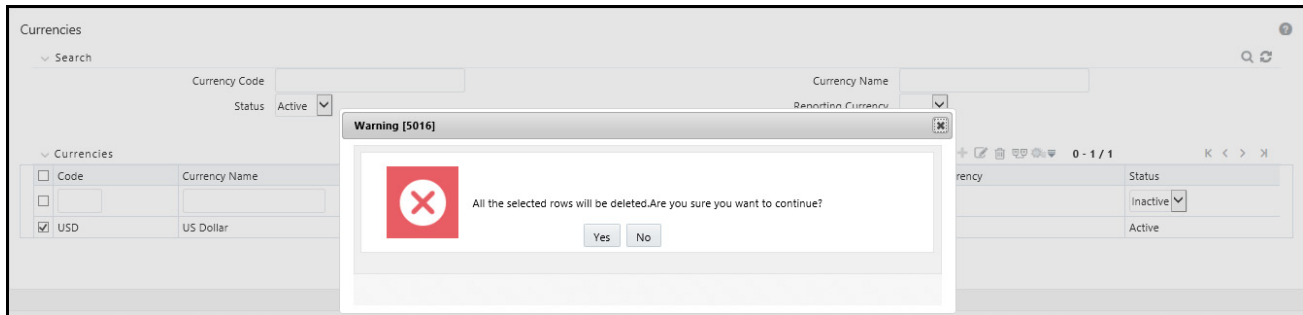
By default, both the Floating Currency Rates container and the Fixed Currency Rates container display the most recent month of historical exchange rate data. You can control the amount of data displayed by selecting a different value from the Effective Date Range drop-down list box in the Currency Selection container.

You may also choose to view a specific range of effective dates by modifying the From Date, the To Date, or both dates within the Currency Rates container.



### Deleting Exchange Rate Data

Clicking on one or more check boxes on the left hand side of any row of data enables the Delete button. After clicking **Delete**, you will be asked to confirm that you wish to delete the rows you have selected.



### Data Loader

The Data Loader Icon executes a function to import historical Currency rates for all defined Currencies. For more information on setting up the automated process, see *Oracle Financial Services Analytical Applications Data Model Utilities User Guide*.

To execute a data load, click on the Data Loader icon. A warning message will appear *Upload all available Currency Rates?*

**Note:** Upon clicking on the data loader icon, the loader will execute all currency rates.

Common Object Maintenance > Rate Management > Currency Rates

Currency Rates As of Date: 25/06/2014

---

**^ Currency Selection**

From Currency*	US Dollar	To Currency *	Albanian Lek
Rate Type*	Floating Rate	Effective Date Range	Last 6 Months
From Date	16/01/2015	To Date	16/07/2015

---

**^ Floating Currency Rates** Exchange Rate per 1 unit of Albanian Lek

<input type="checkbox"/> Effective Date	Albanian Lek	Status	Data Source Code
<input type="checkbox"/>			
<input type="checkbox"/> 13/07/2015	4.000000000	Valid	UI
<input type="checkbox"/> 16/07/2015	20.000000000	Valid	UI

Save Cancel

## Excel Import/Export

Excel import/export functionality is used for adding/editing historical exchange rates.

Common Object Maintenance > Rate Management > Currency Rates

Currency Rates As of Date: 25/06/2014

---

**^ Currency Selection**

From Currency*	US Dollar	To Currency *	Albanian Lek
Rate Type*	Floating Rate	Effective Date Range	Last 6 Months
From Date	16/01/2015	To Date	16/07/2015

---

**^ Floating Currency Rates** Exchange Rate per 1 unit of Albanian Lek

<input type="checkbox"/> Effective Date	Albanian Lek	Status	Data
<input type="checkbox"/>			
<input type="checkbox"/> 13/07/2015	4.000000000	Valid	UI
<input type="checkbox"/> 16/07/2015	20.000000000	Valid	UI

Save Cancel

## Currency Exchange Rate Validation

### Currency Rate Validation – Overview

#### Features:

Exchange Rate Validation contains the following features:

- Movement of historical exchange rates to the Currency Direct Access table.
- Calculation of inverse exchange rates for reporting currencies.
- Calculation of triangulated exchange rates where possible.

**Note:** This feature is equivalent to the Rate Validation capability in OFSA 4.5 Rate Manager.

## Exchange Rate Validation – Feature Description

The goal of exchange rate validation is to ensure that exchange rates from all active currencies to all reporting currencies are available for processing in OFSAA applications. Some of these rates may come from the validated direct input, others are calculated based on relationships with other rates. To support triangulation, all fixed exchange rates are available for all currencies that make up a conversion that needs to be triangulated. In addition, a direct exchange rate between each Child currency and each reporting currency is calculated and supplied to support quick access to exchange rates. If a Child currency is a reporting currency, then exchange rates are calculated for all currencies having an exchange relationship with the Parent currency.

## Currency Rate Validation – Procedure

1. Load currency rates through the Rate Management UI or by using the Historical Rate Data Loader. Notice the initial status is “Not Yet Validated”.
2. Execute the Currency Rate Validation engine.

Navigate to: *Operations > Batch Maintenance > Create Batch*

- Component = “Transform Data”

- Rule Name = “Rate\_Validation”

- Parameter List (Required) = FROM\_DATE and TO\_DATE, for example, '20110101', '20110131'

**Note:** After completion of the validation process, notice the status in the Currency Rates UI changes to **Valid** or in case there is an issue, **Invalid**.

3. View both input and calculated results in the FSI\_EXCHNG\_RATE\_DIRECT\_ACCESS table.
4. All of the OFSAA processing engines read from this table for sourcing historical exchange rates.

**Note:** Executing Rate Validation is a **required** step when multi-currency setup is enabled. All of the OFSAA processing engines require exchange rates to be validated through this process in order to convert results from base currency to a selected reporting currency. If exchange rate validation is not run or required exchange rates are not available, the engines will use default exchange rates = 1.

## Validating Exchange Rate Relationships

Users should run the exchange rate validation process after adding or modifying exchange rate data. You can run the process immediately or schedule one or more to be run in the future.

Each exchange rate is placed in one of the following statuses:

Status	Description
Not Yet Validated	The exchange rate has been input or loaded but not yet validated.
Valid	The exchange rate has been validated.
Invalid	The exchange rate has violated one or more acceptance rules.

Only exchange rates in Valid status are available for processing and they are not subject to future validation unless you edit them.

The rate validation status is displayed in the Rate Management > Currency Rates UI.

### Exchange Rate Validation Criteria

In the rate validation process all exchange rate relationships in the database are examined for compliance with the following criteria. Error messages and warnings are displayed if one or more criteria are not met. (See *Viewing the Messages* section.)

1. If a currency is defined as a Child in a fixed exchange relationship then it must not be in any floating (standard) exchange rate relationship at the same time. Consequently all floating exchange rates to or from the Child currency must be defined through the Parent currency. If this criterion is not met then the following message is displayed : *Invalid fixed relationship—Child currency exists in a standard exchange rate within the same time period.*
2. A Child currency within a fixed relationship must not be a Child currency in any other fixed relationship during the same time period. If this criterion is not met then the following message is displayed : *Invalid fixed relationship—Child currency already exists in a fixed relationship for same time period.*
3. A circular relationship must not exist. In other words, a Child currency cannot link back to its Parent in any other fixed rate relationship within the same time period. If it does, then the following message is displayed : *Invalid fixed relationship creates circular relationship with other fixed exchange rates.*
4. Regarding new floating (standard) exchange rates, From and To currencies must not exist as Child currencies within any fixed exchange rate relationships. If this criterion is not met then the following message is displayed : *From/To/Both currency(ies) in new exchange rate already exist in a fixed relationship for the same time period.*
5. If any exchange rate is equal to 0, then a warning message is displayed. Generally speaking, 0 is a valid value.

You can use it, for example, to designate an exchange rate with a currency of a country that no longer exists. If two exchange rate relationships fail to meet these criteria then both of them will be labeled Invalid. (Exception: if one of the relationships is already in Valid status, then the other one will be labeled Invalid.) For example, if a currency is defined as a Child in a fixed rate relationship and is also defined as being in a floating relationship at the same time, then both fixed and floating rates for that currency will be labeled Invalid.

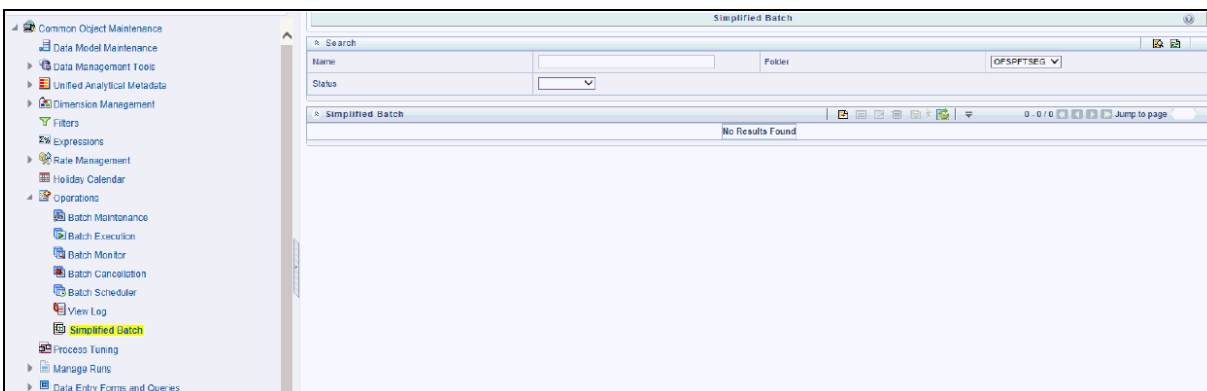
If there are both direct and inverse floating exchange rates defined for any two currencies (in other words, one currency is both a To and a From currency in relation to the other), then both relationships will be marked Valid.

### Running an Exchange Rate Validation

You can run a validation immediately or schedule one or more for later. The validation status is displayed in the Rate Management > Currency Rates UI.

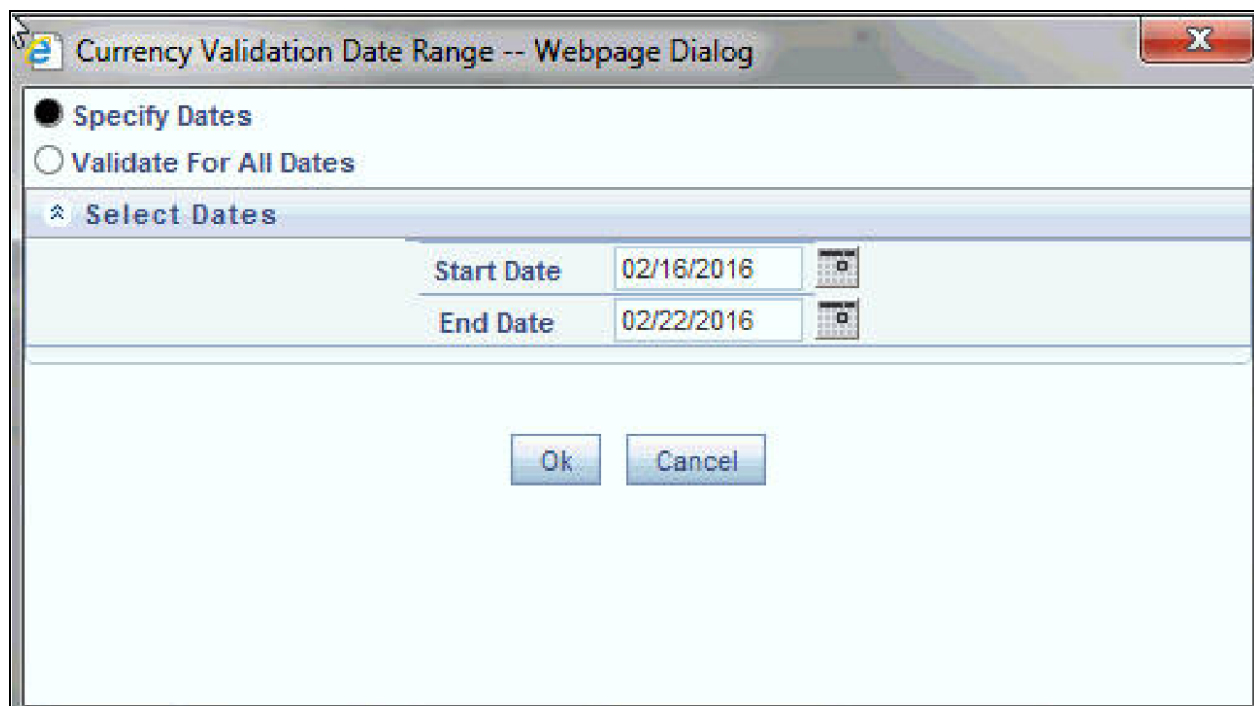
#### Running a Validation Immediately

You can execute exchange rate validation using the "Currency Rates validation" option from the toolbar.



To execute exchange rate validation from the Currency Rates user interface, the following two options are available:

- **Specify Dates:** When you select this option, a **Select Dates** section will be displayed that allows you to input/verify the FROM\_DATE and TO\_DATE parameters and these dates will be passed to the batch for execution.



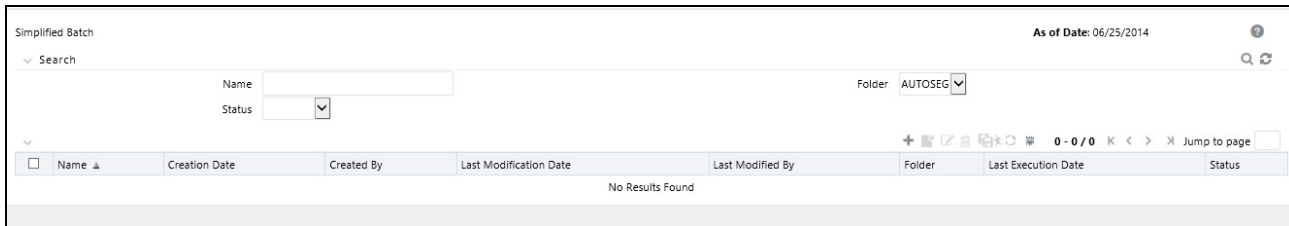
- FROM\_DATE: This is defaulted to the date of last rate validation.
- TO\_DATE: This is defaulted to the current date.
- **Validate For All Dates:** If you select this option, then it will validate all rates irrespective of dates.

**Note:** This option will replace all of the validated exchange rate history and can be a time consuming process depending on the amount history available to be processed.

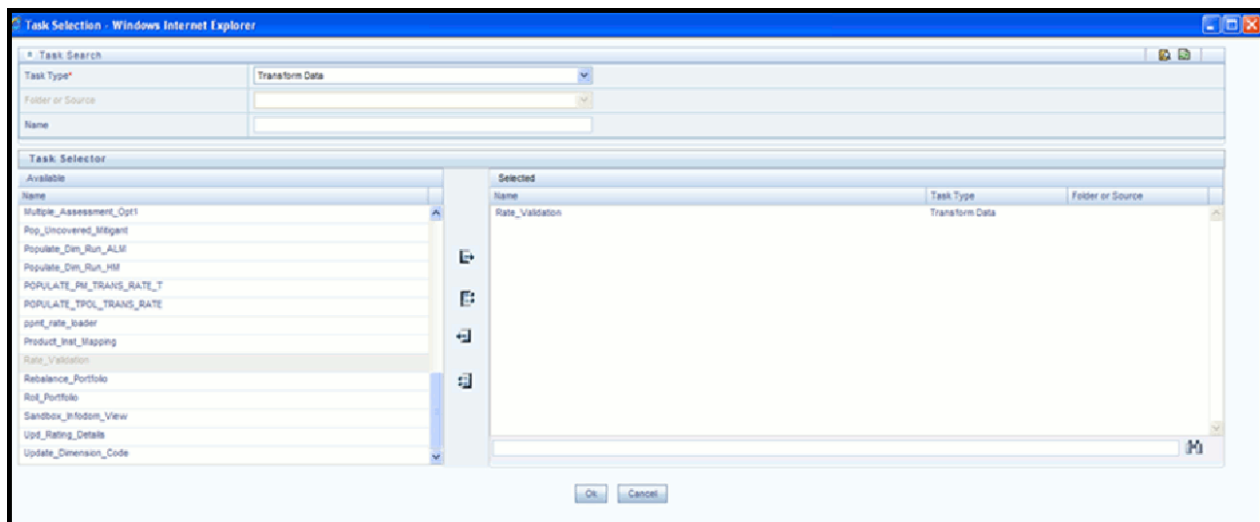
You can execute rate validation using a Simplified Batch process or from the ICC Batch Screen, within the OFSAAI framework. You can also launch it from the Currency Rates UI (Currency Rates Validation toolbar button).

To run the validation using Simplified Batch, perform the following steps:

1. Click Common Object Maintenance-> Simplified Batch -> Add.



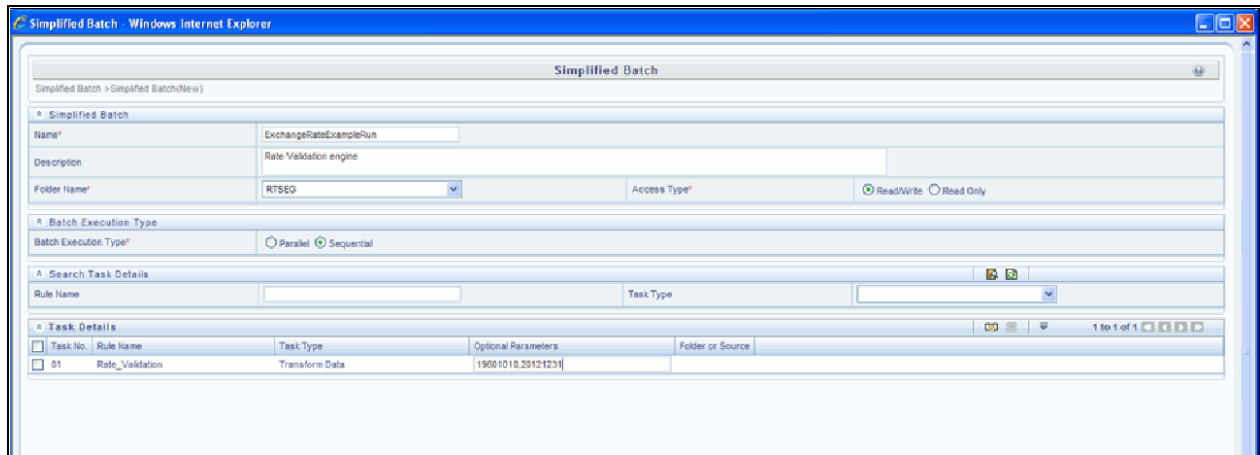
2. From the Task Details block, click Select Task. In the Task Selection window, choose Task Type as Transform Data. Then, click Search. In the Task Selector, Scroll down the left shuttle box (Available tasks) and select Rate\_Validation (move to right shuttle box), then click **Ok**.



3. Back in the Simplified Batch definition window, enter the optional parameters to specify the From and To dates, using the format 'YYYYMMDD','YYYYMMDD'.

**Note:** FROM\_DATE and TO\_DATE should not be the same value. Set FROM\_DATE equal to the last rate validation date and TO\_DATE to the current date. This will ensure that the Effective To Date for the prior record is set correctly.

4. Then click **Save**.



5. In the Simplified Batch summary page, search for and select your batch, then click Run.
6. Click Yes to confirm you want to continue, enter a date and click **Ok** to continue. You will receive a message that it has been successfully launched.
7. Click **Ok**. When the batch is complete, optionally navigate to Operations -> View Log to view the processing log.
8. Any error messages or warnings are displayed in View Log.

**Note:** Simplified Batch does not yet provide access to logs for Transform Data tasks.

To run the validation using the ICC Batch Framework, perform the following steps:

1. Navigate to Operations -> Batch Maintenance -> Add, and create a new batch.
2. Search for the above batch, and select it.
3. In the Task Details block (toolbar), click Add. In the Task Selection window, choose Task Type as Transform Data. Then, click Search. In the Task Selector, Scroll down the left shuttle box (Available tasks) and select Rate\_Validation (move to right shuttle box), then click **Ok**.
4. Back in the Task definition window, select the Rule Name Rate\_Validation and enter the optional parameters to specify the From and To dates, using the format 'YYYYMMDD','YYYYMMDD'.

**Note:** FROM\_DATE and TO\_DATE should not be the same value. Set FROM\_DATE equal to the last rate validation date and TO\_DATE to the current date. This will ensure that the Effective To Date for the prior record is set correctly.

5. Then click **Save**.
6. Navigate to Operations -> Batch Execution. Search for and select your batch, enter Information Date, then click **Execute Batch**.
7. Click **Ok** to confirm you want to continue. A message is displayed stating Batch triggered successfully, also displaying the Batch Run ID.
8. Click **Ok**.
9. To view rates, query the database table FSI\_EXCHNG\_RATE\_DIRECT\_ACCESS after the run is complete.



10. FSI\_EXCHNG\_RATE\_DIRECT\_ACCESS: This output table holds the valid currency codes and their exchange rate with respect to the reporting currency for a period of time say “from date” value to optional “to date” value. When Exchange Rate Validation runs, if data already exists for the selected date range, the Rate Validation package will overwrite the existing values. If no dates are given, the Rate Validation package truncates this table and re-loads data for the entire historical date range. The following describes the structure of the FSI\_EXCHNG\_RATE\_DIRECT\_ACCESS table:

Column Name	Nullable	Data Type
FROM_CURRENCY_CD	Not Null	varchar2(15)
TO_CURRENCY_CD	Not Null	varchar2(15)
EFFECTIVE_FROM_DATE	Not Null	Date
EFFECTIVE_TO_DATE	Not Null	Date
EXCHANGE_RATE	Not Null	Number(15,9)
EXCHANGE_RATE_CONVERT_TYPE_CD	Not Null	Number(5)
EXCHANGE_RATE_CONV_FORMULA	Not Null	Number(15)

11. When the batch is complete, optionally navigate to Operations -> View Log to view the processing log. Any error messages or warnings are displayed in View Log.

### Scheduling One or More Validations

The screenshot shows the 'Batch Scheduler' interface. At the top, there are search filters for 'Batch ID Like', 'Batch Description Like', and 'Module'. Below these are 'Server Time' and 'Current Server Time' (01/05/2018 22:54:58). The main area is a table with columns 'Batch Name', 'Batch ID', and 'Batch Description'. The table lists various batch types, each with an unchecked checkbox in the 'Batch Name' column. The bottom of the interface shows 'Page 1 of 14 (1-15 of 208 items)' and 'Records Per Page 15'.

To schedule a future validation, or to schedule validations on a recurring basis, do the following:

1. Navigate to Operations -> Batch Scheduler. The Batch Scheduler window appears.
2. In the Batch Name block, select the Scheduled Batch ID. An unchecked box means that no validation is scheduled to run.
3. Select New Schedule or Existing Schedule:

Existing Schedule can be selected only if there are existing schedule batches to view.

If New Schedule is selected, the New Schedule grid appears. Enter the Scheduled task name.

\_ To run the validation once, select once. Go to *Step 5*.

\_ To schedule the validation on a recurring basis, select Daily, Weekly, Monthly or Adhoc.

4. In the Schedule Task grid, in the Date field, enter the start date and end date.

5. In the Run Time field, input the time for the next validation to be run.

6. Click **Save** to set the schedule as specified or **Cancel** to drop your changes.

Any error messages or warnings generated during the validation process are displayed in the View Log.

### Viewing the Messages

Any error messages or warnings generated during the exchange rate validation process are displayed in the Log Information window. To access this window: Click Operations -> View Log.

Click the Task ID. The Log Information window will display.

These exception messages could also be seen in FSI\_MESSAGE\_LOG table with the help of the batch\_id which was used during execution.

## Economic Indicators Summary Screen

When you first navigate to Common Object Maintenance > Rate Management > Economic Indicators, an empty screen will be displayed. After you have constructed one or more Economic Indicators, navigating to Common Object Maintenance > Rate Management > Economic Indicators will display a summary screen showing each of the Economic Indicators that you have previously built.

Name	Country	Creation Date	Created By	Last Modification Date	Last Modified By
Econom	Argentina	8/18/2014 07:05:33	JILNA		

### Search Container

A Search container is provided in which you may search for Economic Indicators by Name or by Country. Each Economic Indicators rule is specific to one country.

### Search Control

Enter your desired search criteria and click on the Search control.

### Reset Control

Clicking on the Reset control removes any Name or Country constraint you may have specified and refreshes the screen.

### **Economic Indicators Container**

The Economic Indicators container presents a grid containing all of the Economic Indicators that meet your search criteria. The Economic Indicators summary grid offers several controls that allow you to perform different functions when an Economic Indicators is selected.

To select an Economic Indicators, click on a check box in the first column of the grid. More than one Economic Indicators can be selected at a time but this will cause some of the controls to become disabled. Clicking on a checkbox a second time de-selects the row.

You may select or deselect all of the Economic Indicators in the summary grid by clicking on the check box in the upper left hand corner of the summary grid directly to the left of the Name column header.

#### **Add**

Clicking on the Add control begins the process of building a new Economic Indicator. The Add control is disabled if any rows in the grid have been selected.

#### **View**

Selecting a single row out of the grid enables the View control. Clicking on the View control allows you to view the contents of an Economic Indicator on a read-only basis. The View control is only enabled when a row has been selected.

#### **Edit**

Selecting a single row out of the grid enables the Edit control. Clicking on the Edit control allows you to modify a previously saved Economic Indicator. The Edit control is only enabled when a single row has been selected.

#### **Delete**

Selecting one or more rows out of the grid enables the Delete control. Clicking on the Delete control deletes the Economic Indicators you have selected.

#### **Copy**

Selecting a single row out of the grid enables the Copy control. Clicking the Copy control allows you to create a copy of an existing economic indicator. The Copy control is only enabled when a single economic indicator has been selected.

#### **Check Dependencies**

Select an interest rate curve and then click the Check Dependencies control to generate a report on all rules that utilize your selected economic indicator.

The Check Dependencies control is only enabled when a single economic indicator has been selected.

#### **Data Loader**

The Data Loader Icon executes a function to import historical economic indices for all defined Economic Indicators. For more information on setting up the automated process, see *Oracle Financial Services Analytical Applications Data Model Utilities User Guide*.

To execute a data load, click on the Data Loader icon.

A warning message will appear “Upload all available Economic Indicators?” Click Ok, and all historical indices will be loaded.

### ***Economic Indicators Summary Grid***

The following columns categorize each Economic Indicator in the summary grid:

- Name
- Country
- Creation Date
- Created By
- Last Modification Date
- Last Modified By

#### **Name**

Displays the Economic Indicator's short name. Performing a “mouse-over” on a row within the grid displays the Economic Indicator's detailed description.

#### **Country**

Displays the Country to which an Economic Indicator applies.

#### **Creation Date**

Displays the date and time at which an Economic Indicator was created.

#### **Created By**

Displays the name of the user who created an Economic Indicator.

#### **Last Modification Date**

Displays the date and time at which an Economic Indicator was last modified.

#### **Modified By**

Displays the name of the user who last modified an Economic Indicator.

### ***Economic Indicators Detail Screen***

When you Add, Edit, or View an Economic Indicator, the Economic Indicator Detail Screen is displayed. The Economic Indicator detail screen is comprised of an Economic Indicator Details container, an Economic Indicators – Historical Data Container, and an Audit Trail container.

The Audit Trail container is a standard footer container for every OFSAA rule type. It displays Created By, Creation Date, Last Modified By, and Modification Date on the Audit Trail tab. The User Comments tab may be used to add comments to any rule, subject to a maximum of 4000 characters.

#### ***Economic Indicator Details Container***

When creating a new Economic Indicator, you must supply a Name, a Frequency, a Value Type, and a Country. You may optionally provide a Description.

The screenshot shows a web-based form for 'Economic Indicator Details'. The form has a title bar 'Economic Indicators' and a breadcrumb trail 'Master Maintenance > Rate Management > Economic Indicators'. The form fields are: Name (text input with '10th Feb Economic'), Description (text area with 'hi'), Frequency (dropdown menu with 'Monthly'), Value Type (dropdown menu with 'Numeric'), and Country (dropdown menu with 'United States of America').

## Name

The name you give to your Economic Indicator is the means by which you will subsequently refer to your rule within other OFS Analytical Applications. You may not rename existing Economic Indicators.

## Frequency

The frequency of your Economic Indicator should match the frequency which with the indicator's data is made public. Unemployment statistics, for example, are generally released on a monthly frequency. Select a frequency from the Frequency drop-down list box. Available frequencies are Weekly, Monthly, Quarterly, Semi-Annually, and Annually.

## Value Type

Select a Value Type from the Value Type drop-down list box. Available Value Types are Numeric, Percentage, and Amount.

Value Type	Value Range
Numeric	0-999999
Percentage	-100 to +100
Amount	0-999999

## Country

Select a country to which your Economic Indicator applies from the Country drop-down list box. The value set of Countries is drawn from the seeded Country dimension. OFSAA is seeded with over 70 country values, and you may also add your own user-defined countries (see Chapter 6: OFSAA Dimension Members).

## ***Economic Indicators – Historical Data Container***

Once you have made your selections in the Economic Indicator Details container, you may save your new rule. To build out your historical data, you must enter data within the Economic Indicators – Historical Data container.

The screenshot shows the 'Economic Indicators - Historical Data' container. It features an 'Effective Date Range' dropdown menu set to 'Last Month'. Below this, there are 'From Date' and 'To Date' fields with dates '09.Feb.2013' and '11.Mar.2013' respectively. At the bottom, there is a table with three columns: 'Start Date', 'End Date', and 'Value'. The 'Value' column contains the number '0.000000'.

The Economic Indicators – Historical Data container initially displays a single blank row followed by the most recent period's data (if data has previously been stored in the database).

### Start Date & End Date

Select the calendar control immediately adjacent to the End Date to choose an ending date for your Economic Indicator data point. The application will automatically populate the Start Date based on your Economic Indicator's frequency. For example, if your Economic Indicator is an unemployment statistic that has a monthly frequency, select an end date that is the last day of the month that the unemployment rate describes. In this example, the application will automatically populate the Start Date with the first day of the month you have chosen.

### Value

Enter the value for your Economic Indicator (e.g., the unemployment rate).

### Adding Multiple Data Points

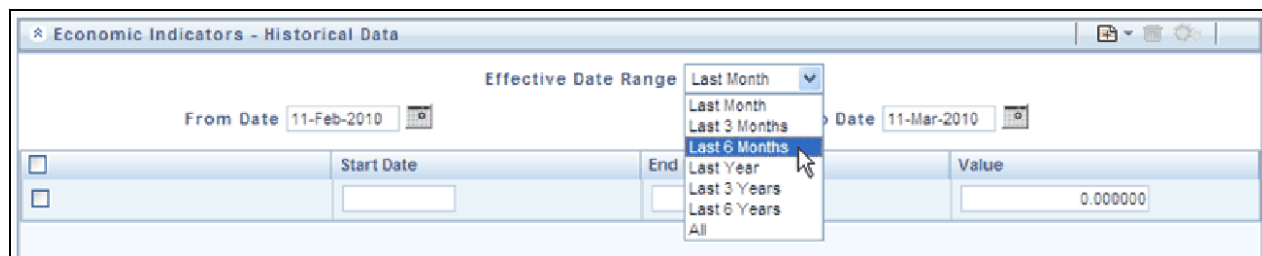
Click on the Add control to add additional blank rows into which you may enter additional Economic Indicator data. When you have finished adding data, click **Save**.

### Editing Economic Indicators – Historical Data

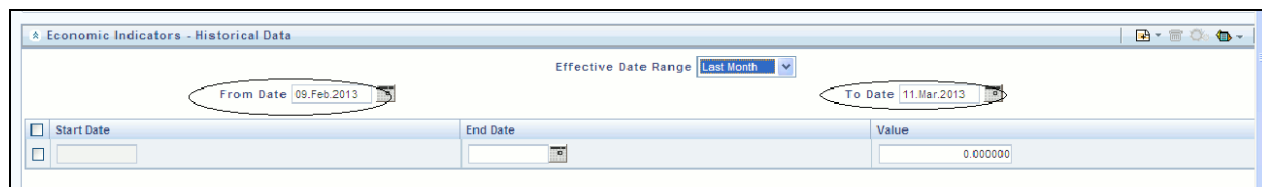
Clicking on a single check box on the left hand side of any row of data enables the Edit control. After clicking on the Edit control, the row you have selected becomes active. You may edit this row and subsequently save your changes.

### Viewing Economic Indicators – Historical Data

By default, the Economic Indicators – Historical Data container displays the most recent month of historical data. You can control the amount of data displayed by selecting a different value from the Effective Date Range drop-down list box.



You may also choose to view a specific date range by modifying the From Date, the To Date, or both dates within the Economic Indicators – Historical Data container.



### Deleting Economic Indicators – Historical Data

Clicking on one or more check boxes on the left hand side of any row of data enables the Delete control. After clicking on the Delete control, you will be asked to confirm that you wish to delete the rows you have selected.

**Data Loader**

The Data Loader Icon executes a function to import historical economic indicators for all defined economic indicators. For more information on setting up the automated process, see *Oracle Financial Services Analytical Applications Data Model Utilities User Guide*.

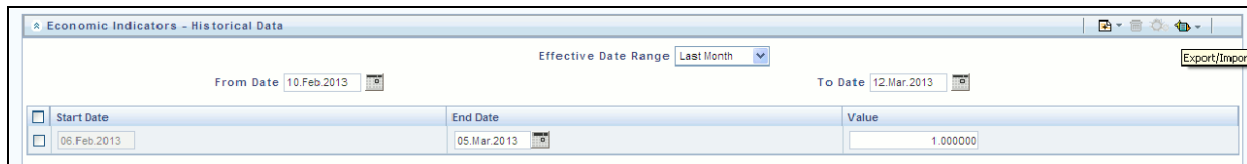
To execute a data load, click the Data Loader icon. A warning message will appear “Upload all available Economic Indicators?”

**Note:** Upon clicking the data loader icon, the loader will execute all economic indicators.



**Excel Import/Export**

Excel import/export functionality is used for adding/editing historical economic indicators.







# *Overview of Profitability Management*

## ***Introduction***

An effective financial services performance management system should include the following key requirements:

- Flexible, efficient access to account level information
- Multiple views of the organization
- Matched rate funds transfer pricing at the account level of detail
- Flexible definition of cost accounting processes with strong accounting controls
- Powerful modeling capabilities

Together with OFSAA Funds Transfer Pricing, OFSAA Profitability Management addresses these requirements by linking general ledger, account-level, and statistical data together in a unique manner.

## ***Multiple Dimensions of Profitability***

OFSAA Profitability Management allows you to construct customized solutions for enriching financial data to generate multidimensional Management Accounting profitability views. Without restricting your ability to select any dimension you choose, the following are representative examples of the types of multi-dimensional performance management views you can construct using OFSAA Profitability Management.

### ***Organizational View***

An Organizational View facilitates the measurement of profitability at any level in your organizational structure, such as Bank, Region, Company, Legal Entity, or Division level; or at lower levels such as Branch, Cost Center or Department. While the Organizational Unit Key Processing Dimension is normally designed to match Responsibility Centers (Cost Centers or Profit Centers) that are found both in your source systems and in your General Ledger extract, you may define additional “organizational” Key Processing Dimensions. For example, you might use the seeded Organizational Unit dimension in reconciling your instrument extracts to your Ledger/Stat data (where your Ledger/Stat table is populated with an extract from your General Ledger) and you might use a second user-defined “organizational” Key Processing Dimension for Legal Entity or Company if one of those dimensions is present in your source General Ledger.

Alternatively, if you have more than one kind of Organizational dimension in your source General Ledger (e.g., Cost Center, Company, Legal Entity, Line of Business), you may be able to concatenate one or more such General Ledger dimensions in your ledger and source system extracts (e.g., Cost Center plus Company) to populate a compound Organizational Unit dimension in your OFSAA data model.

### ***Product View***

A Product View facilitates the measurement of profitability of specific product lines including

- On balance-sheet customer-facing products such as Checking, Savings, Time Deposits, Credit Cards, Commercial Loans, Consumer Loans, Credit Lines, or Mortgages

- On balance-sheet, internal products (non-customer facing products) such as Wholesale Funding or Investments
- Off-balance sheet customer-facing services such as Merchant Card, insurance policies, or mutual fund positions
- Off-balance sheet products that may be either internal or custom facing such as options, other derivatives, etc

### ***Customer View***

A Customer View facilitates the measurement of customer profitability under any definition of customer you choose to implement. Examples of customers or customer segments might include demographic segments, commercial customers, high net worth customers, retail customer segments, etc.

### ***Channel View***

A Channel View facilitates the measurement of the profitability of various methods for delivering products or services to your customer base. As with Customer, you may define your channels any way you wish. Examples of channels might include ATM, Electronic Banking, Telephone, Retail Branch, and Mail. A better understanding of the cost structures of your delivery channels can assist you in steering your customers to lower cost channels or in doing a better job of aligning true costs with revenues through fee structures designed to encourage customers to utilize lower cost channels.

### ***Key Processing Dimensions in the OFSAA Data Model***

Key Processing Dimensions are the keys that the OFSAA engines use to access and segment business data. Every Key Processing Dimension is present in nearly all of your business fact tables, and the usage of processing dimensions permeates most OFSAA rule types.

Seeded dimensions include Financial Element (only found in the Management Ledger), Organizational Unit, General Ledger Account, Common Chart of Accounts, and Product. You may add your own user-defined processing dimensions to your data model.

### ***Financial Element***

Used only in the Management Ledger, Financial Element classifies your data in a fashion not typically found in most General Ledgers. When you initially load your Financial Accounting data from your General Ledger, Financial Elements are used to distinguish between:

- Ending monthly balances (FE 100)
- Average monthly balances (FE 140)
- Interest Income or Expense (FE 420)
- Non-Interest Income (FE 455)
- Non-Interest Expense (FE 457)

### ***Seeded Financial Elements***

The Financial Element dimension is the only seeded Key Processing Dimension that comes with its own seeded dimension member values. Seeded Financial Element values – dimension members from 0 to 10,000 – may not be modified, but they may be used as storage targets for initial load data or as output targets from allocation rules or

other processes. In addition to the five Financial Elements discussed above, OFSAA comes seeded with dozens of additional Financial Elements. For the most part, these other seeded Financial Element values are used as output dimensions member values for:

- The result tables produced by OFSAA Asset/Liability Management (see the *Oracle Financial Services Asset Liability Management (OFSALM) User Guide*.) These Financial Elements are only rarely used within the Management Ledger table (Ledger/Stat).
- Aggregated Instrument-level transfer pricing charges and credits. Such aggregations can be accomplished by using allocation rules or by using application functionality contained within the OFSAA Funds Transfer Pricing application (see the *Oracle Financial Services Funds Transfer Pricing User Guide*.) Seeded Financial Elements that are commonly used as targets in aggregations of FTP data are Financial Element 450 – Transfer Pricing Charge/Credit; and Financial Element 170 – Weighted Average Transfer Rate.
- Direct transfer pricing of Ledger/Stat (again, see the *Oracle Financial Services Funds Transfer Pricing User Guide*; the same Financial Elements (170 and 450) are also used as output targets for direct transfer pricing of Ledger/Stat).
- Storing final, ready to report, fully allocated multidimensional results. While user-defined Financial Elements may also be used for this purpose, there are a series of seeded Financial Elements that you may choose to utilize as they are pre-built to function as “reporting lines” within the Oracle Financial Services Profitability Analytics OBI application.

For a complete listing of all seeded Financial Elements, see [Appendix F, “Seeded Financial Elements”](#). For examples of the usage of user-defined Financial Elements, see the sections “[Initial Loads to the Management Ledger](#)” and “[Usage Examples of User-Defined Financial Elements](#)”.

### **Organizational Unit**

Found in your Management Ledger as well as in all of your Instrument and Transaction Summary tables, Organizational Unit is generally mapped to your General Ledger responsibility center (cost center or profit center) or department.

### **User-Defined Organizational Units**

Additional organizational dimensions may be added as user-defined key dimensions. For example:

- In a Management Ledger implementation, you may wish your profitability implementation to record the organizational source of allocated expense balances to facilitate inbound-outbound center-to-center allocation reporting.
- When loading account level data into the OFSAA Data Model, you will normally map your Book-of-Record cost center to the Organizational Unit dimension (the value that reconciles to your General Ledger). You might also, however, wish to define a Relationship Owning Center or a Company, Line of Business, or Legal Entity dimension.

### **General Ledger Account**

General Ledger Accounts are mapped from your source General Ledger systems. Financial Accounting systems sometimes refer to this dimension as GL Account or Natural Account.

### ***Common Chart of Accounts***

The Common Chart of Accounts dimension is a high-order product dimension that is typically used to link budgeting definitions of product with Transfer Pricing, ALM, or Profitability Management definitions of product (see the User Guides for Funds Transfer Pricing, Asset Liability Management, and Balance Sheet Planning; also see [Appendix G, “Debit and Credit Conventions”](#)).

### ***Product***

Nearly every Profitability Management implementation is interested in a Product view of profitability. Additionally, the seeded Product dimension is normally used for modeling purposes in Funds Transfer Pricing and/or in Asset Liability Management.

### ***User-Defined Key Processing Dimensions***

You may customize your data model by adding your own user-defined Key Processing Dimensions. See the section entitled “Overview of Dimensionality in OFSAA” or the Installation & Configuration Guide for details on adding your own user-defined Key Processing dimensions.

You may also add Standard Dimensions or Simple dimensions to your data model, but the OFSAA Profitability Management engine can only actively utilize Key Processing Dimensions within allocations rule sources, drivers, or outputs.

### ***Additional Profitability Dimensions***

One reason for adding user-defined Key Processing Dimensions is that your institution's design calls for additional profitability dimensions such as Channel, Customer, Geography, Line of Business, or Relationship Owning Unit, Company, or Legal Entity.

### ***Additional Working Dimensions***

Another reason you may wish to add a user-defined Key Processing Dimension is that you need an additional working dimension, i.e., a dimension along which you do not intend to fully develop Management Accounting analytics. For example, you may wish to add a user-defined Key Processing Dimension for Cost Pool, Transaction or Activity. These dimensions may be actively used within your allocation rules but are typically “in process” dimensions, i.e., dimensions used as stepping stones along the path to fully allocated profitability dimensions.

Yet another reason you may wish to add Key Processing Dimensions is that you must map all of the dimensions present in your source General Ledger to OFSAA Key Processing Dimensions. If your source General Ledger has dimensions such as Company or Legal Entity, you may need to add these dimensions to your model. If a dimension present in your source General Ledger is “organizational” in nature (i.e., Company or Legal Entity), it may be possible to concatenate such dimensions into the single OFSAA Organization Unit dimension as you design and build your ETL for loading source data for your OFSAA data model (ledger data, instrument data, or transaction summary data).

## Overview of Profitability Management Methodologies

### Financial Accounting vs. Management Accounting

General Ledger systems are designed to yield Financial Accounting results such as Balance Sheets, Income Statements, Sources & Uses Statements, and other regulatory reports that must be compiled according to international or local accounting standards such as International Financial Reporting Standards (IFRS) or Generally Accepted Accounting Principles (US). In addition to these external reporting functions, GL systems serve internal control and management accountability functions. GL systems need to be able to account for and answer questions such as:

- How much is being spent on occupancy expense?
- How much is being spent on salary and benefits expense?
- How well are we managing IT and network costs?
- Are cost center managers under- or over-budget?
- How are we accounting for cash and receivables?
- How are we accounting for depreciation, goodwill, and other intangibles?
- How are we managing cash flow (cash basis accounting vs. accrual basis accounting)?
- Can we demonstrate appropriate segregation of duties and other internal controls?
- Can we track General Ledger balances back to source transactions and documents?
- Can we satisfy our external stakeholders including auditors, regulators, and investors?

Management accounting systems are often derived in whole or in part from Financial Accounting systems, but are designed to answer a different class of questions such as:

- Which products, customers, geographies, lines of business, divisions, and channels are most profitable?
- How can we influence customer behavior to maximize profitability?
- How well are we controlling risks including interest rate risk, foreign currency risk, credit risk, fiduciary risk, legal risk, operating risk, and market risk?
- Can we measure our profitability in all dimensions on a risk-adjusted basis?
- How can we optimize our external fee structures and internal incentives to optimize risk adjusted profitability?
- Are we properly organized to optimize internal incentives?
- How well are we managing capacity?

### Management Accounting Models

Profitability Management models can generally be divided into two types: Management Ledger level profitability (an aggregated multi-dimensional view) and Customer Account Level profitability (a very detailed view). Management Ledger-level profitability solutions have some advantages over Account Level solutions. They are generally easier to construct and maintain, have shorter processing cycles, and results in smaller, more manageable volumes of data than Account Level profitability solutions.

The primary advantages of Account Level profitability solutions over Management Ledger-level solutions is their ability to segment profitability results using both dimensional measures and non-dimensional attributes & measures found at the account level. Such account level attributes and measures can be exploited to stratify your results; they can also be assembled into “pseudo” dimensions (low cardinality dimensions) that typically would not support hierarchies in a reporting context. Examples of ways in which you might exploit account level attributes to supplement multidimensional results would include

- New Business vs. Total Book of Business
- Repricing Profiles
- Runoff Profiles
- Ranges of Risk
- Customer Level Balance Ranges
- Age of Customer Relationship

Another important advantage of Account Level solutions is that they allow you to construct highly granular capital allocation methodologies. This is particularly valuable in developing Risk Adjusted Profitability Management (RAPM) metrics (you may, however, aggregate your bottom-up equity allocations to the Management Ledger-level).

Institutions often construct models that yield both kinds of profitability results. When both kinds of models are built, they will sometimes reconcile to one another and they will sometimes not reconcile to one another. Whether or not they reconcile is a function of the methodologies you select and your institution's preferences.

## **Management Ledger Profitability Models**

The starting point for most Management Ledger-level profitability solutions is the Financial Accounting data that you import from your General Ledger system. A number of different processes are generally performed to manipulate your initial Financial Accounting data to produce Management Accounting results.

Management Ledger-level models often involve both “top-down” and “bottom-up” kinds of processes. In the Management Ledger context, top-down processes are typically composed of series of allocation rules that operate in a cascading or “waterfall” sequence that begin with your Financial Accounting data.

For example, in constructing Organizational profitability within a Management Ledger, your design might begin with an analysis in which you subdivide all of your Organizational Units (or Responsibility Centers) into either Cost Centers or Profit Centers. Cost Center managers are responsible for managing their expenses and meeting their budgets while Profit Center managers have P & L responsibility. While your General Ledger may support independent P & L's for Divisions, Regions, Lines of Business, or Companies, your objective in a top-down Organizational Profitability solution is to push P & L responsibility down to much lower levels: branches, loan origination centers, etc.

You might continue your analysis by organizing your Cost Centers into functional groups such as Overhead, Indirect Support, and Direct Support. You might then devise a series of allocation rules that allocate Indirect Support expenses to Direct Support cost centers which in turn allocate their (burdened) expenses to Profit Centers. You might finish by allocating Overhead expenses directly to Profit Centers. In some cases, you might also need to allocate some non-interest revenues. Since interest income and interest expense are typically booked to Profit Centers within your General Ledger, it is less likely that you will need to allocate these balances.

You also might wish to build some balance sheet allocations to move non-financial assets & liabilities (such as cash, fixed assets, goodwill, and equity) to the Profit Centers that rely upon those non-financial assets & liabilities to support their businesses. The rationale here is that you generally want to transfer price the entire balance sheet, not

just instrument-level balances. As with your instrument balances, these non-financial balances will generate transfer pricing charges & credits and these secondary transfer pricing charges & credits need to be assigned to the proper Profit Centers.

Frequently, Management Ledger implementations focus on both Organizational Profitability and Product Profitability. Typically, these kinds of implementation will first complete the Organizational Profitability component and then continue with additional allocation rules to develop Product Profitability. Product Profitability is normally developed using both top-down and bottom-up methods.

The above narrative is meant to be illustrative; there are an enormous variety of methodologies you might choose from in constructing Management Ledger-level models for Organizational Profitability, Product Profitability, or multi-dimensional profitability incorporating additional dimensions.

As mentioned above, Management Ledger-level models typically involve both “top-down” and “bottom-up” kinds of processes. One kind of bottom-up process is matched term funds transfer pricing. In matched term funds transfer pricing, every instrument asset record is assigned a “cost of funds used” and every instrument liability record is assigned a “credit for funds provided”. These instrument level charges and credits (expenses and revenues) are subsequently aggregated to the Management Ledger. In implementations where activity based costs (and sometimes revenues) are assigned to each instrument level record, these charges and credits may also be aggregated to the Management Ledger-level.

### **Account Level Profitability Models**

As with Management Ledger-level solutions, Account Level solutions are also frequently constructed using a combination of top-down and bottom-up processes. For example, you might employ top-down allocations to distribute Management Ledger-level income and expenses (expenses being more typical) to the instrument level; and you might employ bottom-up processes such as account level funds transfer pricing and/or allocation rules that employ unit costs and volumes to assign income and expenses (again, expenses being more typical) to instrument level records (note that such low-level, bottom-up income and expense assignments may also be aggregated to the Management Ledger).

### ***Initial Loads to the Management Ledger***

When you initially load the Management Ledger (the Ledger/Stat table) with data extracted from your General Ledger, you need to map each row of extracted data to a Financial Element. Similarly, if or when you load other externally sourced data into Ledger/Stat (i.e., statistical data not extracted from your General Ledger), you must also map each row of your extracted data to a Financial Element.

The recommended Financial Elements for General Ledger data or for externally sourced statistical data is as follows:

- Ending Balances – Financial Element 100
- Average Balances – Financial Element 140
- Interest Income & Expense Balances – Financial Element 420
- Non-Interest Income Balances – Financial Element 455
- Non-Interest Expense Balances – Financial Element 457
- Statistical data – Financial Element 10,000

## **Usage Examples of User-Defined Financial Elements**

You may, of course, create additional user-defined Financial Elements for use in storing:

- Additional statistics that you source from an external system
- Driver statistics that are built up through allocation rule processes and that you intend to use in subsequent allocation rules
- Allocation rule outputs if, for example, you are building up cost pools or other intermediate balances that represent “in process” balances that are not yet fully allocated to final multidimensional results
- Final, fully allocated balances that you wish to report on (i.e., you may use a range of user-defined Financial Elements to build up a “reporting line” structure for fully allocated multidimensional profitability results)

### ***Using Financial Elements as Income Statement Reporting Lines***

You may wish to build up fully allocated, multidimensional profitability balances that represent reporting lines in your final analytical results. For example, if you wish to construct income statements by Organizational Unit and by Product and by Geography, you might choose to construct allocation rules that will drive towards final result balances by Organization Unit, by Product, by Geography, and by Financial Element where Financial Elements are used to store reporting line balances for:

- Interest Income / Interest Expense
- Charge for Funds Used / Credit for Funds Provided
- Fee Income
- Direct Expense
- Indirect Acquisition Expense
- Indirect Servicing Expense
- Direct & Indirect Taxes

Under this approach, Financial Elements that serve as “Reporting Lines” in your Management Accounting income statements take the place of the General Ledger Account dimension that serves the “reporting line” function in pre-allocated Financial Accounting income statement data extracted from your General Ledger. Whereas your Financial Accounting income statements may be very detailed and may be comprised of hundreds or even thousands of GL Accounts, your Management Accounting income statement reporting line structure will likely have far fewer rows, probably 30-60 lines.

A seeded set of Financial Elements that you may wish to use as reporting lines are included in the standard data model (the complete list of all seeded Financial Elements is documented in [Appendix F, “Seeded Financial Elements”](#)).

A seeded Financial Element hierarchy is also provided that contains rollup points for the seeded “reporting line” Financial Elements. These rollup points, which include values such as Total Interest Income, Total Interest Expense, Net Interest Income, Risk Adjusted Net Interest Income, and Risk Adjusted Net Income, are structured to function as subtotals. The seeded “reporting line” Financial Elements (leaf member values plus rollup point values) are pre-built in to the Oracle Financial Services Performance Analytics (OFSPA) Business Intelligence (BI) application. If you choose to develop a strategy that involves allocating balances to Financial Elements that are intended to support such a “reporting line” structure, you may customize the seeded profitability Financial Element structure by (1), adding your own user-defined Financial Elements and (2), modifying the seeded Financial Element hierarchy.



After having customized the seeded profitability Financial Elements and the related subtotalling hierarchy, the changes can be made to flow seamlessly into the analytical dashboards within the OFSPA BI application without any additional customization.

You may, of course, choose not to develop a “reporting line” strategy for developing multidimensional profitability; or you might choose to build out reporting lines under another user-defined Key Processing Dimension.

Regardless of what dimension you choose for storing “reporting line” balances within your Management Ledger (assuming that you do adopt such a strategy), your final “reporting line” balances may be the result of bottom up processes such as:

- Aggregation of Transfer Pricing charges and credits from the instrument level. Such FTP charges and credits may result from charges or credits for principal balances, charges or credits for basis risk or liquidity risk, or charges or credits for secondary balances that you have allocated to instruments (such secondary balances are typically allocated from the Management Ledger to the instrument level; examples would include loan loss reserves, risk equity, central bank reserves; or other allocated debit or credit balances such as plant & equipment, cash & due from banks, goodwill, prepaid expenses, etc.) The charges & credits associated with these allocated balance-sheet balances may be generated using the transfer pricing rates that have already been assigned to each instrument's principal balance; or you may adopt a simple methodology for ascribing transfer rates to these allocated balances (e.g., pre-determined rates that are assigned to each allocated balance as a function of its perceived duration).
- Aggregation of rate times volume results from the Instrument and/or Transaction Summary levels for activity based expenses or revenues
- Aggregation of instrument level revenues or expenses that are derived directly from your source systems (e.g. Accrued Interest Payable or Receivables, Fee income, etc.)

Or they may be the result of top-down processes such as

- Non-interest expense balances (typically in the form of center-to-center or center-to-product allocations)
- Direct Transfer Pricing of Ledger balances (such direct transfer pricing charges and credits may be generated within the Management Ledger using OFSAA Funds Transfer Pricing; OFSAA FTP posts such results to Financial Element 450). For details on direct transfer pricing of ledger balances, see the OFSAA Funds Transfer Pricing User Guide.

### ***Reporting Lines for Selected Balance-Sheet Balances***

You may also wish to build up “reporting line” Financial Elements for selected balance-sheet balances. Although you will not always be able to demonstrate a balanced balance-sheet (for example, it is not meaningful to construct a balanced balance-sheet for a Product line or for a Line of Business), if you build out selected balances, those balances could be used in ratio analysis either within the OFS Profitability Analytics OBI application or within your own reporting tool. The OFS Profitability Analytics OBI application includes predefined cube definitions that generate ratios and formulas that use such balance-sheet balance reporting lines (e.g., for risk adjusted profitability ratios such as return on assets, risk adjusted return on equity, net income after capital charge, etc.)



**CHAPTER 14*****Application Preferences******Overview of Profitability Management Application Preferences***

In addition to Global Preferences, each of the OFS Analytical Applications has its own local application preferences. In the same way as in Global Preferences, Application Preferences (for each application) employs a "delegation" model that allows Administrators to set some preference items for all users while allowing non-administrative users to personalize other preference items.

Application Preferences allow both Administrators and End Users to establish default values and to manage other core application parameters that affect the way business rules are created and the way Profitability Management processes are run.

The procedure for working with and managing Application Preferences includes the following steps:

- Updating Application Preferences.

***Updating Profitability Management Application Preferences***

Navigate to Profitability Management Application Preference to update your preference items.

The following table describes each of the Profitability Management preference items.

Selected Terminology

Term	Description
<b>Select Preference For</b>	
Show Preferences For	<p>There are two modes in which you can access Application Preferences:</p> <p><b>Administrator:</b> If the user has Administrator privileges, he can define preferences for the "All User" group and for his own personal account, which may be the same or different from the "All User" settings. The Administrator can also designate the "All User" preferences as Editable or Non-editable on a row by row basis. If the individual preference is checked as "is Editable", then End Users can update or override the Administrator's default value for their own individual account. If the "is editable" box is not checked, then End Users are not able to change the default for their own account.</p> <p><b>End User:</b> If the user does not have administrator privileges, then certain preference items may have been pre-set by the administrator and the user may not be allowed to change the value. All Application Preference settings are displayed, regardless of access privilege.</p>
Active for Master Maintenance	<p>If a user has more than one application available on their Left Hand menu, then they will need to designate Application Preferences from one of their applications as "Active for Master Maintenance". Certain preferences such as the default folder and read / write access setting will be referenced by items within Master Maintenance based on this selection.</p>
<b>Parameters - General</b>	

Term	Description
Security Map	This option allows you to select an existing security map. Security Maps can be used to control the dimensions and dimension members each user can access when building and executing rules. For more information on Security Mapping, refer to <i>Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) User Guide Release 8.0.4</i> .
As of Date	All processes reference this date at run time to determine the data to include in the process. The As-of-Date value you set in Application Preferences applies to interactive job execution (i.e., when you choose to execute a rule directly from a Summary screen). For batch processing, the As-of-Date is an input parameter.
Show Execution Parameters	If this option is selected, a pop-up window appears whenever you execute a process interactively from a Summary screen. Within this pop-up window, you may confirm or modify your run execution parameters (As-of-Date and Legal Entity).
Legal Entity	<p>Similar to As-of-Date, all processes reference Legal Entity at run time to determine the data to include in the process. The value of Legal Entity you set in Application Preferences applies to interactive job execution (i.e., when you choose to execute a process directly from a Summary screen). For batch processing, Legal Entity is an input parameter.</p> <p><b>Note:</b> Legal Entity is designed to support implementations that require multi-entity or multi-tenant functionality. If your implementation does not require this functionality, you may utilize the Default Legal Entity in all your processes.</p> <p>No additional parameter is required for Legal Entity for command line execution. EPM Engines read the default Legal Entity from the Application preference value saved for the User who is executing from command line.</p> <p>The default legal entity is a per user preference setting. To set this, the following steps are required:</p> <ol style="list-style-type: none"> <li>1. Create a security-map in AAI containing the legal-entity hierarchy as an app-admin.</li> <li>2. Set the default security-map in a user preferences.</li> <li>3. Navigate to "execution parameters" block in user-preferences and choose the default legal-entity member.</li> </ol> <p>If you do not want to use the Multi-Entity feature with Security, you need to save the Application preference for the user executing the batch. The usual choice for Legal Entity would be Default Member in Application preferences.</p> <p>Default implies -1 code.</p> <p>Default value for Legal Entity dimension column in all instrument and ledger table is -1.</p> <p><b>Note:</b> Legal Entity only acts as a run time parameter for Variable Allocation rules. Starting with release 8 Legal Entity is made an optional run time parameter.</p> <p>Legal Entity is not applicable to allocation rules using the Ledger/Stat table.</p>
Disable Legal Entity	If the "Disable Legal Entity" checkbox in application preferences screen is checked, then Legal Entity will no longer be a run time parameter. On disabling Legal Entity, user should edit the already defined Variable Allocation rules and modify the value of Legal Entity dimension in all applicable tabs (Source, Driver and Outputs) in the allocation specification screen. This is an optional parameter.
<b>Processing - General</b>	

Term	Description
Default Total Error Message Limit	Profitability Management processes log error details into the FSI_O_PROCESS_ERRORS table. This parameter defines the limit on the total number of errors that will be logged for any ALM Process.
Default Error Message Limit Per Item	This parameter defines the total number of errors that will be logged for a given type of error.
Debugging Output Level	<p>The debugging output level determines the amount of sql that will be written to the processing log. There are three levels available:</p> <p><b>Do not output SQL to log file:</b> A log file will not be created.</p> <p><b>Show Significant Calculation SQL:</b> Log file is created and will contain those SQLs that are tagged as significant.</p> <p><b>Show all SQL:</b> Log file is created and will contain all the SQL that the engines execute.</p> <p>The Allocation Engine log files can be accessed by the system administrator in the following location on the server:  <b>\$OFSAA_LOG_HOME/&lt;MIS_DATE&gt;/&lt;INFODOM&gt;/Allocation Engine/ folder</b>  The file names will be prefixed with the application initials and will also contain the unique batch run id of the execution request.  for example: <b>Allocation Engine_&lt;INFODOM&gt;_Task1_&lt;Allocation SYSID&gt;.log</b></p> <p>To UNDO log files can be accessed by the system administrator in the following location on the server:  <b>\$OFSAA_LOG_HOME/&lt;SYS_DATE&gt;/&lt;INFODOM&gt;/Allocation Engine/ folder</b>  The file names will be prefixed with the application initials and will also contain the unique batch run id of the execution request.</p>
<b>Processing - Application Specific</b>	
Enable UNDO for Selected Allocations	Set this value to Yes to enable UNDO functionality for individual allocation rules. For details, see Allocation Execution History, page 19-1.
Enable Mass UNDO	Set this value to Yes to enable Mass UNDO functionality. For details, see Allocation Execution History, page 19-1.
Enable UNDO for Selected Batches	Set this value to Yes to enable UNDO for selected batches. For details, see Allocation Execution History, page 19-1.
Enable UNDO for Selected Allocation Models	Set this value to Yes to enable UNDO for selected allocation models. For details, see Allocation Execution History, page 19-1.
Maximum Circular Allocation Iterations	This value governs the maximum number of circular iterations within a Circular Allocation Model. Its purpose is to ensure that circular models will terminate without going into an infinite loop even if the model does not converge on the value of Threshold Source Balance or Threshold Percentage you have selected for your model. For details, see Allocation Models, page 16-1.

Term	Description
Scenario Values	<p>The Application Preference setting for Scenario dimension allows you to run the same rules for different scenarios, such as, Actuals, Budget, and so on. These values are inherited by rules that specify Use Application Preferences in their Sources, Drivers or Outputs. For details, see Allocation Specification, page 15-1.</p> <p><b>Scenario for Source:</b> When an allocation rule's Source is configured as Use Application Preferences in Allocation Source section, the value used by the rule in its Source is the one specified in Application Preferences.</p> <p><b>Scenario for Driver:</b> When an allocation rule's Driver is configured as Use Application Preferences in Allocation Source section, the value used by the allocation rule in its Driver is the one specified in Application Preferences.</p> <p><b>Scenario for Output:</b> When an allocation rule's Output (applicable to both Debit and/or Credit) is configured as Use Application Preferences in Dimension section, the value used by the allocation rule is the one specified in Application Preferences.</p>
<b>Assumption Management Defaults</b>	
Folder Name	<p>This parameter allows you to define the default folder selection. The folder selection for all rule types will be defaulted to this selection within the summary page search screen and when creating a new rule. This selection acts as the starting value for convenience only and users can change to any other available value at their discretion.</p>
Access Type	<p>This parameter allows you to set the default access type setting. Selections include Read / Write and Read Only. This selection acts as the starting value for convenience only and users can change at their discretion.</p>

1. Navigate to the Profitability Management Application Preferences page.

Application Preferences Apply Cancel Reset to Default

▼ Select Preferences For

Show Preferences For PFTUSER

Active for Master Maintenance

▼ Parameters - General

Property Name	Property Value	Is Editable
Security Map	None	<input type="checkbox"/>
As of Date	06/25/2014	<input checked="" type="checkbox"/>
Show Execution Parameters	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Runtime Parameters	Leaf Description	<input type="checkbox"/>
Legal Entity	Default Member	<input type="checkbox"/> Disable Legal Entity

▼ Processing - General

Property Name	Property Value	Is Editable
Default Total Error Message Limit	250	<input checked="" type="checkbox"/>
Default Error Message Limit Per Item	25	<input checked="" type="checkbox"/>
Debugging Output Level	Show All SQL	<input checked="" type="checkbox"/>

▼ Processing - Application Specific

Property Name	Property Value	Is Editable
Enable Undo for Selected Allocations	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input checked="" type="checkbox"/>
Enable Mass Undo	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input checked="" type="checkbox"/>
Enable Undo for Selected Batches	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input checked="" type="checkbox"/>
Enable Undo for Selected Allocation Models	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input checked="" type="checkbox"/>
Maximum circular allocation iterations	5	<input checked="" type="checkbox"/>
Scenario for source	Actual	<input checked="" type="checkbox"/>
Scenario for driver	Actual	<input checked="" type="checkbox"/>
Scenario for output	Actual	<input checked="" type="checkbox"/>

▼ Assumption Management Defaults

Property Name	Property Value	Is Editable
Folder Name	AUTOSEG	<input checked="" type="checkbox"/>
Access Type	<input type="radio"/> Read Only <input checked="" type="radio"/> Read/Write	<input checked="" type="checkbox"/>

2. Input values for all line items.

**Note:** If you are the application administrator, define default values for the “All User” group by making the appropriate selection from the “Show Preference For”, drop list at the top of the page. Pay particular attention to the “Is Editable” status and determine which items require administrative control and which items non-administrative users will be able to set for themselves.

3. Select Apply to confirm changes

4. Select the Reset to Default option if you would like to clear all previously applied inputs and return to the original default state.





**CHAPTER 15*****Profitability Configuration***

This chapter covers the following configuration options for Profitability application:

- [Setup Information](#)
- [Parallel Execution Configuration](#)
- [Setup Parameters](#)
- [Management Ledger Configuration](#)

***Setup Information***

To configure Setup Information, navigate to Profitability Management > PFT Configuration > Setup Information.

Setup Information

▼ Setup Information

Character Set	US7ASCII
Enable Currency Type	<input type="checkbox"/>
Functional Currency	Surinam Dollar ▼
Enable Multi Currency	<input type="checkbox"/>
Output Tablespace	USERS ▼
Signage	Reverse GAAP ▼

▼ Basic Fiscal Configuration

Start Month	November ▼
-------------	------------

Field	Description
Enable Currency Type	Select the check box to enable the currency type.
Functional Currency	Select the relevant functional currency from the drop down list.
Enable Multi Currency	Select the check box to enable multi currency.
Output Tablespace	Not applicable for PFT application.

Field	Description
Signage	This is an "information only" setting describing how balances are carried. The options are: <ul style="list-style-type: none"> <li>● Natural Signage</li> <li>● GAAP Signage</li> <li>● Reverse GAAP Signage</li> </ul>
Start Month	The start month of the fiscal year.

### Parallel Execution Configuration

To configure Parallel Execution, navigate to Profitability Management > PFT Configuration > Parallel Execution Configuration. For more information, see [Profitability Management Parallel Execution](#).

Parallel Execution Configuration

▼ Parallel Execution Configuration

Parallel DML	DISABLE	▼
Parallel Degree		
Parallel Degree Policy	MANUAL	▼
Parallel Query	DISABLE	▼

Save      Reset

Field	Description
Parallel DML	Disable, Enable, or Force
Parallel Degree	Integers between 0 and 99
Parallel Degree Policy	Manual, Limited, or Auto
Parallel Query	Disable, Enable, or Force

### Setup Parameters

To configure setup parameters, navigate to Profitability Management > PFT Configuration > Setup Parameters.

## Setup Parameters

&gt; Setup Parameters

Param Name	Param Value
STRATIFICATION_SCALE_FACTOR	2
STATIC_TABLEID_TEMP_TABLE_CROSS_JOIN_LIMIT	2000000000

Save

Field	Description
STRATIFICATION_SCALE_FACTOR	Not used by PFT application.
STATIC_TABLEID_TEMP_TABLE_CROSS_JOIN_LIMIT	Integers between 0 and 99. Default value as 20 million. If the Cartesian product of the leaf values selected for each dimension exceeds this limit, then the static table driver UI will display only the already defined combinations. Further reducing this parameter value will improve the UI performance, since UI will only display the already existing definitions.

## Management Ledger Configuration

To configure management ledger, navigate to Profitability Management > PFT Configuration > Management Ledger Configuration. For more information, see [Functional Currency](#).

Management Ledger Configuration

> Management Ledger Configuration

Functional Currency	Start Month	Table Name
<input type="checkbox"/> Turkish Lira	August	Management Ledger

Save

Field	Description
Functional Currency	The currency that is applicable for the Table that you select.
Start Month	The start month of the fiscal year.
Table Name	The name of the table.



***Introduction***

This chapter describes Allocation Specification functionality.

- Allocation Specification Summary & Detail Screens
- Navigation within the Allocation Specification Summary Screen
- Navigation within the Allocation Specification Detail Screen
  - Initial Definition Process Tab
  - Source Process Tab
  - Operator Process Tab
  - Driver Process Tab
  - Outputs Process Tab
  - Review Process Tab
- Allocation Examples

***Summary & Detail Screens***

Upon initially navigating to Profitability Management > Rule Specification > Allocation Specification, a summary screen is displayed showing a set of Allocation rules. Using search criteria, you can control the set of Allocation rules that are displayed. When you Add, Edit, or View a rule, a detail screen is displayed.

***Navigation within the Summary Screen***

When you first navigate to the Allocation Specification summary screen, the allocations stored within your current default Folder are presented in a summary grid. The Allocation Specification summary screen has two containers: Search and Allocation Specification.

***Search Container***

Your default Folder functions as a search constraint. The value of your default Folder is set in Application Preferences for Profitability Management. You may select a different Folder or you may remove the Folder constraint entirely by selecting the “blank” Folder, i.e., no Folder. You may also search by Allocation Name or by Allocation Type.

Allocation Specification As of Date: 06/25/2014

Search 🔍

Folder: PFTSEG Name:

Allocation Type:

1 - 20 / 214 Jump to page

<input type="checkbox"/>	Name	Creation Date	Created By	Allocation Type	Last Modified Date	Modified By	Folder	Status
<input type="checkbox"/>	26420257_802010_key_node	09/26/2017 11:49:05	PFTUSER	Static Driver Table	09/26/2017 11:49:05	PFTUSER	AUTOSEG	
<input type="checkbox"/>	26420257_802010_key_only	09/26/2017 11:37:37	PFTUSER	Static Driver Table	09/26/2017 11:37:37	PFTUSER	AUTOSEG	
<input type="checkbox"/>	26420257_802010_key_tgt	09/25/2017 14:58:53	PFTUSER	Static Driver Table	09/26/2017 11:34:33	PFTUSER	AUTOSEG	
<input type="checkbox"/>	26420257_802010_prereq_1	09/25/2017 15:39:26	PFTUSER	Constant	09/26/2017 12:03:20	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_1	08/23/2016 23:15:12	PFTUSER	Dynamic Driver	08/24/2016 00:07:13	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_10	08/25/2016 18:42:28	PFTUSER	Dynamic Driver	08/31/2016 11:26:17	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_11	08/26/2016 11:16:19	PFTUSER	Dynamic Driver	08/26/2016 11:20:32	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_12	08/26/2016 12:32:56	PFTUSER	Dynamic Driver	08/26/2016 12:35:06	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_13	08/26/2016 12:43:14	PFTUSER	Dynamic Driver	08/26/2016 12:43:41	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_2	08/24/2016 12:08:58	PFTUSER	Dynamic Driver	08/24/2016 12:21:25	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_3	08/24/2016 12:24:53	PFTUSER	Dynamic Driver	08/24/2016 12:28:31	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_4	08/24/2016 12:32:33	PFTUSER	Dynamic Driver	08/24/2016 12:34:15	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_5	08/24/2016 12:51:05	PFTUSER	Dynamic Driver	08/31/2016 11:16:00	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_6	08/24/2016 16:21:20	PFTUSER	Dynamic Driver	08/24/2016 21:58:39	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_7	08/25/2016 15:40:55	PFTUSER	Dynamic Driver	08/25/2016 18:24:16	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_8	08/25/2016 18:33:19	PFTUSER	Dynamic Driver	08/25/2016 18:34:13	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_9	08/25/2016 18:37:48	PFTUSER	Dynamic Driver	08/25/2016 18:40:06	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_PREREQ_1	08/30/2016 17:24:52	PFTUSER	Constant	08/30/2016 17:24:52	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_PREREQ_2	08/30/2016 17:25:09	PFTUSER	Constant	08/30/2016 17:25:49	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_14	08/23/2016 12:15:17	PFTUSER	Static Driver	08/23/2016 17:04:07	PFTUSER	AUTOSEG	

Copyright © 1993, 2018 Oracle and/or its affiliates. All rights reserved.

**Search Control**



You may search for Allocation rules by Folder, by Allocation Name, or by Allocation Type. Enter your desired search criteria and click on the Search control.

**Reset Control**



Restores the default Folder, removes any Allocation Name or Allocation Type constraint you may have specified, and refreshes the screen.

**Allocation Specification Container**

The Allocation Specification container presents a grid containing all of the Allocation rules that meet your search criteria. The Allocation Specification summary grid offers several controls that allow you to perform different functions when an Allocation rule is selected. To select an Allocation rule, click on a check box in the first column of the grid. More than one Allocation rule can be selected at a time but this will cause some of the controls to become disabled. Clicking on a checkbox a second time de-selects the Allocation rule.

You may select or deselect all of the Allocation rules in the summary grid by clicking on the check box in the upper left hand corner of the summary grid directly to the left of the Name column header.

**Add**



Clicking on the Add control begins the process of building a new Allocation rule. The Add control is disabled if any rows in the grid have been selected.

### View



Selecting a single row out of the grid enables the View control. Clicking on the View control allows you to view the contents of an Allocation rule on a read-only basis. The View control is only enabled when a single allocation rule has been selected.

### Edit



Selecting a single row out of the grid enables the Edit control. Clicking on the Edit control allows you to modify a previously saved Allocation rule. The Edit control is only enabled when a single allocation rule has been selected.

### Delete

Selecting one or more rows out of the grid enables the Delete control. Clicking on the Delete control deletes the rules you have selected.



### Copy



Selecting a single row out of the grid enables the Copy control. Clicking the Copy control allows you to create a copy of an existing Allocation rule. The Copy control is only enabled when a single Allocation rule has been selected. When you click on Copy, a Save As pop window will appear. Click **Save** after entering the Name, Description, Folder, and Access Type Details.

### Run



After having chosen a single row out of the grid, clicking on the Run control allows you to execute the selected Allocation rule. The As-of-Date that applies to the rule execution is visible in the upper right corner of the screen; it

may be reset in Application Preferences for Profitability Management. The Run control is only enabled when a single allocation rule has been selected.

**View Dependencies**



Clicking on the View Dependencies control generates a report on any rule objects that depend on the Allocation rule you have selected. The View Dependencies control is only enabled when a single allocation rule has been selected.

**Allocation Specification Summary Grid**

The following columns categorize each allocation rule in the summary grid:

- Name
- Creation Date
- Created By
- Allocation Type
- Last Modified Date
- Modified By
- Access Type
- Status

Allocation Specification As of Date: 06/25/2014

Search

Folder: PFTSEG  Name:

Allocation Type:

<input type="checkbox"/>	Name	Creation Date	Created By	Allocation Type	Last Modified Date	Modified By	Folder	Status
<input checked="" type="checkbox"/>	26420257_802010_key_node	09/26/2017 11:49:05	PFTUSER	Static Driver Table	09/26/2017 11:49:05	PFTUSER	AUTOSEG	
<input type="checkbox"/>	26420257_802010_key_only	09/26/2017 11:37:37	PFTUSER	Static Driver Table	09/26/2017 11:37:37	PFTUSER	AUTOSEG	
<input type="checkbox"/>	26420257_802010_key_tgt	09/25/2017 14:58:53	PFTUSER	Static Driver Table	09/26/2017 11:34:33	PFTUSER	AUTOSEG	
<input type="checkbox"/>	26420257_802010_prereq_1	09/25/2017 15:39:26	PFTUSER	Constant	09/26/2017 12:03:20	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_1	08/23/2016 23:15:12	PFTUSER	Dynamic Driver	08/24/2016 00:07:13	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_10	08/25/2016 18:42:28	PFTUSER	Dynamic Driver	08/31/2016 11:26:17	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_11	08/26/2016 11:16:19	PFTUSER	Dynamic Driver	08/26/2016 11:20:32	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_12	08/26/2016 12:32:56	PFTUSER	Dynamic Driver	08/26/2016 12:35:06	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_13	08/26/2016 12:43:14	PFTUSER	Dynamic Driver	08/26/2016 12:43:41	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_2	08/24/2016 12:08:58	PFTUSER	Dynamic Driver	08/24/2016 12:21:25	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_3	08/24/2016 12:24:53	PFTUSER	Dynamic Driver	08/24/2016 12:28:31	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_4	08/24/2016 12:32:33	PFTUSER	Dynamic Driver	08/24/2016 12:34:15	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_5	08/24/2016 12:51:05	PFTUSER	Dynamic Driver	08/31/2016 11:16:00	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_6	08/24/2016 16:21:20	PFTUSER	Dynamic Driver	08/24/2016 21:58:39	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_7	08/25/2016 15:40:55	PFTUSER	Dynamic Driver	08/25/2016 18:24:16	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_8	08/25/2016 18:33:19	PFTUSER	Dynamic Driver	08/25/2016 18:34:13	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_9	08/25/2016 18:37:48	PFTUSER	Dynamic Driver	08/25/2016 18:40:06	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_PREREQ_1	08/30/2016 17:24:52	PFTUSER	Constant	08/30/2016 17:24:52	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_PREREQ_2	08/30/2016 17:25:09	PFTUSER	Constant	08/30/2016 17:25:49	PFTUSER	AUTOSEG	
<input type="checkbox"/>	80204_22963259_14	08/23/2016 13:51:37	PFTUSER	Dynamic Driver	08/23/2016 17:04:07	PFTUSER	AUTOSEG	

**Name**

Displays the Allocation rule's short name. Performing a “mouse-over” on an Allocation Name displays the Allocation rule's long name and system identifier.



**Creation Date**

Displays the date and time at which an Allocation rule was created.

**Created By**

Displays the name of the user who created an Allocation rule.

**Allocation Type**

Displays an Allocation rule's type. Supported rule types are:

- Constant
- Static Driver
- Leaf
- Field
- Dynamic Driver
- Static Driver Table
- Lookup Driver Table

**Last Modified Date**

Displays the date and time at which an Allocation rule was last modified.

**Modified By**

Displays the name of the user who last modified an Allocation rule.

**Access Type**

Displays the “Read/Write” or “Read Only” property of an Allocation rule. Only the creator of a rule may change its Access Type.

**Status**

Prior to having executed an Allocation rule for the first time, Status is blank. After having executed an Allocation rule, the words “View Log” are displayed as a hyperlink. Clicking on the View Log hyperlink opens a log viewer enabling you to view the execution log for the Allocation rule's last run.

## ***Navigation within the Detail Screen***

When you Add, Edit, or View an Allocation rule, the Allocation Specification Detail Screen is displayed. The detail screen is composed of 6 process tabs that are described below. The appearance of the detail screen depends on which tab is active. Regardless of which of tab is currently active, the Process Tabs container and the Allocation Rule Definition container are always the first two containers displayed. Other containers shown depend on which tab is currently active. The Audit Trail container is a standard footer container for every OFSAA rule type. It displays Created By, Creation Date, Last Modified By, and Modification Date on the Audit Trail tab. The User Comments tab may be used to add comments to any rule, subject to a maximum of 4000 characters.

***Process Tabs Container***

Each of the 6 process tabs is designed create, edit, or view different components of an allocation's specification. You may navigate from any tab to any other tab at any time. The 6 process tabs are:

- Initial Definition
- Source
- Operator
- Driver
- Outputs
- Review

### **Allocation Rule Definition Container**

Like the Process tabs container, the Allocation Rule Definition container is always displayed regardless of which process tab is active. The Allocation Rule Definition container, however, is an active container only on the Initial Definition tab; it is displayed in a “read only” format in every other process tab. Its usage on the Initial Definition tab is described below.

The screenshot shows a window titled "Process Tabs" with a horizontal sequence of six tabs: "Initial Definition", "Source", "Operator", "Driver", "Outputs", and "Review". The "Initial Definition" tab is highlighted in orange. Below the tabs is the "Allocation Rule Definition" section, which contains the following fields and controls:

- Rule Name**: A text input field.
- Rule Description**: A large text area.
- Folder**: A dropdown menu with "AUTOSEG" selected.
- Access Type**: Two radio buttons, "Read Only" and "Read/Write", with "Read/Write" selected.
- Variable Rule**: A checkbox that is currently unchecked.

Regardless of which tab is active, you may always return to the Initial Definition table to rename a rule, change its description, its Folder, or its Access Type.

### **Initial Definition Process Tab**

The Initial Definition process tab allows you to specify Rule Name, Rule Description, Folder, Access Type, and Allocation Type.

### **Allocation Rule Definition Container**

Specify the Allocation Rule Name and Description, select a Folder in which the Allocation rule is to be stored, and specify whether you want the Allocation rule to be “Read/Write” or “Read Only” (Access Type). Naming your Allocation rule is required before it can be saved. Default values for Folder and Access Type are stored in Application Preferences for Profitability Management.

### **Variable Rule**

Starting with release 8 Legal Entity is made an optional run time parameter. If the "Disable Legal Entity" checkbox in application preferences screen is checked, then Legal Entity will no longer be a run time parameter.

If Legal Entity is disabled, then while defining a Variable Allocation rule user will have to select a value for the Legal Entity dimension in all applicable tabs in allocation specification screen.

Allocation Rule Definition

Rule Name \*

Rule Description

Folder: AUTOSEG Access Type:  Read Only  Read/Write Variable Rule

For

Table Name \*: Management Ledger Column/Expression:  Column Name  Expression

Column Name: [Current\_Amount]

Allocation Source

Dimension	Value	Description
Financial Element	ALL	ALL
Organizational Unit	ALL	ALL
General Ledger Account	ALL	ALL
Common Chart of Accounts	ALL	ALL

Other Filters

Filter Type:  No Filter  Data Filter  Group Filter  Attribute Filter

For the already defined Variable Allocation rules, user should edit the allocations and modify the value of Legal Entity dimension in all applicable tabs in allocation specification screen.

Dimension	Value	Description
Financial Element	ALL	ALL
Organizational Unit	ALL	ALL
General Ledger Account	ALL	ALL
Common Chart of Accounts	ALL	ALL

Other Filters

The value of Legal Entity used when your rule executes is specified within your batch definition (for batch processes) or is obtained from your Profitability Management Applications Preferences (for interactive executions launched from a Summary screen). If this check box is not selected on the Initial Definition Process tab, then you must specify a value for Legal Entity in your allocation rule's Source, Driver, and Outputs.

**Note:** Legal Entity is designed to support implementations that require multi-entity or multi-tenant functionality. If your implementation does not require this functionality, you may utilize the Default Legal Entity in all your processes and you may declare all of your allocation rules to be Variable. For details, see [Appendix A, “Management Ledger”](#).

Legal Entity is not applicable to allocation rules using the Ledger/Stat table.

Process Tabs

Initial Definition → Source → Operator → Driver → Outputs → Review

Allocation Rule Definition

Rule Name: 26420257\_802010\_key\_only

Rule Description: [Empty]

Folder: AUTOSEG

Access Type:  Read Only  Read/Write

Variable Rule:

Allocation Type

Allocation Type	Description
<input type="radio"/> Constant	Allocates a constant value to a destination.
<input type="radio"/> Static Driver	Performs simple factor calculations.
<input type="radio"/> Leaf	Performs mathematical operations between two groups of data in the ledger table where the two sets differ by the value in a single selected dimension. The source and the By Leaf rows match exactly except for the value in a single dimension.
<input type="radio"/> Field	Performs mathematical operations across a row between two or more columns in Instrument or Transaction tables.
<input type="radio"/> Dynamic Driver	Distributes a balance amount specified by the initial data source condition to a set of specified target rows on a percent or percent-to-total basis.
<input checked="" type="radio"/> Static Driver Table	Uses the dimensionality of the source rows to join with data in a table or set of values. These rules perform either a mathematical operation with the returned value or a distribution and dimensional reclassification of the results.
<input type="radio"/> Lookup Driver Table	Processes source data by adding, multiplying, subtracting, or dividing by one or more statistics or lookup values stored in a user-defined lookup table.

Save Save As Cancel

### Allocation Type Container

When you initially build an Allocation rule, you must select its Allocation Type. Once an Allocation rule has been saved, you may no longer change its type. After you have chosen an Allocation Type on the Initial Definition process tab, the appearance of subsequent process tabs will depend upon the Allocation type you have chosen. The available rule types are Constant, Static Driver, Leaf, Field, Dynamic Driver, Static Driver Table, and Lookup Driver Table.

### Definitions of Static and Dynamic Drivers

Most Allocation rules distribute or aggregate balances using driver data. Examples include:

- Expense allocations as a function of square footage occupied or headcount
- Aggregation of instrument balances to the Management Ledger
- Reclassification of Management Ledger balances to dimensions not found in the original General Ledger data.

Drivers can be stored as components of your overall allocation model or they can be stored as facts within your business data. Headcount and square footage statistics, for example, are frequently stored as memo accounts within your General Ledger. When you load the OFSAA Management Ledger table with your General Ledger data, those headcount and square footage statistics may be utilized as drivers within your allocation rules. These kinds of business-data resident driver are referred to as Dynamic Drivers.

In other cases, you will embed your driver data into an Allocation rule or into a driver table that the allocation engine supports. These kinds of drivers are referred to as Static Drivers. Profitability Management supports three kinds of rules that use static drivers: Static Driver, Static Driver Table, and Lookup Driver Table.

Dynamic Drivers often have many advantages over Static Drivers. An Allocation rule that uses a static driver will take the same value or values every time you use it in a rule, but an Allocation rule that uses a dynamic driver may have different driver sets from day-to-day or from month-to-month. Additionally, whereas you must normally pre-compute your static drivers, dynamic drivers are generated at run time. Dynamic drivers, frequently used in full cost-absorption allocation models, are often generated by other allocation rules.

Static drivers sometimes have advantages over dynamic drivers. For example, you may have pre-computed unit costs that you wish to use to drive your allocations in order to generate partial absorption costing.

## **Allocation Types**

Allocation types are described in detail below.

### **Constant**

A Constant Allocation rule creates a simple balanced transaction consisting of one debit and one credit. You may optionally specify either one debit or one credit (at a minimum, you must supply at least one debit or one credit). The Constant rule type only operates against the Management Ledger. For Constant allocation rules, the Operator and Driver process tabs are disabled; you need only specify a fixed amount in the Source tab and a debit and/or credit in the Outputs process tab.

### **Static Driver**

The Static Driver method enables you to perform simple factor calculations against a set of source balances. Source balances may be drawn from the Management Ledger table, Instrument tables, or Transaction Summary tables. For Static Driver rules, the Driver process tab is disabled. For this kind of rule, you define where to get your source data on the Source process tab, a static driver amount on the Operator process tab, and the resulting debits and/or credits on the Outputs process tab.

### **Leaf**

Leaf type allocations are used only against the Management Ledger table. They are typically used to perform an operation between two sets of rows that differ in a single dimension.

### **Field**

A Field type allocation is typically used to multiply two columns within a single row in an instrument table update Allocation rule.

### **Dynamic Driver**

Typically, Dynamic Driver allocation rules aggregate or distribute balances using dynamic data (business resident driver data) such as headcount, square footage, or instrument-level balances. Dynamic Driver data, however, need not be limited to statistics you may have sourced as part of your ETL load to the OFSAA data model. Dynamic Driver data can be “captured” or developed within an Allocation rule. For example, balances by product within each cost center can normally be obtained from your instrument data. You can build allocation rules to aggregate these statistics from your instrument level data and post them to your Management Ledger for use in subsequent rules; or you can write an allocation that develops this set of driver data by querying your instrument data at runtime. Driver data obtained from your instruments need not be limited to balances. Examples of instrument level dynamic drivers you might use in allocation rules include:

- Number of accounts by product by cost center by year of origination
- Number of loan payments processed by loan processing center by month
- ATM transaction counts by region by month

The most common distribution method for the Dynamic Driver type of allocation is the “Percent-to-Total” or “Force to 100%” method. In addition to the Force to 100% method, Profitability Management also supports a Simple method and a Uniform method. Dynamic Driver allocation rules and methods are described in more detail in the sections entitled Driver Process Tab and Output Process Tab.

### **Static Driver Table**

Static Driver Table allocation rules offer functionality similar to Dynamic Driver allocation rules but use driver data that is stored in a Profitability Management rule type called Static Table Driver. See *Static Table Drivers* for a guide to building Static Table Driver rules and for additional explanation of their use.

### **Lookup Driver Table**

Lookup Driver Table allocation rules support multi-factor allocations at the Instrument level. In addition to supporting dimension based allocation methods, the Lookup Driver Table rule type also supports allocations based on instrument-level balances, rates, codes, or other instrument level measures. Typical use cases include allocation of loan loss reserve, economic provision, or risk equity first as a function of product, a dimension, but also as a function of instrument level measures such as loan-to-value ratio, credit score, loss in event of default, probability of default, remaining term to maturity, etc.

The driver data used in a Lookup Driver Table allocation rule is stored in user-defined lookup tables that are registered into the OFSAA data model. The data from your user-defined lookup table is matched to Instruments tables. The logic governing how your lookup data is matched to Instrument tables is stored in a Lookup Table Driver rule. Lookup Driver Table Allocation rules incorporate Lookup Table Driver rules in the Allocation Specification > Driver process tab.

See *Lookup Table Drivers* for a guide to building Lookup Table Driver rules and additional explanation of their use.

### **Source Process Tab**

Use the Source process tab to specify an Allocation rule's data source. For an Allocation of the Constant type, you simply specify an amount. For any other type of rule, you must specify a source table & column and, optionally, a set of constraints. Alternatively, you may specify an Expression to specify a more complex Source than a simple source table and column.

For Container

Choose a source table and then choose either a column from that table or choose an Expression rule.

### **Management Ledger Source**

The management ledger-level refers to both the Management Ledger table and the Ledger/Stat table. This section describes the usage of the Management Ledger as the Source in allocation rules. The next section describes the usage of Ledger/Stat as the Source in allocation rules. New in release 8, the Management Ledger is a seeded table of a new Management Ledger class of tables (see Oracle Financial Services Analytical Applications Data Model Utilities User Guide for details on adding user-defined dimensions to the Management Ledger or for defining new user-defined Management Ledger tables). The Management Ledger table supports nearly identical allocation functionality as the Ledger/Stat table, but is intended to replace the Ledger/Stat table in a future release. Customers starting new implementations are strongly encouraged to use the new Management Ledger table for ledger-level management accounting. For details on the new Management Ledger table, see [Appendix A, “Management Ledger”](#).

When your Source is the Management Ledger table, you will typically use the <Current Amount> macro as your column name. The Management Ledger table is the default table for new allocation rules (except for the Constant type), and <Current Amount> is the default column. The <Current Amount> macro selects the current month from your Management Ledger on the basis of your As-of-Date and Fiscal Year definitions. If your As-of-Date is set to any day in the month of March, <Current Amount> will generally be interpreted as Fiscal Month number 3. If your fiscal year begins in April, however, your March data is stored in the Management Ledger under Fiscal Month

number 12 since March is the last month in your fiscal year. For details on fiscal year, see [Appendix E, “Fiscal Year Information”](#).

**Note:** Allocation rules that aggregate instrument level data to the Management Ledger table maintain literal As-of-Dates when posting to the management ledger, but when data is retrieved from Management Ledger table (i.e., in Allocation Sources or Allocation Drivers), it is consolidated into a month-to-date balance. For example, if you perform daily instrument level Funds Transfer Pricing and if you use an allocation rule to aggregate daily Funds Transfer Pricing Charges or Credits from the instrument level to the Management Ledger, each day's charges and credits will be posted to the Management Ledger by As-of-Date (by business date). Allocation rules that reference these Management Ledger balances, however, combine the daily postings to obtain month-to-date balances. Similarly, all outputs at the management ledger-level are inherently month-to-date balances.

When your Source is the Management Ledger table, Profitability Management also supports the following macros:

- <Last\_Mo\_Amount>
- <Months\_Ago\_Amt>
- <Current\_Amount>
- <YTD\_Amount>
- <Months\_Ago\_YTD\_Amt>
- <Last\_Mo\_YTD\_Amt>
- <Accrual\_Basis>

<Last Mo\_Amount> selects month-to-date balances from the month prior to your As-of-Date.

<Months\_Ago\_Amount> selects month-to-date balance as of a designated number of months ago. For example, with a typical January to December fiscal year, if today's As-of-Date is March 31, 2015 (Fiscal Month = 3, Fiscal Year = 2015) then 6 months ago in the Enter Months field. This corresponds to September, 2014 (Fiscal Month = 9, Fiscal Year = 2014). The month range for Enter Months is from -99 to 999. For details on fiscal year, see [Appendix E, “Fiscal Year Information”](#).

The screenshot displays the 'Allocation Specification Definition' window, specifically the 'Source' configuration page. The interface includes a process flow diagram at the top with steps: Initial Definition, Source (highlighted), Operator, Driver, Outputs, and Review. Below this, the 'Allocation Rule Definition' section shows 'Rule Name' as 'Test', 'Rule Description' as a text area, 'Folder' as 'AUTOSEG', and 'Access Type' as 'Read/Write'. The 'For' section is configured with 'Table Name' as 'Management Ledger', 'Column/Expression' as 'Column Name', and 'Column Name' as '[Months\_Ago\_YTD\_Amt]'. The 'Enter Months' field is empty. At the bottom, the 'Allocation Source' table lists dimensions and their values:

Dimension	Value	Description
Financial Element	ALL	ALL
Organizational Unit	ALL	ALL
General Ledger Account	ALL	ALL
Common Chart of Accounts	ALL	ALL

## Ledger Stat Source

When your Source is the Ledger/Stat table, you will typically use the <Current Amount> macro as your column name. Except for rules of the Constant type, <Current Amount> is the default column when you build a new rule invoking Ledger/Stat. The <Current Amount> macro selects the current month from Ledger/Stat on the basis of your As-of-Date and Fiscal Year definitions. If your As-of-Date is set to any day in the month of March, <Current Amount> will generally be interpreted as MONTH\_03. If your fiscal year begins in April, however, your March data is stored in Ledger/Stat as MONTH\_12 since March is the last month in your fiscal year. For details on fiscal year, see [Appendix E, “Fiscal Year Information”](#).

In addition to <Current Amount>, Profitability Management also offers macros for:

<YTD Amount>

<Last Month Amount>

<Last Month YTD Amount>

You may also select explicit column names from Ledger/Stat for month end values and YTD values (MONTH\_01 to MONTH\_12 and YTD\_01 to YTD\_12).

## Instrument or Transaction Summary Source

When your Source is an Instrument or Transaction Summary table, you may choose any valid measure in the table. Valid measures include only rates, balances, and numeric statistics such as activity counts.

If you want to simultaneously apply an allocation rule to a group of Instrument tables, select Portfolio Table as Table Name in the For container. After selecting the Portfolio table, a new container called Portfolio Table Details is presented. From this container you may select one, some, or all of the instrument tables that are defined in your environment. Usage of the Portfolio Table is supported for all allocation types except Constant and Leaf.

To select the Instrument tables, follow the steps given below:

1. Select the Table Name as **Portfolio Table**. The **Portfolio Table Details** container will be added, that shows the list of all Instrument tables.



For

Table Name \* Portfolio Table

Column/Expression  Column Name  Expression

Column Name Average Gross Book Balance

Portfolio Table Details

Instrument Table Name
Annuity Contracts
Borrowings
Break Funding Charges
Checking and Savings Accounts

## 2. Click **Select Table**.

Portfolio Table Name

Table Name Portfolio Table

Table Selector

Available Tables

Name
Annuity Contracts
Borrowings
Break Funding Charges
Checking and Savings Accounts
Credit Cards
Credit Lines
Guarantees
Investments
Leases
Ledger Stat Instrument
Loan Contracts
Merchant Cards
Money Market Contracts

Selected Tables

Name
Annuity Contracts
Borrowings
Break Funding Charges
Checking and Savings Accounts
Credit Cards
Credit Lines
Guarantees
Investments
Leases
Ledger Stat Instrument
Loan Contracts
Merchant Cards

Ok Cancel

## 3. Select the Instrument Tables from Table Selector and click **Ok**.

### **Expression Container**

When choosing an Expression to act as the source for your Allocation rule, the expressions available to you are limited to those built upon the table you selected in the For container.

**Note:** Expressions may not be used with Management Ledger tables but will be supported in a future release.

For

Table Name \* Portfolio Table

Column/Expression  Column Name  Expression

Portfolio Table Details

Instrument Table Name
Annuity Contracts
Borrowings
Break Funding Charges
Checking and Savings Accounts

Expression

Folder AUTOSEG

Expression AUTOD\_Expression\_01

### **Allocation Source Container**

The Allocation Source Container is used to provide dimensional constraints on your Source data. For any dimension, you may constrain your source data by selecting a leaf member, a rollup member within a Hierarchy, or a Hierarchy Filter.

At least one leaf level dimension member is required for hierarchies used in allocations.

For allocation rules that source data from the Management Ledger-level, you must select a Source scenario from the Allocation Source container. The default for new allocation rules is <Use Application Preferences>. When you use this default value, Scenario (also called Consolidation Code) is determined by the value that is set in Application Preferences for Profitability Management for the user who is running the rule. If you do not select <Use Application Preferences>, you must select a defined dimension member value (for example, Actual, Budget, Forecast, Forecast Prior). These values are provided with the data model, but you may add additional dimension members in the Consolidation Code dimension.

The Source value for Scenario may also be provided as a task parameter if the rule is executed within:

- a Simplified Batch,
- a standard ICC batch, or
- from the command line

When the Source value for Scenario is passed as a task parameter, it overrides the value set in Application Preferences. For details on passing task parameters, see *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

The screenshot displays the 'Allocation Rule Definition' interface. At the top, a 'Process Tabs' section shows a workflow: Initial Definition → Source → Operator → Driver → Outputs → Review. The 'Allocation Rule Definition' section includes fields for Rule Name (26420257\_802010\_key\_only), Rule Description, Folder (AUTOSEG), Access Type (Read Only, Read/Write), and Variable Rule. The 'For' section shows Table Name (Portfolio Table), Column Name (Average Gross Book Balance), and options for Column/Expression (Column Name, Expression). Below this is a 'Portfolio Table Details' list with items like Instrument Table Name, Annuity Contracts, Borrowings, Break Funding Charges, and Checking and Savings Accounts. The 'Allocation Source' section contains a table with columns for General Ledger Account, Common Chart of Accounts, Product, Legal Entity, and Customer, all set to 'ALL'. At the bottom, there are 'Other Filters' with radio buttons for No Filter, Data Filter, Group Filter, and Attribute Filter, and 'Save', 'Save As', and 'Cancel' buttons.

Some examples of how the Source tab is used to provide input data to an allocation rule include:

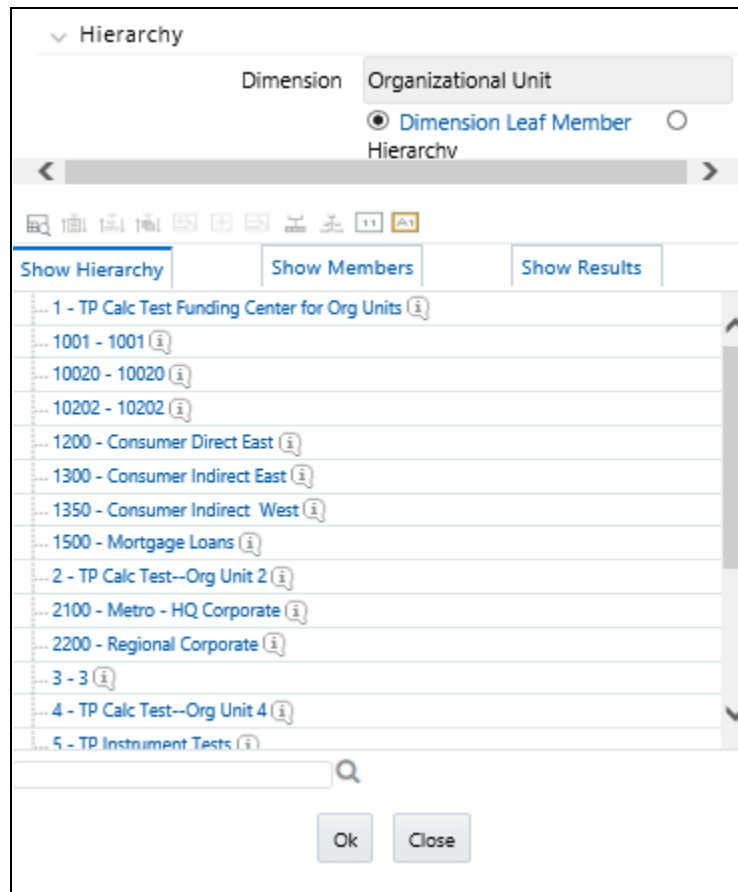
Desired Data	Constraint
Get all expenses for all GL Accounts within a specific cost center	Single leaf constraint on Organizational Unit plus single leaf constraint on Financial Element 457 – Non Interest Expense
Get all current mortgage balances for adjustable rate products originated in the past year	Hierarchy member constraint on the Product dimension plus a Data Filter constraint
Get all initial General Ledger balances plus all allocated balances for a specified set of cost centers for one GL Account	Single leaf constraint on General Ledger Account plus a Hierarchy member constraint on Organizational Unit
Get ending balances for all balance sheet assets for the North, South, and East divisions (but not the West division except for the South-West sub-region)	Single leaf constraint on Financial Element plus Hierarchy member constraint on the GL Account dimension plus a Organizational Hierarchy Filter

The dimensions listed in the Allocation Source container are limited to your Key Processing Dimensions. Financial Element, Organizational Unit, General Ledger Account, Common Chart of Accounts, Legal Entity, and Product are seeded Key Processing Dimensions for all OFS Analytical Applications. You may add your own Key Processing Dimensions (see *Installation & Configuration Guide* for guidance on adding your own processing dimensions). If you have enabled multicurrency, the Currency dimension will also be displayed. For details on multicurrency calculations within Profitability Management, see the [Appendix D, “Multicurrency”](#).

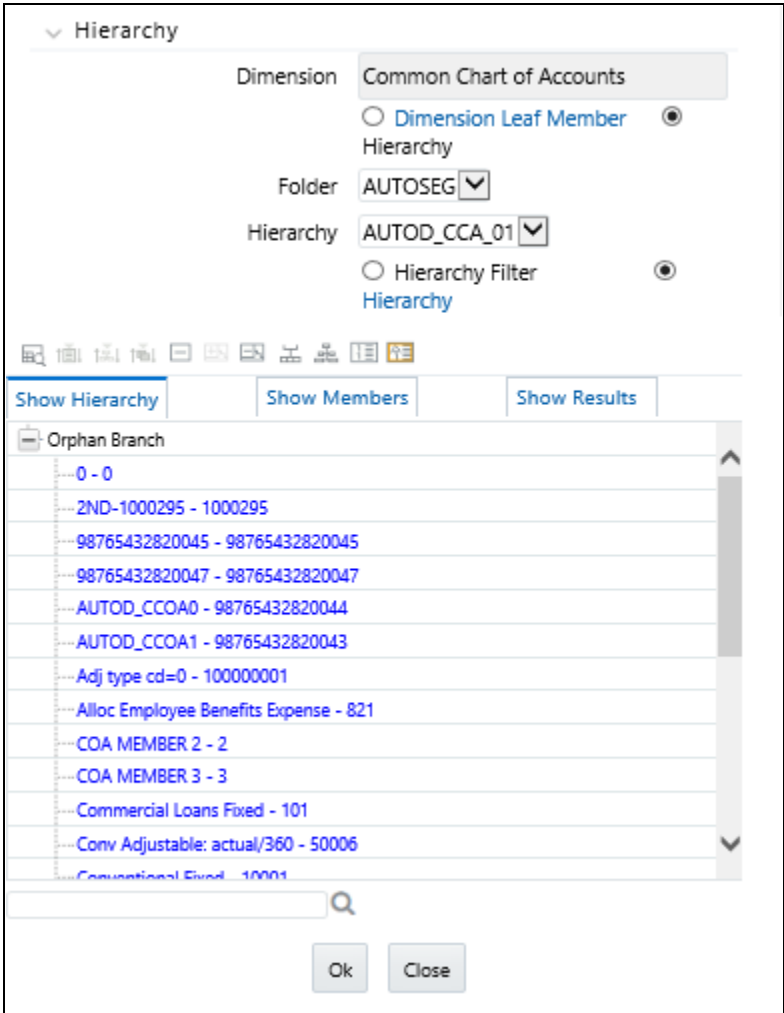
**Note:** Legal Entity is not supported for Ledger/stat allocations.

### ***Hierarchy Browser on the Source Tab***

The default for each dimension is <All> (meaning no constraint). To select a constraint, click on the Hierarchy Browser ellipses (...) next to the dimension you wish to constrain.



The Hierarchy Browser defaults to a list of leaves for the dimension you have chosen (the radio button selector near the top of the browser window defaults to Dimension Leaf Member). You may scroll up and down to find the leaf member you want or you may search for the member's name (short description) using the binoculars at the bottom of the browser window. Additional search functionality is provided by the Search control within the browser. This additional functionality allows you to search by Dimension Member Dimension member, Name, or Code.



To constrain your source using a hierarchy rollup point, click the Hierarchy radio button near the top of the browser window and then search for the hierarchy you wish to use. Once your chosen hierarchy is displayed within the browser window, navigate into the hierarchy until you have found the rollup point or leaf value you want. Click on your selected value and then click OK.

Hierarchy	
Dimension	Organizational Unit
<input type="radio"/> Dimension Leaf Member <input checked="" type="radio"/> Hierarchy	
Folder	GASEG
Hierarchy	Standard Cost Center Hierarchy
<input checked="" type="radio"/> Hierarchy Filter <input type="radio"/> Hierarchy	
Folder	GASEG
Hierarchy Filters	Origination and Processing
<input type="button" value="Ok"/> <input type="button" value="Close"/>	

To constrain your source using a Hierarchy Filter, first select the Hierarchy radio button near the top of the browser window and then search for the hierarchy that supports the Hierarchy Filter you wish to use. Next, click on the Hierarchy Filter radio button near the top of the browser window, search for the Hierarchy Filter you wish to use, and then click OK.

### **Other Filters Container**

You may optionally select a Data Filter, Group Filter or Attribute filter rule to further constrain your Source data.

### **Operator Process Tab**

Generally, the Operator process tab allows you to specify how Source data and Driver data will interact to create results.

#### **Operator Process Tab for Constant Rules**

No Driver is necessary in the specification of a Constant rule. Both the Operator process tab and the Driver process tab are disabled for the Constant rule type.

#### **Operator Process Tab for Static Driver Rules**

For Static Driver rules, the Driver process tab is disabled, but the Operator process tab is enabled in order to allow you to specify a static driver balance. Static amounts are entered into the Factor Operator container.

Allocation Specification Edit  
Allocation Specification (Edit) > Operator

Process Tabs

Initial Definition → Source → **Operator** → Driver → Outputs → Review

Allocation Rule Definition

Rule Name \* 26420257\_802010\_key\_only  
Rule Description  
Folder: AUTOSEG Access Type:  Read Only  Read/Write Variable Rule

Factor Operator

Type:  None  Accrual Basis  Constant  Both

Allocation Operator

Arithmetic Operator \*

Save Save As Cancel

### Operator Process Tab for All Other Rule Types

For all other rule types, the Operator process tab offers both a Factor Operator and an Allocation Operator. The Allocation Operator links the Allocation rule's Source Data with its Driver data. The Factor Operators may be interposed between the Source and Driver.

### Factor Operator Container

Factor Operators may be used to either

- Store static driver amounts and/or accrual basis macros for Static Driver rule types
- Interject constant values and/or accrual basis macros between allocation Sources and Drivers for Leaf, Field, Dynamic Driver, Static Driver Table, or Lookup Driver Table rule types

The Factor Operator allows you to modify Source data by adding, subtracting, multiplying, or dividing Source data by a constant amount, an accrual basis macro, or both.

Allocation Specification Edit  
Allocation Specification (Edit) > Operator

Process Tabs

Initial Definition → Source → **Operator** → Driver → Outputs → Review

Allocation Rule Definition

Rule Name \* 26420257\_802010\_key\_only  
Rule Description  
Folder: AUTOSEG Access Type:  Read Only  Read/Write Variable Rule

Factor Operator

Type:  None  Accrual Basis  Constant  Both  
Arithmetic Operator  Accrual Basis 30/360

Allocation Operator

Arithmetic Operator \*

Save Save As Cancel

### Examples of Usage of the Factor Operator

- Instrument level rate times balance allocations commonly use the “Both” type factor operator in which the first factor operator is “times <accrual-basis> macro” and the second factor operator is “divided by 100” when posting to a monthly income or expense balance.
- In the above example, if you were to choose a 30/360 accrual basis factor, you could equally well specify your factor operator as “divide by 1200”.
- Instrument level rate times balance allocations can also utilize actual instrument level accrual bases instead of applying the same accrual basis to every calculation.
- In a typical percent distribution allocation such as “distribute all Human Resource expense to all cost centers as a function of headcount”, you may sometimes want to distribute less than 100% of total expense. In this example, your Source data would be “all Human Resource expense”, your Driver data would be “head count by cost center” on a percent-to-total basis, and your Factor Operator would be whatever percentage of total expense you are choosing to allocate.

### Allocation Operator Container

For all allocation types except Constant and Static Driver, the Allocation Operator links the Allocation rule's Source Data with its Driver data. The most common form of linkage is multiplication, but both multiplication and division are supported. For some allocation types, addition and subtraction are also supported.

The screenshot displays the 'Allocation Specification Edit' window, specifically the 'Operator' tab. At the top, a process flow diagram shows the sequence: Initial Definition → Source → Operator (highlighted) → Driver → Outputs → Review. Below this, the 'Allocation Rule Definition' section includes fields for Rule Name (26420257\_802010\_key\_only), Rule Description, Folder (AUTOSEG), Access Type (Read Only and Read/Write), and Variable Rule. The 'Factor Operator' section shows Type set to Accrual Basis, Arithmetic Operator as '/', and Accrual Basis as 30/360. The 'Allocation Operator' section shows an Arithmetic Operator set to '\*'. At the bottom, there are Save, Save As, and Cancel buttons.

### Driver Process Tab

The Driver process tab allows you to specify a set of driver data that is combined with Source data to create allocation outputs. The way in which Source data and Driver data interact is a function of the type of Allocation rule you are using and the nature of the Operator you have specified.

The Driver process tab is enabled for Leaf, Field, Dynamic Driver, Static Driver Table, and Lookup Driver Table allocation types. The containers displayed on the Driver process tab vary from rule type to rule type.



## For Container

For the Leaf, Field, and Dynamic Driver allocation types, a For container is displayed in which you can choose either a driver table & column or an Expression to serve as the source of your Driver data.

### Leaf Driver Container

The Leaf Driver container is only displayed for the Leaf allocation type. Use this container to specify the dimension and leaf value you wish to use.

The Legal Entity dimension is not applicable to allocations that use the Ledger Stat table.

The screenshot displays the 'Allocation Specification Definition' window. At the top, a 'Process Tabs' bar shows a sequence of steps: Initial Definition, Source, Operator, Driver (highlighted), Outputs, and Review. Below this, the 'Allocation Rule Definition' section includes fields for Rule Name (new), Rule Description, Follower (PFTUISES), Access Type (Read/Write, Read Only), and Variable Rule. The 'For' section is configured with Table Name 'Ledger Stat' and Column Name '[Current\_Amount]'. The 'Leaf Driver' section shows a dropdown menu for 'Leaf Type' with 'Financial Element' selected, and a 'Leaf Value' field.

### Dynamic Driver Container

For the Dynamic Driver allocation type, use this container to supply dimensional constraints for your driver data. You supply dimensional constraints using the hierarchy browser. From the Dynamic Driver container, the behavior of the hierarchy browser is identical to the hierarchy browser described above under “Allocation Source Container”.

Allocation Specification Definition

Allocation Specification (New) > Driver

Process Tabs: Initial Definition, Source, Operator, Driver, Outputs, Review

Allocation Rule Definition

Rule Name: new

Rule Description:

Folder: PFTLUSE0 Access Type: Read/Write, Read Only Variable Rule:

For

Table Name: Management Ledger Column Expression: Expression Column Name: [Current\_Amount]

Distribution Type

Distribution Type:  Simple  Percent Distribution  Uniform

Dynamic Driver

Dimension	Value	Description
Financial Element	ALL	ALL
Organizational Unit	ALL	ALL
General Ledger Account	ALL	ALL
Common Chart of Accounts	ALL	ALL
Product	ALL	ALL
Legal Entity	ALL	ALL
Scenario	Use Application preferences	Use Application preferences

Other Filters

Filter Type:  No Filter  Data Filter  Group Filter  Attribute Filter

Save Cancel

For Dynamic Driver allocation rules that obtain their driver data from the Management Ledger-level, you must also select a Driver scenario. The default for new allocation rules is <Use Application Preferences>. When you use this default value, Scenario (also called Consolidation Code) is determined by the value that is set in Application Preferences for Profitability Management for the user who is running the rule. If you do not select <Use Application Preferences>, you must select a defined dimension member value (for example, Actual, Budget, Forecast, Forecast Prior). These values are provided with the data model, but you may add additional dimension members in the Consolidation Code dimension.

Given below are the limitations for User Defined Consolidation Code dimension:

- Code name must be a string with alphabetic characters alone (either lower case or upper case).
- Code should not have digits, special characters, blank spaces, or punctuation characters.
- Code should not have the word 'adhoc' in it.

The Driver value for Scenario may also be provided as a task parameter if the rule is executed within:

- a Simplified Batch,
- a standard ICC batch, or
- from the command line

When the Source value for Scenario is passed as a task parameter, it overrides the value set in Application Preferences. For details on passing task parameters, see *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

### **Distribution Type Container**

The Distribution Type container is only displayed for the Dynamic Driver allocation type. Use this container to select the Simple, Percent Distribution, or Uniform distribution method.

Allocation Specification (New) > Driver

Allocation Specification Definition

Process Tabs

Initial Definition → Source → Operator → Driver → Outputs → Review

Allocation Rule Definition

Rule Name: AllocationRule

Rule Description:

Folder: PFTUISECO Access Type: Read/Write Read Only: Variable Rule:

FOR

Table Name: Ledger Stat Column/Expression: Expression Column Name

Column Name: [Current\_Amount]

Distribution Type

Distribution Type:  Simple  Percent Distribution  Uniform

Dynamic Driver

Dimension	Value	Description
Financial Element	ALL	ALL
Organizational Unit	ALL	ALL
General Ledger Account	ALL	ALL
Common Chart of Accounts	ALL	ALL
Product	ALL	ALL
Scenario	Use Application preferences	Use Application preferences

Other Filters

Filter Type:  No Filter  Data Filter  Group Filter  Attribute Filter

Save Cancel

## Percent Distribution Method

The most common distribution method is Percent Distribution (sometimes referred to as Force to 100%). Typical use cases include expense allocations as a function of a driver set that has not been normalized, i.e., converted to percentages of the total driver set. For example, if you wanted to distribute some expense balance to Departments 1, 2, and 3, and if Departments 1, 2, and 3 had headcounts of 100, 200, and 700, you would choose the Percent Distribution method to allocate 10% ( $100/1,000$ ) to Department 1, 20% ( $200/1,000$ ) to Department 2, and 70% ( $700/1,000$ ) to Department 3.

## Simple Method

Use the Simple distribution method in cases where your dynamic drivers are stored as percentages. You might also use the Simple distribution method if your allocation source data were activity counts and your driver data represented unit costs.

### Uniform Method

Use the Uniform distribution method in cases where you want to allocate equal shares of your source data for each destination in your driver set regardless of driver amount. Continuing with the above headcount example, you may wish to allocate equal shares of 10% of total Human Resource department expense to any Department having non-zero headcount. In this case, you would use Human Resource department expenses as your allocation Source, you would specify a Factor Operator of 10%, you would specify your “Headcount by Cost Center” statistic set as Driver, and you would select the Uniform distribution method. Statistical driver sets are frequently stored in the Management Ledger under user-defined Financial Elements.

### Static Table Driver Container

For the Static Driver Table allocation type, specify a Folder and select a Static Table Driver rule. By default, the Distribution Type is set to Force to 100%. For a discussion of their use including examples, see Using Static Table Drivers in [Chapter 18, "Static Table Driver,"](#).

Static Table Driver			
Folder	GASEGR2	Static Table Driver	TEST_STATIC
Static Table Driver Description		Force To 100%	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No

Note that once you have chosen a Static Table Driver rule, a View control is added to the Static Table Driver title bar. Click on this View control to view a read-only version of the Static Table Driver rule you have chosen.

### Lookup Driver Table Container

For the Lookup Driver Table allocation type, specify a Folder and select a Lookup Driver Table rule. Note that the drop down list box for Lookup Table Drivers is limited to Lookup Table Drivers whose source tables match the table specified in the current Allocation rule's Source definition. For a discussion of their use including examples, see [Chapter 19, "Lookup Table Driver,"](#).

Lookup Table Driver			
Folder	GASEGR2	Lookup Table Driver	TEST_LOOKUP
Lookup Table Driver Description			

Note that once you have chosen a Lookup Table Driver rule, a View control is added to the Lookup Table Driver title bar. Click on this View control to view a read-only version of the Lookup Table Driver rule you have chosen.

### Other Filters Container

For the Field and Dynamic Driver allocation types, you may optionally select a Data Filter, a Group Filter or an Attribute Filter rule to further constrain your driver data.

### Outputs Process Tab

The Outputs process tab allows you to specify where the outputs of an Allocation rule are written. When outputting to the Ledger/Stat table, the allocation engine creates Management Ledger debits and/or credits. When outputting to Instrument or Transaction Summary tables, the allocation engine updates target columns.

## Source-Driver Relationship

This container appears in the **Outputs** tab only when the **Allocation Type** is set as **Leaf Allocation** in **Initial Definition** tab.

The screenshot shows the 'Allocation Rule Definition' window. The 'Rule Name' is 'Leaf/AllocationType'. The 'Folder' is 'PFTUSEG'. The 'Access Type' is 'Read/Write'. The 'Variable Rule' checkbox is unchecked. The 'Source-Driver Relationship' section has a dropdown menu for 'Rows to Include' with the following options: 'Include All Rows', 'Include Rows Found Only in Driver', 'Include All Rows', 'Include Only Rows Found in Both Source & Driver', and 'Include Rows Found Only in Source'. The 'Debit' and 'Credit' tabs are visible at the bottom left of the relationship section.

There are four option available:

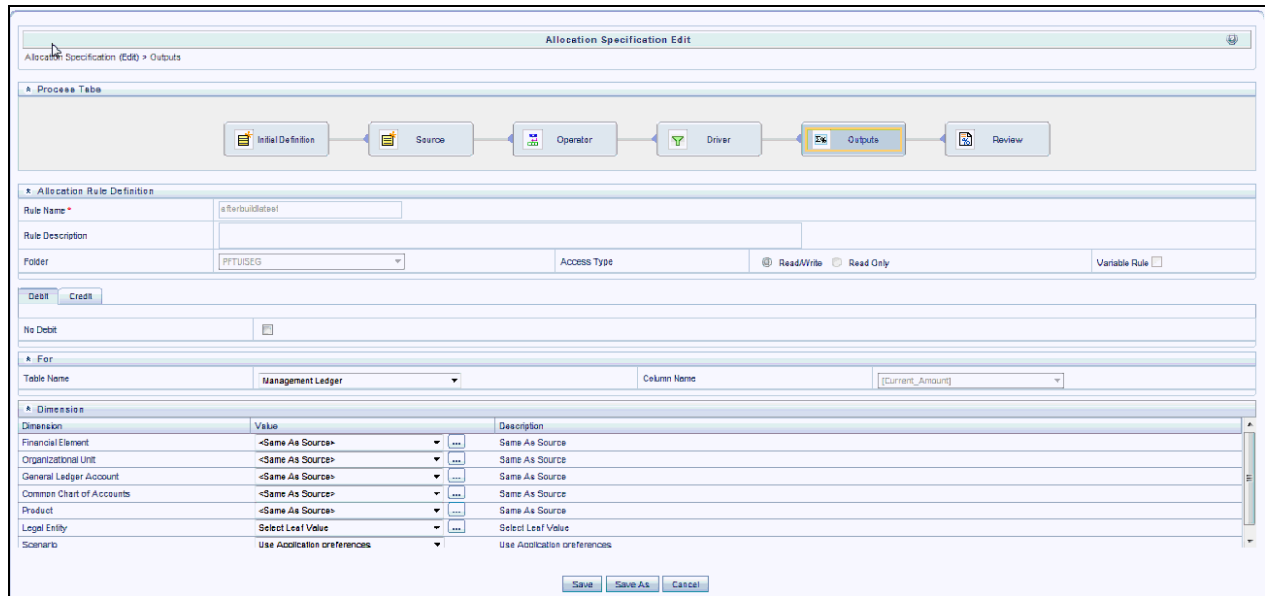
- **Include Rows Found Only in Driver:** When you select this option, the output will include the rows which are available only in Driver.
- **Include Only Rows Found in Both Source & Driver:** When you select this option, the output will fetch the rows from both source and driver based on defined condition(s).
- **Include All Rows:** When you select this option, the output will include all the rows which are available in both Driver and source.
- **Include Rows Found Only in Source:** When you select this option, the output will include the rows which are available only in Source.

## Debit/Credit Tab

Both Debit definitions and Credit definitions are built and maintained on the Outputs process tab. Within the Outputs process tab, the Debit/Credit tabs allow you to navigate back and forth between a rule's Debit definition and its Credit definition. You may also use the Debit/Credit tabs to suppress the output of either Debits or Credits, but you may not suppress the output of both Debits and Credits.

### When Outputting to the Management Ledger-level

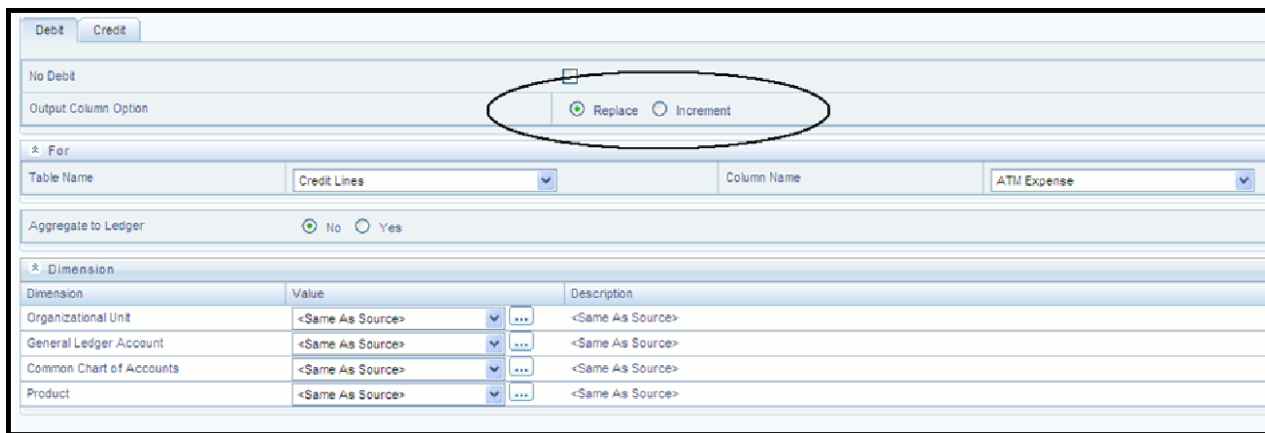
When posting allocation results to the Management Ledger-level, the Profitability Management allocation engine generates a balanced accounting transaction consisting of multiple debits and credits. As few as one debit or credit may be generated, or you might generate thousands of debits and credits.



**When the Output Table is Instrument Level**

When using an allocation rule to update an Instrument or Transaction Summary table, the Profitability Management allocation engine updates your chosen output column for each instrument level account that is found in your Source and for which a matching Driver is found.

When outputting to Instrument or Transaction Summary tables, you may choose to either Replace or Increment your target column values. The default behavior for Allocation rules built prior to release 6.0 is Replace.



**For Container**

Specify the output table and column for the Allocation rule.

- You may only output to < Current Amount > when posting allocation results to the Management Ledger-level.
- For Constant and Leaf type rules, you may only output to the Management Ledger-level.
- For Lookup Driver Table type rules, you may only output to an Instrument or Transaction Summary table.

## Aggregate to Ledger

For Allocation rules that update an Instrument or Transaction Summary tables, you have the option of aggregating your results and posting them to the Management Ledger-level.

The screenshot shows the 'Allocation Specification Definition' form. The 'Aggregate to Ledger' option is set to 'Yes' and is circled in red. The 'Table Name' is 'Credit Lines' and the 'Column Name' is 'ATM Expense'. The 'Dimension' section shows various fields set to '<Same As Source>'. The 'Output Column Option' is set to 'Replace'.

For Allocation rules that update an Instrument or Transaction Summary tables, you have the option of aggregating your results and posting them to the Management Ledger, or Ledger Stat table. To do this, select the Aggregate to Ledger option as Yes and select **Ledger Stat** or **Management Ledger** from the **Ledger Table Name** drop-down list.

The screenshot shows the 'Allocation Specification Definition' form with the 'Process Tabs' section. The 'Output' tab is selected. The 'Aggregate to Ledger' option is set to 'Yes'. The 'Ledger Table Name' dropdown menu is open, showing 'Management Ledger', 'Ledger Stat', and 'Management Ledger' as options. The 'Table Name' is 'Credit Cards' and the 'Column Name' is 'ATM Expense'. The 'Dimension' section shows various fields set to '<Same As Source>'. The 'Output Column Option' is set to 'Replace'.

**Note:** From 8.0.1 release onwards, Lookup Driver Table type allocation rules are capable of outputting to the Management Ledger-level.

## **Dimension Container**

The Dimension container is displayed for every allocation type. The default value for each dimension is generally <Same as Source>. You may choose a specific dimension member value for any dimension for both debits and credits for any allocation type.

## **Output Scenario**

For allocation rules posting to the Management Ledger-level, you must select an output scenario. The default for new allocation rules is <Use Application Preferences>. When you use this default value, Scenario (also called Consolidation Code) is determined by the value that is set in Application Preferences for Profitability Management for the user who is running the rule. If you do not select <Use Application Preferences>, you must select a defined dimension member value (for example, Actual, Budget, Forecast, Forecast Prior). These values are provided with the data model, but you may add additional dimension members in the Consolidation Code dimension.

The output value for Scenario may also be provided as a task parameter if the rule is executed within:

- a Simplified Batch,
- a standard ICC batch, or
- from the command line

When the output value for Scenario is passed as a task parameter, it overrides the value set in Application Preferences. For details on passing task parameters, see *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

## **Specific Leaf Value vs. <Same as Source> Macro**

For allocation rules posting to the Management Ledger-level, <Same as Source> for a particular dimension means that for that dimension, the values found in Source records are passed directly to Output records. For example, you might want to allocate 100% of the expenses from one department to a second department. In your original General Ledger data, expenses for the Source cost center might be posted under hundreds of different General Ledger accounts. In this example, you might specify your Credit Output (expense allocation offset) as:

- <Same as Source> for the Organizational Unit dimension
- <Same as Source> for the General Ledger Account dimension

Written in this fashion, your allocation rule will generate a credit to the original department for every original expense balance. If the source department contained balances under 81 different General Ledger accounts, the allocation rule would generate 81 credit records.

Continuing, you might specify your Debit Output as:

- Target Department (leaf value) for the Organizational Unit dimension
- “Allocated Expense” (leaf value) for the General Ledger Account dimension

In this example, Allocated Expense is a user-defined General Ledger account. You would typically define this dimension member in a reserved range of accounts for use in your Profitability Management model. Note that only one debit row is created in this scenario. Also note: When you choose to output to a specific leaf value, you may not output to a node value. Outputting to node values is not supported.

Other Output Macros

Other output macros include



- <Same as Driver>
- <Match Source & Driver>
- <Same as Table>

The usage of these macros is discussed in the sections below entitled Dynamic Driver Allocation Type, Static Driver Table Allocation Type and Lookup Driver Table Allocation Type.

### Constant Allocation Type

For Constant type allocations, you must specify a target leaf value for each of your processing dimensions for both your debit and your credit. You may optionally suppress either the Debit or the Credit.

### Static Driver Allocation Type

For Static Driver allocations, you may choose either a specific dimension member value or the <Same as Source> macro for each of your processing dimensions.

### Leaf Allocation Type

For Leaf allocations, you may choose either a specific dimension member value or the <Same as Source> macro for each of your processing dimensions.

### Field Allocation Type

For Field allocations, you may choose either a specific dimension member value or the <Same as Source> macro for each of your processing dimensions.

### Dynamic Driver Allocation Type

For Dynamic Driver allocations, for each of your processing dimensions you may choose either a specific dimension member value or

- <Same as Source>
- <Same as Driver>
- <Match Source & Driver>

At least one dimension in either your Debit or Credit specification must be either <Same as Driver> or <Match Source & Driver>.

### Same as Driver Macro

The <Same as Driver> macro is used when you want your outputs to inherit values from your driver data. For example, you might want to build an allocation rule to distribute some kind of processing expense to branches using “Number of Checks Processed per Branch” as your driver statistic set. In this example, your statistics “drive” your processing expense to branches so you would want specify <Same as Driver> in the Organizational Unit dimension of your Debit definition. Since our example is an expense allocation, you might want to construct a Credit definition using <Same as Source> in every dimension.

### Match Source & Driver Macro

The <Match Source & Driver> macro is used when you want to distribute data to one dimension while holding another dimension constant. For example, you might want to build an allocation rule that allocates a Human Resource expense cost pool as a function of headcount but that also allocates an Occupancy Expense cost pool as a function of square footage occupied. If your destinations are cost centers, you can store your statistic sets – one for headcount and one for square footage – on a per Cost Center per Cost Pool basis. In this example, you would use

<Match Source & Driver> on the Cost Pool dimension and <Same as Driver> on the Organizational Unit dimension. This causes the rule engine to create two sets of Debits to Cost Centers:

- One set of Debits from the Human Resource expense Cost Pool using the headcount statistics
- One set of Debits from the Occupancy expense Cost Pool using the square footage occupied statistics

### Static Driver Table Allocation Type

For Static Driver Table allocations, for each of your processing dimensions you may choose either a specific leaf value or you may choose

- <Same as Source>
- <Same as Table>

When you choose <Same as Table> for a dimension, you are telling the allocation engine that you want your rule to inherit its destination dimension member values from the Static Table Driver.

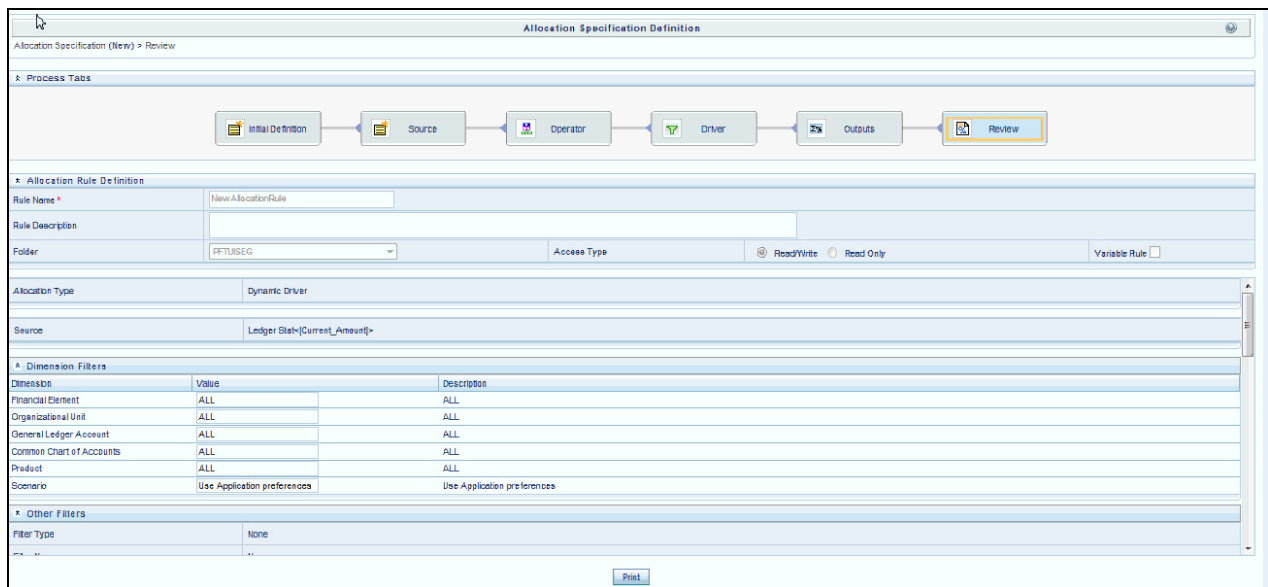
### Lookup Driver Table Allocation Type

For Lookup Driver Table allocations, you may choose either a specific leaf value or the <Same as Source> macro for each of your processing dimensions.

## Review Process Tab

The Review process tab displays a single page, printable report of an Allocation rule's specification.

The review tab shows the dimension values for source, and debit/credit output.



## Allocation Examples

### Examples of Static Driver Allocation Rules

The examples are explained in detail below:

#### From Management Ledger to Management Ledger

For Static Driver allocation rules, Management Ledger-to-Management Ledger is a common use case.

- Allocate 15% of the occupancy expense from one cost center to another cost center. In this example, the static driver is 15%.
- Create a cost pool by aggregating 25% of the expense found under a select group of General Ledger accounts for a region or a division or a department, or a single cost center. In this kind of aggregation, the static driver is 25%.
- Transfer 100% of loan assets from all of the loan origination centers within a region to a regional holding center. In this example, the static driver is 1.

---

**Note:** While such allocations are relatively common, it is more likely when you have a series of such allocations to utilize Static Driver Table rules. Using a Static Driver Table rule, you can accomplish with a single rule what might otherwise require dozens or even hundreds of Static Driver allocation rules.

---

#### From Instrument to Management Ledger

Instrument-to-Management Ledger is a very common use case. Such allocations are inherently aggregative, i.e., multiple rows from the instrument source map to each row posted to the Management Ledger.

You may wish to aggregate your instrument level principal balances (current book balances) to the Management Ledger in order to either enrich your ledger with dimensionality that is present in your instrument data but not present in your initial Financial Accounting data. For example, General Ledgers normally have more constrained dimensionality than is available in your instrument data. Each row of your instrument data may designate an owning Cost Center, a General Ledger corresponding to the instrument's principal balance, its Product, its Customer Segment, etc. Your General Ledger, however, may only have dimensions corresponding to Cost Center and GL Account. In this case, even though your Management Ledger table includes columns for Product and Customer Segment, every row from your initial load from your source General Ledger system will populate a single value for these dimensions the meaning of which is “Not Applicable” or “N/A”.

The following example demonstrates how you can use a Static Driver allocation rule to “reclassify” your Management Ledger data using data from the Checking and Savings (CASA) Instrument table. Build a Static Driver allocation rule as follows:

- Set the Source to Current Book Balance for the CASA Instrument table
- Set the Allocation Operator to “multiply by 1”
- Credit Ledger/Stat for Financial Element 100 (Ending Balance) using <Same as Source> for every dimension
- Debit Ledger/Stat for Financial Element 100 (Ending Balance) using <Same as Source> for the GL Account and Organizational Unit dimensions; set every other Key Processing Dimension to “N/A”

---

**Note:** When allocating debit balances, you must post them using the Debit Output tab; offsets to these debits should be posted using the Credit Output tab. Conversely, when allocating credit balances, you must post them using the Credit Output tab; offsets to these credits should be posted using the Debit Output tab.

---

This allocation effectively eliminates your original balances and replaces them with “enriched” data, i.e., data that is aligned to Product and Customer Segment as well as to Organizational Unit and General Ledger Account. For a further discussion of this kind of aggregation rule, especially in a context in which there are any variances between the sum of your instrument level balances and your initial General Ledger balances, see the section below entitled Management Ledger Reclassification Using Instrument Level Driver Data.

Another very typical use case for aggregating instrument level data to the Management Ledger concerns summarizing Funds Transfer Pricing results. An example for the CASA table might be:

- Set the Source to Transfer Pricing Charge/Credit for the CASA Instrument table
- Set the Allocation Operator to “multiply by 1”
- Credit Ledger/Stat for Financial Element 450 (Transfer Pricing Charge/Credit) using <Same as Source> for every dimension
- Debit Ledger/Stat for Financial Element 450 (Transfer Pricing Charge/Credit) using <Same as Source> for every dimension except for Organizational Unit; for the Organizational Unit dimension, post to your Funding Center

Here, the Funding Center is a shadow cost center that you have established to house all of your transfer pricing off-sets. Your Funding Center acts as an interest rate risk management center. For a typical bank whose weighted asset duration exceeds its weighted liability duration, the Funding Center will normally be a profit center (at least in a normal upward sloping yield curve environment). For a more detailed discussion of the Funding Center and of interest rate risk management in general, see the *Oracle Financial Services Funds Transfer Pricing User Guide*.

---

**Note:** If you use OFSAA Funds Transfer Pricing, you may perform these kinds of aggregation to the Management Ledger within the Funds Transfer Pricing application.

---

### ***From Instrument to Instrument***

Instrument-to-Instrument is a common use case.

- For each instrument, calculate and update a target column as a fixed relationship to some other column. For example, calculate a loan loss reserve as a fixed percentage of the current balance of each mortgage loan.
- For each instrument, calculate a rate times a balance and multiply it by an accrual basis factor and divide it by 100 in order to update a revenue or expense column. This kind of allocation would use as Expression as a Source where the expression contained a rate times balance calculation. The Static Driver would consist of (1) an accrual basis macro and (2) and factor of 0.01.

### ***From Transaction Summary to Management Ledger***

Transaction Summary-to-Management Ledger is a common use case.

- Aggregate Transaction Summary level costs to the Management Ledger; post results to a user defined Financial Element

### **From Transaction Summary to Instrument**

Transaction Summary-to-Instrument is a common use case.

- Aggregate Transaction Summary level costs to an associated Instrument table column. For example, you may record activity level volumes & costs in your CASA (Checking and Savings) Transaction Summary table. You may wish to aggregate a group of ATM related activities such as ATM Withdrawal Expense, ATM Inquiry Expense, ATM Transfer Expense, ATM Deposit Expense, and Other ATM Expense to an instrument column in the CASA table called ATM Expense.

From Transaction Summary to Transaction Summary

Transaction Summary-to-Transaction Summary is an infrequent use case.

- Multiply CASA Transaction Summary volumes by a fixed unit cost and post the result to CASA Transaction Summary costs
- Because Transaction Summary tables commonly store activity volumes, you are more likely to build this kind of rule using Static Table Driver rules that contain unit costs for many activities. To complete your Volume \* Unit Cost  $\diamond$  Cost process, one Static Driver Table allocation rule could take the place of dozens or hundreds of Static Driver allocation rules.

### **Examples of Leaf Allocations**

Leaf allocations only support the Management Ledger-to-Management Ledger use case. Leaf allocations are used to compare a Source set of Management Ledger balances to a Driver set of Management Ledger balances in order to create an Output set of Management Ledger balances.

In this kind of rule, the allocation engine attempts to match each Source row to a Driver row where the two rows share the same values for every Key Processing Dimension but one. For example, in an implementation in which there are 7 Key Processing Dimensions, for each Source row the engine will attempt to find a Driver row which matches the Source row in 6 dimensions but which differs in one dimension. The one dimension in which Source and Driver rows must differ is the dimension chosen in the Driver as the “Leaf” dimension.

- Example 1: You may wish to divide a set of Management Ledger Transfer Pricing Charge/Credit balances (stored under Financial Element 450) by a set of Management Ledger Average Balances (stored under Financial Element 140) in order to generate a third set of Management Ledger Weighted Average Transfer Rates (stored under Financial Element 170). In this case, you would constrain your Source data to Financial Element 450; for your Driver, you would specify Financial Element as your Leaf dimension and you would pick Financial Element 140. Finally, for your output you would choose a Financial Element of 170.
- Example 2: You may wish to subtract a set of “Aggregated Instrument Level Ending Balances” (stored under a user-defined Financial Element such as 10100) from a set of “original General Ledger ending balances” (stored under Financial Element 100) in order to generate a set of variances between your General Ledger data and your instrument data. These variance records might be stored under a second user-defined Financial Element such as 20100.

### **Examples of Field Allocations**

In the Instrument table context, Field allocations perform arithmetic operations on different columns within the same row of data. For example, you might use a Field allocation rule to multiply an instrument level balance times a rate times an accrual basis factor in order to update a rate-related income or expense column. Such an allocation could update a single row or millions of rows depending on your filtering criteria.

In the Management Ledger context, Field allocations are rarely used. When they are used, Field allocations generally perform arithmetic operations on different columns within the same “logical” row of data. For example, you might wish to generate a result set of rows in the Management Ledger that represent changes in asset values from month to month. In this case you build a Field allocation that used the < Current Amount > macro for all Management Ledger asset balances as your Source and that subtracted the < Last Month Amount > in the Driver. In this example, you would suppress the Credit output and write the Debit output to a user-defined Financial Element. If your Source Financial Element were 100 (Ending Balance), you might elect to post your results to a user-defined Financial Element whose name was Month-Over-Month Change.

In this current example, the engine will generally perform its arithmetic calculation (in this case, subtraction) on two different columns within the same physical row within the Management Ledger (the Ledger/Stat table). In the event that you were calculating a Month-over-Month change from December to January, however, the engine would obtain its January value (its Source value) from one row and its December value (the < Last Month Amount > specified in the Driver) from a second row. These two rows would be identical in every respect except for their Year Summary (YEAR\_S) values. Similarly, if you were to build a Field Allocation rule to generate Year-over-Year changes, the engine would always be comparing sets of Source rows that were identical in every respect except for their Year Summary values.

### Examples of Dynamic Allocations

#### Management Ledger Reclassification Using Instrument Level Driver Data

##### Example #1

Most commonly, your General Ledger constitutes a starting point for building up your Management Ledger. One way of enriching your Management Ledger is to exploit your instrument level data to distribute balances to additional dimensions that are not present in your book-of-record General Ledger.

For this example, assume that your General Ledger data is aligned in the Organizational Unit and GL Account dimensions but is not aligned to the Product dimension. For purposes of illustration, imagine a very simple case in which:

- Your General Ledger records principal balances for Commercial Loans and Consumer Loans under 2 GL accounts for Branch 1 and Branch 2
- Your Loan Instrument table contains thousands of loan records for the same 2 GL Accounts (Commercial Loans and Consumer Loans) for Branch 1 and Branch 2 for two different products

The following table summarizes the balances for this example.

Table	GL Account	Branch	Product	Balance	# of Loans
Ledger/Stat	Commercial Loan	1	—	\$1,000	—
Ledger/Stat	Commercial Loan	2	—	\$2,000	—
Ledger/Stat	Consumer Loan	1	—	\$3,000	—
Ledger/Stat	Consumer Loan	2	—	\$4,000	—
Loans	Commercial Loan	1	Land	\$600	214
Loans	Commercial Loan	1	Construction	\$400	659
Loans	Commercial Loan	2	Land	\$1,400	814
Loans	Commercial Loan	2	Construction	\$600	907
Loans	Consumer Loan	1	Auto	\$2,100	273
Loans	Consumer Loan	1	Personal	\$900	622

Table	GL Account	Branch	Product	Balance	# of Loans
Loans	Consumer Loan	2	Auto	\$2,600	861
Loans	Consumer Loan	2	Personal	\$1,400	590

Note that the instrument balances and General Ledger balances reconcile perfectly, e.g., the 214 Land loans and 659 Construction Loans under Branch 1 have balances totaling \$1,000 which reconcile with the General Ledger balance of \$1,000 for Commercial Loans under Branch 1.

In order to “product align” the Management Ledger:

1. Build a Dynamic Driver allocation rule where the Source filters on the Management Ledger for the < Current Amount > macro for the Asset branch of your GL Hierarchy for Financial Element 100 (Ending Balance). Instead of utilizing a rolup node to filter on assets, you could construct a Data Element Filter for the Commercial Loans and Consumer Loans GL accounts. For reasons discussed below, only the Financial Element constraint is truly required.
2. Set the Allocation Operator to Multiply.
3. Set the Dynamic Driver to utilize Ending Balance from your Loan instrument table. Set the Driver's Distribution Type to Percent Distribution. No dimensional constraints or other filters are necessary.
4. Set the Credit Output to Ledger/Stat (note: when posting outputs to Ledger/Stat, you MUST output to the < Current Amount > macro). Set < Same as Source > for each Key Processing Dimension.
5. Set the Debit Output to Ledger/Stat; use < Match Source & Driver > for the GL Account and Organizational Unit dimensions, < Match Driver > for the Product dimension, and < Same as Source > for all other Key Processing Dimensions.

Written in this fashion, the allocation rule will (1) generate credit records that exactly offset the original ledger (debit) balances and (2) aggregate the instrument ending balances on a per GL Account, per Organization Unit, per Product basis and post the results to Management Ledger. The following table summarizes the Management Ledger rows before and after the allocation has been run.

Row Type	GL Account	Branch	Product	Balance
Initial Load	Commercial Loan	1	—	\$1,000
Initial Load	Commercial Loan	2	—	\$2,000
Initial Load	Consumer Loan	1	—	\$3,000
Initial Load	Consumer Loan	2	—	\$4,000
Credit	Commercial Loan	1	—	(\$1,000)
Credit	Commercial Loan	2	—	(\$2,000)
Credit	Consumer Loan	1	—	(\$3,000)
Credit	Consumer Loan	2	—	(\$4,000)
Debit	Commercial Loan	1	Land	\$600
Debit	Commercial Loan	1	Construction	\$400
Debit	Commercial Loan	2	Land	\$1,400
Debit	Commercial Loan	2	Construction	\$600
Debit	Consumer Loan	1	Auto	\$2,100
Debit	Consumer Loan	1	Personal	\$900

Row Type	GL Account	Branch	Product	Balance
Debit	Consumer Loan	2	Auto	\$2,600
Debit	Consumer Loan	2	Personal	\$1,400

Note that:

1. The original Ledger balances are exactly offset by the allocation's Credit records
2. The allocation rule produces a balanced accounting transaction, i.e., a set of Debit and Credit records that sum to zero
3. The allocation rule's Debit records effectively “product align” the Management Ledger

Also note that it was not strictly necessary to supply any kind of GL Account or Organizational Unit filter in your allocation's Source specification. The reason that doing so is not strictly speaking required is that your rule is written to < Match Source & Driver > in the GL Account and Organizational Unit dimensions. Since only 2 GL Accounts (Commercial Loans and Consumer Loans) and 2 Organizational Units (Branch 1 and Branch 2) are found in the driver data (the instrument records), the Source is effectively constrained to these values even if you do not explicitly filter on them in the Source specification.

### Example #2

Note that the exact same results from Example #1 above could have been obtained from a Static Driver rule:

- Source = Instrument ending loan balances
- Allocation Operator = “times 1.00”
- Debit = < Same as Source > for all dimensions
- Credit = < Same as Source > for Organization Unit and GL Account and N/A for Product

Since the exact same results could be obtained more simply, why use the Dynamic Driver rule type to perform Management Ledger “reclassifications”? The answer is that your instruments will not always reconcile to your General Ledger due to timing differences or other reasons. Your institution will have a threshold tolerance level for such out-ages and tolerance levels will vary from institution to institution. Moreover, you may wish to product align expenses, fees, or other Management Ledger balances using instrument balances (or other measures) as proxies that would never reconcile to the Management Ledger balances.

The following initial data illustrates this second example where the instrument data does not reconcile to the General Ledger data:

Table	GL Account	Branch	Product	Balance	# of Loans
Ledger/Stat	Commercial Loan	1	—	\$1,000	—
Ledger/Stat	Commercial Loan	2	—	\$2,000	—
Ledger/Stat	Consumer Loan	1	—	\$3,000	—
Ledger/Stat	Consumer Loan	2	—	\$4,000	—
Loans	Commercial Loan	1	Land	\$603	214
Loans	Commercial Loan	1	Construction	\$399	659
Loans	Commercial Loan	2	Land	\$1,401	814
Loans	Commercial Loan	2	Construction	\$604	907
Loans	Consumer Loan	1	Auto	\$2,106	273



Table	GL Account	Branch	Product	Balance	# of Loans
Loans	Consumer Loan	1	Personal	\$903	622
Loans	Consumer Loan	2	Auto	\$2,597	861
Loans	Consumer Loan	2	Personal	\$1,399	590

Note that total Commercial Loans under Branch #1 is now \$1,002 whereas the ledger balance is only 1,000. A simple Static Driver allocation that aggregated these balances to the Management Ledger would create one credit record for \$1,002 and two debit records totaling \$1,002. This would leave a net “unaligned” balance of \$2. The Dynamic Driver allocation, however, would still create a single credit record for Commercial Loans under Branch 1 in the Management Ledger in the amount of \$1,000; and it would still create a two debit records for Commercial Loans under Branch 1 totaling \$1,000.

The following table summarizes the data that the Dynamic Driver allocation would create given the original (Example #2) balances shown above:

Row Type	GL Account	Branch	Product	Balance
Initial Load	Commercial Loan	1	—	\$1,000
Initial Load	Commercial Loan	2	—	\$2,000
Initial Load	Consumer Loan	1	—	\$3,000
Initial Load	Consumer Loan	2	—	\$4,000
Credit	Commercial Loan	1	—	(\$1,000)
Credit	Commercial Loan	2	—	(\$2,000)
Credit	Consumer Loan	1	—	(\$3,000)
Credit	Consumer Loan	2	—	(\$4,000)
Debit	Commercial Loan	1	Land	\$601.80
Debit	Commercial Loan	1	Construction	\$398.20
Debit	Commercial Loan	2	Land	\$1,397.51
Debit	Commercial Loan	2	Construction	\$602.49
Debit	Consumer Loan	1	Auto	\$2,099.70
Debit	Consumer Loan	1	Personal	\$900.30
Debit	Consumer Loan	2	Auto	\$2,599.60
Debit	Consumer Loan	2	Personal	\$1,400.40

### **Management Ledger Allocations Using Statistics**

#### **Percent Distribution**

Examples #1 and #2 above utilize instrument-level statistics as Driver data for rules whose Source is the Management Ledger (Ledger/Stat) and which Output to the Management Ledger. It is, of course, also possible to use the Management Ledger as Source, the Management Ledger as the source of Driver data, and the Management Ledger as your output target. Multiple examples of such rules can be found in this User Guide. Two classic examples would be distributing Human Resource expenses to Cost Centers as a function of (Management Ledger resident) headcount statistics and distributing Occupancy expenses to Cost Centers as a function of (Management Ledger resident) square footage statistics (i.e., space occupied by the target cost centers).

## Uniform

See Uniform Method under Driver Process Tab in Allocation Specification for an example of the Uniform allocation method.

## Simple

Your institution might obtain volumetric statistics for different kinds of activities either from your source systems or as memo accounts within your General Ledger. In either case, if you have such activity counts stored within your Management Ledger, you could build allocation rules to develop cost pools in order to subsequently build other rules to develop unit costs for each of your activities. For example, beyond general marketing expense, your institution might track advertising expense for Time Deposits under a single General Ledger account and record “Number of CD's Sold” for each Time Deposit product under a General Ledger memo account (likely stored in Ledger/Stat under Financial Element 10,000: Statistic). In this case, you could build a Dynamic Driver allocation rule that used the “Time Deposit Advertising Expense” GL Account as its Source, that divided by the “Number of CD's Sold” on a Percent to Total basis, and that debited a new, user-defined Financial Element 10,100: CD Acquisition Unit Costs (for this allocation, you would want to set your debit GL Account and Org Unit and all other Key Processing Dimension values to a dummy value the meaning of which was “N/A” or “Not Applicable”). In creating these unit costs, you might also have used instrument level data to dynamically obtain your “Number of CD's Sold” statistic. To accomplish this, your driver would look to the Record Count column (the Record Count column contains the number “1”) of the Time Deposit table (FSI-D-TERM-DEPOSITS); would include a Data Element Filter that isolated new accounts; and would include a Hierarchy Filter on the Product dimension that included only the relevant Time Deposits products.

Under either approach, your result set would be a series of unit costs by Product for acquiring new CD's. Note that in this example, we assumed that the only costs included in acquiring new Time Deposits were the advertising costs directly related to Time Deposit products. In a more realistic example, you might first build a series of allocation rules that created a cost pool for this expense category; or you might have build a more complex Source expression to capture all of the relevant costs dynamically. Moreover, your institution might capture these unit costs within your General Ledger or might develop these costs in an external model. Regardless of how you acquired the unit costs, the following example demonstrates how you might utilize unit costs using a Dynamic Driver allocation rule under the “Simple” method.

Having acquired (or developed) your unit cost statistics, you could build a Dynamic Driver allocation rule as follows:

- Source: Record Count (1) from the Time Deposits table
- Allocation Operator: Multiplication
- Driver: CD Acquisition Unit Costs under Financial Element 10,100 using the Simple method
- Debit Outputs: A user-defined Acquisition Costs column within the Time Deposits table; set the Product dimension to < Match Source & Driver >; set all other Key Processing Dimensions to < Same as Source >
- Credit Outputs: None

Written in this fashion, the allocation engine will read each record, match it to the appropriate unit cost for the record's Product, and update the record with that appropriate unit cost.

---

**Note:** You might also set the Credit Output to aggregate total allocated costs in order to offset the Management Ledger GL Account or Cost Pool containing the original costs.

---

Note that in achieving the objective of distributing activity based costs, it is not strictly necessary to either build cost pools or unit costs. You might be able to simply define your cost pool dynamically within an allocation rule and

allocate those costs directly to your instruments on a Percent to Total basis using appropriate instrument-level drivers (in this example, number of new accounts).

One reason you might want to take the more complicated path of actually developing unit costs is to be able to more readily report directly on those unit costs. Another reason you may prefer to deal directly with unit costs for your activities is that you have obtained those unit costs from an independent cost study or from an external Activity Based Costing application.

Finally, you might also decide that burdening new Time Deposits with their entire Acquisition Cost (1) in the month in which there were originated and (2) based on the current month's advertising costs was not economically “fair” or realistic. Instead, you might choose to develop unit costs reflecting the average of your YTD or “rolling 12” advertising expense; and you might choose to allocate not simply to new accounts but to all Time Deposit accounts. Choosing either of these methods would complicate the task of reconciling your total account level profitability back to your General Ledger, but choosing “economic” allocation methods for allocating expenses to the account level is common.

### ***From Management Ledger to Instrument***

As seen from some of the examples above, Dynamic Driver allocation rules can also update balance or rate columns at the instrument-level. The following example uses the Management Ledger as a Source while using an Instrument column as Driver in order to post to the instrument level.

In this example, your objective is to distribute Item Processing expense from your Management Ledger to individual customer account records. If each of your individual customer account records for every demand deposit account carried a statistic called Number of Items Processed, that statistic would make an excellent Percent Distribution driver for Item Processing expense; your target column for such a rule would be a user-defined instrument column called Item Processing Expense. For this rule, you would likely utilize < Match Driver > for each Key Processing Dimension in your Outputs. Note that instrument level allocations can only alter the target balance or rate column. Instrument level allocations cannot alter Key Processing Dimension values; the instrument-level Key Processing Dimensions can only be used as lookup keys. If you were to use either < Same as Source > or < Match Source & Driver > on one dimension, your rule would be forced to exclude any instrument rows that did not match your Management Ledger for the dimension in which you chose < Match Source & Driver >.

### ***Transaction Summary Tables***

Each row within an Instrument table describes a unique customer account or position at a point in time. Instrument rows are “wide” or “horizontal”, i.e., they contain potentially hundreds of columns containing attributes or measures. By contrast, Transaction Summary tables (each Instrument table has a corresponding Transaction Summary table) also describe unique customer accounts or positions at a point in time, but they include one or more (meaningful) dimension in their primary keys that are not populated in the corresponding Instrument table. In this sense, Instrument tables and Transaction Summary tables have a parent-child relationship; each row in an Instrument table may have one or more child rows in its corresponding Transaction Summary table; parent and child records share the same business date and “account identifier” (ID-NUMBER), but the “child” records vary in the “differentiating” dimension or dimensions. Each child row in a Transaction Summary table typically contains only two fact columns: Volume & Cost (you may, however, customize your Transaction Summary tables). Unlike Instrument tables, Transaction Summary tables are “tall” or “vertical”.

One way of conceptualizing Transaction Summary tables is as vertical expressions of Instrument tables. Each numeric measure within an Instrument row could be expressed as a single row within a Transaction Summary table. Used in this fashion, you might define a Transaction Summary table to have the same primary key as its parent Instrument table with the addition of one additional Key Processing Dimension called “Measure Name”. In this

case, each member in the Measure Name dimension would correspond to a column in the parent instrument table. Note that Key Processing Dimensions are present in all of your business fact tables (Instrument tables, Transaction Summary tables, and the Management Ledger). When you actively utilize a Key Processing Dimension within a Transaction Summary table, you typically do not “actively” use that dimension at the instrument level (i.e., the value in the Instrument table would typically be “N/A”). Although this is not how Transaction Summary tables are intended to be used, it may help in understanding their structure.

Another way of conceptualizing Transaction Summary tables is as follows. At the instrument-level, the value of General Ledger account for a given row is meant to express the principal balance General Ledger account for that row. When you aggregate all instrument-level current book balances, the resulting total balance should reconcile to your General Ledger principal balance. You may, however, want to reconcile balances other than simply principal balances. You might wish to reconcile average book balances, par balances, deferred balances, interest income or expense balances, accrued interest receivable or payable balances, or fee balances. To accomplish this, you might store all of your balances in instrument records, but store selected balances in child Transaction Summary tables under their respective General Ledger accounts that will reconcile back to your General Ledger.

Mortgage Instrument Record (hundreds of additional columns not depicted)						
Loan #	As-of-Date	GL Account	Book Balance	Par Balance	Interest Income	Fee Income
1	Jan, 2011	Mortgages, Book Balance	100,000	99,734	713	14

Associated Transaction Summary Child Records for Selected Balance			
Loan #	As-of-Date	GL Account	Balance
1	Jan, 2011	Mortgages, Book Balance	100,000
1	Jan, 2011	Mortgages, Par Balance	99,734
1	Jan, 2011	Mortgages, Interest Income	713
1	Jan, 2011	Mortgages, Fee Income	14

The examples above are meant to help you understand the basic structure of Transaction Summary tables. The primary purpose of Transaction Summary tables, however, is to support bottom-up profitability models. For example, the “differentiator” between an Instrument table and its child Transaction Summary table might be a user-defined Key Processing Dimension called Transaction or Activity. If your source systems can provide account level volume statistics for different kinds of activities, you might develop unit costs for each activity in order to calculate account level costs for each activity. For example, imagine that you are able to collect the following account level statistics (counts over time, typically over a month) from your source systems:

- ATM Inquiries
- ATM Withdrawals
- ATM Deposits
- ATM Transfers, In Network
- ATM Transfers, Out of Network
- Other ATM Transactions
- Direct Deposits (Electronic)

- E-Banking Auto-transfers
- E-Banking Bill Pay
- E-Banking Transfers, In Network
- E-Banking Transfers, Out of Network
- Teller Inquiries
- Teller Withdrawals
- Teller Deposits
- Teller Transfers
- Checks Processed
- Overdrafts Processed
- Paper Statements Processed

You could store these volume statistics as user-defined columns within the CASA (Checking & Savings) instrument table. The listing of such activities (transaction summary counts) might number in the dozens or even in the hundreds, and each activity would require its own extended cost column (typically populated in “rate times volume allocations”). Moreover, when you have a large number of such activities, many activities might have a count of zero resulting in wasted storage in your instrument columns. Finally, if your list of activities changes over time, you would have to modify your CASA table to remove columns corresponding to activities you no longer use and you would have to modify your CASA table to add new columns whenever you introduced new activities.

Alternatively, you could store these volume statistics in the CASA Transaction Summary table utilizing a user-defined Key Processing Dimension called Activity to differentiate child records from parent records. The dimension members within the Activity dimension would correspond to your list of activities.

There are many other advantages to this Transaction Summary approach. First, since your Activity dimension would be a Key Processing Dimension, you could construct an Activity hierarchy. The Activity hierarchy might be useful in a reporting context, but more importantly, higher level rollup points within your Activity hierarchy are likely to be much more stable than individual activities (leaf members within the Activity dimension). For example, you may wish to construct an account level profitability model for demand deposits in which you want to calculate and report on higher level cost elements that have a channel orientation such as ATM Expense, Branch Expense, and E-Banking Expense. You might choose to store your volumes and compute your costs for each (leaf level) activity at the Transaction Summary level and then construct Instrument level columns for ATM Expense, Branch Expense, and E-Banking Expense. Using unit costs, you can construct allocation rules to compute your Transaction Summary levels costs. Subsequently, you can use other allocation rules to roll up your Transaction Summary levels costs to target columns within your Instrument table that correspond to rollup points in your Activity hierarchy. The sections that follow continue with the above example to describe such allocation rules.

Note that using this approach you will not pay any storage penalty if many activities frequently have a zero count for any given account (you will not have wasted instrument columns that have zero counts and zero costs; and Transaction Summary rows will only exist for non-zero counts). Also note that if you add new activities you need only construct a new Activity member and update your Activity hierarchy to indicate its rollup point. No further maintenance would be required either in terms of your data model or your allocation rules.

### ***Updating Transaction Summary Tables***

To update a Transaction Summary table with unit costs held in your Management Ledger, construct an allocation rule that uses the Volume column of your Transaction Summary table as its Source, that uses your Ledger/Stat statistics as your driver (using the Simple method), and that debits the Cost column in your Transaction Summary table. In your debit specification, use < Match Source & Driver > for the Activity dimension and < Same as Source > for every other Key Processing Dimension. Your (Ledger/Stat resident) unit cost drivers would likely be stored under a user-defined Financial Element (one unit cost for each Activity). Note, however, that for each Activity you might potentially have different unit costs for different products. If your statistics were stored under a single Financial Element but varied by Activity and by Product, you could construct your rule to < Match Source & Driver > for both Activity and Product.

Note that the sample list of Activities used in this discussion has a “Channel” orientation. You might wish to construct a smaller list of more fundamental Activities that vary by channel. For example, the list of activities presented above could be re-expressed as follows:

- Inquiries
- Debits Processed
- Credits Processed
- Other Transactions
- Reversals Processed
- Statements Processed

Under this smaller set of Activities, you could choose to store your unit costs in Ledger/Stat by Activity and by Channel (another user-defined Key Processing Dimension). In this scenario, you would define your Transaction Summary table to utilize both Activity and Channel to differentiate it from its parent Instrument table.

Note that this example uses a Dynamic Driver allocation rule to update the Transaction Summary table. In this example and in examples of updates to Instrument tables, you could also use Static Driver Table allocation rules. One advantage to the Static Driver Table kind of rule is that its drivers are static. Even if your unit costs are the same from month-to-month, they generally need to be stored for every month in Ledger/Stat (although you can store your unit costs under a fixed month, say, January and then hard code your allocation rules to always use the January balance). For additional examples of using Static Driver Table allocation rules, see [Using Static Table Drivers in Chapter 18, "Static Table Driver,"](#).

### ***Updating Instrument Tables from Transaction Summary Tables***

Once you have run your allocation rule to update the Cost column in your Transaction Summary table, run other allocation rules to aggregate costs to the instrument level. You will need one allocation rule for each instrument-level target column, but each of these aggregation allocations would be structurally very similar. For example, to aggregate the costs associated with each of the ATM-related activity based cost in your Transaction Summary table, build a Static Driver allocation rule that uses the Transaction Summary table's Cost column as your Source, that multiplies by 1.00 in the Allocation Operator, and that debits the ATM Expense column in your target instrument table. The Source specification would also utilize the ATM Expense rollup point within your Activity hierarchy. Each subsequent allocation would have the same structure, but would vary its hierarchy rollup point filter in its Source specification and its Instrument target column in its debit specification.

### **Examples of Static Driver Table Allocations**

See Using Static Table Drivers in [Chapter 18, "Static Table Driver,"](#)

### **Examples of Lookup Driver Table Allocations**

See [Chapter 19, "Lookup Table Driver,"](#) for examples of using Lookup Driver Table allocation rules.

## **Aggregate Instrument Table**

The Aggregate Instrument Table functionality allows you to avoid the writing of multiple rules that was one rule per table, where the rule logic is essentially identical except for the specification of the Instrument Table.

User selects the “Portfolio” table as the Source table, Driver table, or Output table in any allocation rule that is capable of operating against the instrument level (that is, not including Constant rules or Leaf rules). After selecting the “Portfolio” table, the columns available in the column drop-down list box will include all the columns that have the Portfolio property, that is, all columns that are by definition found in every instrument table. The allocation engine is able to determine which table or tables are relevant given the Source, Driver, or Output tab's other constraints.

You can aggregate instrument-level data into tables other than Ledger/Stat which have instrument-like qualities, such as, many balances and attributes per row, very high row counts. The aggregate instrument tables or “super-instrument tables” are intended to facilitate both processing and, eventually, reporting on customer, household, relationship manager, and so on.

- Support the flexible methods for aggregating the instrument level data via the allocation engine. The types of aggregation would include by Customer, by Household, or by Relationship Manager.
- Support the ability to aggregate up from instrument tables to aggregate instrument tables; to push down from aggregate instrument tables to normal instrument tables; push down from Ledger/Stat to aggregate instrument tables.
- Support the ability to aggregate up from Transaction Summary tables to aggregate instrument tables. This is a less important objective and may be dropped in Function or Technical designs
- Support the ability to load aggregated instrument tables from Staging
- Support bulk aggregations via T2T rule or other Infrastructure capabilities if possible.
- Support the ability to aggregate instrument data – one column at a time – using a MERGE function so that rows may be INSERTED where they do not already exist and may be UPDATED where they already do exist.
- Support both Static Table Driver rules and Lookup Table Driver rules. For Static Table Driver rules, support Target Leaf functionality.
- Support the ability to UNDO allocations to the aggregate instrument table level using the same paradigm as is used for UNDO against instrument tables or transaction summary tables (i.e., using IDENTITY-CODE-CHG).

### **Aggregation to Ledger/Stat**

An allocation rule is written to aggregate one column from an instrument table to Ledger/Stat. Common aggregations include ending or average balances or interest income or transfer charges/credits to specific Financial

Elements in Ledger/Stat. Frequently, there are many other non-interest income or non-interest expense columns that users might also want to aggregate to Ledger/Stat. For any given column to be aggregated, the operation for CASA is functionality identical to the same aggregation from any other instrument table. Users need one such rule for each instrument table (one rule per table per column to be aggregated). Additionally, if the user adds a new instrument table, he must add another rule. If he has a family of such rules, for example, 10 rules per table (to aggregate 10 columns), adding another instrument table means adding 10 more rules.

### ***Instrument to Instrument***

Another common rule type performs column-wise calculations such as “rate x balance x accrual basis factor” that is again identical for different instrument tables.

### ***Ledger/Stat to Instrument***

Another common rule type allocates from Ledger/Stat to the instrument level.



## ***Overview of Allocation Models***

An Allocation Model consists of a list of individual allocation rules that can be executed as a single unit. Profitability Management supports two types of Allocation Models:

- Standard Models
- Circular Models

### ***Standard Models***

A Standard Allocation Model consists of a list of individual allocation rules that run sequentially and that may be executed as a single unit of work. Standard Allocation Models are useful when you wish to assemble a logical grouping of allocation rules into a single executable rule.

### ***Circular Models***

A Circular Allocation Model is similar to a Standard Allocation Model, but consists of two lists of allocation rules: a list of “circular” rules and a list of “sweep” rules. The list of circular rules executes first; generally, the circular list will run multiple times. Each rule within the circular list of rules runs sequentially. Once the circular list has run one or more times, each of the sweep rules also runs sequentially, but the list of sweep rules only runs once.

The primary purpose of the Circular Allocation Model is to process allocation scenarios in which the allocation rules may Source data that result from previous allocations and that send data back to Sources that had previously been “eliminated”. One common example of this kind of situation occurs in center-to-center kinds of allocation rules that involve providers of shared services. For example, a block of allocation rules might be designed to push expenses from a series of providers of shared services such as IT, Finance, HR, Payroll, Accounting, Treasury, etc. to a “downstream” series of direct support centers and profit centers. An issue that you might encounter here is that HR allocates most of its expense to direct support centers and to profit centers, but also allocates some of its expense to other providers of shared services such as Payroll or Accounting. These centers, in turn, might allocate some of their expense back to HR when the first allocation had already cleared all of the expense from the HR center.

## ***Summary and Detail Screens***

Upon initially navigating to Profitability Management > Rule Specification > Allocation Model Specification, a summary screen is displayed showing a set of Allocation Models. Using search criteria, you can control the set of Allocation Models that are displayed. When you Add, Edit, or View a rule, a detail screen is displayed.

### ***Navigation within the Summary Screen***

When you first navigate to the Allocation Model summary screen, the Allocation Models stored within your current default Folder are presented in a summary grid. The Allocation Model Specification summary screen has two containers: Search and Allocation Model Specification.

Allocation Model As of Date: 06/25/2014

Search Q ↻

Folder: AUTOSEG

<input type="checkbox"/>	Name	Creation Date	Created By	Last Modified Date	Modified By	Model Type
<input type="checkbox"/>	PFT_AUTO_AM_11_Edit	04/30/2018 10:54:36	PFTUSER	04/30/2018 11:48:42	PFTUSER	Circular
<input type="checkbox"/>	test	04/30/2018 16:13:10	PFTUSER	04/30/2018 16:13:10	PFTUSER	Circular

**Search Container**

Your default Folder functions as a search constraint. The value of your default Folder is set in Application Preferences for Profitability Management. You may select a different Folder or you may remove the Folder constraint entirely by selecting the “blank” Folder, that is, no Folder. You may also search by Name or by Model Type.

**Search Control**



You may search for Allocation Models by Folder, by Model Name, or by Model Type. Enter your desired search criteria and click the Search control.

**Reset Control**



Restores the default Folder, removes any Allocation Model Name or Allocation Model Type constraint you may have specified, and refreshes the screen.

**Allocation Model Specification Container**

The Allocation Model Specification container presents a grid containing all of the Allocation Models that meet your search criteria. The Allocation Model Specification summary grid offers several controls that allow you to perform different functions when an Allocation Model is selected.

To select an Allocation Model, click a check box in the first column of the grid. More than one Allocation Model can be selected at a time but this will cause some of the controls to become disabled. Clicking a check box a second time de-selects the Allocation Model.

You may select or deselect all of the Allocation Models in the summary grid by clicking the check box in the upper left hand corner of the summary grid directly to the left of the Name column header.

**Add**

Clicking the Add control begins the process of building a new Allocation Model. The Add control is disabled if any rows in the grid have been selected.

**View**

Selecting a single row out of the grid enables the View control. Clicking the View control allows you to view the contents of an Allocation Model on a read-only basis. The View control is only enabled when a single Allocation Model has been selected.

**Edit**

Selecting a single row out of the grid enables the Edit control. Clicking the Edit control allows you to modify a previously saved Allocation Model. The Edit control is only enabled when a single Allocation Model has been selected.

**Delete**

Selecting one or more rows out of the grid enables the Delete control. Clicking the Delete control deletes the models you have selected.

**Run**

After having chosen a single row out of the grid, clicking the Run control allows you to execute the selected Allocation Model. The As-of-Date that applies to the model's execution is visible in the upper right corner of the screen; it may be reset in Application Preferences for Profitability Management. The Run control is only enabled when a single Allocation Model has been selected.

**Check Dependencies**

Clicking the Check Dependencies control generates a report on any rule objects that depend on the Allocation Model you have selected. The Check Dependencies control is only enabled when a single Allocation Model has been selected.

### ***Allocation Model Specification Summary Grid***

The following columns categorize each Allocation Model in the summary grid:

- Name
- Creation Date
- Created By
- Last Modified Date
- Modified By
- Access Type
- Folder
- Model Type

#### **Name**

Displays the Allocation Model's short name. Performing a “mouse-over” on an Allocation Model's Name displays the model's long name and system identifier.

#### **Creation Date**

Displays the date and time at which an Allocation Model was created.

#### **Created By**

Displays the name of the user who created an Allocation Model.

#### **Last Modified Date**

Displays the date and time at which an Allocation Model was last modified.

#### **Modified By**

Displays the name of the user who last modified an Allocation Model.

#### **Access Type**

Displays the “Read/Write” or “Read Only” property of an Allocation Model. Only the creator of a model may change its Access Type.

#### **Folder**

Displays the name of the Folder in which an Allocation Model is stored.

#### **Model Type**

Displays whether a model is a Standard Allocation Model or a Circular Allocation Model.

## Navigation within the Detail Screen

When you Add, Edit, or View an Allocation Model, the Allocation Model Specification Detail Screen is displayed. The detail screen is composed of 4 containers:

- Allocation Model Definition Details
- Model Type
- Model Details
- Audit Trail / User Comments

### Allocation Model Definition Details Container

The screenshot shows a web form titled "Allocation Model Definition" with a sub-header "Allocation Model (New)". Below this is a section titled "Allocation Model Definition Details" which is expanded. It contains the following fields:
 

- Name \***: A text input field.
- Description**: A larger text input field.
- Folder Name**: A dropdown menu currently showing "AUTOSEG".
- Access Type**: Two radio buttons, "Read Only" and "Read/Write". The "Read/Write" option is selected.

The Allocation Model Definition Details container allows you to specify a model's Name, Description, Folder, and Access Type.

### Model Type Container

The appearance of the Model Type container depends on whether you are building a Standard Model or a Circular Model.

#### Standard Model

If you wish to construct a Standard Allocation Model, simply select Standard from the Model Type drop-down list box.

The screenshot shows a section titled "Model Type" which is expanded. It contains a label "Type" followed by a dropdown menu that has "Standard" selected.

#### Circular Model

If you wish to construct a Circular Allocation Model, select Circular from the Model Type drop-down list box.

The most typical use case for Circular Allocation Models involves sequences of center-to-center rules where your objective is to transfer a series of expense balances in which each allocation rule allocates expense from one center to a series of target centers. In this scenario, it is very common to credit back to the source using <Same as Source> in your specification for each dimension within your allocation Credit specification. Sometimes, however, especially in a shared services context, centers that you have cleared of expense will themselves become the targets of allocations that come later in a sequence of allocation rules. Typically, each center will allocate out 100% of its original expense, but by the end of the series of shared services allocations, each center will have some small expense balances that have been allocated back to it from other centers. The amount of expense that “backwashes” to centers that have already been cleared of expense will vary, but typically will be in the range of 1% to 10% of the

expenses originally found in each center. If the typical center in a series of such rules ended up with 10% of its original expense after the full sequence of rules had been run one time, then after running the same sequence of rules a second time the typical center would have 1% of its original expense remaining and after two cycles would have 1/10th of 1% of its original expense remaining.

The purpose of Circular Allocation Models is to allow you to organize such lists of allocation rules that you wish to run iteratively for multiple cycles to reduce to near zero the balances that were originally present. Once the remaining balances have reached an acceptable threshold, you can simply “sweep” the remaining balances from all sources with one or more additional rules that will run only once.

After selecting “Circular” from the Model Type drop-down list box, you must select how you want to control the number of cycles or iterations you want your “circular” rules to run. To do this you must select one of the following three choices:

- Number of cycles
- Threshold Source Balance
- Threshold Percentage

### **Number of Cycles**

Specify the number of iterations or cycles you want the circular part of your Allocation Model to execute.

### **Threshold Source Balance**

Specify the residual threshold source balance you would like to achieve within your Allocation Model.

After each rule within the “circular” container has been executed once, the engine calculates the remaining Total Source Balance. The remaining Total Source Balance is equal to the sum of the Source specification of each of the rules within the Allocation Model’s “circular” container. This value is compared to your Threshold Source Balance. If your threshold balance is greater than the remaining Total Source Balance, the iterative process stops and each rule within the “sweep” container is run once sequentially.

If your threshold balance is less than the remaining Total Source Balance, each of your “circular” rules will run again in sequence. After each cycle, your threshold balance is compared to the remaining Total Source Balance until either (1) it is greater than the remaining Total Source Balance or (2) the maximum number of iterations (defined in the following sections) is reached.

### **Threshold Percentage**

Specify the residual threshold percent balance you would like to achieve within your Allocation Model.

Threshold Percentage is similar to Threshold Source Balance except that instead of comparing the remaining Total Source Balance to a Threshold Source Balance, the remaining Total Source Balance after each cycle is compared to the original Total Source Balance. The original Total Source Balance is calculated only once before any of the circular rules have been executed. After each cycle through the circular rules, the remaining Percent Balance is calculated as

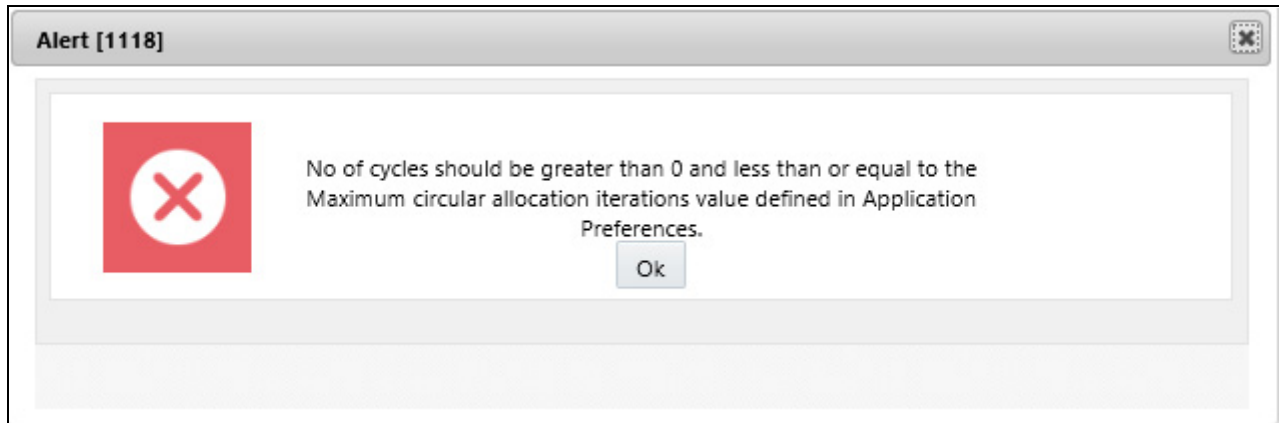
“remaining Total Source Balance” divided by “original Total Source Balance”. The remaining Percent Balance is compared to your threshold Percentage to determine whether to exit the loop and proceed to the sweep rules or to continue cycling through the circular rules. As with Threshold Source Balance, the circular section of rules will never run more than the “maximum number of iterations” (defined in the following section).

### Maximum Circular allocation Iterations

In order to ensure that you never experience an infinite loop of allocation rules that never finish, Allocation Models are governed by an overall limit on the maximum number of iterations or cycles that can be run before the circular part of a model completes. This maximum number of iterations is set in Application Preferences for Profitability Management.

Application Preferences		
Apply Cancel Reset to Default		
Select Preferences For		
Show Preferences For	PFTUSER	
Active for Master Maintenance	<input checked="" type="checkbox"/>	
Parameters - General		
Property Name	Property Value	Is Editable
Security Map	None	<input type="checkbox"/>
As of Date	06/25/2014	<input checked="" type="checkbox"/>
Show Execution Parameters	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Runtime Parameters	Leaf Description	<input type="checkbox"/>
Legal Entity	Default Member	<input type="checkbox"/> Disable Legal Entity
Processing - General		
Property Name	Property Value	Is Editable
Default Total Error Message Limit	250	<input checked="" type="checkbox"/>
Default Error Message Limit Per Item	25	<input checked="" type="checkbox"/>
Debugging Output Level	Show All SQL	<input checked="" type="checkbox"/>
Processing - Application Specific		
Property Name	Property Value	Is Editable
Enable Undo for Selected Allocations	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input checked="" type="checkbox"/>
Enable Mass Undo	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input checked="" type="checkbox"/>
Enable Undo for Selected Batches	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input checked="" type="checkbox"/>
Enable Undo for Selected Allocation Models	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input checked="" type="checkbox"/>
Maximum circular allocation iterations	5	<input checked="" type="checkbox"/>
Scenario for source	Actual	<input checked="" type="checkbox"/>
Scenario for driver	Actual	<input checked="" type="checkbox"/>
Scenario for output	Actual	<input checked="" type="checkbox"/>
Assumption Management Defaults		
Property Name	Property Value	Is Editable
Folder Name	AUTOSEG	<input checked="" type="checkbox"/>
Access Type	<input type="radio"/> Read Only <input checked="" type="radio"/> Read/Write	<input checked="" type="checkbox"/>

Within the specification of an Allocation Model, if you attempt to save a model using a value for “Number of Cycles” that either (1) exceeds the value found in Application Preferences or (2) is less than or equal to zero, the user interface will display the following message:

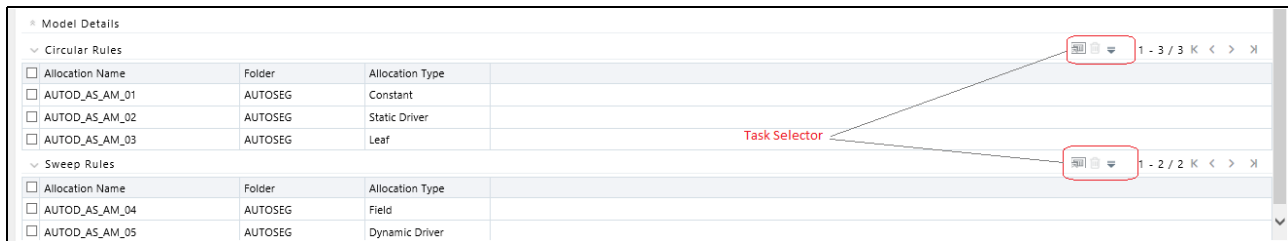


**Model Details**

If you choose to construct a Standard Allocation Model, the Model Details container will present a single list of allocation rules. If you choose to construct a Circular Allocation Model, the Model Details container will present a list of “circular rules” and a list of “sweep rules”.

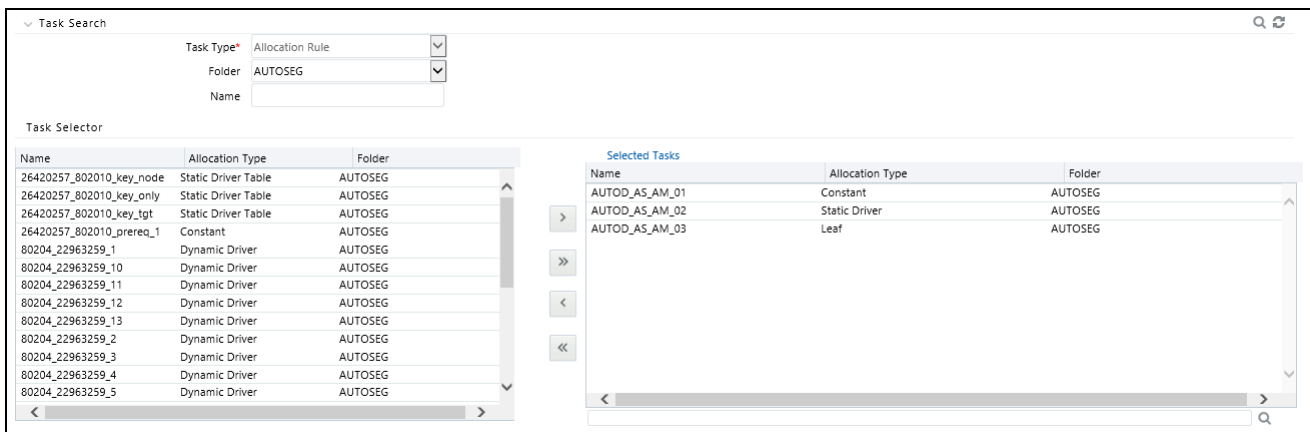
Selecting Allocation Rules to Include

Click the Task Selector to choose the rules you want to include.



Task selector consists of two containers

- Task Search
- Task Selector





## Task Search

Task Search

Task Type\* Allocation Rule

Folder AUTOSEG

Name

Task Search contains Task Type, Folder and Name. The only Task Type presented for Allocation Models are allocation rules (the Allocation Model Task Selector is otherwise identical to the Task Selector used within Simplified Batches). Select a Folder and click the Search control to view all of the allocation rules within that Folder. You may also select “blank” or “no Folder” in order to search for all allocation rules. You may also search for rules by name. After executing a search, qualifying rules are displayed in the left hand side of the Task Selector.

## Task Selector

Name	Allocation Type	Folder
26420257_802010_key_node	Static Driver Table	AUTOSEG
26420257_802010_key_only	Static Driver Table	AUTOSEG
26420257_802010_key_tgt	Static Driver Table	AUTOSEG
26420257_802010_prereg_1	Constant	AUTOSEG
80204_22963259_1	Dynamic Driver	AUTOSEG
80204_22963259_10	Dynamic Driver	AUTOSEG
80204_22963259_11	Dynamic Driver	AUTOSEG
80204_22963259_12	Dynamic Driver	AUTOSEG
80204_22963259_13	Dynamic Driver	AUTOSEG
80204_22963259_2	Dynamic Driver	AUTOSEG
80204_22963259_3	Dynamic Driver	AUTOSEG
80204_22963259_4	Dynamic Driver	AUTOSEG
80204_22963259_5	Dynamic Driver	AUTOSEG

Name	Allocation Type	Folder
AUTOD_AS_AM_04	Field	AUTOSEG
AUTOD_AS_AM_05	Dynamic Driver	AUTOSEG

Ok Cancel

The Allocation Model Task Selector has the same user interface as is used for selecting tasks in Simplified Batches. For details on using the Task Selector for Allocation Models, see Task Selection in Simplified Batches.

## Audit Trail / User Comments

The Audit Trail container is a standard footer container for every OFSAA rule type. It displays Created By, Creation Date, Last Modified By, and Modification Date on the Audit Trail tab. The User Comments tab may be used to add comments to any rule, subject to a maximum of 4000 characters.



## CHAPTER 18

*Static Table Driver***Overview**

This chapter describes Static Table Driver functionality.

- Static Table Driver Summary & Detail Screens
- Navigation within the Summary Screen
- Navigation within the Detail Screen
- Using Static Table Drivers

**Summary & Detail Screens**

Upon initially navigating to Profitability Management > Rule Specification > Static Table Driver, a summary screen is displayed showing a set of Static Table Driver rules. Using search criteria, you can control the set of rules that are displayed. When you Add, Edit, or View a rule, a detail screen is displayed.

The screenshot shows the 'Static Table Driver' summary screen. At the top, there is a search bar with a 'Folder' dropdown set to 'AUTOSEG' and a 'Name' input field. Below the search bar is a table with the following columns: Name, Creation Date, Created By, Description, Last Modified Date, and Modified By. The table contains 16 rows of data, including rules like '25136153\_key\_only', '26420257\_802010\_key\_node', and '81\_PORTFOLIO\_STD\_BOTH'. The 'As of Date' is set to 06/25/2014.

Name	Creation Date	Created By	Description	Last Modified Date	Modified By
25136153_key_only	02/28/2017 13:03:00	PFTUSER		02/28/2017 14:54:33	PFTUSER
26420257_802010_key_node	09/26/2017 11:43:56	PFTUSER		09/26/2017 11:43:56	PFTUSER
26420257_802010_key_only	09/26/2017 11:36:09	PFTUSER		09/26/2017 11:36:09	PFTUSER
26420257_802010_key_tgt	09/25/2017 14:56:59	PFTUSER		09/26/2017 11:35:08	PFTUSER
81_PORTFOLIO_STD_BOTH	04/26/2016 13:23:50	PFTUSER		04/26/2016 13:23:50	PFTUSER
81_PORTFOLIO_STD_BOTH_SIMPLE	05/06/2016 17:55:06	PFTUSER		05/06/2016 17:55:06	PFTUSER
81_PORTFOLIO_STD_KEY	04/26/2016 13:20:20	PFTUSER		04/26/2016 13:21:10	PFTUSER
81_PORTFOLIO_STD_TGT	04/26/2016 13:22:17	PFTUSER		04/26/2016 13:22:17	PFTUSER
81_PORTFOLIO_STD_TGT_SIMPLE	05/06/2016 18:02:08	PFTUSER		05/06/2016 18:02:08	PFTUSER
AUTOD_STD_BOTH_03	02/15/2018 11:31:17	PFTUSER		02/15/2018 11:31:17	PFTUSER
AUTOD_STD_KEY_01	10/19/2017 12:10:28	PFTUSER	TESTING	10/19/2017 12:10:28	PFTUSER
AUTOD_STD_TARGET_02	10/19/2017 15:35:02	PFTUSER		10/19/2017 15:35:02	PFTUSER
PFT_AUTO_STD_3_Copy	04/30/2018 09:44:00	PFTUSER		04/30/2018 09:44:00	PFTUSER
PFT_AUTO_STD_3_Edit	04/30/2018 08:47:52	PFTUSER	TESTING	04/30/2018 08:57:08	PFTUSER

**Navigation within the Summary Screen**

When you first navigate to the Static Table Driver summary screen, the rules stored within your current default Folder are presented in a summary grid. The Static Table Driver summary screen has two containers: Search and Static Table Driver.

**Search Container**

Your default Folder functions as a search constraint. The value of your default Folder is set in Application Preferences for Profitability Management. You may select a different Folder or you may remove the Folder

constraint entirely by selecting the “blank” Folder, that is, no Folder. You may also search by a rule's Name or Description.

**Search Control**

You may search for Static Table Driver rules by Folder, Name, or Description. Enter your desired search criteria and click on the Search control.

**Reset Control**

Restores the default Folder, removes any Name or Description constraint you may have specified, and refreshes the screen.

***Static Table Driver Container***

The Static Table Driver container presents a grid containing all of the Static Table Driver rules that meet your search criteria. The Static Table Driver summary grid offers several controls that allow you to perform different functions when a rule is selected.

To select a rule, click on a check box in the first column of the grid. More than one rule can be selected at a time but this will cause some of the controls to become disabled. Clicking on a checkbox a second time de-selects the rule.

You may select or deselect all of the Static Table Driver rules in the summary grid by clicking on the check box in the upper left hand corner of the summary grid directly to the left of the Name column header.

**Add**

Clicking on the Add control begins the process of building a new Static Table Driver rule. The Add control is disabled if any rows in the grid have been selected.

**View**

Selecting a single row out of the grid enables the View control. Clicking on the View control allows you to view the contents of a Static Table Driver rule on a read-only basis. The View control is only enabled when a single rule has been selected.

**Edit**



Selecting a single row out of the grid enables the Edit control. Clicking on the Edit control allows you to modify a previously saved Static Table Driver rule. The Edit control is only enabled when a single rule has been selected.

### Delete



Selecting one or more rows out of the grid enables the Delete control. Clicking on the Delete control deletes the Static Table Driver(s) you have selected. OFSAAI will not allow you to delete Static Table Driver rules which have any dependencies (see View Dependencies below). When you click on Delete, a message is displayed: Are you sure you want to delete this record

### Copy



Selecting a single row out of the grid enables the Copy control. Clicking the Copy control allows you to create a copy of an existing Static Table Driver rule. The Copy control is only enabled when a single Static Table Driver rule has been selected. When you click on Copy, a Save As pop window will appear. Click Save after entering the Name, Description, Folder, and Access Type Details.

### View Dependencies



Clicking on the View Dependencies control generates a report on any Allocation rule that depend on the Static Table Driver rule you have selected. The View Dependencies control is only enabled when a single rule has been selected.

### Refresh



Click the Refresh control to refresh the Static Table Driver rule summary grid, including updates to the Last Modified Date.

### **Static Table Driver Summary Grid**

The following columns categorize each allocation rule in the summary grid:

- Name
- Creation Date

- Created By
- Description
- Last Modified Date
- Modified By
- Access Type
- Folder

Static Table Driver As of Date: 06/25/2014

Search  Folder: AUTOSEG  Name:

Description:

<input type="checkbox"/>	Name	Creation Date	Created By	Description	Last Modified Date	Modified By
<input type="checkbox"/>	25136153_key_only	02/28/2017 13:03:00	PFTUSER		02/28/2017 14:54:33	PFTUSER
<input type="checkbox"/>	26420257_802010_key_node	09/26/2017 11:43:56	PFTUSER		09/26/2017 11:43:56	PFTUSER
<input checked="" type="checkbox"/>	26420257_802010_key_only	09/26/2017 11:36:09	PFTUSER		09/26/2017 11:36:09	PFTUSER
<input type="checkbox"/>	26420257_802010_key_tgt	09/25/2017 14:56:59	PFTUSER		09/26/2017 11:35:08	PFTUSER
<input type="checkbox"/>	81_PORTFOLIO_STD_BOTH	04/26/2016 13:23:50	PFTUSER		04/26/2016 13:23:50	PFTUSER
<input type="checkbox"/>	81_PORTFOLIO_STD_BOTH_SIMPLE	05/06/2016 17:55:06	PFTUSER		05/06/2016 17:55:06	PFTUSER
<input type="checkbox"/>	81_PORTFOLIO_STD_KEY	04/26/2016 13:20:20	PFTUSER		04/26/2016 13:21:10	PFTUSER
<input type="checkbox"/>	81_PORTFOLIO_STD_TGT	04/26/2016 13:22:17	PFTUSER		04/26/2016 13:22:17	PFTUSER
<input type="checkbox"/>	81_PORTFOLIO_STD_TGT_SIMPLE	05/06/2016 18:02:08	PFTUSER		05/06/2016 18:02:08	PFTUSER
<input type="checkbox"/>	AUTOD_STD_BOTH_03	02/15/2018 11:31:17	PFTUSER		02/15/2018 11:31:17	PFTUSER
<input type="checkbox"/>	AUTOD_STD_KEY_01	10/19/2017 12:10:28	PFTUSER	TESTING	10/19/2017 12:10:28	PFTUSER
<input type="checkbox"/>	AUTOD_STD_TARGET_02	10/19/2017 15:35:02	PFTUSER		10/19/2017 15:35:02	PFTUSER
<input type="checkbox"/>	PFT_AUTO_STD_3_Copy	04/30/2018 09:44:00	PFTUSER		04/30/2018 09:44:00	PFTUSER
<input type="checkbox"/>	PFT_AUTO_STD_3_Edit	04/30/2018 08:47:52	PFTUSER	TESTING	04/30/2018 08:57:08	PFTUSER

**Name**

Displays the rule's short name. Performing a “mouse-over” on a Name displays the Static Table Driver rule's description and system identifier.

**Creation Date**

Displays the date and time at which a rule was created.

**Created By**

Displays the name of the user who created a rule.

**Description**

Displays the rule's Description.

**Last Modified Date**

Displays the date and time at which a rule was last modified.

**Modified By**

Displays the name of the user who last modified a rule.

**Access Type**

Displays the “Read/Write” or “Read Only” property of a Static Table Driver rule. Only the creator of a rule may change its Access Type.

**Folder**

Displays the name of the Folder in which a Static Table Driver rule is stored.

## ***Navigation within the Detail Screen***

When you Add, Edit, or View a Static Table Driver rule, the Static Table Driver Detail screen is displayed.

In addition to Name, Description, Folder, and Access Type, the definition of a Static Table Driver includes Key Leaves, a Target Leaf, and coefficients values. Once a Static Table Driver has been defined and saved, you will generally only edit an existing rule to update your driver coefficients (addition or deletion of Key and Target dimensions are not possible).

The Audit Trail container is a standard footer container for every OFSAA rule type. It displays Created By, Creation Date, Last Modified By, and Modification Date on the Audit Trail tab. The User Comments tab may be used to add comments to any rule, subject to a maximum of 4000 characters.

### **Static Table Driver Functionality**

Static Table Drivers are used in conjunction with Allocation rules for two purposes:

- To distribute balances at the Management Ledger-level
- To perform a lookup table function against instrument tables

### ***Distribution with the Management Ledger-level***

Static Driver Table Allocation rules that distribute balances at the management ledger-level function similarly to Dynamic Driver allocation rules. In your Debit or Credit outputs

- Key Leaves function similarly to Dynamic Driver allocation rules that utilize < Match Source & Driver >
- Target Leaves function similarly to Dynamic Driver allocation rules that utilize < Match Driver >

### ***Instrument Level Update***

Static Driver Table Allocation rules perform a lookup table function against an instrument table to match dimension values for each instrument row against the Key Leaf values you define in your Static Table Driver rule. When matching rows are found, the allocation performs an arithmetic operation combining source balance columns and a coefficient value you specify in order to update a result column.

### ***Static Table Driver Container***

Specify the Static Table Driver rule's Name and Description, select a Folder in which the Static Table Driver rule is to be stored, and specify whether you want the Static Table Driver rule to be "Read/Write" or "Read Only" (Access Type). Naming your Static Table Driver rule is required before it can be saved. Static Table Driver rule Name does not accept Special characters (&, @, ~, +, Single quote). Default values for Folder and Access Type are stored in Application Preferences for Profitability Management.

### ***Static Table Driver Definition Container***

The first thing you must do when creating a new Static Table Driver is to define its structure. Once a Static Table Driver has been defined, typical maintenance is usually limited to updating driver coefficients. The most critical aspects of a Static Table Driver's structural definition are its:

- Key dimension
- Key Leaf values for each Key Leaf



- Target dimension
- Target Leaf values for each Target Leaf
- Defining Key Leaf values and Target Leaf values
- Static Table Driver Coefficients

You need not specify both a Key Leaf and a Target Leaf, but you must specify at least one or the other. That is, you should select a Key or Target dimension while defining a new Static Table rule.

---

**Note:** You cannot add or delete the Key/Target dimensions after defining the Static Table Driver (edit mode). You can add or delete the dimension members.

---

### **Key Leaves**

For both Management Ledger-level and Instrument allocations, Key Leaves perform a lookup function. Rows from the Source of the Allocation rule are joined with the Key Leaf values from the Static Table Driver. If an input row leaf value matches a Key Leaf value for the first Key Leaf dimension specified in the Static Table Driver, the row is then checked against the second Key Leaf values, and so on. If a match is found for every Key Leaf dimension, the row is processed by the Allocation rule. Profitability Management supports a maximum of 3 Key Leaf dimensions.

### **Target Leaf**

Target leaves are used only in allocations that distribute to the Management Ledger-level. Distribution allocations can be viewed as first performing a lookup on one or more Key Leaves. When a match is found for each Key Leaf dimension, the matching input row's amount is distributed to all dimension values specified as Target Leaf values. This means that data is added to the Management Ledger for each dimension member value specified as a Target Leaf value. The leaf values specified in the Allocation rule's Debit typically use the < Same As Source > macro for all Key leaf dimensions and use the < Same As Table > macro for the Target Leaf dimension (very similar to the < Same as Driver > macro used in Dynamic Driver allocation rules).

### **Coefficients**

You must specify coefficient values for every Static Table Driver rule. A part of the coefficient specification process is to enter the values for each distinct combination of each of your Key Leaves and your Target Leaves. Where only Key Leaves are used, you must enter values for each distinct combination of each of your Key Leaves.

For Static Table Drivers with a Target Leaf Dimension

For a Static Table Driver that uses a Target Leaf dimension, you must supply one coefficient value for each distinct combination of each of your Key Leaves and your Target Leaves. Again, Static Table Drivers that use a Target Leaf dimension are only supported for Management Ledger allocation rules.

Target Leaf coefficient values represent distribution statistics. You would typically use these statistics on a Percent-to-Total basis. Static Table Drivers support both the Percent-to-Total method and the Simple method.

The definition of the Target Leaf dimension may be preceded by the definition of one or more Key Leaf dimensions. Key Leaves are never required unless there is no Target Leaf defined. When Key Leaves are present, they operate as lookup keys as described above. For a Static Table Driver that uses a Target Leaf dimension, the Key leaves function analogously to the Dynamic Driver allocation function of < Match Source & Driver > and the Target Leaf functions analogously to the Dynamic Driver allocation function of < Match Driver >.

### For Static Table Drivers with Key Leaf Dimensions But No Target Leaf

For Static Table Drivers that do not use a Target Leaf dimension, you must supply one coefficient value for each distinct combination of each of your Key Leaves. Static Table Drivers that use Key Leaves but do not use a Target Leaf can support both distributive Management Ledger allocation rules and Instrument level update rules.

For an Instrument level update rule using a Static Table Driver, the Key Leaves function as lookup keys and the coefficients are used as arithmetic factors in updating a target column in an instrument table. For Management Ledger allocation rules, the Key Leaves function analogously to the Dynamic Driver allocation function of < Match Source & Driver >.

## Creating a New Static Table Driver – Sample Workflow

This section describes a typical workflow for defining a Static Table Driver rule.

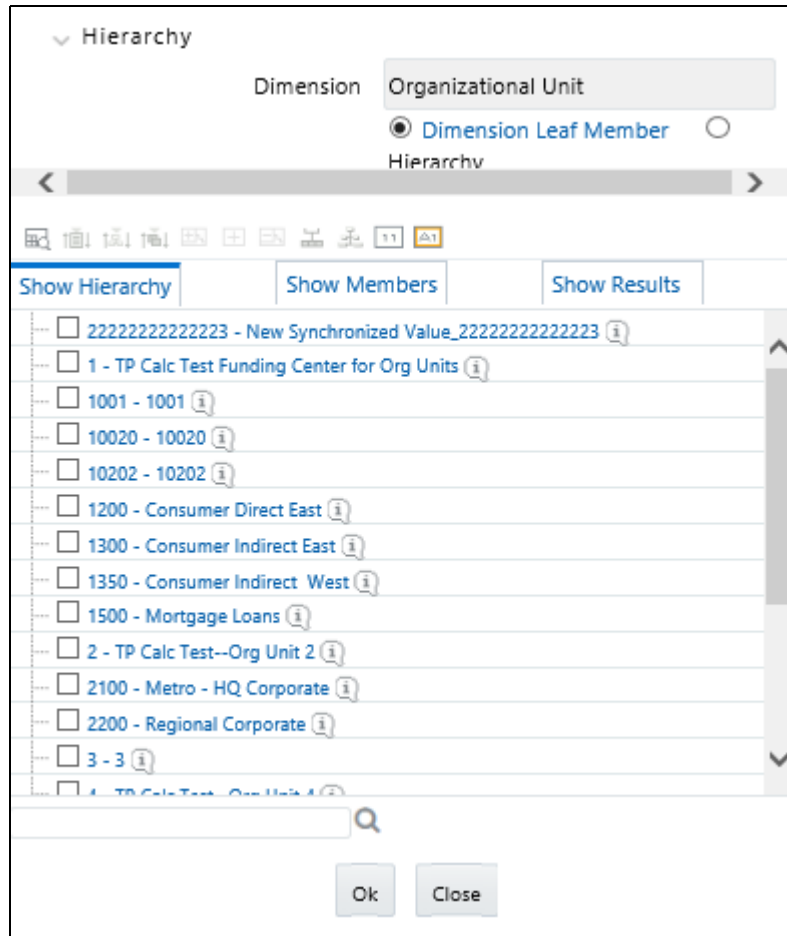
### Create a New Key Leaf

Start the setup process by defining a Key Leaf dimension.

1. Choose the dimension you want to define as a Key Leaf from the **Add Leaf** drop-down list.
2. Click on the **Key** radio button.
3. Click on the **Add** control (+). The Key Leaf dimension will be added to Key Dimension section.

4. Click the Hierarchy Browser ellipses (...) that appear next to your Key Leaf. The Hierarchy Browser defaults to a list of leaves for the Key Leaf dimension you have chosen (the radio button selector near the top of the browser window defaults to Dimension Leaf Member). You may scroll up and down to find the leaf member or leaf members you want. You may also search for a dimension member's name (short description) using the binoculars at the bottom of the browser window. Additional search functionality is provided by the Search control within the browser. This additional functionality allows you to search by Dimension Member Dimension member, Name, or Code. Once you have selected the Key Leaf values you want, click **OK**.

5. You may also define Key Leaf values to be rollup members within a hierarchy. To select hierarchy rollup point members, click the Hierarchy radio button near the top of the browser window and then search for the hierarchy you wish to use. Once your chosen hierarchy is displayed within the browser window, navigate into the hierarchy until you have found the rollup points you want. Select the value or values you want and then click OK.



6. Once you have clicked OK within the Hierarchy Browser window, that window closes and you will be directed back to Static Table Driver Definition tab. If you again open the Hierarchy Browser of dimension, it will show the members as checked which were already selected while defining a key leaf.

You may now repeat this process to add a second or third Key Leaf if desired. If no additional Key Leaves are needed, you may proceed to either (1) adding a Target Leaf dimension or (2) providing coefficients for your chosen Key Leaf value(s).

**Note:** When you select hierarchy rollup points, all of the members, you select must come from the same level within the underlying hierarchy. Hierarchy Filters, which may include leaves and rollup nodes from different levels within a hierarchy, are not supported in Static Table Driver rules.

Target leaves are used only in allocations that distribute to the Management Ledger.

## Creating a New Target Leaf

Continuing with the typical setup workflow, the following narrative assumes that you have added only one Key Leaf dimension.

1. Choose the dimension you want to define as a Target Leaf from the **Add Leaf** drop-down list.
2. Click on the **Target** radio button.
3. Click on the **Add** control (+)
4. The Target Leaf dimension will be added under the Target Dimension section.

Static Table Driver Definition  
Static Table Driver (New)

Static Table Driver

Name\*

Description

Folder: AUTOSEG

Access Type:  Read Only  Read/Write

Add Leaf: Organizational Unit

Key Dimension

Organizational Unit

Target Dimension

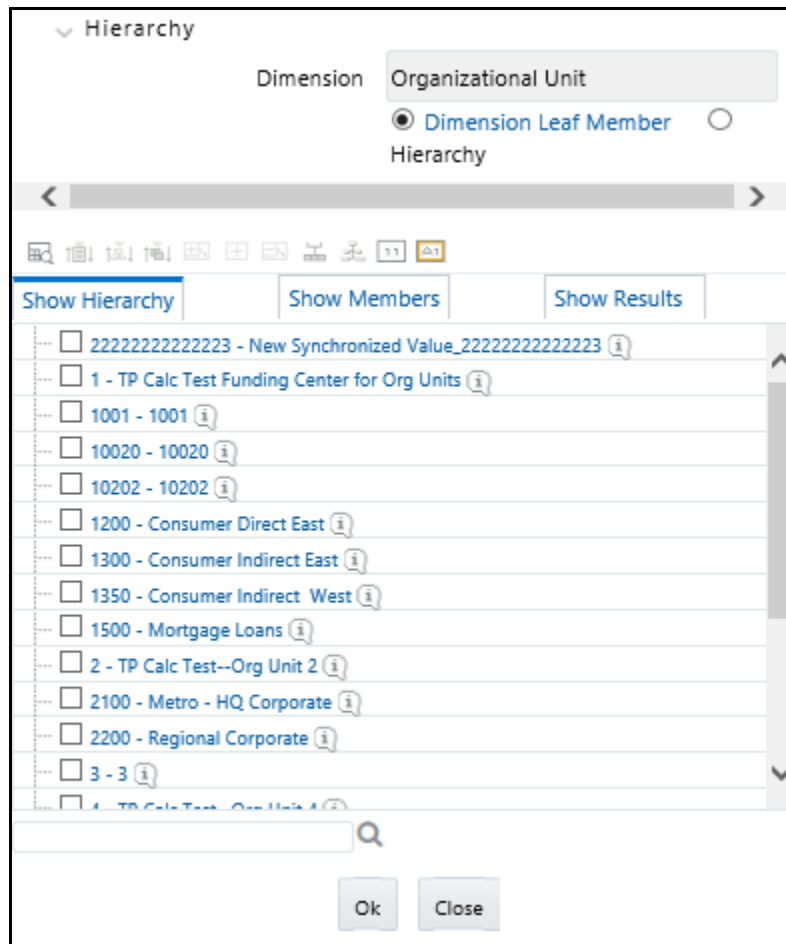
Organizational Unit

Coefficient:

Force to 100%  Yes  No

Save Cancel

5. Click on the Hierarchy Browser ellipses (...) that appear on your Target Leaf.
6. Select the Target Leaf values (the selection process is identical to the process of selecting Key Leaf values described above) from Hierarchy Browser and click **OK**.



- Once you have clicked on OK within the Hierarchy Browser window, that window closes and you will be directed back to Static Table Driver detail screen. However, while the screen still displays your Key Leaf values under the Key Dimension section as rows, it now displays your Target Leaf values under the Target Dimension section as rows.

### **Defining Coefficient Values**

Navigate to the Coefficient tab to assign coefficients to combinations of Key leaf (or node) values and Target leaf values. Continuing with the typical setup workflow, the following narrative assumes that you have now added one Key Leaf dimension and one Target Leaf dimension, have chosen your Key leaf dimension values and your Target Leaf dimension values. These steps are the normal setup steps required when building a new rule.

- Click on Coefficient to navigate to the Coefficient tab.
- Enter the coefficient values.

Static Table Driver Definition  
Static Table Driver (New)

Static Table Driver

Name\*

Description

Folder: AUTOSEG

Access Type:  Read Only  Read/Write

Static Table Driver Definition

Organizational Unit	Coefficient
1001	10020

Apply

Definition

Force to 100%  Yes  No

Save Cancel

3. Click **Apply**. Click **Definition** button to view or edit the Key Leaf and Target Leaf dimensions' definition. Deleting Key Leaf or Target Leaf Dimensions

Select the Delete control to remove any Key Leaf or Target Leaf dimension. When you click Delete, a message box is displayed: Are you sure you want to delete this record? Click OK to continue.

### **Force to 100%**

Select either “Yes” or “No” for the value of the “Force to 100%” allocation method (see radio buttons near the bottom of the detail screen). The default value for “Force to 100%” is “Yes”.

Static Driver Table rules are similar in many ways to Dynamic Driver rules. Select “Yes” for the value of Force to 100% when you want to perform a distribution using non-normalized statistics such as headcount, square footage of occupancy, etc. If you are using a normalized set of driver statistics (statistics that sum to 1.00 or 100%) or if you are combining activity counts with activity rates, you need not use Force to 100%.

When you use Force to 100% in a Static Table Driver rule, balances will be distributed on a percent-to-total basis within each row visible on your Coefficient data entry grid.

For example, for the above coefficient values, balances would be distributed from Company A to Cost Center 1 and Cost Center 2 in proportions of 10% and 90%; and balances would be distributed from Company B to Cost Center 2 and Cost Center 3 in proportions of 40% and 60%.

### **Save**

Click on Save to validate and save your rule.

### **Cancel**

Click on Cancel to close the rule without saving any changes you may have made. This control is only active when you are in Edit mode.

## Excel Export

Use this feature to export your Static Table Driver data to a spreadsheet. Within the spreadsheet, you can update the coefficient values for any existing Key ' Target combination and you can add new Key ' Target combinations and their coefficient values. Afterwards, you can import your spreadsheet back into Profitability Management..

The screenshot shows the 'Static Table Driver Definition' window. At the top, there are fields for 'Name\*', 'Description', and 'Folder' (set to 'AUTOSEG'). Below these are radio buttons for 'Access Type' with 'Read/Write' selected. A table below shows columns for 'Organizational Unit' and 'Coefficient'. The first row has '1001' and '10020' under 'Organizational Unit' and an empty box under 'Coefficient'. An 'Apply' button is centered below the table. At the bottom left, there is a 'Definition' button and radio buttons for 'Force to 100%' with 'Yes' selected. At the bottom right, there are 'Save' and 'Cancel' buttons. The 'Export' icon in the top right toolbar is highlighted with a red box.

To export to a spreadsheet:

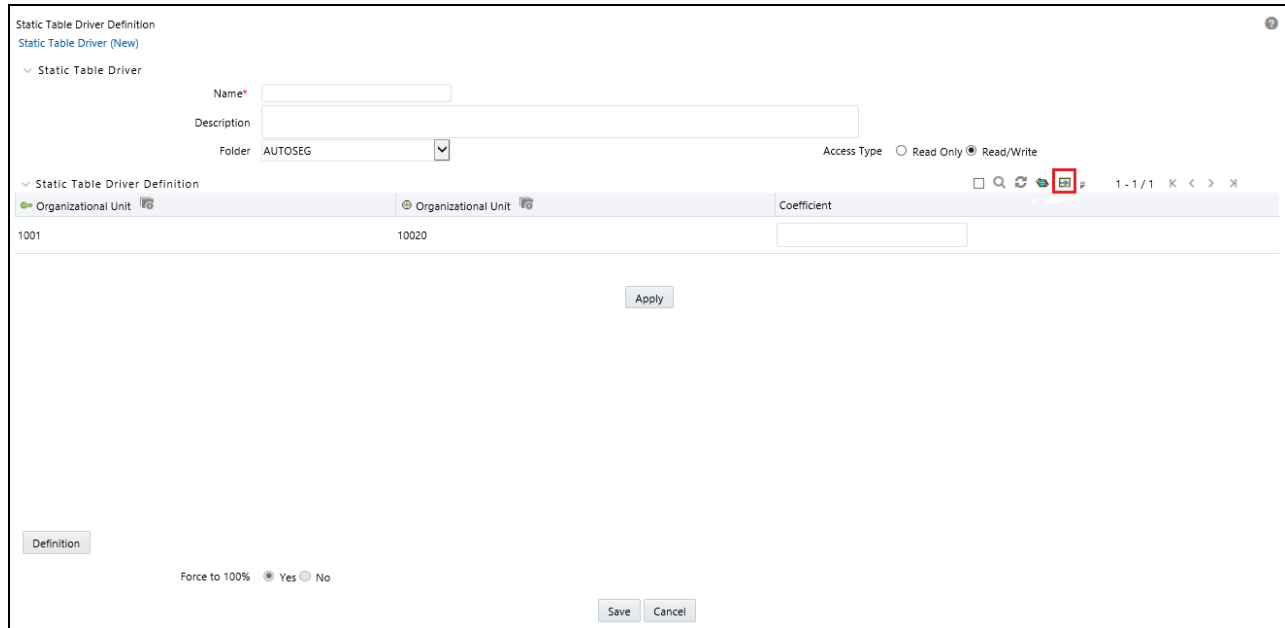
1. Click on Export.
2. Open using Excel or save the file to the name & location of your choice.

### Excel Import

Excel import functionality is used for adding / editing leaf data in static table drivers.

To import a file, follow the below steps:

1. Click **Import** to trigger an Upload File dialog.



2. **Browse** for the spreadsheet that you want to import.
3. Select the file and click **Open**.
4. Click **Upload File**.

### Search

The Search option allows you to search the Leaf values those have coefficient values defined on Static Table Driver Definition page. This functionality works only if **Only Coefficient** option is enabled. You can export this extracted list using the Export functionality.

To perform the search, follow the below steps:

1. Enable the **Only Coefficient** using corresponding check-box.



Static Table Driver Definition  
Static Table Driver (New)

Static Table Driver

Name\*

Description

Folder PFTSEG

Access Type  Read Only  Read/Write

Static Table Driver Definition

Organizational Unit	Financial Element	Organizational Unit	Financial Element	Coefficient
TP Calc Test Funding Center for Org Units	Statistical	10020	Ending Number of Accounts	<input type="text" value="1"/>

Apply

Definition

Force to 100%  Yes  No

Save Cancel

## 2. Click Search.

Static Table Driver Definition  
Static Table Driver (New)

Static Table Driver

Name\*

Description

Folder PFTSEG

Access Type  Read Only  Read/Write

Static Table Driver Definition

Organizational Unit	Financial Element	Organizational Unit	Financial Element	Coefficient
TP Calc Test Funding Center for Org Units	Statistical	10020	Ending Number of Accounts	<input type="text" value="1"/>

Apply

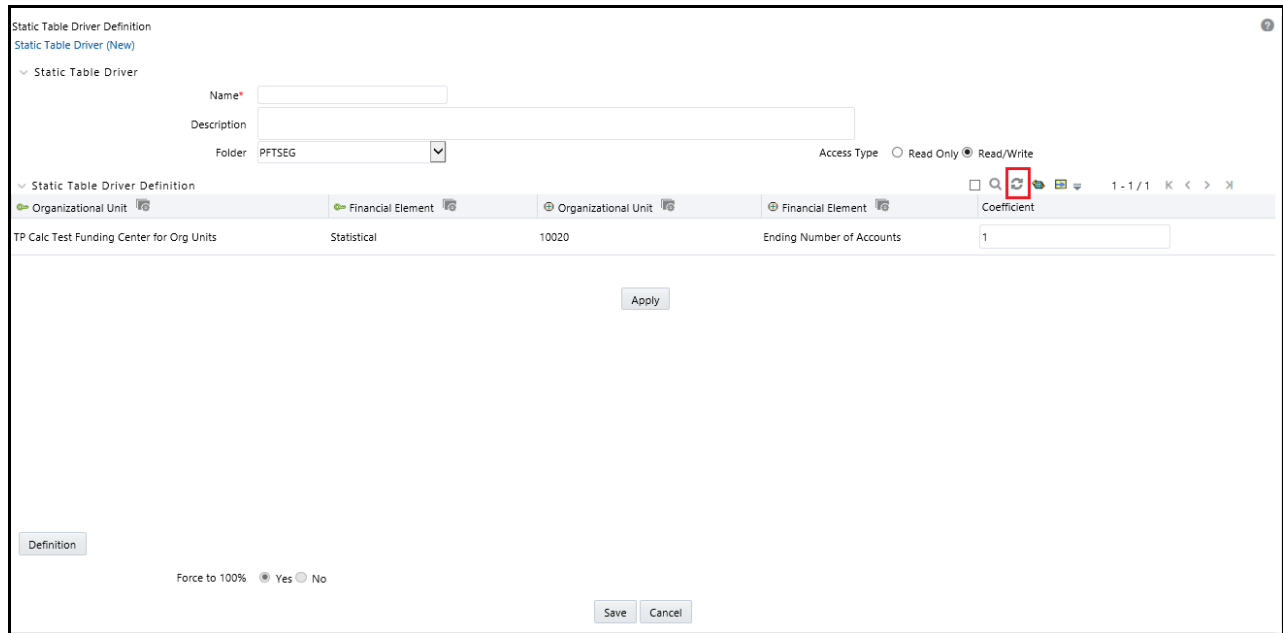
Definition

Force to 100%  Yes  No

Save Cancel

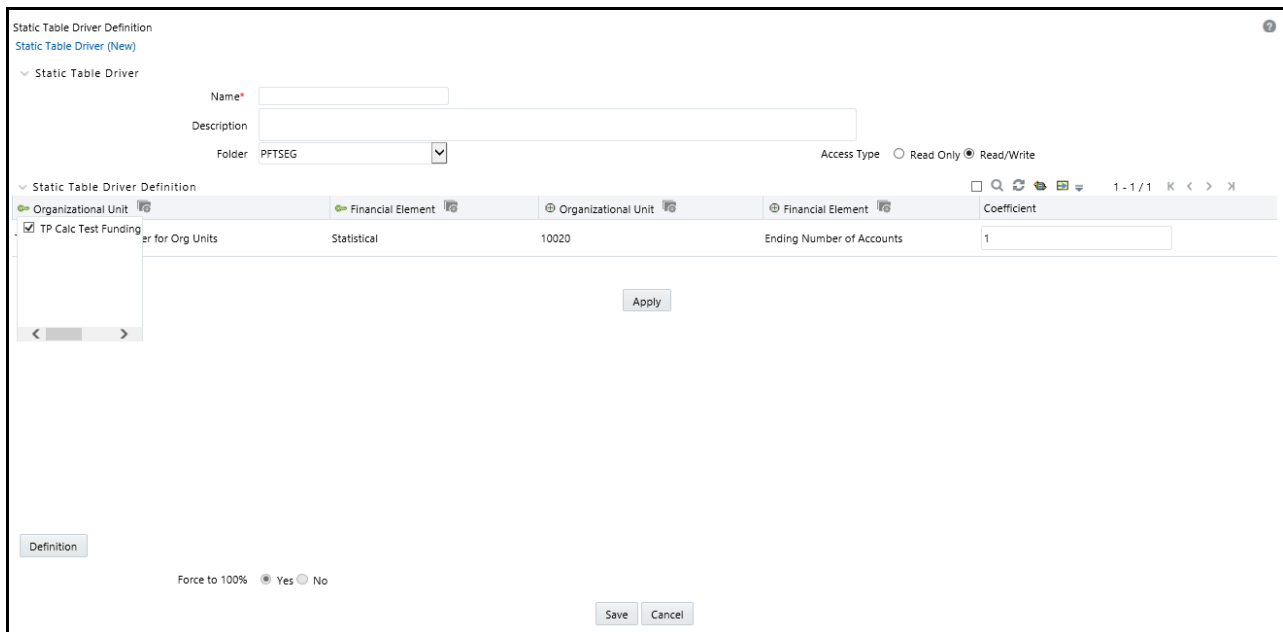
3. It shows the list of Leaf values those have coefficient defined. Also, a red dot (.) will be added next to **Only Coefficient** option. You can also export this list using the Export functionality.

4. Use Reset button to remove the filter.



The Search option allows you to search for specific Leaf.

1. Click ellipses (...) that is available next to Key Leaf or Target Leaf dimension, a list of defined Key Leaf values or Target Leaf values will display. Select the required Leaf values using the corresponding check-boxes.



2. Click **Search**. Only selected Leaf values will be displayed. You can also export this list using the Export functionality.

Static Table Driver Definition

[Static Table Driver \(New\)](#)

Static Table Driver

Name:

Description:

Folder: PFTSEG

Access Type:  Read Only  Read/Write

Static Table Driver Definition

Organizational Unit	Financial Element	Organizational Unit	Financial Element	Coefficient
TP Calc Test Funding Center for Org Units	Statistical	10020	Ending Number of Accounts	<input type="text" value="1"/>

Apply

Definition

Force to 100%  Yes  No

Save Cancel

You can also use the Search option to search the specific leaf with coefficients defined using the **Only Coefficient** check box. You can also export this list using the Export functionality.

### Reset

The Reset option on Static Table Driver Definition page removes any coefficient you may have specified, and refreshes the screen. It also allows you to reset the Search on Static Table Driver Definition page.

### Example

For other uses, you may want only a subset of all possible combinations of Key Leaf values and Target Leaf values. For example, you may wish to allocate a series of Management Ledger balances from cost center to cost center. If you wanted to allocate balances from Company A to Cost Centers 1 and 2; and if you wanted to allocate balances from Company B to Cost Centers 2 and 3, then your

- Key Leaf dimension would be Organizational Unit
- Key Leaf values would be Company A and Company B
- Target Leaf dimension would also be Organizational Unit
- Target Leaf values would be Cost Centers 1, 2, & 3
- Enabled combinations would be A-1, A-2, B-2, and B-3

Click on the Coefficient control. The detail screen now displays a data entry grid. Enter the values for valid Key Leaf value and Target Leaf value combinations.

Static Table Driver Definition  
Static Table Driver (New)

Static Table Driver

Name\*

Description

Folder PFTSEG

Access Type  Read Only  Read/Write

Static Table Driver Definition

Organizational Unit	Financial Element	Organizational Unit	Financial Element	Coefficient
TP Calc Test Funding Center for Org Units	Statistical	10020	Ending Number of Accounts	1

Apply

Definition

Force to 100%  Yes  No

Save Cancel

You may toggle back and forth between specification of valid combinations and entry of coefficient values (Coefficient / Map). You may also return to your Key Leaf dimension tab and modify its definition by adding or removing values. More generally, you are free to navigate at any time between any of your Key Leaf definitions, your Target Leaf definition, your “enabled combinations” definitions, and your coefficient values.

## Working with Exported Static Table Driver Data

At time of initial set up on the Definition tab, a Static Table Driver rule will have a set of possible Key ' Target combinations equal to the cross product of each of the Key and Target leaves you have defined for your rule. For example, a Static Table Driver rule having one Key leaf dimension with 10 defined members and one Target leaf dimension with defined 100 members will have a total of 1,000 possible Key ' Target combinations. Of these 1,000 possible combinations, you may have defined 75, or 125, or any number coefficient values up to 1,000.

The Excel Export feature exports "active" Key ' Target combinations, i.e., combinations for which you have already established coefficients.

The structure of your exported Static Table Driver depends on how many Key dimensions and Target dimensions are used in your rule. The following example shows an exported data a Static Table Driver having one Key leaf dimension and one Target leaf dimension with 11 defined coefficients.

	A	B	C	D	E
1	F0_ID	F0_DESC	T1_ID	T1_DESC	COEFF
2	8100	Executive	8100	Executive	10
3	8100	Executive	8400	Facilities Management	35
4	8100	Executive	8200	Finance Department	60
5	8100	Executive	8300	Human Resources	85
6	8100	Executive	8500	Information Technology Unit	110
7	8400	Facilities Management	8100	Executive	135
8	8400	Facilities Management	8400	Facilities Management	160
9	8400	Facilities Management	8200	Finance Department	185
10	8400	Facilities Management	8300	Human Resources	210
11	8400	Facilities Management	8500	Information Technology Unit	235
12	8200	Finance Department	8100	Executive	260

In this example, the first two columns (F0\_ID and F0\_Desc) contain the leaf identifiers and descriptions for the Key leaf dimension. For a Static Table Driver having a 2nd and 3rd Key leaf dimension, the exported spreadsheet would include ID's and descriptions for each Key leaf dimension (F0, F1, and F2 represent "from" dimensions, i.e., Key leaf dimensions, while T1, T2, and T3 represent your first, second, and third Target leaf dimensions).

### Updating Exported Static Table Driver Data

You may update the coefficient value for any combination in the spreadsheet (yellow cells in example above).

### Adding New Combinations to Exported Data

To add a new combination to your data:

1. Enter a new leaf ID value for each Key leaf dimension you have defined in your rule
2. Enter a new leaf ID value for each Target leaf dimension you have defined in your rule
3. Assign a coefficient to the new combination

The example below (yellow cells) highlights a new combination for Key leaf =8200, Target leaf = 8400, and Coefficient = 285.

---

**Note:** Descriptions are included in the download for convenience. You are not required to provide Descriptions for any Key or Target leaves. Any description values you provide will be ignored when Import into Profitability Management.

---

	A	B	C	D	E
1	F0_ID	F0_DESC	T1_ID	T1_DESC	COEFF
2	8100	Executive	8100	Executive	10
3	8100	Executive	8400	Facilities Management	35
4	8100	Executive	8200	Finance Department	60
5	8100	Executive	8300	Human Resources	85
6	8100	Executive	8500	Information Technology Unit	110
7	8400	Facilities Management	8100	Executive	135
8	8400	Facilities Management	8400	Facilities Management	160
9	8400	Facilities Management	8200	Finance Department	185
10	8400	Facilities Management	8300	Human Resources	210
11	8400	Facilities Management	8500	Information Technology Unit	235
12	8200	Finance Department	8100	Executive	260
13	8200		8400		285

## Validating the Imported Data

On Import, the system validates each row in your spreadsheet. Rows failing validation will not be imported. The validation requirements are as follows:

1. The spreadsheet structure must match the definition of the rule to which it is being imported. This is not an important restriction because the typical workflow for maintaining a Static Driver Table in a spreadsheet begins with an Export of the rule that you want to edit.
2. Each Key leaf value and each Target leaf value must be part of the definition of your Static Table Driver rule. In the case of the example above, the Key leaf value of 8200 and the Target leaf value of 8400 must have been included in the initial set up of the Static Table Driver rule (on the Definition tab).

**Note:** You must define a Key ' Target combination on the Static Table Driver Definition tab before you can add that combination to your spreadsheet.

---

## Large Cross Product Static Table Drivers

Static Table Drivers are generally used in a distributive fashion, and the Static Table Driver user interface is engineered for this typical use case. Some features of the user interface are limited for Static Table Drivers having large numbers of Key leaf values (or node values) in conjunction with large numbers of Target leaf values. For these cases, you must maintain your large Static Table Driver rules using Excel Export/Import functionality.

### Limit on Large Cross Products

The Static Table Driver user interface is designed to allow you specify coefficient values for any combination - or even for every combination - of your Key leaves and Target leaves. For example, a simple Static Table Driver having 50 Key Leaf values (in one Key Leaf dimension) and 100 Target Leaf values (in one Target Leaf dimension) could have as many as 5,000 "active" combinations (i.e., combinations for which coefficients are defined). Typically, the number of defined combinations is a small fraction of the number of possible combinations.

The number of possible combinations is determined by the Cartesian product of the number of leaf values in each dimension in your Static Table Driver rule. For example, a Static Table Driver rule having 5,000 Key Leaf values (in one Key Leaf dimension) and 5,000 Target Leaf values (in one Target Leaf dimension) has a cross product of 25 million. The performance and response time of the Static Table Driver user interface can become degraded with extremely large cross products. For this reason, the application applies a limit of 20 million on the size of the cross product. You may modify the default limit by manually updating `SETUP_PARAMETERS_MASTER.STATIC_TABLEID_TEMP_TABLE_CROSS_JOIN_LIMIT`.

When you are working with a Static Table Driver whose cross product exceeds the cross join limit, the Coefficients tab of the Static Table Driver user interface will display only "defined" combinations, i.e., combinations for which coefficients have previously been established.

## Managing Large Cross Product Static Table Drivers

### New Rules

To build a Static Table Driver that will have a very large cross product, follow the normal steps of building out your Key leaf (or node) values and your Target leaf values on the Static Table Driver Definition tab. If you start with a modest number of defined Key and Target leaves, the Coefficients tab will operate normally and you may save coefficients for any possible combination of Key and Target leaves that you have defined on the Definition tab.

If you navigate to the Definition tab and add more Key leaves and more Target leaves and then return to the Coefficients tab, the Coefficients tab will again allow you to edit coefficients for any possible combination of your selected Key & Target leaves so long as you have not yet exceeded the cross product limit. Once you have added enough combinations on the Definition tab for the rule to exceed to cross product limit, the Coefficients tab will no longer show unmapped combinations. In this state, you can edit the coefficient values for any previously mapped combination but you can no longer establish new combinations & coefficient values within the Coefficients tab.

---

**Note:** Instead of starting the build process with a modest number of Key leaves and Target leaves, you might also begin by specifying a very large number of Key and Target leaf values. If you specify enough values in the Definition tab to exceed the cross product limit, then when you initially transition to the Coefficients tab it will appear blank.

---

Regardless of whether your Static Table Driver rule begins as a small rule and evolves into a large cross product rule or your Static Table Driver rule was "large" at the time it was initially defined, the Coefficients tab will only display "defined" combinations once you have exceeded the cross product limit.

## **Editing Existing Coefficient Values**

You can edit the coefficient value for any defined combination within the user interface. Alternatively, you can also export your data and edit coefficient values offline.

## **Adding New Combinations & Coefficients**

1. Navigate to the Definition tab and add new Key & Target leaf values.
2. Navigate to the Coefficient tab and export your rule to a spreadsheet.
3. Add your new combinations and their coefficients to the spreadsheet.
4. Save and import the spreadsheet.

## **Removing Combinations & Coefficients**

You can remove coefficients within the Coefficient tab by deleting the coefficient and saving the rule. Once you have removed the coefficient for a combination of Key ' Target leaves, that combination will no longer be updatable within the Coefficient tab, but the combination remains defined (i.e., you could still add a coefficient to this combination offline).

To completely remove defined Key and Target leaf values, navigate to the Definition tab and remove the leaves that are no longer required.

## ***Using Static Table Drivers***

Static Table Drivers are declared as drivers in the Driver process tab for allocation rules of the type Static Driver Table (see [Chapter 16, "Allocation Specification,"](#)).

Static Drivers Table allocation rules are similar in many ways to Dynamic Driver allocation rules. Both are used to distribute balances, but while Dynamic Driver allocation rules obtain their driver data directly from your business data, Static Driver Table allocation rules obtain their driver data from a Static Table Driver rule.

In the example that follows, your goal is to “product align” your Management Ledger data (your initial General Ledger data is aligned to Organizational Unit and to General Ledger Account, but not to Product). You need to write an allocation rules that distributes expenses within each cost center to Product. In this example, you are focusing on expenses incurred in two rollup points within your Organizational Unit hierarchy: Mortgage Origination (a rollup point of multiple regional origination centers) and Statement Processing (a rollup point of multiple statement processing centers).



Organizational Unit	Product	Coefficient
Mortgage Organization	30 Year Fixed Mortgage	55
Mortgage Organization	15 Year Fixed Mortgage	45
Mortgage Organization	Savings	
Mortgage Organization	Time Deposits	
Statement Processing	30 Year Fixed Mortgage	
Statement Processing	15 Year Fixed Mortgage	
Statement Processing	Savings	45
Statement Processing	Time Deposits	55

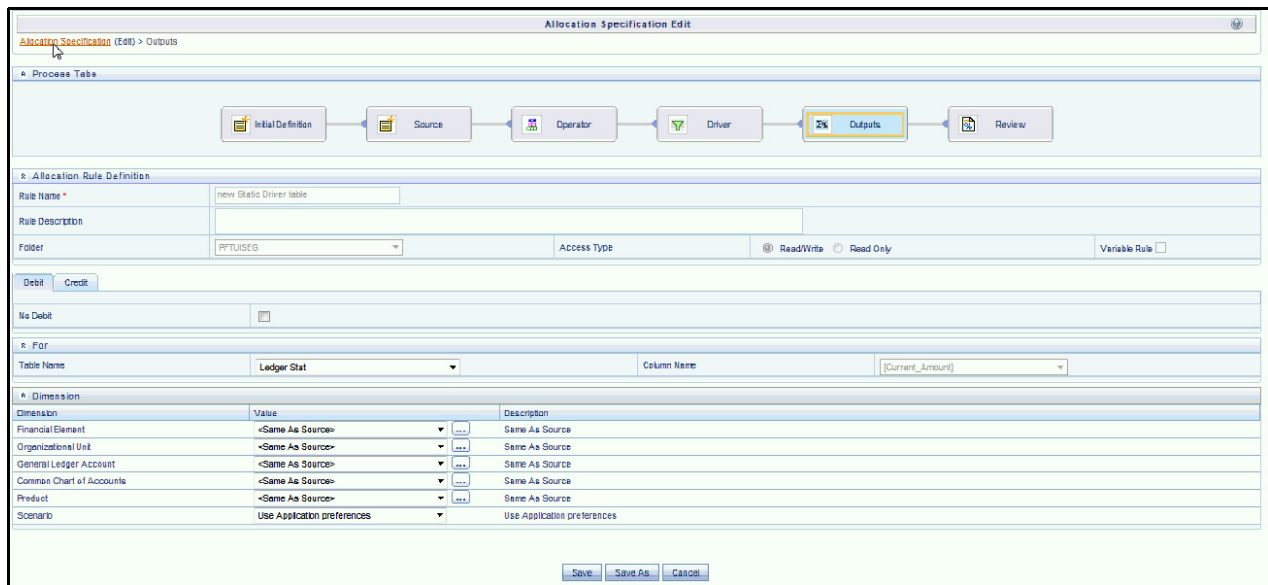
Your cost studies have told you that 55% of mortgage origination expense is attributable to your 30 Year Fixed Mortgage product, and 45% to your 15 Year Fixed product.. At the same time, historical balance reports tell you that 55% of your retail deposits are Savings, and 55% are Time Deposits. On the assumption that balance ratios are a good way to distribute Statement Processing expense, you decide to use a 45-55 split.

To build this allocation, start by constructing a Static Table Driver rule that uses an Organizational Unit Key Leaf and a Product Target Leaf. Select the Mortgage Origination and Statement Processing rollup points from your organizational hierarchy for your Key Leaf values. Select the 3 mortgage products plus the Checking, Savings, and Time Deposits products as your Target Leaf values. Finally, enable the appropriate combinations of Key Leaf and Target Leaf and enter your coefficient values.

Next, build a new Static Driver Table allocation rule. In the Driver tab, select the Static Table Driver you just built. In the Source tab, constrain the General Ledger Account dimension with a rollup point whose meaning is Total Non Interest Expense; alternatively, you might specify Financial Element leaf value 457 – Non Interest Expense.

Note that you could specify an Organizational Unit constraint in your Source specification narrowing the source data down to just the Mortgage Origination and Statement Processing rollup points. Doing so is not strictly necessary as the allocation rule will insist on matching of Source cost centers and Driver cost centers and you have already constrained the Driver cost centers in your Static Table Driver rule.

In your allocation Output Debit, specify < Same as Table > for the Organizational Unit dimension and specify < Same as Source > for every other dimension.



Note the similarity between the Static Driver Table allocation rule defined above and a very similar allocation built using dynamic drivers. If the statistics we used in the Static Table Driver were available from an external source that we could load every month to the data model, we could achieve the same results with a Dynamic Driver allocation rule. The Output Debits in the table below show the values you would use for the two rule types.

Dimension	Dynamic Driver Allocation Debit Definition	Static Table Driver Leaf Type	Static Driver Table Allocation Debit Definition
Organizational Unit	< Match Source & Driver >	Key Leaf	< Same as Source >
General Ledger Account	< Same as Source >	Not defined	< Same as Source >
Product	< Match Driver >	Target Leaf	< Same as Table >

---

**CHAPTER 19**

# Lookup Table Driver

## Overview

This chapter describes Lookup Table Driver functionality.

- Lookup Table Driver Summary & Detail Screens
- Navigation within the Summary Screen
- Navigation within the Detail Screen
- Using Lookup Table Drivers

## Summary & Detail Screens

Upon initially navigating to Profitability Management > Rule Specification > Lookup Table Driver, a summary screen is displayed showing a set of Lookup Table Driver rules. Using search criteria, you can control the set of rules that are displayed. When you Add, Edit, or View a rule, a detail screen is displayed.

## Navigation within the Summary Screen

When you first navigate to the Lookup Table Driver summary screen, the rules stored within your current default Folder are presented in a summary grid. The Lookup Table Driver summary screen has two containers: Search and Lookup Table Driver.

### Search Container

Your default Folder functions as a search constraint. The value of your default Folder is set in Application Preferences for Profitability Management. You may select a different Folder or you may remove the Folder constraint entirely by selecting the “blank” Folder, i.e., no Folder. You may also search by a rule's Name or Description.

### Search Control



You may search for Lookup Table Driver rules by Folder, Name, or Description. Enter your desired search criteria and click on the Search control.

### Reset Control



Restores the default Folder, removes any Name or Description constraint you may have specified, and refreshes the screen.

### **Lookup Table Driver Container**

The Lookup Table Driver container presents a grid containing all of the Lookup Table Driver rules that meet your search criteria. The Lookup Table Driver summary grid offers several controls that allow you to perform different functions when a rule is selected.

To select a rule, click on a check box in the first column of the grid. More than one rule can be selected at a time but this will cause some of the controls to become disabled. Clicking on a check box a second time de-selects the rule.

You may select or deselect all of the Lookup Table Driver rules in the summary grid by clicking on the check box in the upper left hand corner of the summary grid directly to the left of the Name column header.

#### **Add**



Clicking on the Add control begins the process of building a new Lookup Table Driver rule. The Add control is disabled if any rows in the grid have been selected.

#### **View**



Selecting a single row out of the grid enables the View control. Clicking on the View control allows you to view the contents of a Lookup Table Driver rule on a read-only basis. The View control is only enabled when a single Lookup Table Driver rule has been selected.

#### **Edit**



Selecting a single row out of the grid enables the Edit control. Clicking on the Edit control allows you to modify a previously saved Lookup Table Driver rule. The Edit control is only enabled when a single Lookup Table Driver rule has been selected.

#### **Delete**



Selecting one or more rows out of the grid enables the Delete control. Clicking on the Delete control deletes the Lookup Table Driver(s) you have selected. OFSAAI will not allow you to delete Lookup Table Driver rules which have any dependencies (see View Dependencies below).

#### **Copy**



Selecting a single row out of the grid enables the Copy control. Clicking the Copy control allows you to create a copy of an existing Static Table Driver rule. The Copy control is only enabled when a single Static Table Driver rule has been selected. When you click on Copy, a Save As pop window will appear. Click Save after entering the Name, Description, Folder, and Access Type Details.

### View Dependencies



Clicking on the View Dependencies control generates a report on any Allocation rules that depend on the Lookup Table Driver rule you have selected. The View Dependencies control is only enabled when a single rule has been selected.

### Lookup Table Driver Summary Grid

The following columns categorize each allocation rule in the summary grid:

- Name
- Creation Date
- Created By
- Description
- Last Modified Date
- Modified By
- Access Type
- Folder

### Lookup Table Driver Summary Screen

Lookup Table Driver							As of Date: 06/25/2014
<input type="text" value="Search"/>							<input type="text" value="Name"/>
Folder: <input type="text" value="AUTOSEG"/>		<input type="text" value="Description"/>					<input type="text" value="Name"/>
Name	Creation Date	Created By	Description	Last Modified Date	Modified By		
<input type="checkbox"/> 81_EXPR_MGMT_LED_1	05/12/2016 20:24:12	PFTUSER		05/25/2016 12:58:46	PFTUSER		
<input type="checkbox"/> 81_EXPR_MGMT_LED_2	05/12/2016 20:24:46	PFTUSER		05/30/2016 13:59:51	PFTUSER		
<input type="checkbox"/> 81_PORTFOLIO_LOOKUP	05/17/2016 08:19:48	PFTUSER		06/01/2016 10:47:21	PFTUSER		
<input type="checkbox"/> AUTOD_LTD_01	02/15/2018 14:41:24	PFTUSER		02/15/2018 14:41:24	PFTUSER		
<input type="checkbox"/> AUTOD_LTD_02	12/06/2017 10:52:57	PFTUSER	TESTING	12/06/2017 10:52:57	PFTUSER		
<input type="checkbox"/> AUTOD_LTD_03	02/15/2018 15:16:40	PFTUSER		02/15/2018 15:17:27	PFTUSER		
<input type="checkbox"/> PFT_AUTO_LTD_34_copy	04/30/2018 08:43:35	PFTUSER		04/30/2018 08:43:35	PFTUSER		
<input type="checkbox"/> PFT_AUTO_LTD_34_Edit	04/27/2018 16:53:43	PFTUSER	TESTING	04/27/2018 17:01:33	PFTUSER		

## Navigation within the Detail Screen

When you Add, Edit, or View a Lookup Table Driver rule, the Lookup Table Driver Detail screen is displayed. In addition to Name, Description, Folder, and Access Type, the definition of a Lookup Table Driver includes the specification of a source table and a lookup table, the mapping of source columns to lookup table columns, a lookup table filter (optional), and a lookup table return column.

The Audit Trail container is a standard footer container for every OFSAA rule type. It displays Created By, Creation Date, Last Modified By, and Modification Date on the Audit Trail tab. The User Comments tab may be used to add comments to any rule, subject to a maximum of 4000 characters.

### Lookup Table Driver Definition Container

Specify the Lookup Table Driver rule's Name and Description, select a Folder in which the Lookup Table Driver rule is to be stored, and specify whether you want the Lookup Table Driver rule to be “Read/Write” or “Read Only” (Access Type). Naming your Lookup Table Driver rule is required before it can be saved. Default values for Folder and Access Type are stored in Application Preferences.

### Lookup Table Driver Functionality

Lookup Table Driver rules are used in conjunction with Allocation rules (of the Lookup Driver Table type) to match Instrument level data with data from user defined lookup tables. Each Instrument table row retrieved within the Allocation rule's Source definition is matched with your lookup table in order to return a lookup table factor. For each row, the resulting lookup table factor is arithmetically combined (typically multiplication) with the column specified in the Allocation rule's Source definition in order to update another column within the same row. A very typical use case might be the allocation of Loan Loss Reserves, Economic Loan Loss Provision, or Credit Risk Capital to each of your commercial loan instruments as a function of Product, Remaining Term to Maturity, and Credit Rating. Static Driver Table rules also support this kind of “matching”, but only for key processing dimensions (only for Product in this example). Lookup Table Driver rules extend the functionality of Static Table Driver rules by allowing you to also match on Instrument level measures or attributes (Remaining Term to Maturity and Credit Rating in this example).

Additional examples of how you might use a Lookup Table Driver rule include the following kinds of assignments:

- Risk equity as a function of Product (a dimension), Division (a rollup point within a dimension), Credit Score (an instrument level attribute), and Remaining Term to Maturity (also an instrument level attribute)
- Loan Loss Reserve or Economic Provision (expected loss) as a function of Product (a dimension), Amortization Type (an instrument level attribute), and Loan to Value Ratio (also an instrument level attribute).
- Account Maintenance Expense as a function of Product (a dimension) and Current Net Book Balance (an instrument level measure)

## Source and Lookup Tables Container

Select a Source instrument table and a Lookup table. The tables available in the Lookup drop down list box are limited to tables that you have registered as Lookup Tables within your data model (see *Oracle Financial Services Analytical Applications Data Model Utilities User Guide* for details on creating and registering Lookup Tables.)

The screenshot shows the 'Lookup Table Driver Definition' form. It is divided into three main sections:

- Lookup Table Driver Definition:** Contains fields for Name, Description, Folder (set to PFTSEG), and Access Type (Read Only and Read/Write radio buttons).
- Source and Lookup Tables:** Contains fields for Source (Guarantees) and Lookup (PFT Lookup.Template.Table).
- Source - Lookup Mapping:** A table with columns for Range Lookup, Account classification, and Maximum Average Balance. The first three rows are 'Range Lookup' with 'Account classification' and 'Maximum Average Balance' dropdowns. The last row is 'Select LookupType' with 'Select Source Column' and 'Credit Status Code' dropdowns.

When you select a Lookup table, the Source – Lookup Mapping container (described in the following sections) responds by displaying one row for each lookup column within your registered lookup table.

Click the following button adjacent to the **Lookup** drop-down list to navigate to the Lookup Table Data Information page.



Below is an example of the Lookup Table Data Information page. The actual columns will depend on the Lookup Table you selected.

**Lookup Table Data Information**

▼ PFT Lookup Template Table + 🔄 1 - 20 / 1253 ⌂ ⏪ ⏩

<input type="checkbox"/>	Maximum Average Balance	Minimum Average Balance	Credit Status Code	Maximum Dealer Number	Minimum Dealer Number	Maximum Issue Date	Minimum Issue Date	Organization Unit Code	Return Value 1
<input type="checkbox"/>	6000000	4000001	1	125	100	1993-09-30	1993-06-30	400130004	2
<input type="checkbox"/>	6000000	4000001	1	125	100	1993-12-31	1993-09-30	400130001	3
<input type="checkbox"/>	6000000	4000001	1	125	100	1993-12-31	1993-09-30	400130004	3
<input type="checkbox"/>	6000000	4000001	1	125	100	1994-03-31	1993-12-31	400130001	1
<input type="checkbox"/>	6000000	4000001	1	125	100	1994-03-31	1993-12-31	400130004	4
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-03-31	1993-01-31	400130001	3
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-03-31	1993-01-31	400130004	1
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-06-30	1993-03-31	400130001	5
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-06-30	1993-03-31	400130004	5
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-09-30	1993-06-30	400130001	2
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-09-30	1993-06-30	400130004	4
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-12-31	1993-09-30	400130001	3
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-12-31	1993-09-30	400130004	4
<input type="checkbox"/>	6000000	4000001	1	150	125	1994-03-31	1993-12-31	400130001	3
<input type="checkbox"/>	6000000	4000001	1	150	125	1994-03-31	1993-12-31	400130004	3
<input type="checkbox"/>	6000000	4000001	1	175	150	1993-03-31	1993-01-31	400130001	4

### Adding a Row

You can add a new row to selected Lookup table using the below button:



A new row will be added. Enter the values for row and click **Save**.

**Lookup Table Data Information**

▼ PFT Lookup Template Table + 🔄 1 - 20 / 1253 ⌂ ⏪ ⏩

<input type="checkbox"/>	Maximum Average Balance	Minimum Average Balance	Credit Status Code	Maximum Dealer Number	Minimum Dealer Number	Maximum Issue Date	Minimum Issue Date	Organization Unit Code	Return Value 1
<input type="checkbox"/>	6000000	4000001	1	125	100	1993-09-30	1993-06-30	400130004	2
<input type="checkbox"/>	6000000	4000001	1	125	100	1993-12-31	1993-09-30	400130001	3
<input type="checkbox"/>	6000000	4000001	1	125	100	1993-12-31	1993-09-30	400130004	3
<input type="checkbox"/>	6000000	4000001	1	125	100	1994-03-31	1993-12-31	400130001	1
<input type="checkbox"/>	6000000	4000001	1	125	100	1994-03-31	1993-12-31	400130004	4
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-03-31	1993-01-31	400130001	3
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-03-31	1993-01-31	400130004	1
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-06-30	1993-03-31	400130001	5
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-06-30	1993-03-31	400130004	5
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-09-30	1993-06-30	400130001	2
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-09-30	1993-06-30	400130004	4
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-12-31	1993-09-30	400130001	3
<input type="checkbox"/>	6000000	4000001	1	150	125	1993-12-31	1993-09-30	400130004	4
<input type="checkbox"/>	6000000	4000001	1	150	125	1994-03-31	1993-12-31	400130001	3
<input type="checkbox"/>	6000000	4000001	1	150	125	1994-03-31	1993-12-31	400130004	3
<input type="checkbox"/>	6000000	4000001	1	175	150	1993-03-31	1993-01-31	400130001	4



### Deleting Rows

You can delete the row(s) from selected Lookup table using the below button:



### Excel Export / Import

Excel export / import functionality is used for adding / editing Lookup Table Data Information in Lookup Table Drivers. Click the following button to access the Export/Import functionality.



Lookup Table Data Information									
1	125	100	1994-03-31	1993-12-31	400130004	4			
1	150	125	1993-03-31	1993-01-31	400130001	3			
1	150	125	1993-03-31	1993-01-31	400130004	1		3	
1	150	125	1993-06-30	1993-03-31	400130001	5		5	
1	150	125	1993-06-30	1993-03-31	400130004	5		5	
1	150	125	1993-09-30	1993-06-30	400130001	2		2	
1	150	125	1993-09-30	1993-06-30	400130004	4		3	
1	150	125	1993-12-31	1993-09-30	400130001	3		5	
1	150	125	1993-12-31	1993-09-30	400130004	4		1	
1	150	125	1994-03-31	1993-12-31	400130001	3		2	
1	150	125	1994-03-31	1993-12-31	400130004	3		3	
1	175	150	1993-03-31	1993-01-31	400130001	4		1	
1	175	150	1993-03-31	1993-01-31	400130004	3		3	
1	175	150	1993-06-30	1993-03-31	400130001	3		2	
1	175	150	1993-06-30	1993-03-31	400130004	5		5	
1	175	150	1993-09-30	1993-06-30	400130001	1		5	

### Source – Lookup Mapping Container

Lookup Table Driver rules support three types of matching:

- Range Lookup
- Exact Match
- Hierarchy Match

Lookup Type	Source	Hierarchy			Single Lookup	Range Lookup
		Folder	Hierarchy Name	Level Name		
Range Lookup	Account classification				Maximum Average Bi	Maximum Average Bi
Range Lookup	Account classification				Maximum Average Bi	Maximum Average Bi
Range Lookup	Account classification				Maximum Average Bi	Maximum Average Bi

## Range Lookup

One of your options in defining a lookup table is to define a minimum column and a maximum column to be employed in a “Range Lookup” against each row of selected instrument data. For example, you may wish to assign an Account Maintenance fee against certain checking account products as a function of balances ranges, e.g., one fee amount for accounts having balances between 0 and 1,000 and a different fee amount for balances between 1,000 and 5,000, and a third fee amount for accounts having balances greater than 5,000.

If the lookup table you chose in the “Source and Lookup Table” container (discussed above) includes range lookup columns, a Range Lookup row (i.e., a row whose Lookup Type is Range Lookup) will automatically be generated within the Source – Lookup Mapping container. Within this automatically generated row, select the source column that you want to compare to the Range Lookup columns from your lookup table. In the example described above, you might want to compare Average Net Book Balance for each account with the range values from your lookup table. Range Lookup supports all instrument level measures, attributes, and dimension members (numbers, dates, or strings).

**Note:** Range Lookups require that your lookup data not have overlapping ranges that would lead to ambiguous lookup values.

## Exact Match

An exact match is a literal database join. Exact Match supports all instrument level measures, attributes, and dimension members (numbers, dates, or strings).

## Hierarchy Match

Similar to Static Table Drivers, Lookup Table Drivers allow you to match leaf values from a Source instrument table with any leaf member that belongs to a hierarchical rollup point of that leaf. You might have, for example, sets of driver statistics that vary by region where regions can be defined as rollup points in an Organizational Unit hierarchy. If you had North, South, East, and West regions, you could store your lookup data in 4 regional sets. If there were 300 cost centers in the West region, by using Hierarchy Match functionality you avoid the repetition of 299 sets of otherwise identical driver data for the West region. Hierarchy matching is only supported for Key Processing dimensions.

## Lookup Table Filters Container

You may constrain the data within your selected lookup table by applying a Lookup Table Filter. Chose No Filter, Data Filter, or Group Filter. If you have chosen either Data Filter or Group Filter, continue by selecting a Folder and a Filter Name. Note that the Filter Name drop down list box will only display filters that are applicable to your chosen lookup table.

## Lookup Return Value

Specify the column within your lookup table from which to return a value for each lookup.

Lookup Table Return Value		Column/Expression <input checked="" type="radio"/> Column Name <input type="radio"/> Expression
Lookup Table Name	PFT Lookup Template Table	
Lookup Return Value	Return Value 1	

Lookup tables may contain multiple lookup columns. For example, you may define a lookup table called Risk Factors that contains return columns for Credit Risk Factor, Operating Risk Factor, Economic Loan Loss Provision Factor, and Loan Loss Reserve Factor. In this example, you could subsequently define 4 separate Lookup Table Driver rules to be used within 4 separate Allocation rules (one Lookup Table Driver rule and one Allocation rule for each defined lookup column). In this example, each of your Allocation rules might utilize the same instrument column source (as defined in each Allocation rule's Source definition), e.g., Average Balance.

As another example, you might define a lookup table called Expense Factors that contains return columns for Account Maintenance Expense, Account Origination Expense, ATM Transaction Unit Cost, and Check Processing Unit Cost. In this example, you might develop 4 Lookup Table Driver rules and 4 Allocation rules. Here, you would probably utilize different Source columns within your Allocation rule definitions. For maintenance expense and origination expense, you might choose to allocate a flat amount for each account (e.g., use the value of 1.00 for each account; you may accomplish this using Record Count as the Source column since the Record Count column within Instrument tables is typically set to 1). For ATM expense and Check Processing expense, you might utilize Instrument source columns of ATM Transaction Count and Number of Checks Processed.



***Allocation Execution History Functionality***

This chapter describes Allocation Execution History functionality.

- UNDOing the results of previously executed allocation rules
- Viewing rule definitions in effect when an allocation rule was executed
- Viewing inline reports of data generated by an allocation rule's Source, Driver, and Output components for a selected rule execution
- Viewing listings of distinct data sources that contributed to an allocation's Source
- Viewing listings of subsequent allocation runs which included any of the current allocation rule's Outputs in their Source queries
- Cross referencing to or from the execution history of any allocation rule

This chapter presents the following topics:

- Allocation Execution History User Interface
- UNDO Functionality
- Allocation Execution Audit Functionality
- Source Tab
- Operator Tab
- Driver Tab
- Outputs Tab
- Trace Allocation Tab

***Allocation Execution History User Interface***

Allocation Execution History allows you to review the runtime history of any allocation rule. You may see the order in which allocation rules completed, you may UNDO the results generated by any allocation rule, and you may drill in to the details of individual allocation executions to audit what a rule did at the time that it ran.

The Allocation Execution History summary screen presents a grid that lists a series of allocation runs sorted by As-of-Date and by Completion Date. You may sort on any column you choose by clicking on a column title, but the results will always be sorted first by As-of-Date and second by the column you have chosen.

When you first enter the Allocation Execution History summary screen, your results are filtered down to the most recent As-of-Date for which there is any allocation rule history. You may select a different As-of-Date from a drop-down list, but the only As-of-Date's that are offered are those for which allocation history exists. You may optionally select "All As-of-Dates".

## As of Date

Controls how much rule execution history is displayed in the body of the summary grid. Here, the latest As of Date is displayed as default. When you select As of Date option as **All As of Dates**, the **Allocation Execution History Summary Grid** will display the Allocation Names based on **As of Date** sorting and then **Completion Date** (that is, based on As of Date).

## Search Container

The search container is used to specify criteria that will reduce the number of rule executions that will be displayed in the Allocation Execution History summary grid. You may search based on:

- Allocation Name
- Batch Name
- Ranges of completion Date and Time
- Allocation Model

These constraints function as radio buttons, and the functionality of each type of search is explained below.

As of Date	Folder	Allocation Name	Table	Column	Record Count	Start Date	Completion Date	Elapsed Time	Batch Name	Status	Allocation Model Name	Allocation Model Sequence
09/12/2017	AUTOSEG	test_constant_01	FSI_D_MANAGEMENT_LEDGER	[Current_Amount]	2	09/12/2017 20:39:58	09/12/2017 20:39:59	00:00:01	OFSPFTINFO_Task1_200305_EXEC_20170912_1	Completed	Test_am_01	1
09/12/2017	AUTOSEG	Test_LTD_01	FSI_D_ANNUITY_CONTRACTS	ACCRUED_INTEREST	0	09/12/2017 20:26:13	09/12/2017 20:26:16	00:00:03	OFSPFTINFO_Task1_200278_EXEC_20170912_1	Completed		
09/12/2017	AUTOSEG	test_field	FSI_D_MANAGEMENT_LEDGER	[Current_Amount]	4	09/12/2017 20:24:42	09/12/2017 20:24:43	00:00:01	OFSPFTINFO_Task1_200260_EXEC_20170912_1	Completed		
09/12/2017	AUTOSEG	Test_leaf_01	LEDGER_STAT	*	0	09/12/2017 20:23:45	09/12/2017 20:23:48	00:00:03	OFSPFTINFO_Task1_200251_EXEC_20170912_1	Completed		
09/12/2017	AUTOSEG	Test_SD_01	FSI_D_MANAGEMENT_LEDGER	[Current_Amount]	4	09/12/2017 20:22:29	09/12/2017 20:22:31	00:00:02	OFSPFTINFO_Task1_200242_EXEC_20170912_1	Completed		
09/12/2017	AUTOSEG	test_constant_01	FSI_D_MANAGEMENT_LEDGER	[Current_Amount]	2	09/12/2017 20:20:55	09/12/2017 20:20:56	00:00:01	OFSPFTINFO_Task1_200203_EXEC_20170912_1	Completed		

### Allocation Name

Performs a wild card search on Allocation Name. The search is executed when the user selects the Search button.

### Batch

Performs a wild card search on Batch Name. The search is executed when the user selects the Search button.

### Completion Date

Restricts the number of rules displayed on the Allocation Execution History summary screen based on a user's specification of a single execution date and a “no earlier than” starting time and a “no later than” ending time. The search is executed when the user selects the search button.

### Allocation Model

Allows you to view the allocation rules those are grouped under a allocation model. Allocation Model consists of a list of individual allocation rules that can be executed as a single unit.

**Search**

After entering search criteria, the user selects the Search icon to perform the search function. The As-of-Date from the first container on the screen is always applicable when the Summary screen is initially built or is refreshed. You may, however, select the “All As-of-Dates” option that effectively removes the As-of-Date as a constraint.

**Reset**

Removes values entered as search criteria and refreshes the screen.

**Allocation Execution History Container**

The Allocation Execution History summary grid offers several controls that allow the user to perform different functions when an allocation rule is selected. More than one Allocation Rule can be selected at a time but this will cause some of the icons to become disabled.

**View**

After having chosen a single row out of the grid, selecting the View icon drills into an audit trail for the currently selected rule. The View icon is disabled if no rows have been selected or if multiple rows have been selected. See “Allocation Execution Audit Functionality” below for details.

---

**Note:** Allocation Execution History View functionality for the allocations using the Management Ledger table is not yet supported but will be supported in a future release.

---

**UNDO**

This icon is used to UNDO or reverse one or more allocation runs. For details, see UNDO Functionality below.

**Refresh**

This icon refreshes the contents of the Summary Grid. Users may want to do this if they have requested an UNDO or if they are simply watching as allocation rules currently running in a batch complete. Selecting Refresh will not alter any Search criteria you may have selected.

**Pagination Options**

As with all Summary screens, the maximum number of rows displayed per page is generally governed by the Pagination Records value you set in Global Preferences. You may use this local “Pagination Options” control to override your normal default value. You may set your local value using Pagination Options of 10, 25, 50, or any value between 1 and 50.

**VCR Buttons**

VCR Buttons allow you to navigate between pages.

**Allocation Execution History Summary Grid**

The following columns categorize each allocation rule in the summary grid:

You can sort any of these columns by clicking on the text in the column header. Explanations of the meaning of each column header are as follows:

- As of Date
- Folder

- Allocation Name
- Table
- Column
- Record Count
- Start Date
- Completion Date
- Elapsed Time
- Batch Name
- Status
- Allocation Model Name
- Allocation Model Sequence

**As-of-Date** - Displays the As-of-Date for which the allocation rule was executed.

**Folder** - Displays the name of the Folder to which the allocation rule belong.

**Allocation Name** - Displays the allocation rule's short name. A “mouse over” on the rule's name displays the rule's long description as well as the Identity Code generated or used by that allocation. Identity Codes may be used in queries against the database to identify rows generated by the rule.

**Table** - This denotes the target table name in the database where the allocation either posted or updated records. For example, Ledger Stat, Management Ledger, and so on.

**Column** - This denotes the target column name in the database where the allocation either posted or updated records.

**Record Count** - Displays the Record count as 1 or 0. To allocate a flat amount for each account, use Record Count as 1.

**Start Date** - Displays the date and time at which the each allocation rule started.

**Completion Date** - Displays the date and time at which each allocation rule completed.

**Elapsed Time** - Displays the elapsed time required for each allocation rule to complete shown in hours, minutes & seconds.

**Batch Name** - Displays the Batch name under which each allocation rule was executed. For rules executed directly from the Allocation Specification user interface, the system automatically generates synthetic batch names.

**Status** - Denotes the status of each allocation rule. Values include “Completed” and “Marked for UNDO”. Rules that are “Marked for UNDO” are in the process of being UNDONE. This status will normally disappear in just a few seconds or minutes. To refresh the status of allocations that are “Marked for UNDO”, click on the Refresh control.

**Allocation Model Name** - Displays the allocation model name under which the allocation rule is grouped. Allocation Model consists of a list of individual allocation rules that can be executed as a single unit.

**Allocation Model Sequence** - Displays the sequence number that is, how many times the Allocation Model was executed.



---

**Note:** **Allocation Model Name** and **Allocation Model Sequence** columns will be blank for any **Allocation Rule** that was not executed from within an **Allocation Model**.

---

### **UNDO Functionality**

For Allocations that insert or update data in the Management Ledger and Ledger Stat tables, an UNDO operation reverses the effect of that rule run. For allocations that update data in instrument or transaction tables, an UNDO operation updates the column targeted by that allocation run with a value of zero.

Because UNDOING a rule effectively removes completed allocation data from your environment, your management may wish to restrict your access to UNDO functionality. For details on restricting access to UNDO functionality, see Application Preferences. The UNDO icon is enabled whenever you select one or more rows from the summary grid. Upon requesting an UNDO operation, a pop-up dialog appears offering the following UNDO options:

- Selected Allocations
- Selected Batches
- Selected Allocation Models
- Mass UNDO
- If you choose "Selected Allocations", each row that you have selected from the summary grid will be **UNDONE**. This functionality works only if **Enable Undo for Selected Allocations** option in the Application Preferences is set to Yes.
- If you choose "Selected Batches", all rule executions that belong to any of the batches you have selected will be **UNDONE**. This functionality works only if **Enable Undo for Selected Batches** option in the Application Preferences is set to Yes.
- If you choose "Selected Allocation Models", all rule executions that belong to any of the Allocation Models that you have selected will be **UNDONE**. This functionality works only if **Enable Undo for Selected Allocation Models** option in the Application Preferences is set to Yes.

Starting from the release 8.0.6.0.0, UNDO of Allocation Models is supported using Batch Execution screen.

For creating the Batch for UNDO of Allocation Models:

In Batch Maintenance screen create a Batch and a corresponding Task with component as "RUN EXECUTABLE", and provide the following as the "Executable" property value.

```
PFTUNDO.sh,<INFODOM>_<BATCH_RUN_ID>
```

```
ParameterID=<ALLOCATION_MODEL_SYS_ID>-<ALLOCATION_MODEL_SEQ_NUM> A
```

where:

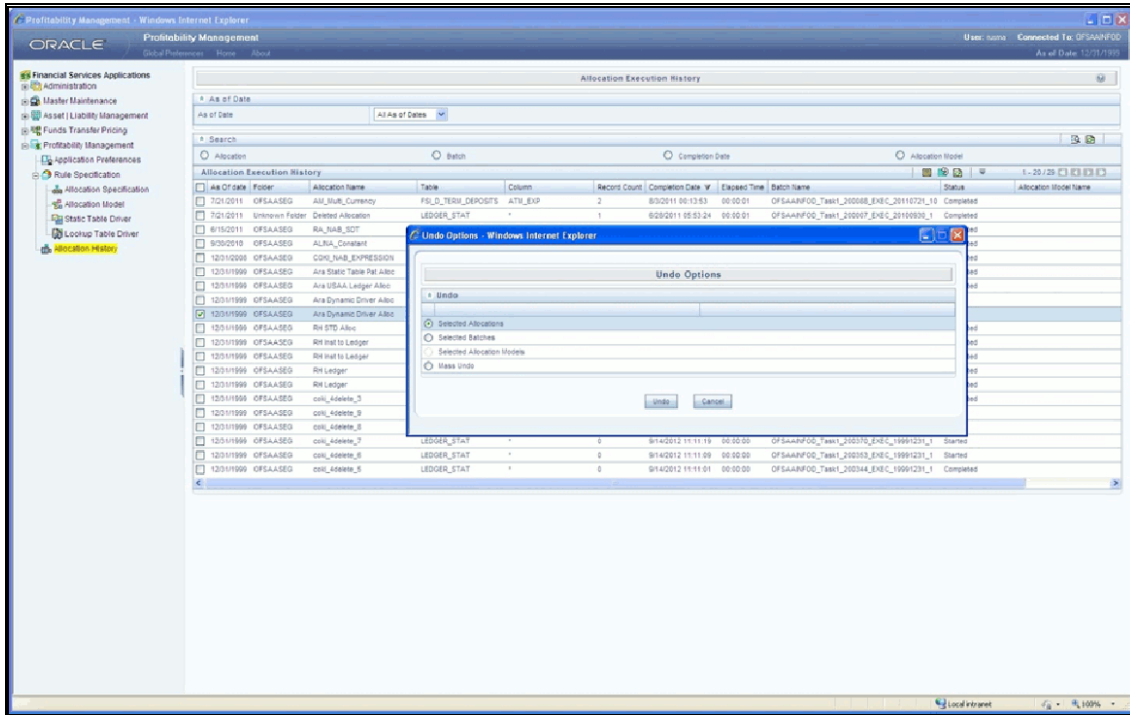
INFODOM - Information Domain name

BATCH\_RUN\_ID - Any unique identifier to identify the batch execution

ALLOCATION\_MODEL\_SYS\_ID - Sys ID number of the allocation model to be undone

ALLOCATION\_MODEL\_SEQ\_NUM - Sequence number to specify which execution of the allocation model is to be undone. This can be obtained from Allocation History screen.

- If you choose “Mass UNDO”, every rule execution whose "Completion Date" (a time-stamp value) is chronologically later than the earliest row that you have selected from the summary grid will be UNDONE. This functionality works only if **Enable Mass Undo** option in the Application Preferences is set to Yes.



## Allocation Execution Audit Functionality

After having selected a single row from the summary grid and selecting the View icon, a series of screens are displayed showing the allocation rule as it appeared at the time the rule was executed. If the rule has been modified since it was executed, the body of the rule will not be displayed. Future releases of Profitability Manager will include rule versioning so that you will always be able to see the structure of any rule at the time it was executed.

### Common Components

The following three sections – Process Tabs, Allocation Header, and Audit Trail/User Comments – are common to each of the five Allocation Execution History View (audit) screens.

#### Process Tabs

The Process Tabs display the process flow of the underlying allocation rule definition. The Initial Definition and Review tabs from the specification process flow are not displayed here, but one additional tab is added: Trace Allocation. The tabs displayed are: Source, Operator, Driver, Outputs, and Trace Allocation. When choosing these tabs the specific process page is displayed.

#### Allocation Header

This section contains header level information about the rule run including the rule's Name, Description, and Folder, the As-of-Date for which the rule was run, the Batch Name under which the rule was run, and its Completion Time. The Allocation Header also displays the allocation's type.

### **Audit Trail/User Comments**

The Standard Audit Trail/User Comments section is common to all detail screens. You may add additional comments when viewing allocation execution history if your user ID is assigned the Auditor role, subject to a maximum of 4000 characters.

### **Inline Reports**

The audit functionality provided within Allocation Execution History includes inline reports for Sources, Drivers, and Outputs of allocation rules. These inline reports are described in detail in the following sections.

---

**Note:** Trace and Inline Reporting functionality for allocations using Management ledger table is supported starting with release 8.

---

### **Source Tab**

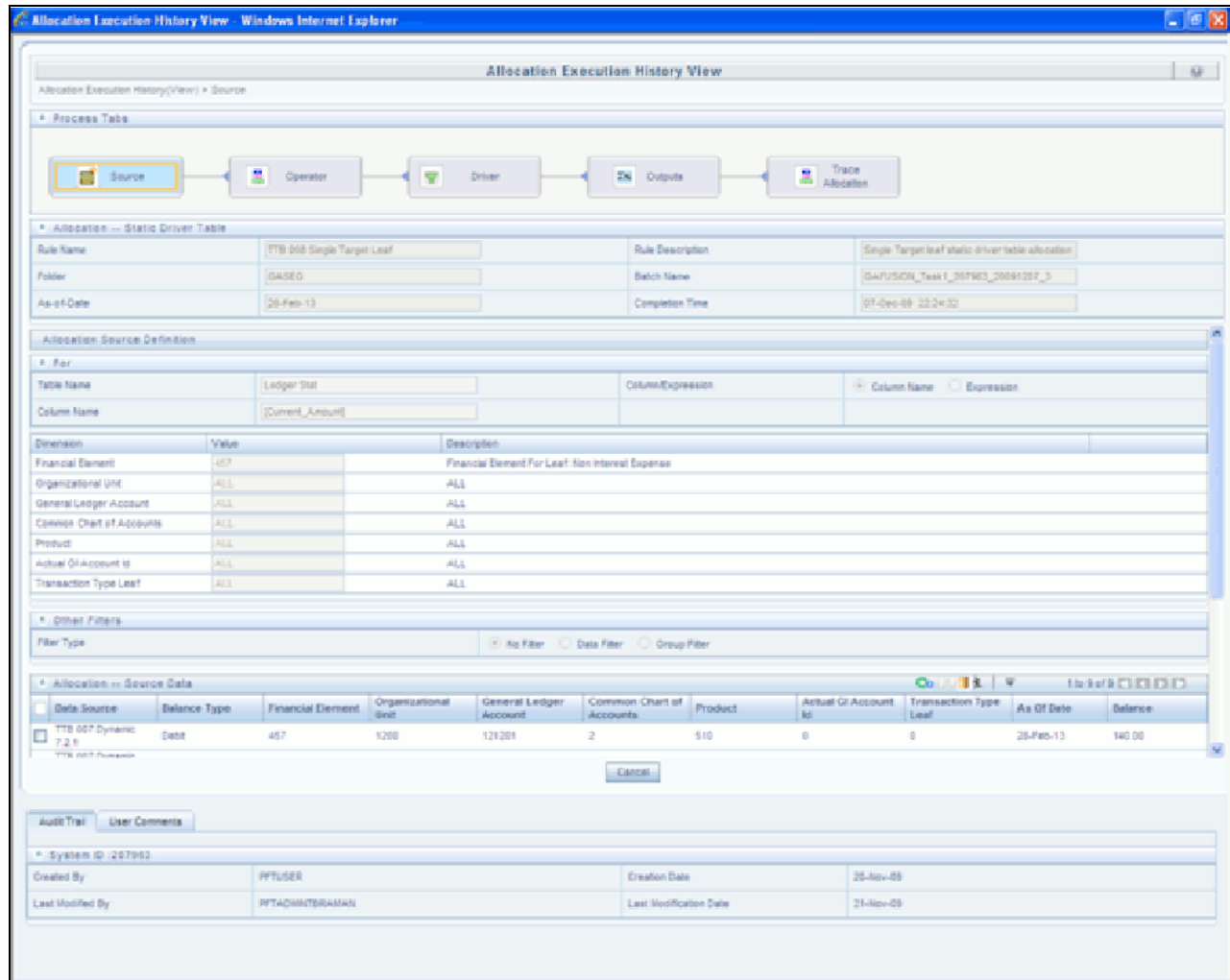
The Source tab is the first to be displayed when the user chooses to View a specific allocation run on the Allocation Execution History Summary page. This tab displays the details pertaining to the allocation's Source specification at the time that it was run. This view-only screen also offers an inline report of the Source data generated by the allocation rule at the time that it was run and offers the user the ability to trace the sources of the data that fed into the current allocation.

### **Allocation Source Definition**

The For section of the Allocation Execution History Source tab displays a read-only version of the For section of the underlying allocation rule's Source specification. The Other Filters section of the Allocation Execution History Source tab displays a read-only version of the Other Filters section of the underlying allocation rule's Source specification.

### **Allocation Source Data**

This section of the screen is initially empty. If you wish to see an inline report for the allocation's source data, you must select the Run icon. Actual queries generated by the allocation engine are persisted at runtime so that the inline reports will always be available (regardless of whether or not the rule has been modified since it was executed). The Source inline report excludes any data generated by allocations or other processes that ran after the rule run currently being examined.



### Allocation Source Data Controls

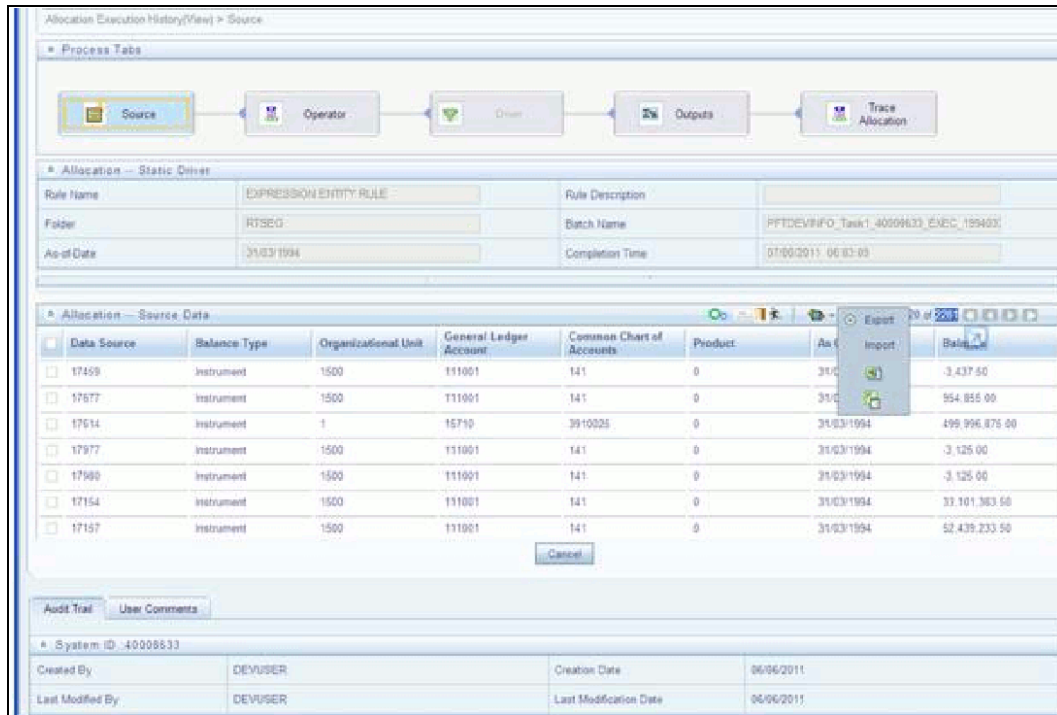
These include: Options, Trace, Show SQL, and Run.

#### Run

Select the Run icon to invoke the Source Inline Report.

#### Excel Export

Excel export functionality is used for adding / editing source data in the source tab of Allocation Execution History.



### Show SQL

Select the Show SQL icon to view the query that generates the inline report. That is, it displays the Source Query that has been prepared and stored in database while running the allocation. You may copy and paste this SQL to any query tool.

### Trace

After having run the inline report, you may trace to another allocation rule that contributed to the Source of the current allocation rule by checking a select box on one of the current report rows and then selecting the trace icon. Non-allocation sources are disabled; only allocations may be traced.

### Options

After having executed an inline report, selecting the Options icon invokes a History Options dialog. This dialog shows two containers: Dimension Display and Hide/Collapse Dimensions.

#### Dimension Display

This container offers you radio buttons with options to display short dimension names, long dimension names, or both within your inline report.

#### Hide/Collapse Dimensions

This container offers you the option of “collapsing” or “hiding” selected dimensions from your inline report. Dimensions containing only a single value within your report may be “hidden”, i.e., removed from the inline report. Dimensions containing multiple values within your report may be “collapsed” upon. A “collapsed” dimension is one that is removed from both the SELECT and GROUP BY clauses of the inline report query. If you “collapse” a dimension, it is removed from your inline report and the number of rows within the report is reduced.

### Operator Tab

This tab displays the allocation's Operator specification at the time that the rule was run.

### Driver Tab

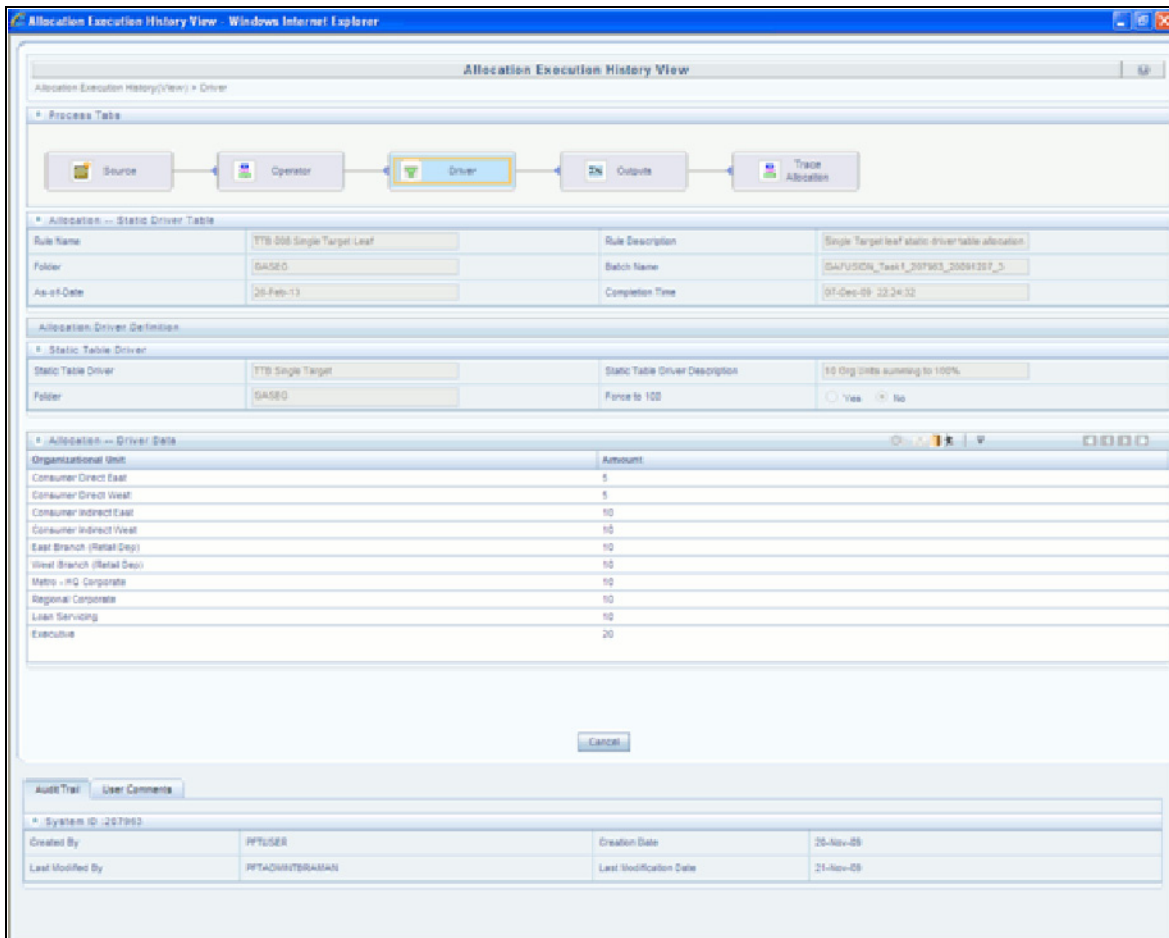
This tab displays the details pertaining to the allocation's Driver specification at the time that it was run. This view-only screen also offers an inline report of the Driver data generated by the allocation rule at the time that it was run and offers the user the ability to trace the sources of the date that fed into the current allocation.

### Allocation Driver Definition

The For section of the Allocation Execution History Driver tab displays a read-only version of the For section of the underlying allocation rule's Driver specification. The Other Filters section of the Allocation Execution History Driver tab displays a read-only version of the Other Filters section of the underlying allocation rule's Driver specification.

### Allocation Driver Data

This section of the screen is initially empty. If you wish to see an inline report for the allocation's driver data, you must select the Run icon. Actual queries generated by the allocation engine are persisted at runtime so that the inline reports will always be available (regardless of whether or not the rule has been modified since it was executed). The Driver inline report excludes any data generated by allocations or other processes that ran after the rule run currently being examined.



## Allocation Driver Data Controls

These controls, including Options, Trace, Show SQL, and Run, operate identically to those described above under Allocation Source Data Controls.

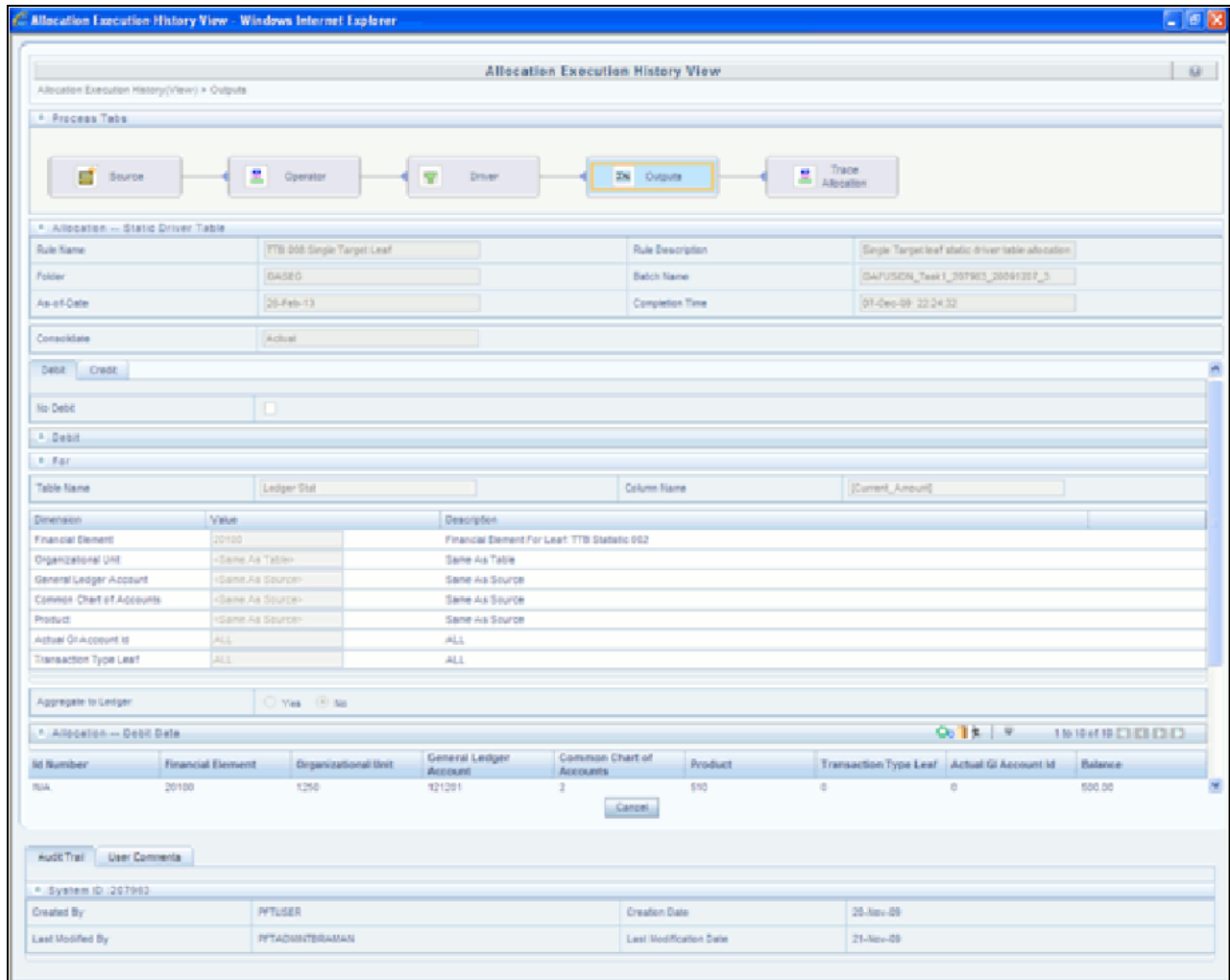
## Excel Export

Excel export functionality allows you to export Allocation Execution History source data from the source tab.

General Ledger Account	Organizational Unit	Amount
fiscal year gl-adw	10010	0.5
fiscal year gl-adw	10020	0.4
fiscal year gl-avg	10010	0.3
fiscal year gl-avg	10020	0.2
fiscal year gl-add	10010	0.6
fiscal year gl-add	10020	0.3

## Outputs Tab

The Outputs tab displays the details pertaining to the allocation's Output specification at the time that it was run. This view-only screen also offers an inline report of the Output data generated by the allocation rule at the time that it was run.



### Trace Allocation Tab

The Trace Allocation tab displays two containers, one showing distinct upstream data sources and the other showing affected downstream allocations.

### Upstream Data Sources

The Upstream Data Sources container displays all distinct Sources of data that fed into the current allocation run. You may trace any upstream allocation, but you may not trace non-allocation sources (initial ledger loads, transfer pricing rate migrations, etc). The All, Debit, Credit, and Others tabs within the Upstream Data Sources container allow you to discriminate by Balance Type.

### Affected Downstream Allocations

The Affected Downstream Allocations container displays all downstream allocation rules that included the outputs of the current rule run in their Source queries. You may trace to any downstream allocation. The All, Debit, Credit, and Others tabs within the Affected Downstream Allocations container allow you to discriminate by Balance Type.



The screenshot displays the 'Allocation Execution History View' interface. At the top, a process flow diagram shows the sequence: Source → Operator → Driver → Output → Trace Allocation. Below this, the 'Allocation -- Static Driver Table' section provides details for a specific rule: Rule Name 'TTB 000 Single Target Leaf', Folder 'DASEG', As-of Date '28-Feb-13', Rule Description 'Single Target leaf static driver table allocation', Batch Name 'SAPFUSION\_Task1\_207903\_0001206\_3', and Completion Time '28-Dec-09 14:51:26'. The 'Upstream Data Sources' table lists several allocation rules with their respective balance types and amounts. The 'Affected Downstream Allocations' table is currently empty. The 'Audit Trail' section at the bottom shows system ID '267903', created by 'PFTUSER', and last modified by 'PFTADHITHIRAMAN' on '21-Nov-09'.

Allocation Name	Balance Type	Row count	Balance
<input type="checkbox"/> TTB-000 Static x 75%	Credit	1	-75.00
<input type="checkbox"/> TTB-001 Create 100	Debit	1	100.00
<input type="checkbox"/> TTB-007 Dynamic 7.2.1	Debit	3	100.00
<input type="checkbox"/> TTB-007 Dynamic 7.2.1	Credit	2	-100.00
<input type="checkbox"/> TTB-003 Static x 25%	Debit	1	25.00
<input type="checkbox"/> TTB-003 Static x 75%	Debit	1	75.00
<input type="checkbox"/> TTB-003 Static x 25%	Credit	1	-25.00
<b>Total</b>		<b>10</b>	<b>100.00</b>

Allocation Name	Balance Type	Row count	Balance
<input type="checkbox"/>			

Created By	Creation Date
PFTUSER	28-Nov-09
Last Modified By	Last Modification Date
PFTADHITHIRAMAN	21-Nov-09

### Hyperlinking Between Audit Reports

Selection checkboxes often appear in the first column of the Upstream Data Sources and Affected Downstream Allocations containers on the Trace Allocation tab. Similarly, selection checkboxes often appear in the first column of the Source and Driver inline reports.

These checkboxes appear whenever a data source is itself an allocation rule. You may click on any checkbox and then click on the Trace control to hyperlink to the Allocation Execution History of the allocation source you just chose.



## CHAPTER 21

# Viewing PFT Objects in Metadata Browser

## Metadata Browser (Object and Application View)

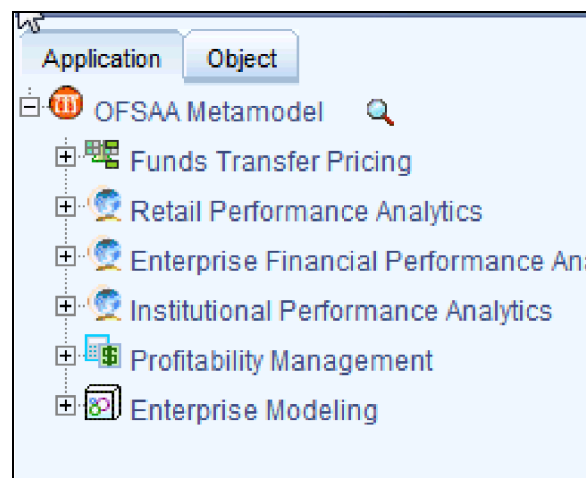
Metadata Browser (Object and Application View)

Metadata Browser (Object and Application View) provides common repository of metadata Objects created in OFSAAI and applications hosted in OFSAAI. Using this view, you can identify the usage of base Objects in higher level Objects and the mapping of Objects to Application. It enables to perform traceability and impact analysis. It also allows you to view the data flow and the work flow of the application and understand the usage of Objects within the application.

Metadata Browser (MDB) supports Application view and Object view: In Application view, you can browse through the metadata created using the applications hosted in OFSAAI. In Object view, you can view the metadata created in OFSAAI.

To access the Metadata Browser (Object and Application View), your role Business Analyst (SYSBAU) must be mapped to the SCR\_MDB function. For more information on mapping functions to a role, see *Function - Role Map*.

The Application view provides the top-down approach to view the defined metadata. The Application view shows the list of Applications at the parent level, and the Child elements are Runs for the application. This list will grow dynamically based on the Run definitions mapped to Application, through this tab user can view Run execution details in Metadata Browser.

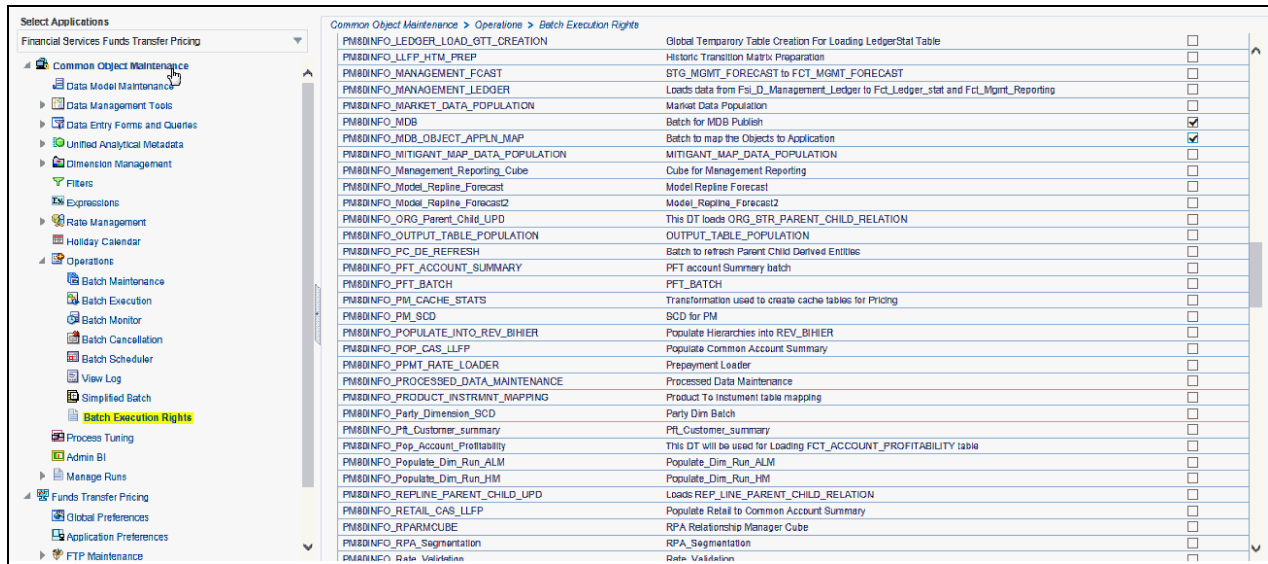


For more information on Metadata Browser, refer to Metadata Browser section of *Oracle Financial Services Analytical Application Infrastructure (OFSAAI) User Guide 8.0.4.0.0*.

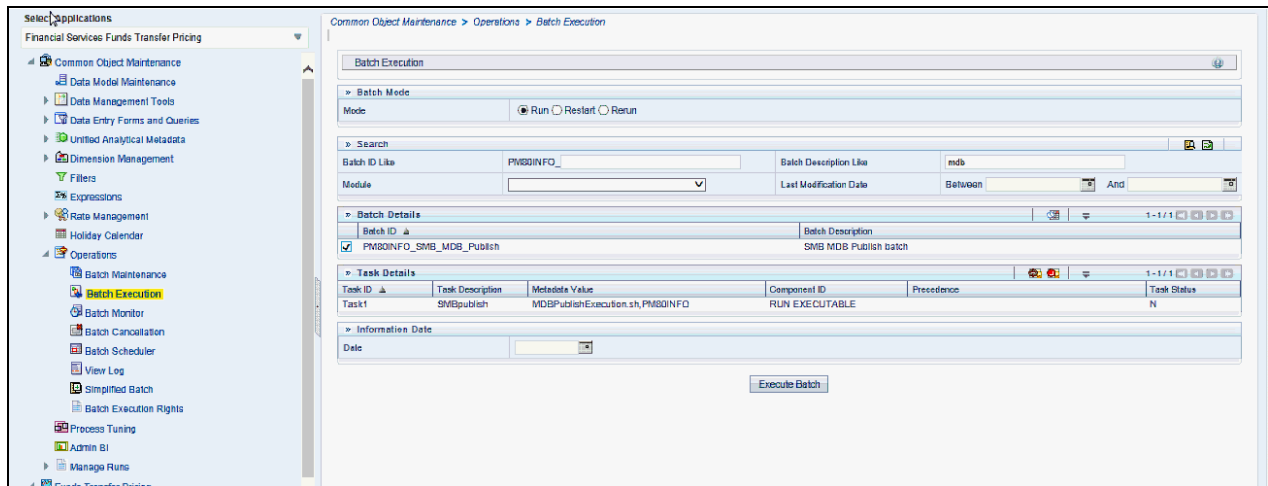
### Configuring Metadata Browser

You can configure the Metadata Browser for ALM using the following steps:

1. Map the MDB batch to User - Group using the Common Object Maintenance > Operations> Batch Execution Rights as mentioned below:



2. For more information, refer to *Command Line Utility to Publish Metadata in Metadata Browser* section in OFSAAI User Guide.
3. Execute the Batch for MDB publish.



For more information, refer to ICC Batch Execution section of *Oracle Financial Services Analytical Application Infrastructure (OFSAAI) User Guide 8.0.4.0.0*.

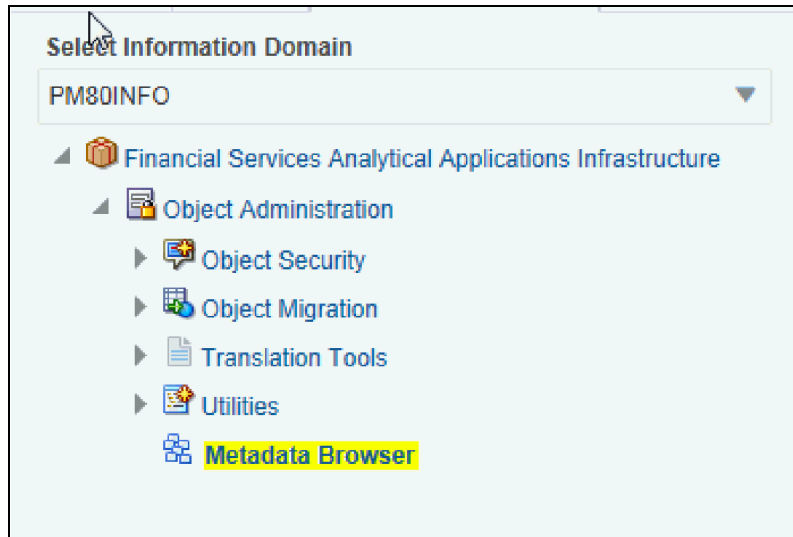
4. Navigate to Metadata Browser (Object and Application) window of respective Application.

## Viewing PFT Objects in Metadata Browser

Profitability Management under Oracle Financial Services Analytical Applications has the Metadata Browser (MDB).

To view PFT applications and objects in MDB, perform the following steps:

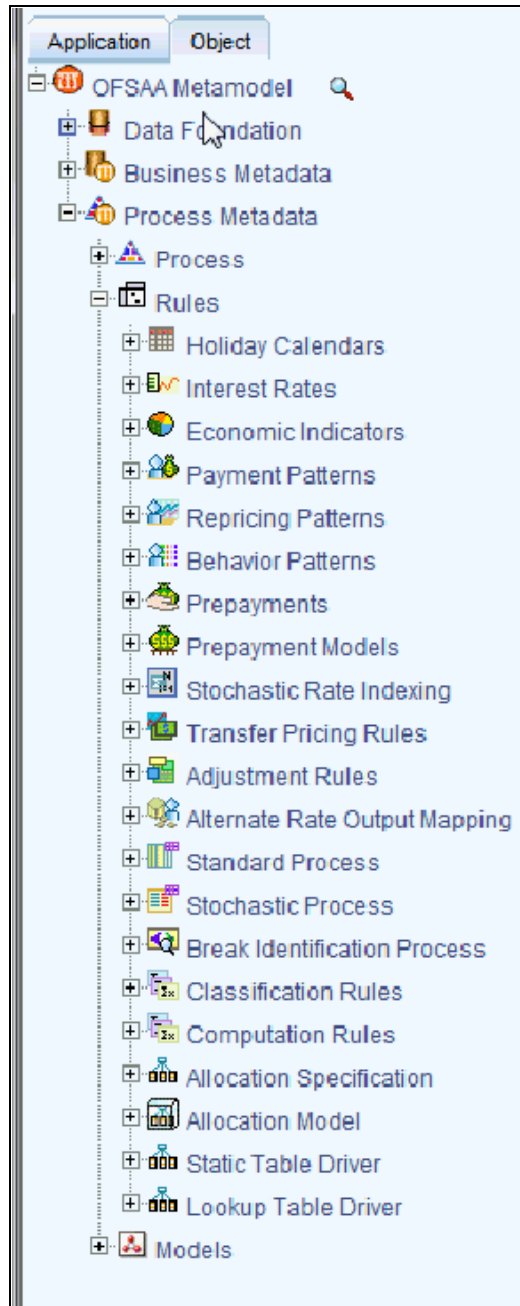
1. After successful execution of the Batch for MDB Publish, in the Object Administration tab of the Oracle Financial Services Analytical Applications window, in the Left-Hand Side (LHS) menu under Object Administration, choose Metadata Browser..



2. Click the Application tab under Metadata Browser window to view the PFT application.



3. Click the Object tab under Metadata Browser window to view PFT objects:



4. Under Process Metadata > Rules, all the rules defined under PFT window are displayed.

**APPENDIX A**

# Management Ledger

## Statement of Direction

New in release 8, the Management Ledger table is intended to replace OFSAA's Ledger/Stat table as the central repository for management accounting within OFSAA Enterprise Performance Management solutions.

The new Management Ledger supports the same functionality as historically has been supported by Ledger/Stat and stores the same kinds of data as has always been housed in Ledger/Stat including:

### General Ledger Data

- Reconciled to instrument level data
- Typically provides a starting point for management accounting processes

### Budget or Forecast Data

- From OFSAA Balance Sheet Planning or from other budgeting and forecasting sources

### Multidimensional Statistical or Volumetric Data

- Used or created by allocation rules

### Value-Added data From

- OFSAA Profitability Management allocation rules
- OFSAA Funds Transfer Pricing processes
- Other value-added processes

In the future, functional enhancements to Profitability Management will be directed towards the Management Ledger table class. New enhancements may or may not be extended to apply to the Ledger/Stat table.

## Frequently Asked Questions

*How do OFSAA Infrastructure and Applications interact with the Management Ledger table?*

### Loading Staging Data into the Management Ledger Table

- OFSAA Infrastructure 8.0.4.0.0 includes a T2T rule (T2T\_MANAGEMENT\_LEDGER) for loading the Management Ledger table from STG\_GL\_DATA

### OFSAA Profitability Management

- Distribution of balances within the management ledger level
- Aggregation and posting of instrument or transaction summary level data to the management ledger level
- Distribution of ledger-level balances to the instrument or transaction summary level
- Any other kind of Profitability Management process involving management ledger level data

**OFSAA Funds Transfer Pricing**

- Funds Transfer Pricing 8 supports the same functionality with the Management Ledger as historically has been available with the Ledger/Stat table
  - Direct transfer pricing of ledger-level balances
  - Aggregation/migration of instrument-level transfer pricing results to the ledger-level (weighted average rates and charges & credits for multiple FTP components)

**OFSAA Enterprise Financial Performance Analytics**

- OFSAA Enterprise Financial Performance Analytics 8 includes data movement processes for loading its results tables from either Ledger/Stat or Management Ledger

*What are the functional advantages of the new table structure?*

**Enhanced multi-currency model**

- Functional & Entered balances for each row
- More straightforward integration to multi-currency General Ledger systems
- Simplified reconciliation of ledger-level data with multi-currency instrument data
- All value-add processes post simultaneously to both Entered & Functional balances
  - Funds Transfer Pricing
  - Profitability Management

**Daily posting/monthly processing**

- Supports daily posting to the Management Ledger level
  - Particularly useful for customers who want to post daily FTP results to the ledger level

**Multi-Entity/Multi-Tenant functionality**

- Multi-entity functionality, also introduced in release 8, is available with the Management Ledger table but is not supported for Ledger/Stat.

**Future Enhancements**

- All future enhancements to OFSAA applications that utilize ledger-level data will be built to support the Management Ledger structure



*What are the technical advantages of the new table structure?*

- The structure of the Management Ledger table eliminates of the need to
  - Generate YTD balances for every row at time of insertion
  - Refresh YTD balances for every row at time of update
  - Store YTD balances on every row
- YTD balances may still be referenced in Profitability Management rules
  - The engine can easily generate this data when needed but the cost of calculation and storage for every row is eliminated

*Can I have multiple Management Ledger tables?*

Yes. The Management Ledger table is a seeded table but Management Ledger is also a table class. If you do establish multiple Management Ledger tables, you must manage the data in each table and its relationship to instrument level data or to other data in your Information Domain. See “Adding Management Ledger Class Tables” in the OFSAA Data Model Utilities User Guide for Release 8.0.

*Will customers need multiple Management Ledger tables?*

- Most customers will not need multiple Management Ledger tables. Utilization of multiple Management Ledger tables might be useful for some multi-entity or multi-tenant implementations.

*Can different Management Ledger tables have different dimensions?*

Yes, different Management Ledger tables may have different dimensionality.

*Can I move data between Ledger/Stat and the Management Ledger table?*

No. All rules and processes that post to or read from the ledger-level can utilize only one ledger table.

*How long will the Ledger/Stat table be supported?*

- Release 8 as well as all previous OFSAA releases will always support the Ledger/Stat table and its related application functionality
- Ledger/Stat and its related application functionality may be de-supported in a future release, but Ledger/Stat and its related application functionality will always be supported for current releases and Oracle intends to continue supporting Ledger/Stat for the foreseeable future. New application enhancements are unlikely to be extended to the Ledger/Stat table.

*Which table does Oracle recommend that customers use?*

### **New Customers**

- Customers beginning new implementations are advised to use the new Management Ledger table and are advised not to use the Ledger/Stat table
  - Customers who elect to use Ledger/Stat should be aware that support for application functionality that utilizes the Ledger/Stat table will be withdrawn in a future release
  - Customers who want to migrate to a future release for which Ledger/Stat is no longer supported will be required to migrate their Ledger/Stat processes to the Management Ledger prior to upgrading to that release

- The need to eventually move to the Management Ledger structure can be avoided by starting with the Management Ledger table

### **Upgrading Customers**

- Customers upgrading or considering upgrading to release 8 from a prior release are advised to continue using the Ledger/Stat table
  - This will minimize the time and effort required for the migration to release 8 by retaining all rules and processes that currently involve the Ledger/Stat table
- Customers upgrading to release 8 should be aware that that support for application functionality that utilizes the Ledger/Stat table may be withdrawn in a future release
  - Application rules and processes that involve the Ledger/Stat table may need to be migrated from Ledger/Stat to the Management Ledger at some point in the future

**APPENDIX B**

# Archive-Restore

## Command Line Migration

Using the command line utility, you can migrate (export/ import) PFT metadata objects across different OFSAA environments. You can specify one or more objects within an object type or within multiple object types.

**Note:** PFT does not support command line object migration of .CSV format files.

For the list of objects that can be migrated, see the table below.

## Command Line Migration Utility

PFT uses the same utility that is used by the OFSAA infrastructure. For more details, refer to the section "Command Line Utility to Migrate Objects" in the AAI User Guide's Object Migration chapter.

## Supported Objects

The following objects are supported for migration:

Object Name	Object Type ID	Support for wild card Select all option	Support for implicit Dependency	Object Code	Remarks
Application Preferences	201	No	No	CONFIG	Entire data set is exported and replaced in the target. No validation and its dependency objects are handled
Interest Rates	801	Yes	Yes	User defined unique code	Historical rates from FSI_IRC_RATE_HIST table are not migrated.
Currency rates	803	No	No		Currency rates are not supported.
Currencies	802	Yes	No	User defined Currency code	

Object Name	Object Type ID	Support for wild card Select all option	Support for implicit Dependency	Object Code	Remarks
Allocation Rule	0	Yes	Yes	System generated code	
Allocation Model	7	Yes	Yes	System generated code	
Static Table Driver	17	Yes	Yes	System generated code	
Lookup Table Driver	31	Yes	Yes	System generated code	
Simplified Batch	32	Yes	Yes	System generated code	Simplified Batch migration is supported from the Release v8.0.6.1.10.

### Input Values for OBJECTMIGRATION.xml

The following table describes the input values required for OBJECTMIGRATION.xml file:

Object Name	Object Type Id	Input value in OBJECTMIGRATION.XML for wild card	Input value in OBJECT_MIGRATION.xml for Object Code
Application Preferences	201	<OBJECT Code="CONFIG" Type="201"/>	<OBJECT Code="CONFIG" Type="201"/>
Interest Rates	801	<OBJECT Code="" Type="801"/>	<OBJECT Code="User defined unique code" Type="801"/>
Currencies	802	<OBJECT Code="" Type="802"/>	<OBJECT Code="User defined Currency code" Type="802"/>
Allocation Specification	0	<OBJECT Code="" Type="0"/>	<OBJECT Code="System generated code" Type="0"/>
Allocation Model	7	<OBJECT Code="" Type="7"/>	<OBJECT Code="System generated code" Type="7"/>

Object Name	Object Type Id	Input value in OBJECTMIGRATION.XML for wild card	Input value in OBJECT_MIGRATION.xml for Object Code
Static Table Driver	17	<OBJECT Code="*" Type="17"/>	<OBJECT Code="System generated code" Type="17"/>
Lookup Table Driver	31	<OBJECT Code="*" Type="31"/>	<OBJECT Code="System generated code" Type="31"/>

## Logging

1. <War\_deployed\_location>/logs/ offline\_migration.log
2. <War\_deployed\_location>/logs/FUSIONSUBSYSTEMSService.log



**APPENDIX C***Multi Language Support***Overview**

With the introduction of Multi Locale Support, the user interfaces can be viewed in various languages. The display language is a selection in the Login window.



In addition to the base language (American English, for example, en\_US), the following languages are currently supported for Profitability Management, by installing the associated Language Pack.

Language Name	Java Locale
Arabic	AR_EG
Portuguese	PT_BR
French	FR_FR
German	DE_DE
Indonesian	IN_ID
Italian	IT_IT
Japanese	JA_JP
Korean	KO_KR

Language Name	Java Locale
Russian	RU_RU
Simplified Chinese	ZH_CN
Spanish	ES_ES
Thai	TH_TH
Traditional Chinese	ZH_TW
Vietnamese	VI_VN
Turkish	TR_TR
Hebrew	iw_IL
French Canada	fr_CA

Depending on the type of the rule, the Name/Description specified during create/edit is either attached to the login-locale or is stored independent of the login-locale. The selection of the login-locale therefore governs which assumption/rules are visible, and which are not visible.

The following illustrates the visibility of assumptions and rules, across languages in the 8 release. This applies to all Summary screens, embedded selections in subsequent screens and so on.

**Note:** Starting from AAI 7.3.3.0.0, all Dimension Management Objects as well as Filters and Expressions would be visible in all languages (and therefore available for embedding), regardless of the language in which they were created.

Assumption/ Rules	Visible across all languages	Visible only in the creation-language*
Administration		
Process Tuning	Yes	
Simplified Batch		Yes
Master Maintenance		
Interest Rate Codes**	Yes	
Currencies**	Yes	
Funds Transfer Pricing		
Rule Specification		
Allocation Specification		Yes
Allocation Model		Yes
Static Table Driver		Yes
Lookup Table Driver		Yes
Allocation History	Yes	

- \* A future release (TBD) will expand visibility for many of these application object types, to provide visibility regardless of logged-in locale.
- \*\* These object-types are visible across all languages because the Name and Descriptions are not stored in translation-compatible structures, which is different from the storage format for AAI objects like Dimension Management, Filters and Expressions.



---

**APPENDIX D**

# *Multicurrency*

This Appendix covers:

- Implementation of multiple currencies in OFSAA solutions
- How currency translation is used in OFSAA Profitability Management calculations
- Usage of multiple-currency output from Profitability Management in relation to the Management Ledger, Instrument, and Transaction Summary tables

This appendix covers the following topics:

- OFSAA Rate Management
- Functional Currency
- Currency Codes
- 002 code
- Loading the Management Ledger Table
- Multiple Currencies in Allocation Rules
- Using the Currency Dimension as a Filtering Condition
- Allocation Engine Process Processing Flow
- Selecting Rows from Ledger/Stat
- Treatment of Constants and Coefficient Values
- Selecting Rows from Detail Tables
- Inserting Into or Updating Ledger/Stat
- Updating Detail Tables
- Notes on Usage of Statistics and Other Balances
- Monetary Balances and Balance Weighted Objects
- Statistics & Standard Rates
- Examples

## ***OFSAA Rate Management***

Documented in OFSAA Rate Management chapter is a rate management utility that enables users to manage interest rates, exchange rates, and currency data with a high degree of security and control. As part of OFSAA Infrastructure, OFSAA Rate Management handles all currency definitions for OFS Analytical Applications. Currencies are conveniently referred to by code and written description. A comprehensive list of ISO-defined currencies is included, and the user can define and add his or her own. For details on using OFSAA Rate Management, see [Chapter 12, "OFSAA Rate Management"](#) of this user guide.

Upon installation, one currency is active and ready for processing: the one identified by your organization as the functional currency (see the section below entitled Functional Currency for more information). You may activate other currencies and define and activate newly created currencies as needed.

You may define, input, and maintain exchange rates between any two active currencies. In addition to the standard floating exchange relationships, special fixed relationships are available (For example, used prior to 2000 in the European Monetary Union). OFSAA Rate Management complies with EMU structure and legislation, and supports changes in currency rates as well as additions and deletions of the currencies of member countries.

Exchange rates between currencies used in Profitability Management allocations are reciprocal. The rate of exchange between two currencies must be arbitrage free. Profitability Management does not calculate any currency exchange offsets for non-reciprocal exchange rates.

## **Functional Currency**

At the time of installation, OFSAA Rate Management requires that you designate a functional, or primary, currency for your organization. A Swiss multinational bank, for example, would designate the Swiss franc as its functional currency. The functional currency is always an active currency. Once you have assigned your functional currency, it cannot be changed. To assign your functional currency, manually update FSI\_DB\_INFO.FUNCTIONAL\_CURRENCY\_CD with your selected currency.

---

**Note:** This one row configuration table also contains your fiscal year definition; see Appendix D for more information on Fiscal Year definition.

---

Most customers need only one functional currency, but if you require multiple functional currencies and/or multiple fiscal year definitions, you may establish ledger-specific selections of functional currency and fiscal year by manually updating LEDGER\_CCY\_AND\_FISCAL\_YR\_INFO (one row per defined Management Ledger table).

Processes operating against the Ledger/Stat table always determine Functional Currency and Fiscal Year by consulting FSI\_DB\_INFO.

Processes operating against a Management Ledger table generally determine Functional Currency and Fiscal Year by consulting LEDGER\_CCY\_AND\_FISCAL\_YR\_INFO. However, any process operating against a Management Ledger table for which no row has been set up in LEDGER\_CCY\_AND\_FISCAL\_YR\_INFO determines its Functional Currency and Fiscal Year by consulting FSI\_DB\_INFO (i.e. FSI\_DB\_INFO serves as a fallback in the event that you have not set up ledger-specific currencies and calendars in LEDGER\_CCY\_AND\_FISCAL\_YR\_INFO).

---

**Note:** This one row configuration table also contains your fiscal year definition; see Appendix D for more information on Fiscal Year definition.

---

The designation of functional currency is stored as an ISO currency code in FSI\_DB\_INFO.FUNCTIONAL\_CURRENCY\_CD / LEDGER\_CCY\_AND\_FISCAL\_YR\_INFO.FUNCTIONAL\_CURRENCY, as applicable.

Profitability Management allocations can read and write in multiple currencies. However, Profitability Management allocations perform internal calculations in the functional currency.

When multicurrency is disabled, all ISO Currency Code values are written in the Functional Currency.

## Currency Codes

ISO Currency Codes are defined for the world's major currencies and are seeded in the OFSAA database at installation. ISO Currency Codes are Simple Dimension members (see [Dimension Types](#) in Overview of OFSAA Infrastructure chapter) that define the currency of any row in the database. ISO Currency Code values used in any of the OFS Analytical Applications must be designated as active in OFSAA Rate Manager. ISO Currency Codes may also be user-defined. User-defined ISO Currency Codes are set up in OFSAA Rate Management (again, for details, see [Chapter 12, "OFSAA Rate Management"](#)).

Users must designate ISO Currency Codes as reporting currencies in order to activate the calculation of conversion rates. Simply designating the currency as active is not sufficient.

### 002 code

The ISO Currency Code value of 002 represents statistical data in the Management Ledger. An ISO Currency Code value of 002 means “no currency basis”. For example, a row in the Management Ledger representing a square footage statistic does not need a currency association and would therefore have a 002 ISO Currency Code. Assigning an ISO Currency Code to a statistic is useful for matching rows that have the same leaf values but differ only by ISO Currency Code (Leaf type Allocation rules).

## Loading the Management Ledger Table

OFSAA Infrastructure includes a process for loading data to the Ledger/Stat table (see Ledger Data Loader in the OFSAA Data Model Utilities User Guide). The following describes the behavior of the Ledger Data Loader as regards ISO Currency Code values:

If multicurrency is Disabled

The ISO Currency Code of the Functional currency is always loaded into Ledger/Stat for every row found in the staging table.

If multicurrency is Enabled

Staging rows that are populated with valid ISO Currency Codes are loaded to Ledger/Stat. Staging rows that are not populated with valid ISO Currency Codes are loaded as follows:

- If a row's Financial Element has the column property of Balance or Balance Weighted Object, it is loaded to Ledger/Stat with the ISO Currency Code of the Functional Currency.
- If a row's Financial Element has the column property of Statistic or Standard Rate, it is loaded to Ledger/Stat with the ISO Currency Code '002' (the "non-currency" or "statistical" currency code).

## Multiple Currencies in Allocation Rules

OFSAA Profitability Management allocation engine supports processing of balances in multiple currencies. When multiple currency functionality is enabled (`FSI_DB_INFO.MULTI_CURRENCY_ENABLED_FLG = 1`), a Currency dimension is enabled in each of the Allocation Specification dialogs (i.e., Sources, Drivers, and Outputs) that lists dimension member values. ISO Currency Code functions similarly to other dimension members in the Profitability Management user interfaces except that the ISO Currency Code dimension is a Simple Dimension and

therefore does not support hierarchies. When specifying ISO Currency Codes in an Allocation rule, you may only specify a leaf value (an active ISO Currency Code) or a macro value.

- For all tabs the allocation rule dialogs support the Functional Currency macro ('001'), the Statistical Currency macro (non-Currency Basis of '002'), and all active ISO Currency Codes.
- The Source and Driver tabs also support an All Currencies macro. All Currencies is the default value for Source and Driver tabs.
- The Debit and Credit pages within the Output tab also support a <Same as Source> macro. For Dynamic Driver rules, the Debit and Credit pages also support <Same as Driver >and <Match Source & Driver>

Output Tab Showing Multicurrency Options for a Dynamic Driver Rule

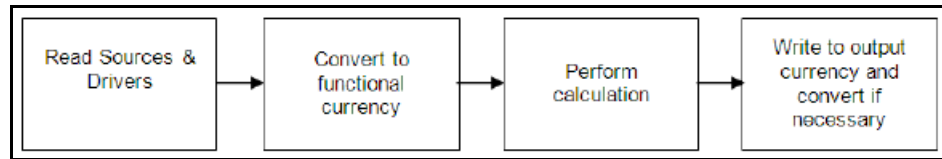
Dimension	Value	Description
Financial Element	<Same As Source>	<Same As Source>
Organizational Unit	<Same As Driver>	Same As Driver
General Ledger Account	<Same As Source>	Same As Source
Common Chart of Accounts	<Same As Source>	Same As Source
Product	<Same As Source>	Same As Source
Currency	-99300 - <Same As Source>	Same As Source

## Using the Currency Dimension as a Filtering Condition

As with other Key Processing Dimensions, the use of any value other than <All> will constrain the data that serves as Sources or Drivers to allocation rules. For example, if you were to specify GL Account 12345 – Salary Expense within the Source of an allocation rule, only rows having GL Account 12345 would be included in the Source of your rule. Similarly, if you were to specify US Dollars within the Source of an allocation rule, only rows denominated in US Dollars would be included.

## Allocation Engine Process Processing Flow

In Sources and Drivers, the Profitability Management allocation engine can read multiple input currencies. When posting to the Ledger/Stat table, the engine can output multiple result currencies. When posting to a detail table, outputs are converted, if necessary, to match the ISO\_CURRENCY\_CD on each detail record to be updated.



The preceding diagram describes the basic allocation engine processing flow for multicurrency allocations. Input data for monetary balances are converted to the functional currency. By having all inputs in the same currency, calculations, such as percent distribution, are created on equal basis. After converting all input data to functional currency, the allocation calculations are performed. The output results of the allocation can be in any currency. The results of calculations are converted from functional currency to the appropriate debit or credit currency for that allocation. As noted earlier you may direct the output currency to your choice for Ledger/Stat outputs; for detail tables, outputs always match the currency of the records to be updated.

Input rows to allocation rules are checked to determine whether currency translation is necessary. The checks for whether an item needs translation vary between the Ledger/Stat table and detail tables (Instrument and Transaction Summary tables).

## Selecting Rows from Ledger/Stat

When reading rows from Ledger/Stat, the allocation engine decides whether or not it needs to translate a Month-to-Date or Year-to-Date value based on the “Column Property” characteristic of each row's Financial Element ID. This decision process applies equally to Sources and Drivers and is made on a row-by-row basis (one decision point for each distinct Financial Element found). OFSAA Infrastructure supports four Column Properties for Financial Elements: Balance, Balance Weighted Object, Statistic, and Standard Rate.

- If the Column Property of an input row's Financial Element is Balance or Balance Weighted Object then the value is translated to the Functional Currency
- If the Column Property of an input row's Financial Element is Statistic or Standard Rate then the value is NOT translated (that is, the value has no currency specific basis)

**Note:** You may execute the following query to see the Column Property for each Financial Element:

```

SELECT A.FINANCIAL_ELEM_ID,
C.FINANCIAL_ELEM_NAME,
B.COLUMN_PROPERTY
FROM DIM_FINANCIAL_ELEMENTS_ATTR A, REV_COLUMN_PROPERTY_DSC B,
DIM_FINANCIAL_ELEMENTS_TL C
WHERE A.ATTRIBUTE_ID = 5004
AND C.LANGUAGE = <Specify your language here or simply use 'US'>
AND A.DIM_ATTRIBUTE_NUMERIC_MEMBER = B.COLUMN_PROPERTY_CD
AND A.FINANCIAL_ELEM_ID = C.FINANCIAL_ELEM_ID
ORDER BY 1;
  
```

**Note:** When you generate a user-defined Financial Element, you must assign a Column Property attribute value.

## Treatment of Constants and Coefficient Values

The following constant values are assumed to be either statistical in their nature or are assumed to be entered in Functional currencies and are therefore never translated:

- Input values on the Source tab for Constant type allocation rules
- Constants found in the Operator for any rule
- Values in Static Table Driver rules
- Values found by Lookup Table Driver rules in user-defined Lookup Tables

## Selecting Rows from Detail Tables

When reading a column from a detail table (an Instrument table or a Transaction Summary table), the allocation engine decides whether or not it needs to translate a value based on the column's Column Property characteristic.

- If a column's Column Property is Balance then the value is translated to the Functional Currency
- If a column's Column Property is anything OTHER than Balance then the value is NOT translated (that is, the value has no currency specific basis)

For detail level columns, OFSAA Infrastructure supports the following Column Properties:

Property	Description
BALANCE	Monetary balance
CHAR	Fixed-length alpha-numeric data
CODE	Defined Alpha or Numeric Code Value
CODE_NUM	Undefined Numeric Code Value
DATE	Date value
FLAG	True/False value
FREQ	Recurring period of time
LEAF	Leaf column
IDENTITY	Reserved
ID_NUMBER	Reserved
MULT	Multiplier
NUMBER	Numeric Value
NUMERIC	Numeric Value
RATE	Interest rate
TERM	Non-recurring period of time
VARCHAR2	Variable-length alpha-numeric value
PCT	Percent
DEFAULT	Default Datatype

**Note:** You may execute the following query to see the Column Property for any column in an Instrument or Transaction Summary table:

```

SELECT TABLE_NAME,
COLUMN_NAME,
A.REV_DATA_TYPE_CD,
B.REV_DATA_TYPE,
B.DESCRPTION
FROM REV_TAB_COLUMNS A, FSI_DATA_TYPE_DESC B
WHERE TABLE_NAME = <Insert table name here>
AND A.REV_DATA_TYPE_CD = B.REV_DATA_TYPE_CD
ORDER BY 1, 3, 2;

```

## ***Inserting Into or Updating Ledger/Stat***

Allocations inserting into or updating the Ledger/Stat table treat Currency as a Key Processing Dimension. For example, in a Dynamic Driver type of allocation that outputs to Ledger/Stat, if the allocation's Debit value in the Currency dimension is <Same as Source> and if five distinct currencies are found in the Source data, then the engine will output to five currencies in its Debit. If the allocation's Debit value in the currency dimension is <Same as Driver> and if six distinct currencies are found in the Driver data and if the allocation's Driver is constructed to use Force to 100%, then the allocation engine will perform a percent-distribution calculation to the six currencies.

The engine decides how to Debit & Credit the Currency dimension based on the instructions you specify in an allocation rule. The decision as to whether or not to translate output values to the currency basis of the target output currencies is a completely separate one.

For any given row in the output of an allocation rule, the engine decides whether or not to convert to a target currency based on the Column Property of the Financial Element for that row. As was the case when selecting data from Ledger/Stat:

- If the Column Property of a target row's Financial Element is Balance or Balance Weighted Object then the value is translated from the Functional Currency to the target currency.
- If the Column Property of a target row's Financial Element is Statistic or Standard Rate then the value is NOT translated that is, the value has no currency specific basis)

## ***Updating Detail Tables***

When updating a Balance type column within any of the detail tables, outputs are translated to match the currency of the records to be updated.

## ***Notes on Usage of Statistics and Other Balances***

It is important to understand the details of the behavior of the allocation engine with respect to currencies before building your rules. Some elements that you may regard as being statistical in nature are registered within the system as Balances or as Balance Weighted Objects and are therefore always translated (both when serving as inputs to rules and when output from rules). You may wish to store other statistical elements as a function of currency, that is, elements having different values for each currency, but which you do not want the engine to ever translate. Finally,

you may wish to store elements that you regard as being statistical in nature but which vary as a function of currency and which you do want the engine to translate.

## ***Monetary Balances and Balance Weighted Objects***

Monetary Balances (or simply Balances at the detail table level) have a strong currency basis that is relatively obvious. Ending balances and average balances for Balance Sheet elements and expense & revenue balances are the most common examples.

In a multicurrency implementation, instrument level monetary balances are inherently stored as “transacted” balances, that is, as balances stored in a local currency. The monetary balances that you source from your general ledger system and which you store in Ledger/Stat will generally be 100% functional. You may, however, sometimes wish to aggregate instrument level balances to your Ledger/Stat table and to store them there in their local currency amounts and/or generate non-functional balances in your Ledger/Stat table.

Balance Weighted Objects include elements such as weighted average Transfer Rate (Financial Element 140) at first glance appear to have a statistical nature but have a direct relationship to underlying monetary balances and therefore have the same strong currency basis as Monetary Balances.

More commonly, you will write your allocation rules that debit or credit Ledger/Stat using the 001 – Functional Currency macro or using the ISO Currency Code that represents your Functional Currency (these are equivalent definitions).

## ***Statistics & Standard Rates***

The Statistics and Standard Rates Column Property Financial Elements are never translated – neither on input (Sources and Drivers) nor on output (Debits and Credits).

Similarly, at the instrument level, columns whose Column Property is other than Balance are never translated.

The meaning of the term “statistic” is contextual. Statistics often lack a currency basis (that is, you do not want the allocation engine ever to translate them, neither on input nor on output). Most often, transactional volumes, activity volumes, unit costs, and other statistics do not vary by currency in their meaning. For example, you may wish to utilize headcount statistics as a driver in one or more allocation rules. Your headcount statistic would almost always be invariant as to currency.

On the other hand, you may wish to store some other kinds of “statistics” as a function of currency. Weighted average interest rates on loans and deposits or weighted average transfer rates, for example, have a strong currency basis.



## Examples

### Example #1

In this example, a “debit only” Constant allocation rule generates a single row in Euros in an environment in which the Functional currency is US dollars.

Application Data	Nature of Data	Implication
Source Data: Constant value of 100	Input value is a constant	As a constant, the engine treats this as \$100 (since USD is the Functional currency)
Driver Data: None	Constant rule types do not support drivers	There is no driver data
Debit: One leaf dimension value is supplied for each dimension	Debit: Record is output in Euros to Financial Element = Ending Balance	Since the Column Property of the Ending Balance Financial Element is “Balance”, the output value is translated and the one row created is denominated in Euros

In this example, if the USD to EUR exchange rate is \$1.2987 per Euro then the allocation rule's Debit produces an output value of €77.00 (100/1.2987).

### Example #2

Similar to example #1, in this example a “debit only” Constant allocation rule generates a single row in Currency Code “002 – Non Currency Basis” in an environment in which the Functional currency is US dollars.

Allocation Data	Nature of Data	Implication
Source Data: Constant value of 100	Input value is a constant	As a constant, the engine treats this as \$100 (since USD is the Functional currency)
Driver Data: None	Constant rule types do not support drivers	There is no driver data
Debit: One leaf dimension value is supplied for each dimension	Debit: Record is output to Currency = '002' and to a user-defined Financial Element = Headcount (a statistical Financial Element)	Since the Column Property of the output Financial Element is “Statistic”, the output value is NOT translated and the one row created is denominated in '002', Non Currency Basis

In this example, note that even if the allocation had output to a Balance type Financial Element (Ending Balance, for example), no translation would have occurred since '002' means “No currency basis”. The debit value in this example is 100.

**Example #3**

Example #3 is an allocation rule that distributes an equal amount of ledger-level expense to an instrument level balance column.

Allocation Data	Nature of Data	Implication
Source Data: Ledger/Stat for Statement Processing Expense	Various Statement Processing Expenses sourced from Ledger/Stat for Checking and Savings products. In this example, all of the Source rows are denominated in USD which is the Functional currency.	No translation is necessary since all of the source expense is already denominated in the Functional currency
Driver Data: Record Count column in the Checking and Savings instrument table	The instrument-level Record Count column will nearly always have a value of 1 and is intended to assist, among other purposes, as a driver for “equal distribution” kinds of allocation rules. The Column Property of the Record Count column is Numeric.	As a Numeric column, Record Count has non-currency specific meaning and therefore is not translated.
Debit: Instrument level Statement Processing Expense column	In this example, 3 distinct currencies are found within the Checking and Savings table for the current As-of-Date: USD, EUR, and GBP.	No translation is required for data written to the Statement Processing Expense column for the rows denominated in USD, but translation is required for rows denominated in EUR and GBP.

In this example, we are allocating a pool of ledger-level Statement Processing expense down to the instrument level for Checking and Savings products. The methodology we have selected is to allocate an equal share of the ledger-level expense to every row in the CASA (Checking and Savings Accounts) table on the assumption that every row generates an equal amount of expense for the bank.

Once completed, we would expect that the total USD-equivalent of the amount of the Statement Processing Expense allocated to the instrument level would be exactly equal to the amount originally sourced from the ledger-level.

**Example #4**

Example #4 is an allocation rule that distributes ledger-level expense to an instrument level balance column on a percent-to-total basis of another instrument level balance column.

Allocation Data	Nature of Data	Implication
Source Data: Ledger/Stat resident Loan Loss Reserves for Mortgages	All Loan Loss Reserve data for Mortgage products denominated in USD (the Functional currency)	No translation is necessary since all of the source expense is already denominated in the Functional currency
Driver Data: Average Balance column in the Mortgages instrument table (percent-to-total)	The Column Property of the Average Balance column is Balance. In this example, rows are found that are denominated in USD and JPY	As a Balance column, Average Balance is translated to the Functional Currency
Debit: Instrument level Loan Loss Reserve column	The Column Property of the Loan Loss Reserve column is also Balance	Once again, translation is required but only for the rows denominated in JPY

In this example we started with Source data (ledger-level) Loan Loss Reserve in the amount of \$40,000,000 that was to be allocated to detail level data in the Mortgages table for a portfolio of 10,000 USD-denominated loans the total

average balance of which was \$2 billion; and 10,000 JPY-denominated loans the total average balance of which was ¥ 200 billion. In this case the current exchange between JPY and USD is 100:1 so the USD equivalent average balance of the JPY mortgage portfolio is equal to the USD mortgage portfolio – each portfolio is valued at 2 billion USD.

Since the JPY and USD portfolios are of equal size, we would expect that each portfolio should receive an equal distribution of total Loan Loss Reserve (\$20 million to each portfolio). Moreover, we would expect that each dollar of USD average balances would receive an amount equal to each ¥ 100 of JPY average balances; and that after the allocation rule had been run, we would expect to find \$20 million in allocated Loan Loss Reserve associated with USD mortgage rows and ¥ 2 billion in allocated Loan Loss Reserve associated with JPY mortgage rows.



**APPENDIX E*****Fiscal Year Information***

Oracle Financial Services Analytical Applications (OFSAA) support both calendar year (January to December) and fiscal year configurations. If your organization operates on a fiscal year calendar, this appendix documents how to configure your fiscal year within OFSAA and how a fiscal year configuration affects processes that insert into or update the Ledger/Stat table. Note that in the context of this appendix, a fiscal year denotes a non-calendar financial year (that is, a year that does not contain 12 months and/or does not begin in January and end in December).

This appendix covers the following topics:

- Configuring a Fiscal Year
- How YTD Calculations Are Affected by a Fiscal Year Configuration
- Examples of Calendar Year and Fiscal Year Configurations
- Using the Undo Function with a Fiscal Year Configuration

***Configuring a Fiscal Year***

For a Ledger/Stat setup, Fiscal year configuration data is held in FSI\_FISCAL\_YEAR\_INFO, a special one-row, two column table. This table is seeded with values corresponding to a standard 12-month, January to December calendar year. For a Management Ledger setup, Fiscal year configuration data is held in 'LEDGER\_CCY\_AND\_FISCAL\_YR\_INFO'. If the Users fail to populate the aforesaid table, but do populate the FSI\_FISCAL\_YEAR\_INFO, this table acts as a fall back source for the Management Ledger/s.

FSI_FISCAL_YEAR_INFO / LEDGER_CCY_AND_FISCAL_YR_INFO table	
FISCAL_PERIOD	START_MONTH
12	1

If your organization operates on a fiscal year basis, the values in this table should be updated at the time OFSAA is installed. In the FSI-FISCAL-YEAR-INFO / LEDGER\_CCY\_AND\_FISCAL\_YR\_INFO table, the values in the Fiscal-Period and Start-Month columns determine the duration and beginning month of your financial year, respectively. All application processes that insert into, update, or delete from the Ledger/Stat table read the configuration data in FSI-FISCAL-YEAR-INFO / LEDGER\_CCY\_AND\_FISCAL\_YR\_INFO in order to properly maintain the 12 month-to-date and the 12 year-to-date columns in Ledger/Stat.

**Fiscal-Period Column**

This column holds a value that represents the duration of the financial year in months. For a calendar year, this value is always 12 months. For a fiscal year, this value is typically 12 but it may have a different value. Fiscal year configurations give you the flexibility to set shorter financial periods within a 12-month period. For example, if your organization has two 6-month fiscal periods over a 12-month span, then the value in this column would be 6.

For Oracle Financial Services Analytical Applications, the default setting is a calendar year and the value in the fiscal period column is 12. If you set your fiscal period to any value other than 12, the system automatically treats this setting as a fiscal year configuration. Valid fiscal periods include 1, 2, 3, 4, 6, or 12 months. Note that fiscal period values of other than 1, 2, 3, 4, 6, or 12 months are not supported and will cause application runtime errors.

### **Start-Month Column**

This column holds a value that represents the first month of your financial year. Calendar years, which always begin in January, have a Start Month value of 1. Start month would have a value of 3 if your fiscal year were to begin in March and a value of 7 if your fiscal year were to begin in July.

For Oracle Financial Services Analytical Applications, the default setting is a calendar year so the starting month value is 1. Note that starting month values of less than 1 or greater than 12 are not supported and will cause will cause application runtime errors.

## **How YTD Calculations Are Affected by a Fiscal Year Configuration**

The Ledger/Stat table contains each of your Key Processing dimensions, a number of administrative columns that are also components of the table's primary key, and 24 “fact” columns (balance columns). The first 12 fact columns (MTD-01 to MTD-12) contain monthly balances; the next 12 fact columns (YTD-01 to YTD-12) contain year-to-date balances. When you are running a standard calendar year configuration, MTD\_01 corresponds to January data and MTD\_12 corresponds to December data. If, however, your Start-Month column were set to 9 (indicating that your fiscal year begins in September), then the MTD\_01 in Ledger/Stat would correspond to September and MTD\_12 would correspond to August.

Year-to-date calculations begin at the start date of the fiscal period rather than January of the year in which the calculations are performed. For example, if July were the start date for your organization's fiscal period, and if you were to load General Ledger data to Ledger/Stat for July, August, and September, then the YTD\_03 column would contain fiscal year-to-date data for September, the 3rd month of your fiscal year. In addition to the Ledger/Stat load process, every OFSAA process that interacts with Ledger/Stat will reference FSI-FISCAL-YEAR-INFO / LEDGER\_CCY\_AND\_FISCAL\_YR\_INFO to properly maintain both the month columns and the YTD columns.

This is an important consideration for the following calculations:

- Allocation rules that insert into or update the Ledger/Stat table.
- The allocation UNDO process that performs updates or deletes from Ledger/Stat.
- Loading data into Ledger/Stat through the OFSAA Ledger Data Loader utility (For more details, see *Oracle Financial Services Analytical Applications Data Model Utilities User Guide*).
- Transfer pricing rate migration from instrument tables to the Ledger/Stat table through OFSAA Transfer Pricing.
- Direct transfer pricing of the Ledger/Stat table through OFSAA Transfer Pricing.

## **Examples of Calendar Year and Fiscal Year Configurations**

The following examples demonstrate different configurations in the FSI-FISCAL-YEAR-INFO / LEDGER\_CCY\_AND\_FISCAL\_YR\_INFO table and show how these configurations affect application functionality involving inserts into, updates of, or deletions from the Ledger/Stat table.

## Using a Calendar Year Configuration

Under the default calendar year configuration, the FSI-FISCAL-YEAR-INFO/ LEDGER\_CCY\_AND\_FISCAL\_YR\_INFO table contains a Fiscal Period value of 12 and a Start Month value of 1. For the following calendar year example, calendar months for 2011 correspond to the Ledger/Stat monthly buckets as shown in the following table:

Monthly Data for Calendar Year 2011											
MTD_01	MTD_02	MTD_03	MTD_04	MTD_05	MTD_06	MTD_07	MTD_08	MTD_09	MTD_10	MTD_11	MTD_12
Jan2011	Feb2011	Mar2011	Apr2011	May2011	Jun2011	Ju2011	Aug2011	Sep2011	Oct2011	Nov2011	Dec2011

If you have set your As-of-Date to June, 2011, then any OFSAA process that addresses Ledger/Stat will:

- Set or re-set the MTD\_06 column (updates and inserts)
- Set or re-set YTD\_01 to YTD\_12 (updates, inserts, and deletes)

Continuing with this calendar year example, the 12 Ledger/Stat YTD buckets correspond to 2011 calendar month ranges as shown in the following table:

Year-to-Date Data for Calendar Year 2011											
YTD_01	YTD_02	YTD_03	YTD_04	YTD_05	YTD_06	YTD_07	YTD_08	YTD_09	YTD_10	YTD_11	YTD_12
Jan2011	Jan to Feb 2011	Jan to Mar 2011	Jan to April 2011	Jan to May 2011	Jan to June 2011	Jan to July 2011	Jan to Aug 2011	Jan to Sep 2011	Jan to Oct 2011	Jan to Nov 2011	Jan to Dec 2011

For details on the relationships between Month-to-Date columns and Year-to-Date columns, see [Appendix A, “Management Ledger.”](#)

## Using a Fiscal Year Configuration with a 12-Month Duration

In this example, the FSI-FISCAL-YEAR-INFO/ LEDGER\_CCY\_AND\_FISCAL\_YR\_INFO table holds the following values:

- Start Month = 4
- Fiscal Period = 12

Since the fiscal period in this example is 12 months long so we can refer to the fiscal period as a fiscal year. In this fiscal year example, the months for the fiscal year correspond to Ledger/Stat monthly buckets as shown in the following table:

Monthly Data for Fiscal Year 2011												
YEAR_S value	MTD_01	MTD_02	MTD_03	MTD_04	MTD_05	MTD_06	MTD_07	MTD_08	MTD_09	MTD_10	MTD_11	MTD_12
2011	Apr 2011	May 2011	June 2011	July 2011	Aug 2011	Sep 2011	Oct 2011	Nov 2011	Dec 2011	Jan 2012	Feb 2012	Mar 2012

For this fiscal year example, the first calendar month of the 2011 fiscal year is April, 2011, and the last calendar month of the 2011 fiscal year is March, 2012. Note that all business data for fiscal year 2011 is stored in Ledger/Stat

where every row has the same YEAR-S value of 2011. If you were to set your As-of-Date to January 31, 2012 and then execute an allocation rule that output to Ledger/Stat, the allocation engine would write its month-to-date results to the MTD-10 column with YEAR-S equal to 2011.

Continuing with this fiscal year example, the 12 Ledger/Stat YTD buckets correspond to 2011 fiscal year YTD ranges as shown in the following table:

Year-to-Date Data for Fisca Year 2011												
YEAR _S value	YTD _01	YTD _02	YTD _03	YTD _04	YTD _05	YTD _06	YTD _07	YTD _08	YTD _09	YTD _10	YTD _11	YTD _12
2011	Apr 2011	Apr 2011 to May 2011	Apr 2011 to Jun 2011	Apr 2011 to Jul 2011	Apr 2011 to Aug 2011	Apr 2011 to Sep 2011	Apr 2011 to Oct 2011	Apr 2011 to Nov 2011	Apr 2011 to Dec 2011	Apr 2011 to Jan 2012	Apr 2011 to Feb 2012	Apr 2011 to Mar 2012

For details on the relationships between Month-to-Date columns and Year-to-Date columns, see [Appendix H, “Financial Element Aggregation Methods.”](#)

### Using a Fiscal Year Configuration with a 6-Month Duration

In this example, the FSI-FISCAL-YEAR-INFO/ LEDGER\_CCY\_AND\_FISCAL\_YR\_INFO table holds the following values:

- Start Month = 4
- Fiscal Period = 6

Since in this example the Fiscal Period is 6, we are no longer dealing with a Fiscal Year, but with two fiscal periods within a 12-month time span. In this fiscal period example, there are two 6-month fiscal periods within the 12 month window beginning April, 2011. Just as in the previous 12-month fiscal year example, the months for the two fiscal periods correspond to Ledger/Stat monthly buckets as shown in the following table:

Monthly Data for Fiscal Period #1, 2011 and Fiscal Period #2, 2011												
YEAR _S value	MTD _01	MTD _02	MTD _03	MTD _04	MTD _05	MTD _06	MTD _07	MTD _08	MTD _09	MTD _10	MTD _11	MTD _12
2011	Apr 2011	May 2011	June 2011	July 2011	Aug 2011	Sep 2011	Oct 2011	Nov 2011	Dec 2011	Jan 2012	Feb 2012	Mar 2012

Continuing with this fiscal period example, the 12 Ledger/Stat YTD buckets correspond to the two 2011 fiscal period YTD ranges (actually, fiscal period-to-date ranges) as shown in the following table:

Fiscal Period-to-Date Data for Fiscal Period #1, 2011 and Fiscal Period #2, 2011												
YEAR _S value	YTD _01	YTD _02	YTD _03	YTD _04	YTD _05	YTD _06	YTD _07	YTD _08	YTD _09	YTD _10	YTD _11	YTD _12
2011	Apr 2011	Apr 2011 to May 2011	Apr 2011 to Jun 2011	Apr 2011 to Jul 2011	Apr 2011 to Aug 2011	Apr 2011 to Sep 2011	Oct 2011	Oct 2011 to Nov 2011	Oct 2011 to Dec 2011	Oct 2011 to Jan 2012	Oct 2011 to Feb 2012	Oct 2011 to Mar 2012



Note that in this fiscal period example:

- The first calendar month of the first fiscal period is April, 2011
- The last calendar month of the first fiscal period is September, 2011
- The first calendar month of the second fiscal period is October, 2011
- The last calendar month of the second fiscal period is March, 2012

Also note that, as in the previous fiscal year example, all business data for this two fiscal-period example is stored in Ledger/Stat where every row has the same YEAR-S value of 2011. If you were to set your As-of-Date to January 31, 2012 and then execute an allocation rule that output to Ledger/Stat, the allocation engine would write its month-to-date results to the MTD-10 column with YEAR-S equal to 2011. These results are identical to the earlier 12 month fiscal year example. The YTD calculations, however, for the two fiscal periods are different.

When performing year-to-date calculations where the fiscal period value is 6 months, OFSAA segments the financial year into two 6-month periods. “Year-to-Date” in the context of a multi-period fiscal configuration really means “fiscal period-to-date”.

## ***Using the Undo Function with a Fiscal Year Configuration***

The allocation UNDO function removes the results of allocations from the Ledger/Stat table. The database function that operates against Ledger/Stat may be either an UPDATE or a DELETE. Generally speaking, the UNDO function operates as follows:

- Examine each row generated by the allocation rule a specific month of which we intend to UNDO.
- For each row, determine how many MTD columns in months other than the month currently being UNDONE contain non-zero values.
- If the answer is zero, delete the row. If the answer is not zero, update the MTD column that is to be UNDONE with a value of zero and re-set all YTD columns in the fiscal period of the month we are UNDOING.

### **Examples**

The following examples assumes you have run an allocation rule that posts data to Ledger/Stat for January, February, and March. When you perform an UNDO operation on the March execution:

- For any row in which either the January MTD column or the February MTD column contains a non-zero value.
  - The engine will update the March MTD column with a value of zero.
  - The engine will re-set (update) the YTD columns for each YTD column within the fiscal period of the month being UNDONE.
- The engine will delete any row in which neither the January MTD column nor the February MTD column contains a non-zero value.

With a Calendar Year configuration or with a Fiscal Year configuration, the re-setting of the YTD columns will affect all 12 YTD columns. However, with a Fiscal Period configuration (a configuration in which the Fiscal Period column in FSI-FISCAL-YEAR-INFO/ LEDGER\_CCY\_AND\_FISCAL\_YR\_INFO is set to 1, 2, 3, 4, or 6), the re-setting of YTD columns will only affect the YTD columns that fall within the same fiscal period as the month of the allocation that is being UNDONE.



**APPENDIX F***Seeded Financial Elements****Financial Elements Listed By Number***

<b>Financial Element Number</b>	<b>Description</b>	<b>Primary Usage</b>
60	Beginning Balance	ALM Output
70	Beginning Gross Rate	ALM Output
80	Beginning Net Rate	ALM Output
90	Beginning Transfer Rate	ALM Output
91	Beginning Liquidity Adjustment	FTP Output
92	Beginning Basis Risk Cost Rate	FTP Output
93	Beginning Pricing Incentive Rate	FTP Output
94	Beginning Other Adjustment Rate	FTP Output
100	End Balance	Initial Load; or Profitability Management or ALM Output
110	Ending Gross Rate	ALM Output
120	Ending Net Rate	ALM Output
130	Ending Transfer Rate	ALM Output
131	Ending Liquidity Adjustment Rate	FTP Output
132	Ending Basis Risk Cost Rate	FTP Output
133	Ending Pricing Incentive Cost Rate	FTP Output
134	Ending Other Adjustment Rate	FTP Output
140	Average Bal	Initial Load; or Profitability Management, FTP or ALM Output
141	Average Account Balance	ALM Output

Financial Element Number	Description	Primary Usage
142	Average Total Balance	ALM Output
143	Average Percentage Active	ALM Output
144	Average Volume Total	ALM Output
150	Average Gross Rate	ALM Output
160	Average Net Rate	ALM Output
170	Average Transfer Rate	Initial Load; or Profitability Management, FTP, or ALM Output
171	Average Historic Option Cost	ALM Output
172	Average Rem Term Transfer Rate	ALM Output
173	Average Current Option Cost	ALM Output
174	Average Liquidity Adjustment Rate	FTP Output
175	Average Basis Risk Cost Rate	FTP Output
176	Average Pricing Incentive Rate	FTP Output
177	Average Other Adjustment Rate	FTP Output
178	Inflation Adjustment to Principal	ALM Output
179	Inflation Adjustment to Interest	ALM Output
180	Prepay Runoff - Positive	ALM Output
181	Timing of Prepay Runoff - Positive	ALM Output
182	Prepay Runoff - Negative	ALM Output
183	Timing of Prepay Runoff - Negative	ALM Output
184	MOA Prepay Runoff	ALM Output
185	Timing of MOA Prepay Runoff	ALM Output
186	Writeoff - Positive	ALM Output

Financial Element Number	Description	Primary Usage
187	Timing of Writeoff - Positive	ALM Output
188	Writeoff - Negative	ALM Output
189	Timing of Writeoff - Negative	ALM Output
190	Payment Runoff - Positive	ALM Output
191	Timing of Payment Runoff - Positive	ALM Output
192	Payment Runoff - Negative	ALM Output
193	Timing of Payment Runoff - Negative	ALM Output
195	Maturity Runoff - Positive	ALM Output
196	Timing of Maturity Runoff - Positive	ALM Output
197	Maturity Runoff - Negative	ALM Output
198	Timing of Maturity Runoff - Negative	ALM Output
200	Non Maturity - Core Runoff	ALM Output
201	Timing of Non Maturity - Core Runoff	ALM Output
202	Non Maturity - Volatile Runoff	ALM Output
203	Timing of Non Maturity - Volatile Runoff	ALM Output
204	Devolvement Runoff	ALM Output
205	Timing of Devolvement Runoff	ALM Output
206	Recovery Runoff	ALM Output
207	Timing of Recovery Runoff	ALM Output
208	Non-Performing Asset Runoff	ALM Output

Financial Element Number	Description	Primary Usage
209	Timing of Non-Performing Asset Runoff	ALM Output
210	Total Runoff - Positive	ALM Output
211	Timing of Total Runoff - Positive	ALM Output
212	Total Runoff - Negative	ALM Output
213	Timing of Total Runoff - Negative	ALM Output
220	Total Runoff Gross Rate	ALM Output
225	Prepay Runoff Gross Rate	ALM Output
230	Total Runoff Net Rate	ALM Output
235	Prepay Runoff Net Rate	ALM Output
240	Total Runoff Transfer Rate	ALM Output
245	Prepay Runoff Transfer Rate	ALM Output
250	Repricing Balance	ALM Output
255	Repricing Balance At End	ALM Output
260	Before Repricing Gross Rate	ALM Output
270	After Repricing Gross Rate	ALM Output
280	Before Repricing Net Rate	ALM Output
290	After Repricing Net Rate	ALM Output
300	Before Reprice Transfer Rate	ALM Output
310	After Reprice Transfer Rate	ALM Output
320	Fully Indexed Gross Rate	ALM Output
330	Fully Indexed Net Rate	ALM Output
340	New Add Balance	ALM Output

Financial Element Number	Description	Primary Usage
341	Rollover Percentage	ALM Output
342	Net New Business	ALM Output
350	New Add Gross Rate	ALM Output
360	New Add Net Rate	ALM Output
370	New Add Transfer Rate	ALM Output
375	New Add Spread	ALM Output
380	Roll Add Balance	ALM Output
390	Roll Add Gross Rate	ALM Output
400	Roll Add Net Rate	ALM Output
410	Roll Add Transfer Rate	ALM Output
420	Interest	Initial Load; or Profitability Management or ALM Output
425	Interest Amount Gross	ALM Output
430	Interest Cash Flow	ALM Output
435	Interest Cash Flow Gross	ALM Output
437	Interest Cash Flow T-Rate	ALM Output
438	Interest CF (Without Offset)	ALM Output
440	Interest Accrued	ALM Output
441	Interest Accrued Net	ALM Output
442	Accrued Interest (Without Offset)	ALM Output
443	Accrued Interest Net (Without Offset)	ALM Output
445	Interest Accrued Gross	ALM Output

Financial Element Number	Description	Primary Usage
446	Interest Accrued Gross (current basis)	ALM Output
447	Accumulated Interest CF Net	ALM Output
448	Accumulated Interest CF Gross	ALM Output
449	Accumulated Interest CF Transfer Rate	ALM Output
450	Charge/Credit	Initial Load; or Profitability Management, FTP, or ALM Output
451	Historic Option Cost Charge/Credit	FTP Output
452	Charge/Credit Rem Term	FTP Output
453	Current Option Cost Charge/Credit	FTP Output
454	Interest Accrued Transfer Rate (Cur Bas)	ALM Output
455	Non Interest Income	Initial Load or Profitability Management
457	Non Interest Expense	Initial Load or Profitability Management
460	Accrued Interest Ending Balance	ALM Output
465	Total Currency Gain/Loss (Principal)	ALM Output
470	Accrued Interest Average Balance	ALM Output
475	Realized Currency Gain/Loss (Principal)	ALM Output
480	Interest Credited	ALM Output
485	Realized Currency Gain/Loss (Interest – Net)	ALM Output
486	Realized Currency Gain/Loss (Interest – Gross)	ALM Output



Financial Element Number	Description	Primary Usage
487	Realized Currency Gain/Loss (Interest – T-Rate)	ALM Output
490	Discount Rate	ALM Output
491	Timing of Cash Flow (in days)	ALM Output
500	WARM	ALM Output
510	Annual Prepayment Rate	ALM Output
515	Balance Before PrePay	ALM Output
520	Deferred End Balance	ALM Output
530	Deferred Average Balance	ALM Output
540	Deferred Runoff	ALM Output
550	Period Cap Balance	ALM Output
560	Period Cap Effect - Rate	ALM Output
570	Period Cap Effect - Amount	ALM Output
580	Life Cap Balance	ALM Output
590	Life Cap Effect - Rate	ALM Output
600	Life Cap Effect - Amount	ALM Output
610	Tease Balance	ALM Output
620	Tease Effect - Rate	ALM Output
630	Tease Effect - Amount	ALM Output
640	Neg-Am Balance	ALM Output
650	Neg-Am Interest	ALM Output
660	Gap Runoff	ALM Output
661	Gap Principal Runoff	ALM Output

Financial Element Number	Description	Primary Usage
662	Gap Repricing Runoff	ALM Output
663	Gap Deferred Runoff	ALM Output
670	Gap Runoff Term	ALM Output
671	Gap Interest Cash Flow Gross	ALM Output
672	Gap Interest Cash Flow Net	ALM Output
673	Gap Interest Cash Flow Transfer	ALM Output
674	Gap Accrued Interest Gross	ALM Output
675	Gap Accrued Interest Net	ALM Output
676	Gap Accrued Interest Transfer	ALM Output
677	Gap Interest Credited	ALM Output
680	Gap Runoff Gross Rate	ALM Output
681	Interest Accrued Gross Hist Fx Basis	ALM Output
682	Interest Accrued Net Hist Fx Basis	ALM Output
683	Interest Accrued Transfer Hist Fx Basis	ALM Output
690	Gap Runoff Net Rate	ALM Output
700	Gap Runoff Transfer Rate	ALM Output
710	Market Value	ALM Output
720	Duration	ALM Output
730	Convexity	ALM Output
760	Cur Pos Reprice Balance	ALM Output
765	Adj Cur Pos Reprice Balance	ALM Output

Financial Element Number	Description	Primary Usage
768	Cur Pos Before Reprice Rate	ALM Output
770	Cur Pos After Reprice Rate	ALM Output
775	Adj Cur Pos Reprice Rate	ALM Output
778	Cur Pos Before Reprice T-Rate	ALM Output
780	Cur Pos After Reprice T-Rate	ALM Output
785	Adj Cur Pos Reprice Transfer Rate	ALM Output
790	Cur Pos Runoff Balance	ALM Output
795	Adj Cur Pos Runoff Balance	ALM Output
800	Cur Pos Runoff Rate	ALM Output
805	Adj Cur Pos Runoff Rate	ALM Output
810	Cur Pos Runoff Transfer Rate	ALM Output
815	Adj Cur Pos Runoff Transfer Rate	ALM Output
820	Total Maturity/Repricing	ALM Output
830	Total Mat/Repricing Rate	ALM Output
840	Total Mat/Repricing Transfer Rate	ALM Output
900	Fee Income on Int. Bearing Acct.	ALM Output
905	Loan Fee Percentage	ALM Output
910	Tax Exempt Adjustment	ALM Output
920	Other Interest Income Adjustment	ALM Output
930	Federal Taxes	ALM Output
935	Local Taxes	ALM Output
940	Dividends	ALM Output

Financial Element Number	Description	Primary Usage
950	Accumulated Translation Amount	ALM Output
1001	Allocated_Amt	ALM Output
1101	Ending Balance per Account	ALM Output
1102	Ending Number of Accounts	ALM Output
1103	Percent of Active Ending Accounts	ALM Output
1104	Number of Active Accounts	ALM Output
1105	Ending Balance Unearned Discount	ALM Output
1106	Ending Gross Balance	ALM Output
1141	Average Account Balance (FE1141)	ALM Output
1142	Average Number of Accounts	ALM Output
1143	Average Percent Active Accounts	ALM Output
1144	Average Number of Active Accounts	ALM Output
1342	Net New Business (FE1342)	ALM Output
1343	New Balance Per Account	ALM Output
1344	Number of New Accounts	ALM Output
1660	Liquidity GAP Runoff (1661 + 1663)	ALM Output
1661	Liquidity GAP Principal Runoff	ALM Output
1663	Liquidity GAP Deferred Runoff	ALM Output
1670	Liquidity GAP Runoff Term	ALM Output
1671	Liquidity GAP Interest Cash Flow Gross	ALM Output

Financial Element Number	Description	Primary Usage
1672	Liquidity GAP Interest Cash Flow Net	ALM Output
1673	Liquidity GAP Interest Cash Flow Transfer Rate	ALM Output
1674	Liquidity GAP Accrued Interest Gross	ALM Output
1675	Liquidity GAP Accrued Interest Net	ALM Output
1676	Liquidity GAP Accrued Interest Transfer Rate	ALM Output
1677	Liquidity GAP Interest Credited	ALM Output
1678	Liquidity GAP Runoff Gross Rate	ALM Output
1679	Liquidity GAP Runoff Net Rate	ALM Output
1680	Liquidity GAP Runoff Transfer Rate	ALM Output
1794	Average Maturing Account Balance	ALM Output
1796	Number of Maturing Accounts	ALM Output
1823	Maturing Balance per Account	ALM Output
1824	Number of Maturing Active Accounts_24	ALM Output
1825	Percent of Active Maturing Accounts	ALM Output
1826	Number of Maturing Active Accounts_26	ALM Output
2001	Static Gross Par Balance	ALM Output
2002	Static Net Par Balance	ALM Output
2003	Static Deferred Balance	ALM Output
2004	Static Accrued Interest Balance	ALM Output

Financial Element Number	Description	Primary Usage
2005	Static Net Rate	ALM Output
2006	Static Transfer Rate	ALM Output
2007	Static Remaining Term	ALM Output
2008	Static Market Value	ALM Output
2009	Static Duration	ALM Output
9001	Interest Income	OFSPA Reporting Line
9002	Customer Break Funding Fees	OFSPA Reporting Line
9003	Amortization of Discount for Asset	OFSPA Reporting Line
9004	Amortization of Premium for Liability	OFSPA Reporting Line
9005	Pricing Incentive	OFSPA Reporting Line
9006	Credit for Other Allocated Liabilities	OFSPA Reporting Line
9007	Transfer Pricing Credit	OFSPA Reporting Line
9008	Credit for Liquidity	OFSPA Reporting Line
9009	Credit for Break Funding	OFSPA Reporting Line
9010	Central Bank Int. Income	OFSPA Reporting Line
9011	Credit for Float	OFSPA Reporting Line
9012	Early Redemption Fee	OFSPA Reporting Line
9013	Transfer Pricing Charge	OFSPA Reporting Line
9014	Charge for Liquidity	OFSPA Reporting Line
9015	Charge for Basis Risk	OFSPA Reporting Line
9016	Charge for Optionality	OFSPA Reporting Line
9017	Charge for Break Funding	OFSPA Reporting Line

Financial Element Number	Description	Primary Usage
9018	Charge for Other Allocated Assets	OFSPA Reporting Line
9019	Interest Expense	OFSPA Reporting Line
9020	Charge for Central Bank Reserves	OFSPA Reporting Line
9021	Amortization of Discount for Liability	OFSPA Reporting Line
9022	Amortization of Premium for Asset	OFSPA Reporting Line
9023	Economic Provision	OFSPA Reporting Line
9024	Deposit Insurance	OFSPA Reporting Line
9025	Credit for Loan Loss Reserve	OFSPA Reporting Line
9026	Credit for Equity	OFSPA Reporting Line
9027	Gross Fee Income	OFSPA Reporting Line
9028	Waived Fees	OFSPA Reporting Line
9029	Indirect Non-interest Income	OFSPA Reporting Line
9030	Direct Account Expense	OFSPA Reporting Line
9031	Indirect Distribution Expense	OFSPA Reporting Line
9032	Indirect Processing Expense	OFSPA Reporting Line
9033	Tax Expense	OFSPA Reporting Line
9034	Other Indirect Non-Interest Expense	OFSPA Reporting Line
9035	Average Balance	OFSPA Reporting Line
9036	Ending Balance	OFSPA Reporting Line
9037	Original Balance	OFSPA Reporting Line
9038	Float	OFSPA Reporting Line

<b>Financial Element Number</b>	<b>Description</b>	<b>Primary Usage</b>
9039	Allocated Loan Loss Reserve	OFSPA Reporting Line
9040	Other Allocated Assets	OFSPA Reporting Line
9041	Central Bank Reserve	OFSPA Reporting Line
9042	Other Allocated Liabilities	OFSPA Reporting Line
9043	Operating Risk Capital	OFSPA Reporting Line
9044	Credit Risk Capital	OFSPA Reporting Line
9045	Market Risk Capital	OFSPA Reporting Line
9046	Other Allocated Capital	OFSPA Reporting Line
9047	Interest Rate Risk Capital	OFSPA Reporting Line
9048	Liquidity Risk Capital	OFSPA Reporting Line
9050	Risk Weighted Average Balance	OFSPA Reporting Line



**APPENDIX G*****Debit and Credit Conventions******Conventions within the Management Ledger******Standard Accounting Conventions***

Under standard double entry accounting rules, accounting transactions must contain balanced debits and credits. Additionally, when you post a debit to a debit account, you increase the running balance for that debit account; and when you post a credit to a debit account, you reduce the running balance for that debit account. Conversely, when you post a credit to a credit account, you increase the running balance for that credit account; and when you post a debit to a credit account, you reduce the running balance for that credit account.

Posting A	To A	Effect
Debit	Debit Account	Increases the running balance
Credit	Debit Account	Decreases the running balance
Credit	Credit Account	Increases the running balance
Debit	Credit Account	Decreases the running balance

For example, if you generate an accounting transaction for a new loan, your transaction would debit the appropriate asset GL account (a debit account) for the new loan thus increasing the running balance for the Loan GL account; and would credit cash (also a debit account) thus reducing the running balance for the Cash GL account.

Debit	Credit
New loan \$500	Cash (\$500)

***Profitability Management Allocation Engine***

The Profitability Management allocation engine generates complex accounting transactions that have the effect of increasing or decreasing running balances for the accounts that are debited or credited. A single allocation rule may generate dozens, hundreds, or even thousands of debits and credits. In constructing allocation rules, however, you must take care to ensure (1) that the resulting debits and credits constitute a balanced transaction and (2) that your debits and credits match your business intent when you constructed the rule.

The following example shows the result of an allocation rule that gathers expenses for all expense GL Accounts from Cost Center 100 (a total of \$4,500 of expense), debits three destination centers (Cost Centers 1, 2, and 3), and credits Cost Center 100. In this example, only 3 GL accounts have been booked to Cost Center 100 (expense accounts 1, 2, and 3). The effect of this rule is to assign all of the expenses in Cost Center 100 to Cost Centers 1, 2, and 3; and to “relieve” (credit) Cost Center 100 by the same amount.

Debit	Credit
Expense Account 1 for Center 1: \$300	Expense Account 1 for Center 100: (\$1,200)
Expense Account 1 for Center 2: \$400	Expense Account 2 for Center 100: (\$1,500)
Expense Account 1 for Center 3: \$500	Expense Account 3 for Center 100: (\$1,800)
Expense Account 2 for Center 1: \$400	
Expense Account 2 for Center 2: \$500	

Debit	Credit
Expense Account 2 for Center 3: \$600	
Expense Account 3 for Center 1: \$500	
Expense Account 3 for Center 2: \$600	
Expense Account 3 for Center 3: \$700	

In the example above, only expense accounts (debit accounts) are being debited or credited because the allocation rule's Source was defined to focus on expense accounts. As a second example, imagine a similar allocation but for revenues booked to Cost Center 100 that you wish to distribute to Cost Centers 1, 2, and 3.

Debit	Credit
Revenue Account 1 for Center 100: (\$1,200)	Revenue Account 1 for Center 1: \$300
Revenue Account 2 for Center 100: (\$1,500)	Revenue Account 1 for Center 2: \$400
Revenue Account 3 for Center 100: (\$1,800)	Revenue Account 1 for Center 3: \$500
	Revenue Account 2 for Center 1: \$400
	Revenue Account 2 for Center 2: \$500
	Revenue Account 2 for Center 3: \$600
	Revenue Account 3 for Center 1: \$500
	Revenue Account 3 for Center 2: \$600
	Revenue Account 3 for Center 3: \$700

In this second example, only revenue accounts (credit accounts) are being debited or credited because the allocation rule's Source was defined to focus on revenue accounts. Note that in both examples, you need to take care in how you construct the definition of your allocation debits and credits to obtain the desired results. Generally, when you are allocating expenses or assets, you “target” receiving centers (or products or other dimensions) in your debit specification; and you offset the sources of those debit accounts in your credit specification. Conversely, when you are allocating revenues or liabilities or equity, you “target” receiving centers (or products or other dimensions) in your credit specification; and you offset the sources of those credit accounts in your debit specification.

Note that in both examples, the allocation rule's outputs were limited to only one type of account (debit accounts in the first example and credit accounts in the second example). When designing allocation rules, you generally need to take care that your allocation sources contain only one type of account. More specifically, you need to analyze your target accounts types as will be explained below.

### **Allocation Engine Logic**

When an allocation rule is configured to generate Ledger/Stat debits and credits, it generates two result sets: a debit result set and a credit result set. Generally, each row within a debit result set will be entirely composed of either debit balances or credit balances. Similarly, each row within a credit result set will generally be entirely composed of either debit balances or credit balances. Any given result set, however, may be composed of a mixture of both debit balances and credit balances. For this reason, it is important to explain in some detail the behavior of the allocation engine.

### **Debit Result Sets**

In evaluating debit result sets, the allocation engine examines each row of output in order to determine its account type, i.e., to determine whether the row represents a debit balance or a credit balance. If a row represents debit balance, the allocation engine preserves the signage of the row (i.e., multiplies it by plus one); if the row a row represents credit balance, the allocation engine reverses the signage of the row (i.e., multiplies it by minus one).

Allocation Output Tab	Account Type of Output Row	Rule
Debit	Debit	Multiply by +1
Debit	Credit	Multiply by -1

### **Credit Result Sets**

Similarly, in evaluating credit result sets, the allocation engine examines each row of output in order to determine its account type, i.e., to determine whether the row represents a debit balance or a credit balance. If a row represents credit balance, the allocation engine preserves the signage of the row (i.e., multiplies it by plus one); if the row a row represents debit balance, the allocation engine reverses the signage of the row (i.e., multiplies it by minus one).

Allocation Output Tab	Account Type of Output Row	Rule
Credit	Debit	Multiply by -1
Credit	Credit	Multiply by +1

### **Determining Account Type**

In examining any output row (whether from an allocation's debit specification or its credit specification) the allocation engine employs the following logic:

Examine the row's Financial Element

Determine the value of the Financial Element's Account Type attribute

If the Financial Element's Account Type attribute cannot determine account type, examine the row's Common Chart of Accounts (Common COA) value

Determine the value of the Common COA's Account Type attribute

Determine the Account Type of the Financial Element

Account Type is a “dimension type” attribute that is shared by both the Financial Element and Common Chart of Accounts dimensions. Dimension type attributes are based upon simple dimensions, i.e., lists of dimension member values. The values of the Account Type dimension constitute detailed account types each of which is fundamentally either a debit type or a credit type.

### Detailed Account Type Values

In order to determine whether a target output row is a Debit row or a Credit row, for every row, the allocation engine examines the Account Type associated with the row's Financial Element.

Account Type Dimension Members	
Fundamental Account Type	Detailed Account Type
Debit	Dividends
	Earning Assets
	Interest Expense
	Non-Interest Expense
	Off Balance Sheet Payable
	Other Asset
	Statistical
	Taxes
	Equity
Credit	Interest Bearing Liabilities
	Interest Income
	Non-Interest Income
	Off Balance Sheet Receivable
	Other Liability

In some cases, however, the Account Type associated with a row's Financial Element is ambiguous. For example, Financial Element 457 – Non-Interest Expense is clearly a debit account type (its detailed Account Type is Non-Interest Expense). The Account Type of Financial Element 100 – Ending Balance, however, is ambiguous. An Ending Balance could be either an asset or a liability. In such cases, the Account Type associated with a Financial Element is a special value called < Use Detailed Leaves >. What this means is that the engine should fall back upon the Common COA dimension to determine a row's Account Type.

Determine the Account Type of the Common Chart of Accounts

When a row's fundamental account type (DR or CR) cannot be determined from the Account Type attribute associated with its Financial Element (i.e., when the Account Type attribute takes on the special < Use Detail Leaves > value), the allocation engine determines the row's debit or credit status by examining the Account Type attribute associated with its Common Chart of Account value.

### Practical Implications

#### Designing Allocation Rules

One approach for avoiding undesired allocation results is to ensure that your allocation Sources are limited to either debit accounts or to credit accounts. Another approach is to output your results to unambiguous Financial Elements, i.e., to Financial Elements whose account types are known with certainty. For example, when allocating balance-sheet balances, you need to ensure that your allocation Source definition is limited to 100% asset balances or to 100% liability balances. On the output side of an asset allocation, you might choose to debit a user-defined Financial Element whose Account Type is unambiguous (e.g., “Allocated Assets”).

## ***Designing General Ledger Extracts***

When coding your General Ledger extract, you must map each of your ledger balances to specific Financial Element and Common Chart of Account values. Financial Element values are straightforward and easy to map, but if your ledger lacks a product dimension, it is not clear how you should map each source balance to a Common COA value. One approach you might use is to build 14 Common COA dimension member values each of which you associate with one of the 14 detailed Account Types (see table above). With this approach, you can easily map your initial ledger balances.

### ***Signage Methods***

When OFSAA is initially installed, you must choose one of three available Signage Methods that indicate how you have chosen to store balances in the Ledger/Stat table.

### ***Natural Signage Convention***

#### ***GAAP Signage Convention***

#### ***Reverse GAAP Signage Convention***

One of these conventions is likely to reflect how balances are stored in your General Ledger, but you need not necessarily adopt the same convention in your OFSAA data model as was adopted in your General Ledger. This is a decision you need to make as you are designing your GL data extract.

### ***Natural Signage Convention***

Under the Natural Signage Convention, balances initially loaded into Ledger/Stat are almost always positively signed. The only time an initially loaded balance is negatively signed is when the balance in question is in a “contra” state. Accounts described as contra-accounts are normally in a “contra” state, i.e., their balances are normally negative. Examples of such accounts include loan loss reserves, depreciation and amortization. Any account, however, may be in a contra state, i.e., carry a negative balance.

#### ***GAAP Signage Convention***

Under the GAAP signage convention, debit accounts (asset and expense balances) are generally stored with positive signage and credit accounts (liability, equity, and revenue balances) are generally stored with a negative signage. As with the Natural Signage convention, the exceptions are for balances that are in a “contra” state. Some prefer the GAAP signage convention over the Natural Signage convention because it is perceived to be easier to construct balance sheets when assets and liabilities are oppositely signed.

#### ***Reverse GAAP Signage Convention***

Under the Reverse GAAP signage convention, debit accounts (asset and expense balances) are generally stored with negative signage and credit accounts (liability, equity, and revenue balances) are generally stored with a positive signage. As with the Natural Signage convention, the exceptions are for balances that are in a “contra” state. Some prefer the Reverse GAAP signage convention over either the Natural Signage convention or the GAAP Signage convention because it is perceived to be easier to construct income statements when revenues are positively signed and expenses are negatively signed.

## ***Practical Implications***

### ***Choosing a Signage Convention***

Signage conventions have evolved over time in order to simplify reporting processes. With modern reporting tools, your choice is not highly significant. The OFS Profitability Analytics BI solution, for example, automatically takes your choice of Signage into account when it builds multidimensional cubes from your Ledger/Stat table. Regardless of the convention you have chosen, OFSPA displays balance sheets and income statements using a visually pleasing “natural” convention (i.e., all balances, with understood exceptions, are shown as positive balances).

The Natural Signage convention is the most commonly recommended method for OFSAA implementations.

### ***Implications for Allocation Rules***

Your choice of Signage convention has no effect on the design of your allocation rules. To see why this is the case, examine the following examples.

- If you have chosen to store your expenses as negative numbers (Reverse GAAP convention), an allocation of expense from one cost center to another cost center will debit the receiving center with negative expense (signage is preserved) and will credit the source center with positive expense (signage is reversed)
- If you have chosen to store your liabilities as negative numbers (GAAP convention), a liability allocation from one cost center to another cost center will credit the target center with a negative liability balance (signage is preserved) and will debit the source center with a positive liability balance (signage is reversed)

In short, the allocation engine is indifferent to (and actually has no knowledge of) the Signage convention you have adopted.

**APPENDIX H**

# *Financial Element Aggregation Methods*

## **Overview**

There are a number of processes that insert into, update, or delete from the Ledger/Stat table.

- Ledger load processes
- Allocation rules
- Allocation UNDOs
- Direct transfer pricing of Ledger/Stat
- Transfer pricing rate migration to Ledger/Stat

Every process that inserts into, updates, or deletes from Ledger/Stat targets a single month-to-date balance (MTD\_01 to MTD\_12). At the same time, each such process calculates or re-calculates appropriate values for each of the 12 year-to-date balances (YTD\_01 to YTD\_12). For every row in the Ledger/Stat table, the relationship between month-to-date values and year-to-date values is governed by the Aggregation Method attribute of the Financial Element dimension.

## **Aggregation Methods**

Aggregation Methods are assigned as attributes to Financial Elements. A dimension attribute, Aggregation Method is itself a simple dimension, i.e., it is defined by a list of values. The Aggregation Method dimension is a reserved dimension (meaning that you may not modify its membership). Additionally, Aggregation Method is a required attribute of the Financial Element (meaning that an Aggregation Method must be assigned to each Financial Element). An Aggregation Method is already assigned to each reserved Financial Element and you may not modify reserved Financial Elements. You must, however, assign an Aggregation Method to any user-defined Financial Element that you create. The members of the Aggregation Method dimension are:

- Add
- Average
- Average by Days Weighted
- Beginning
- Last

The following descriptions are general in that they assume a normal calendar year configuration (for details on configuring a fiscal year calendar that does not begin in January and/or has a duration other than 12 months, see *Fiscal Year Information*, on page 371).

### **Add**

For Financial Elements having the “Add” Aggregation Method, each YTD value is calculated as the sum of each preceding month. For example, the value of YTD\_04 for any row in Ledger/Stat for which the Financial Element

has the Aggregation Method of “Add” is MONTH\_01 + MONTH\_02 + MONTH\_03 + MONTH\_04. The “Add” method is assigned to all income and expense type Financial Elements.

### ***Average***

For Financial Elements having the “Average” Aggregation Method, each YTD value is calculated as the average of each preceding month. For example, the value of YTD\_04 for any row in Ledger/Stat for which the Financial Element has the Aggregation Method of “Average” is  $(\text{MONTH\_01} + \text{MONTH\_02} + \text{MONTH\_03} + \text{MONTH\_04}) / 4$ .

### ***Average by Days Weighted***

For Financial Elements having the “Average by Days Weighted” Aggregation Method, each YTD value is calculated as the weighted average of each preceding month. For example, the value of YTD\_04 for any row in Ledger/Stat for which the Financial Element has the Aggregation Method of “Average by Days Weighted” is  $[(\text{MONTH\_01} \times 31) + (\text{MONTH\_02} \times 28) + (\text{MONTH\_03} \times 31) + (\text{MONTH\_04} \times 30)] / (31 + 28 + 31 + 30)$ . “Average by Days Weighted” calculations are not adjusted to use 29 days in leap years.

### ***Beginning or Last***

For Financial Elements having either the “Beginning” or “Last” Aggregation Method, each YTD value is equal to the value for MTD for the same month. For example, the value of YTD\_04 for any row in Ledger/Stat for which the Financial Element has the Aggregation Method of “Beginning” or “Last” is simply MONTH\_04.



**APPENDIX I**

# *Performance Tuning for Profitability Management*

## **SQL Hints**

### **Overview**

A typical Profitability Management implementation involves dozen, hundreds, or sometimes even thousands of distinct allocation rules. There is commonly a bell-curve distribution of processing times for each allocation rule. Some finish in seconds, some finish in minutes, and others may take hours to complete. An allocation's run time is normally closely related to the volume of data it generates, but there may be times when an allocation runs slower than it should due to stale database statistics, skews in the data being queried, a lack of appropriate indexes, or a poor execution plan generated by the database optimizer. Your DBA may be able to enhance the overall performance of the database by gathering database statistics more frequently, generating additional indexes, or by other tuning techniques. One such additional tuning technique is to add **hints** to your queries.

Oracle Financial Services Profitability Management allows users to exploit SQL Hints to tune the performance of allocation rules. SQL Hints provide a mechanism to instruct the optimizer to choose a desired query execution plan based on specified hint criteria. The Profitability Management allocation engine reads user-defined hints that are stored in the FSI\_SQL\_HINTS\_OPTIONS table.

Profitability Management supports 4 types of SQL hint:

- DML\_HINT
- PERCENT\_DIST\_HINT
- TREE\_FILTER\_HINT
- FILTER\_ON\_HINT

Note:

- Hints must be defined by users; there are no default hints
- There is currently no user interface for the definition of hints. Your DBA can assist you in defining, testing, and optimizing SQL Hints. SQL Hints must be manually inserted into the FSI\_SQL\_HINTS\_OPTIONS table.
- The FSI\_SQL\_HINTS\_OPTIONS table is uniquely indexed by (1) the system identifier (SYS\_ID\_NUM) of the rule to which the hint applies and (2) by the allocation engine's PROCESS\_ENGINE\_CD (zero). Currently, only the Profitability Management engine supports SQL Hints; additional analytical application engines may support SQL Hints in future releases.
- You may determine an allocation's SYS\_ID\_NUM by performing a **mouse-over** on the allocation rule's name in the Allocation Specification summary screen.
- Hints are limited to 250 characters
- DML\_HINT and FILTER\_ON\_HINT are applicable to all types of allocations.

- TREE\_FILTER\_HINT is applicable only to allocation rules that include a Hierarchy Filter.
- PERCENT\_DIST\_HINT is applicable only to Dynamic Driver allocations.

## Types of SQL hints

Profitability Management supports four types of SQL hints.

### Filter On Hint

A hint inserted into the FILTER\_ON\_HINT column will be applied to the SELECT query on the allocation's source table (LEDGER\_STAT, Instrument Table, or Transaction Summary Table).

---

**Note:** Effective with release 8.0.5, hints are also supported for Management Ledger tables.

---

### Tree Filter Hint

For an allocation rule using a Hierarchy Filter, the hint text specified in the TREE\_FILTER\_HINT column will be applied to sub-selects from the OFSA\_IDT\_ROLLUP table.

### Percent Distribution Hint

A hint inserted to the PERCENT\_DIST\_HINT column will be applied to the SELECT statement that acquires driver data for Dynamic Driver allocation rules.

### DML Hint

A hint inserted into the DML\_HINT column will be applied to DML operations that insert into or update target tables, that is, Debit or Credit operations. Note that hints that specify database parallelism for DML operations will be ineffective unless you have enabled DML parallelism in FSI\_PARALLEL\_PARAMETERS see [Profitability Management Parallel Execution](#).

## How to use SQL Hints

To enable one or more SQL Hints for an allocation rule, you must generate appropriate entries into the FSI\_SQL\_HINTS\_OPTIONS table for the allocation rule to which you want your hints to apply. For general guidance on SQL Hints, see *Oracle Database Performance Tuning Guide*.

Following is a sample query to insert a FILTER ON HINT and a TREE FILTER HINT into FSI\_SQL\_HINTS\_OPTIONS:

```
INSERT INTO FSI_SQL_HINTS_OPTIONS
  (SYS_ID_NUM,
   PROCESS_ENGINE_CD,
   FILTER_ON_HINT,
   DML_HINT,
   TREE_FILTER_HINT,
   PERCENT_DIST_HINT)
VALUES
```

```
(201230,
0,
'/*+ parallel(ledger_stat, 8)*/',
,
'/*+ parallel(ofsa_idt_rollup, 2)*/'
,
)
```

In this example:

- SYS\_ID\_NUM is the system identifier for the allocation rule to which the hint is applicable. You may determine the system identifier for an allocation rule by performing a **mouse-over** on the rule's name in the Allocation Specification summary screen.
- PROCESS\_ENGINE\_CD is set to a value of zero for the Profitability Management engine. Other Oracle Financial Services Analytical Applications (OFSAA) engines may support SQL Hints in future releases.

In designing hints to enhance the performance of slow running rules, your DBA will typically begin by capturing the slow running SQL. You may capture the SQL generated by an allocation rule from the OFSAA log files (first set Profitability Management -> Application Preferences -> Debugging Output Level to Show All SQL) or through generic database tracing (For more information about using Application Tracing Tools, see *Oracle Database Performance Tuning Guide*). Once your DBA has captured the slow running SQL and its execution plan, he or she can experiment offline to determine what form of hint will provide the best performance.

## Sample Queries

Several examples follow of DML statements that the allocate engine will issue to generate debits and credits. These examples illustrate where the four kinds of supported hints would be inserted into the queries. Note that while these examples all demonstrate hints that suggest parallelism to the database optimizer, you may utilize any valid form of hint.

### Filter on Hint:

```
MERGE INTO LEDGER_STAT TARGET
USING (
SELECTC.*
FROM (
SELECT 11002.000000 identity_code, 1994 year_s, 'D'
accum_type_cd, 100 consolidation_cd, 101 balance_type_cd, 'USD'
ISO_CURRENCY_CD, 457.000000 FINANCIAL_ELEM_ID, ORG_UNIT_ID,
GL_ACCOUNT_ID, 821.000000 COMMON_COA_ID, PRODUCT_ID, (
SOURCE_AMOUNT + leaf_amount ) * 1 BALANCE_AMOUNT
from (SELECT /*+ parallel(ledger_stat, 8) */row_number() over (partition by
a.PRODUCT_ID,
a.GL_ACCOUNT_ID, a.ORG_UNIT_ID order bya.PRODUC T_ID, a.GL_ACCOUNT_ID,
a.ORG_UNIT_ID)rw , SUM(DECODE(COMMON_COA_ID, 101, 0,
```

```
DECODE(a.year_s, 1994, a.month_03, 0)) ) over (partition by
a.PRODUCT_ID, a.GL_ACCOUNT_ID, a.ORG_UNIT_ID) SOURCE_AMOUNT,
SUM(DECODE(COMMON_COA_ID, 101, DECODE(a.year_s, 1994, a.month_03,
0), 0)) over ( partition bya.PRODUCT_ID, a.GL_ACCOUNT_ID,
a.ORG_UNIT_ID ) leaf_amount , a.PRODUCT_ID, a.GL_ACCOUNT_ID,
a.ORG_UNIT_ID
from LEDGER_STAT a
where(a.COMMON_COA_ID=820
OR a.COMMON_COA_ID=101)
and (((a.IDENTITY_CODE = 0)
or (a.IDENTITY_CODE = 1)
or (a.IDENTITY_CODE = 700)
or (a.IDENTITY_CODE = 779)))
and (a.identity_code) IN (select distinct a.parent_identity_code
from FSI_M_DATA_IDENTITY_DETAIL a
where a.identity_code in (11002)
and a.as_of_date = '03/ 31/ 1994'
and a.src_drv_type in ('S' , 'D') )
and a.year_s IN (1994) )
where rw = 1 ) C ) SOURCE ON (TARGET.FINANCIAL_ELEM_ID =
457.000000
AND TARGET.ORG_UNIT_ID = SOURCE.ORG_UNIT_ID
AND TARGET.GL_ACCOUNT_ID = SOURCE.GL_ACCOUNT_ID
AND TARGET.COMMON_COA_ID = 821.000000
AND TARGET.PRODUCT_ID = SOURCE.PRODUCT_ID
AND TARGET.YEAR_S = 1994
AND TARGET.IDENTITY_CODE = 11002.000000
AND TARGET.ISO_CURRENCY_CD = SOURCE.ISO_CURRENCY_CD
AND TARGET.CONSolidATION_CD = 100
AND TARGET.ACCUM_TYPE_CD = 'D' ) WHEN MATCHED THEN
UPDATE
SET TARGET.MONTH_03 = TARGET.MONTH_03 + SOURCE.BALANCE_AMOUNT,
BALANCE_TYPE_CD = 101, ytd_03 = ytd_02 + SOURCE.BALANCE_AMOUNT , ytd_04 = ytd_03 +
month_04 , ytd_05 = ytd_04 + month_05 , ytd_06
= ytd_05 + month_06 , ytd_07 = ytd_06 + month_07 , ytd_08 =
ytd_07 + month_08 , ytd_09 = ytd_08 + month_09 , ytd_10 = ytd_09
+ month_10 , ytd_11 = ytd_10 + month_11 , ytd_12 = ytd_11 +
month_12 WHEN NOT MATCHED THEN
```

```

INSERT (FINANCIAL_ELEM_ID, ORG_UNIT_ID, GL_ACCOUNT_ID,
COMMON_COA_ID, PRODUCT_ID, ACCUM_TYPE_CD, BALANCE_TYPE_CD,
CONSOLIDATION_CD, CURRENCY_TYPE_CD, ISO_CURRENCY_CD,
IDENTITY_CODE, MONTH_01, MONTH_02, MONTH_03, MONTH_04, MONTH_05,
MONTH_06, MONTH_07, MONTH_08, MONTH_09, MONTH_10, MONTH_11,
MONTH_12, YTD_01, YTD_02, YTD_03, YTD_04, YTD_05, YTD_06, YTD_07,
YTD_08, YTD_09, YTD_10, YTD_11, YTD_12, YEAR_S )
VALUES ( 457.000000, SOURCE.ORG_UNIT_ID, SOURCE.GL_ACCOUNT_ID,
821.000000, SOURCE.PRODUCT_ID, 'D' , 101 , 100 , 0 ,
SOURCE.ISO_CURRENCY_CD, 11002.000000 , 0.0, 0.0,
SOURCE.BALANCE_AMOUNT, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0, 0, SOURCE.BALANCE_AMOUNT , SOURCE.BALANCE_AMOUNT ,
SOURCE.BALANCE_AMOUNT , SOURCE.BALANCE_AMOUNT ,
SOURCE.BALANCE_AMOUNT , SOURCE.BALANCE_AMOUNT ,
SOURCE.BALANCE_AMOUNT , SOURCE.BALANCE_AMOUNT ,
SOURCE.BALANCE_AMOUNT , SOURCE.BALANCE_AMOUNT , 1994)

```

**Tree Filter Hint:**

```

MERGE INTO LEDGER_STAT TARGET
USING (
SELECT 11040.000000 identity_code, 1994 year_s, 'D'
accum_type_cd , 100 consolidation_cd , 101 balance_type_cd, 'USD'
ISO_CURRENCY_CD, 10900.000000 FINANCIAL_ELEM_ID, ORG_UNIT_ID,
GL_ACCOUNT_ID, COMMON_COA_ID, PRODUCT_ID, SOURCE_AMOUNT * 1
BALANCE_AMOUNT
from (SELECT row_number() over (partition by a.PRODUCT_ID,
a.COMMON_COA_ID
order by a.PRODUCT_ID, a.COMMON_COA_ID)rw, SUM(DECODE(a.year_s, 1994,
a.month_03, 0) * 2 ) over (partition by a.PRODUCT_ID,
a.COMMON_COA_ID) SOURCE_AMOUNT , a.PRODUCT_ID, a.COMMON_COA_ID ,
a.GL_ACCOUNT_ID, a.ORG_UNIT_ID
from LEDGER_STAT a
where a.FINANCIAL_ELEM_ID=100
and a.ORG_UNIT_ID=2100
and a.GL_ACCOUNT_ID IN (SELECT /*+ parallel (ofsa_idt_rollup,
8) */ leaf_node
FROM OFSA_IDT_ROLLUP
WHERE OFSA_IDT_ROLLUP.sys_id_num = 2000779

```

```
and (OFSA_IDT_ROLLUP.leaf_node = 101301))
and (((a.CONSolidATION_CD = '100'))
and ((a.IDENTITy_CODE = 0)))
and (a.identity_code) IN (select distinct a.parent_identity_code
from FSI_M_DATA_IDENTITy_DETAIL a
where a.identity_code in (11040)
and a.as_of_date = '03/ 31/ 1994'
and a.src_drv_type = 'S' )
and a.year_s IN (1994) )
where rw = 1 ) SOURCE ON (TARGET.FINANCIAL_ELEM_ID = 10900.000000
AND TARGET.ORG_UNIT_ID = SOURCE.ORG_UNIT_ID
AND TARGET.GL_ACCOUNT_ID = SOURCE.GL_ACCOUNT_ID
AND TARGET.COMMON_COA_ID = SOURCE.COMMON_COA_ID
AND TARGET.PRODUCT_ID = SOURCE.PRODUCT_ID
AND TARGET.YEAR_S = 1994
AND TARGET.IDENTITy_CODE = 11040.000000
AND TARGET.ISO_CURRENCY_CD = SOURCE.ISO_CURRENCY_CD
AND TARGET.CONSolidATION_CD = 100
AND TARGET.ACCUM_TYPE_CD = 'D' ) WHEN MATCHED THEN
UPDATE
SET TARGET.MONTH_03 = TARGET.MONTH_03 + SOURCE.BALANCE_AMOUNT,
BALANCE_TYPE_CD = 101, ytd_03 = ((ytd_02* 2) +
SOURCE.BALANCE_AMOUNT )/ 3 , ytd_04 = ((ytd_03* 3) + month_04 )/
4 , ytd_05 = ((ytd_04* 4) + month_05 )/ 5 , ytd_06 = ((ytd_05* 5)
+ month_06 )/ 6 , ytd_07 = ((ytd_06* 6) + month_07 )/ 7 , ytd_08
= ((ytd_07* 7) + month_08 )/ 8 , ytd_09 = ((ytd_08* 8) + month_09
/ 9 , ytd_10 = ((ytd_09* 9) + month_10 )/ 10 , ytd_11 =
((ytd_10* 10) + month_11 )/ 11 , ytd_12 = ((ytd_11* 11) +
month_12 )/ 12 WHEN NOT MATCHED THEN
INSERT (FINANCIAL_ELEM_ID, ORG_UNIT_ID, GL_ACCOUNT_ID,
COMMON_COA_ID, PRODUCT_ID, ACCUM_TYPE_CD, BALANCE_TYPE_CD,
CONSolidATION_CD, CURRENCY_TYPE_CD, ISO_CURRENCY_CD,
IDENTITy_CODE, MONTH_01, MONTH_02, MONTH_03, MONTH_04, MONTH_05,
MONTH_06, MONTH_07, MONTH_08, MONTH_09, MONTH_10, MONTH_11,
MONTH_12, YTD_01, YTD_02, YTD_03, YTD_04, YTD_05, YTD_06, YTD_07,
YTD_08, YTD_09, YTD_10, YTD_11, YTD_12, YEAR_S )
VALUES ( 10900.000000, SOURCE.ORG_UNIT_ID, SOURCE.GL_ACCOUNT_ID,
```

```

SOURCE.COMMON_COA_ID, SOURCE.PRODUCT_ID, 'D' , 101 , 100 , 0 ,
SOURCE.ISO_CURRENCY_CD, 11040.000000 , 0.0, 0.0,
SOURCE.BALANCE_AMOUNT, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0, 0, (SOURCE.BALANCE_AMOUNT) / 3, ((
(SOURCE.BALANCE_AMOUNT) / 3)* 3) / 4, (((
(SOURCE.BALANCE_AMOUNT) / 3)* 3) / 4)* 4) / 5, ((((((
(SOURCE.BALANCE_AMOUNT) / 3)* 3) / 4)* 4) / 5)* 5) / 6, (((((((
(SOURCE.BALANCE_AMOUNT) / 3)* 3) / 4)* 4) / 5)* 5) / 6)* 6) / 7,
((((((((((( (SOURCE.BALANCE_AMOUNT) / 3)* 3) / 4)* 4) / 5)* 5) /
6)* 6) / 7)* 7) / 8, (((((((((((((( (SOURCE.BALANCE_AMOUNT) / 3)*
3) / 4)* 4) / 5)* 5) / 6)* 6) / 7)* 7) / 8)* 8) / 9,
((((((((((((((((((( (SOURCE.BALANCE_AMOUNT) / 3)* 3) / 4)* 4) / 5)* 5)
/ 6)* 6) / 7)* 7) / 8)* 8) / 9)* 9) / 10, ((((((((((((((((((((((
(SOURCE.BALANCE_AMOUNT) / 3)* 3) / 4)* 4) / 5)* 5) / 6)* 6) / 7)*
7) / 8)* 8) / 9)* 9) / 10)* 10) / 11, ((((((((((((((((((((((((((
(SOURCE.BALANCE_AMOUNT) / 3)* 3) / 4)* 4) / 5)* 5) / 6)* 6) / 7)*
7) / 8)* 8) / 9)* 9) / 10)* 10) / 11)* 11) / 12 , 1994)

```

**Pct Distr Hint:**

```

MERGE INTO FSI_D_MERCHANT_CARDS TARGET USING (SELECT SUM
(IVW_SOURCE.IVW_TARGET_AMOUNT) TARGET_AMOUNT , IDENTITY_CODE,
ID_NUMBER
FROM(
SELECT ( Src.SOURCE_AMOUNT * Drv.driv_factor) IVW_TARGET_AMOUNT,
Drv.IDENTITY_CODE, Drv.ID_NUMBER
FROM (SELECT SUM(DECODE(a.year_s, 1994, a.month_03, 0) )
SOURCE_AMOUNT
FROM LEDGER_STAT a
where a.FINANCIAL_ELEM_ID=100
and a.ORG_UNIT_ID=2100
and a.GL_ACCOUNT_ID IN (SELECT leaf_node
FROM OFSA_IDT_ROLLUP
WHERE OFSA_IDT_ROLLUP.sys_id_num = 2000779
and (OFSA_IDT_ROLLUP.leaf_node = 101301))
and ((a.CONSOLIDATION_CD = '100'))
and((a.IDENTITY_CODE = 0)))
and (a.identity_code) IN (select distinct a.parent_identity_code

```

```
from FSI_M_DATA_IDENTITY_DETAIL a
where a.identity_code in (10316)
and a.as_of_date = '03/ 31/ 1994'
and a.src_drv_type = 'S' )
and a.year_s IN (1994)) Src , (SELECT ( ColVal / mTotal )
drv_factor , IDENTITY_CODE, ID_NUMBER
from (SELECT /*+ parallel(FSI_D_MERCHANT_CARDS, 8) */
a.IDENTITY_CODE, a.ID_NUMBER, (TOTAL_TRANSACTIONS) ColVal, (SUM
(TOTAL_TRANSACTIONS) OVER()) mTotal
from FSI_D_MERCHANT_CARDS a
where a.GL_ACCOUNT_ID=101301
and a.as_of_date='03/ 31/ 1994')
WHERE mTotal <> 0) Drv) IVW_SOURCE
WHERE IVW_TARGET_AMOUNT <> 0
GROUP BY IDENTITY_CODE , ID_NUMBER ) SOURCE ON
(SOURCE.IDENTITY_CODE = TARGET.IDENTITY_CODE
AND SOURCE.ID_NUMBER = TARGET.ID_NUMBER ) WHEN MATCHED THEN
UPDATE
SET IDENTITY_CODE_CHG = 10316.000000 , TARGET.ATM_EXP =
DECODE(identity_code_chg, 10316.000000, ATM_EXP, 0)+
(SOURCE.TARGET_AMOUNT * 1)
```

**DML Hint:**

```
MERGE /*+ parallel(ledger_stat, 4) */ INTO LEDGER_STAT TARGET
USING (SELECT (SUM (IVW_SOURCE.IVWSRC_AMOUNT) * 1 )
BALANCE_AMOUNT, ORG_UNIT_ID, GL_ACCOUNT_ID, PRODUCT_ID
FROM (SELECT (Src.src_amount * Drv.drv_factor) IVWSRC_AMOUNT,
Src.ORG_UNIT_ID, Src.GL_ACCOUNT_ID, Src.PRODUCT_ID
FROM (SELECT SUM(DECODE(a.year_s, 1994, a.month_03, 0) )
src_amount , a.GL_ACCOUNT_ID, a.PRODUCT_ID, a.ORG_UNIT_ID
from LEDGER_STAT a
where ((a.IDENTITY_CODE = 0)
or (a.IDENTITY_CODE = 1)
or (a.IDENTITY_CODE = 700)
or (a.IDENTITY_CODE = 779)))
and (a.identity_code) IN (select distinct a.parent_identity_code
from FSI_M_DATA_IDENTITY_DETAIL a
```



```

where a.identity_code in (10993)
and a.as_of_date = '03/ 31/ 1994'
and a.src_drv_type = 'S' )
and a.year_s IN (1994)
GROUP BY a.GL_ACCOUNT_ID, a.PRODUCT_ID, a.ORG_UNIT_ID) Src JOIN
(SELECT (ColVal/ mTotal) Drv_factor, ORG_UNIT_ID
FROM (SELECT ROW_NUMBER() OVER (PARTITION BY a.ORG_UNIT_ID
ORDER BY a.ORG_UNIT_ID) rw, (SUM(DECODE(a.year_s, 1994,
a.month_03, 0)) over (PARTITION BY a.ORG_UNIT_ID ))ColVal,
(SUM(DECODE(a.year_s, 1994, a.month_03, 0)) over(PARTITION BY
a.ORG_UNIT_ID ))mTotal, a.ORG_UNIT_ID
from LEDGER_STAT a
where ((a.IDENTITY_CODE = 0)
or (a.IDENTITY_CODE = 1)
or (a.IDENTITY_CODE = 700)
or (a.IDENTITY_CODE = 779)))
and a.year_s IN (1994) AND(a.ORG_UNIT_ID) IN (SELECT ORG_UNIT_ID
FROM (SELECT SUM(DECODE(a.year_s, 1994, a.month_03, 0) )
src_amount , a.GL_ACCOUNT_ID, a.PRODUCT_ID, a.ORG_UNIT_ID
from LEDGER_STAT a
where ((a.IDENTITY_CODE = 0)
or (a.IDENTITY_CODE = 1)
or (a.IDENTITY_CODE = 700)
or (a.IDENTITY_CODE = 779)))
and (a.identity_code) IN (select distinct a.parent_identity_code
from FSI_M_DATA_IDENTITY_DETAIL a
where a.identity_code in (10993)
and a.as_of_date = '03/ 31/ 1994'
and a.src_drv_type = 'S' )
and a.year_s IN (1994)
GROUP BY a.GL_ACCOUNT_ID, a.PRODUCT_ID, a.ORG_UNIT_ID) ))
WHERE RW = 1
AND mTotal <> 0
AND ColVal <> 0) Drv ON (Src.ORG_UNIT_ID = Drv.ORG_UNIT_ID )
IVW_SOURCE
where IVWSRC_AMOUNT <> 0
GROUP BY ORG_UNIT_ID, GL_ACCOUNT_ID, PRODUCT_ID ) SOURCE ON

```

```
(TARGET.FINANCIAL_ELEM_ID = 457.000000
AND TARGET.ORG_UNIT_ID = SOURCE.ORG_UNIT_ID
AND TARGET.GL_ACCOUNT_ID = SOURCE.GL_ACCOUNT_ID
AND TARGET.COMMON_COA_ID = 821.000000
AND TARGET.PRODUCT_ID = SOURCE.PRODUCT_ID
AND TARGET.YEAR_S = 1994
AND TARGET.IDENTITY_CODE = 10993.000000
AND TARGET.ISO_CURRENCY_CD = 'USD'
AND TARGET.CONSolidATION_CD = 100
AND TARGET.ACCUM_TYPE_CD = 'D' ) WHEN MATCHED THEN
UPDATE
SET TARGET.MONTH_03 = TARGET.MONTH_03 + SOURCE.BALANCE_AMOUNT,
BALANCE_TYPE_CD = 101, ytd_03 = ytd_02 + SOURCE.BALANCE_AMOUNT ,
ytd_04 = ytd_03 + month_04 , ytd_05 = ytd_04 + month_05 , ytd_06
= ytd_05 + month_06 , ytd_07 = ytd_06 + month_07 , ytd_08 =
ytd_07 + month_08 , ytd_09 = ytd_08 + month_09 , ytd_10 = ytd_09
+ month_10 , ytd_11 = ytd_10 + month_11 , ytd_12 = ytd_11 +
month_12 WHEN NOT MATCHED THEN
INSERT (FINANCIAL_ELEM_ID, ORG_UNIT_ID, GL_ACCOUNT_ID,
COMMON_COA_ID, PRODUCT_ID, ACCUM_TYPE_CD, BALANCE_TYPE_CD,
CONSOLIDATION_CD, CURRENCY_TYPE_CD, ISO_CURRENCY_CD,
IDENTITY_CODE, MONTH_01, MONTH_02, MONTH_03, MONTH_04, MONTH_05,
MONTH_06, MONTH_07, MONTH_08, MONTH_09, MONTH_10, MONTH_11,
MONTH_12, YTD_01, YTD_02, YTD_03, YTD_04, YTD_05, YTD_06, YTD_07,
YTD_08, YTD_09, YTD_10, YTD_11, YTD_12, YEAR_S )
VALUES ( 457.000000, SOURCE.ORG_UNIT_ID, SOURCE.GL_ACCOUNT_ID,
821.000000, SOURCE.PRODUCT_ID, 'D' , 101 , 100 , 0 , 'USD' ,
10993.000000 , 0.0, 0.0, SOURCE.BALANCE_AMOUNT, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0, 0, SOURCE.BALANCE_AMOUNT ,
SOURCE.BALANCE_AMOUNT , SOURCE.BALANCE_AMOUNT ,
SOURCE.BALANCE_AMOUNT , SOURCE.BALANCE_AMOUNT ,
SOURCE.BALANCE_AMOUNT , SOURCE.BALANCE_AMOUNT ,
SOURCE.BALANCE_AMOUNT , SOURCE.BALANCE_AMOUNT ,
SOURCE.BALANCE_AMOUNT , 1994)
```

## Profitability Management Parallel Execution

### Overview

Profitability Management implementations frequently employ dozens, hundreds or even thousands of allocation rules. Frequently, many of these rules must be executed sequentially and in larger implementations, end-to-end execution times may exceed allowable thresholds. One way to reduce end-to-end runtimes is to execute as many processes as possible in parallel. For example, running Funds Transfer Pricing against multiple instrument tables simultaneously. You may also find that you can run some allocations processes concurrently. Other ways to reduce your end-to-end runtimes include tuning your database, employing hints on your slowest running allocations (see [SQL Hints](#)), or upgrading your hardware (more CPUs, more memory, faster I/O subsystems, and so on). Having a more powerful database server, however, will not improve your performance if hardware resources are left idle. When executing allocation rules serially, you will generally utilize only a single CPU on your database server.

Oracle Financial Services Profitability Management allows you to exploit Oracle Parallel Execution. To capture what might otherwise be idle system resources, Oracle Parallel Execution subdivides SQL statements into multiple independent units of work each of which can run in parallel.

The Oracle database supports parallel execution of SQL statements through the use of:

- The PARALLEL clause for selected tables and indexes
- Parallel hints
- Alter Session statements to enable subsequent SQL operations to run in parallel

Working with your DBA, you may choose to establish specific degrees of parallelism for selected tables and indexes utilizing the PARALLEL clause. You may also elect to employ **parallel** SQL Hints on selected allocation rules (see [SQL Hints](#)). Either of these approaches may be employed to encourage the database to run SELECT statements in parallel (Parallel Query), but the third mechanism is required if you want to encourage the database to run DML statements in parallel (UPDATE, INSERT, DELETE, MERGE, and so on). Employing Parallel Query can help, but most allocation execution time is consumed by DML queries where you are modifying the database. In order to exploit Parallel DML, the **session** from which a SQL query is issued must be specifically enabled for Parallel DML.

Each time an allocation rule is executed, a new database session is established. Each allocation starts by issuing a series of Alter Session statements based on parameters that you define in the FSI\_PARALLEL\_PARAMETERS table. This parameter table contains four parallel parameters as follows:

- Parallel Query
- Parallel DML
- Parallel Degree
- Parallel Degree Policy

There should be only one row in the FSI\_PARALLEL\_PARAMETERS table. Upon installation, this one row is seeded with the following values:

Column Name	Valid Values	Seeded Value
PROCESS_ENGINE_CD	0 for Profitability Management	0
PARALLEL_QUERY	Disable, Enable, or Force	Disable
PARALLEL_DML	Disable, Enable, or Force	Disable
PARALLEL_DEGREE	Integers between 0 and 99	Null
PARALLEL_DEGREE_POLICY	Manual, Limited, or Auto	Manual

The parameters values found in FSI\_PARALLEL\_PARAMETERS apply to all allocation rules. Currently, Oracle Parallel Execution is only supported for Profitability Management. OFSAA Funds Transfer Pricing and OFSAA Asset Liability Management employ a different technique for parallelism: OFSA Application Multiprocessing (for details on OFSA Application Multiprocessing, see the *Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration Guide*).

```
UPDATE FSI_PARALLEL_PARAMETERS SET
PARALLEL_QUERY = 'ENABLE',
PARALLEL_DML = 'FORCE',
PARALLEL_DEGREE = NULL,
PARALLEL_DEGREE_POLICY = 'AUTO';
```

The specific Alter Session statements generated by the allocation engine for the different possible parallel parameter values are discussed as follows.

## Parallelism Options

The following discusses the behavior of the Profitability Management engine given different possible configuration options of the parameters found in FSI-PARALLEL-PARAMETERS. Oracle Parallel Execution refers to a rich set of core database functionality. For a thorough discussion of Oracle Parallel Execution, including Parallel Query and Parallel DML, see the listing of documentation resources found at the end of this appendix.

### PARALLEL\_QUERY

#### **DISABLE**

When the PARALLEL\_QUERY option in the FSI\_PARALLEL\_PARAMETERS table is set to DISABLE, the engine will begin each rule by issuing the following statement:

```
ALTER SESSION DISABLE PARALLEL QUERY;
```

Setting Parallel Query to DISABLE will ensure that no SELECT statement issued by the allocation engine will run in parallel.

## **ENABLE**

When the `PARALLEL_QUERY` option in the `FSI_PARALLEL_PARAMETERS` table is set to `ENABLE`, the engine will begin each rule by issuing the following statement:

```
ALTER SESSION ENABLE PARALLEL QUERY;
```

When Parallel Query is enabled, the database may elect to execute a `SELECT` statement in parallel if it is run against an object (table or index) that has been declared or altered through the use of the `PARALLEL` clause.

Two examples of the use of the `PARALLEL` clause to alter a table follow:

```
ALTER TABLE LEDGER_STAT PARALLEL 8;
```

```
ALTER TABLE LEDGER_STAT PARALLEL (DEGREE DEFAULT);
```

In the earlier first example, the database may elect to parallelize a `SELECT` statement against the Ledger/Stat table using a Degree of Parallelism equal to 8. In the second example, the database may elect to parallelize a `SELECT` statement against the Ledger/Stat table using the **default** Degree of Parallelism as determined by the database but only if `PARALLEL_DEGREE_POLICY` is set to `Auto` (default parallelism and `PARALLEL_DEGREE_POLICY` are discussed further as follows).

The database may also elect to execute a `SELECT` statement in parallel if a parallel SQL hint is included in the `SELECT` statement. A parallel SQL hint will override the Degree of Parallelism assigned to a table or index through the `PARALLEL` clause. If the tables and indexes against which a `SELECT` statement is executed have not been defined using the `PARALLEL` clause and if the `SELECT` statement does not include a parallel hint, then the SQL statement will always run serially (in a single thread).

If the tables and indexes against which a `SELECT` statement is executed have been defined using the `PARALLEL` clause or if the `SELECT` statement includes a parallel hint, the database may elect to run in parallel if it is possible to do so. The conditions necessary for the database to choose parallelize a query are complex and beyond the scope of this discussion – for details, see the listing of documentation resources found at the end of this appendix.

## **FORCE**

When the `PARALLEL_QUERY` option in the `FSI_PARALLEL_PARAMETERS` table is set to `FORCE` and the `PARALLEL_DEGREE` parameter is set to `NULL`, the engine will begin each rule by issuing the following statement:

```
ALTER SESSION FORCE PARALLEL QUERY;
```

After this `ALTER SESSION` statement, subsequent `SELECT` statements are executed with a **default** Degree of Parallelism unless (a) the objects against which the `SELECT` statement operates have been defined as having a specific Degree of Parallelism through the use of the `PARALLEL` clause or (b) the `SELECT` statement includes a parallel hint. Again, a parallel hint will override the Degree of Parallelism specified through a `PARALLEL` clause.

The default Degree of Parallelism is determined by the database and is generally a function of the number of CPU's on your server. For details on how the Oracle database determines the default Degree of Parallelism, see *Oracle Database VLDB and Partitioning Guide*.

### **FORCE with PARALLEL\_DEGREE**

When the `PARALLEL_QUERY` option in the `FSI_PARALLEL_PARAMETERS` table is set to `FORCE` and the `PARALLEL_DEGREE` parameter is set to a non-NULL integer value (8 in this example), the engine will begin each rule by issuing the following statement:

```
ALTER SESSION FORCE PARALLEL QUERY PARALLEL 8;
```

The subsequent behavior of Parallel Query is the same as where no specific Degree of Parallelism has been specified except that the database will employ the Degree of Parallelism you have set in your `PARALLEL_DEGREE` parameter instead of using the database's default Degree of Parallelism. Your `PARALLEL_DEGREE` parameter will override any Degree of Parallelism stemming from a `PARALLEL` clause associated with a table or index, but a Degree of Parallelism specified in a parallel hint will override your `PARALLEL_DEGREE` parameter setting.

Regardless of its value, the `PARALLEL_DEGREE` parameter is ignored when the `PARALLEL_QUERY` parameter is set to either `DISABLE` or `ENABLE`.

### **PARALLEL\_DML**

#### **Limitations in Using Parallel DML**

A Parallel DML operation's lock requirements are very different from the serial DML requirements. For these and other reasons, the database imposes some restrictions on Parallel DML operations.

One such restriction is that while a single transaction can contain multiple parallel DML statements that modify different tables, after a parallel DML statement modifies a table no subsequent statement (DML or query) can access the same table again in the same transaction. For this reason, Parallel DML is disabled, regardless of your parameter settings, for any allocation rule that both debit and credit the same table. For allocation rules that both debit and credit the same table and which you want to ensure run in parallel, it is recommended that you divide your one allocation rule into two rules (one rule for the debit side and one rule for the credit side).

Parallel DML can not be set (regardless of the parameter settings) for any allocation rule where the Output tables (Credit/Debit) are the same tables as the Source/Driver tables.

For more information regarding the restrictions on Parallel DML, see *Oracle Database Data Warehousing Guide*.

### **DISABLE**

When the `PARALLEL_DML` option in the `FSI_PARALLEL_PARAMETERS` table is set to `DISABLE`, the engine will begin each rule by issuing the following statement:

```
ALTER SESSION DISABLE PARALLEL DML;
```

Setting Parallel DML to `DISABLE` will ensure that no DML statement issued by the allocation engine will ever run in parallel.

### **ENABLE**

When the `PARALLEL_DML` option in the `FSI_PARALLEL_PARAMETERS` table is set to `ENABLE`, the engine will begin each rule by issuing the following statement:

```
ALTER SESSION ENABLE PARALLEL DML;
```

When Parallel DML is enabled, the database may elect to execute a DML statement in parallel if it is run against an object (table or index) that has been declared (or altered) using the `PARALLEL` clause.

The database may also elect to execute a DML statement in parallel if a parallel SQL hint is included in the DML statement. A parallel SQL hint will override the Degree of Parallelism assigned to a table or index through the PARALLEL clause. If the tables and indexes against which a SELECT statement is executed have not been defined using the PARALLEL clause and if the SELECT statement does not include a parallel hint, then the SQL statement will always run serially (in a single thread).

### **FORCE**

When the PARALLEL\_DML option in the FSI\_PARALLEL\_PARAMETERS table is set to FORCE and the PARALLEL\_DEGREE parameter is set to NULL, the engine will begin each rule by issuing the following statement:

```
ALTER SESSION FORCE PARALLEL DML;
```

After this ALTER SESSION statement, subsequent DML statements are executed with the default Degree of Parallelism for the database unless (a) the objects against which the SELECT statement operates have been defined as having a specific Degree of Parallelism through the use of the PARALLEL clause or (b) the SELECT statement includes a parallel hint. A parallel hint will override the Degree of Parallelism specified through a PARALLEL clause. Again, for details on how the Oracle database determines the default Degree of Parallelism, see *Oracle Database VLDB and Partitioning Guide*.

### **FORCE with PARALLEL\_DEGREE**

When the PARALLEL\_DML option in the FSI\_PARALLEL\_PARAMETERS table is set to FORCE and the PARALLEL\_DEGREE parameter is set to a non-NULL integer value (4 in this example), the engine will begin each rule by issuing the following statement:

```
ALTER SESSION FORCE PARALLEL DML PARALLEL 4;
```

The subsequent behavior of Parallel DML is the same as where no specific Degree of Parallelism has been specified except that the database will employ the Degree of Parallelism you have set in your PARALLEL\_DEGREE parameter instead of using the database's default Degree of Parallelism. Your PARALLEL\_DEGREE parameter will override any Degree of Parallelism stemming from a PARALLEL clause associated with a table or index, but a Degree of Parallelism specified in a parallel hint will override your PARALLEL\_DEGREE parameter setting.

Regardless of its value, the PARALLEL\_DEGREE parameter is ignored when the PARALLEL\_QUERY parameter is set to either DISABLE or ENABLE.

### **PARALLEL\_DEGREE\_POLICY**

A new feature in Oracle 11gR2, PARALLEL\_DEGREE\_POLICY specifies whether or not automatic Degree of Parallelism, statement queuing, and in-memory parallel execution will be enabled. These 3 topics are covered in depth by the documentation resources found at the end of this appendix.

### **MANUAL**

When the PARALLEL\_DEGREE\_POLICY option in the FSI\_PARALLEL\_PARAMETERS table is set to MANUAL, the engine will begin each rule by issuing the following statement:

```
ALTER SESSION SET PARALLEL_DEGREE_POLICY = 'MANUAL';
```

The subsequent behavior of both Parallel Query & Parallel DML is the same as it would have been prior to Oracle 11gR2. SQL statements will only be processed in parallel for objects declared with a PARALLEL clause or if a

parallel SQL hint is included. Additionally, the Profitability Management PARALLEL\_QUERY parameter must be set to either ENABLE or FORCE (for Parallel Query) and the PARALLEL\_DML parameter must be set to either ENABLE or FORE (for Parallel DML).

### **LIMITED**

When the PARALLEL\_DEGREE\_POLICY option in the FSI\_PARALLEL\_PARAMETERS table is set to LIMITED, the engine will begin each rule by issuing the following statement:

```
ALTER SESSION SET PARALLEL_DEGREE_POLICY = 'LIMITED';
```

Under the LIMITED policy, automatic Degree of Parallelism is enabled for some statements but statement queuing and in-memory Parallel Execution are disabled. Automatic Degree of Parallelism is only applied to those statements that access tables or indexes decorated explicitly with the DEFAULT degree of parallelism using the PARALLEL clause. Statements executed against tables or indexes defined using an explicit Degree of Parallelism through the PARALLEL clause or statements including a parallel SQL hint may also run in parallel.

### **AUTO**

When the PARALLEL\_DEGREE\_POLICY option in the FSI\_PARALLEL\_PARAMETERS table is set to AUTO, the engine will begin each rule by issuing the following statement:

```
ALTER SESSION SET PARALLEL_DEGREE_POLICY = 'AUTO';
```

Under the AUTO policy, the automatic Degree of Parallelism, statement queuing, and in-memory parallel execution features are all enabled. Under this policy, the Oracle database automatically decides if a statement should execute in parallel or not and what Degree of Parallelism it should use. The database also determines if the statement can be executed immediately or if it should be queued until more system resources are available. Finally, the database decides if the statement can take advantage of the aggregated cluster memory or not. For more details, see the documentation resources as follows.

## **Parallel Execution Documentation Resources**

Oracle Database Parallel Execution Fundamentals: An Oracle White Paper – October 2010

<http://www.oracle.com/technetwork/articles/datawarehouse/twp-parallel-execution-fundamentals-133639.pdf>

Oracle® Database Data Warehousing Guide, 11g Release 1

Part Number B28313-02

[http://download.oracle.com/docs/cd/B28359\\_01/server.111/b28313.pdf](http://download.oracle.com/docs/cd/B28359_01/server.111/b28313.pdf)

See in particular:

- Enabling Parallel DML
- Space Considerations for Parallel DML
- Locks for Parallel DML
- Restrictions on Parallel DML

Oracle® Database New Features Guide, 11g Release 2

Part Number E10881-01

[https://docs.oracle.com/cd/E11882\\_01/server.112/e41360/toc.htm](https://docs.oracle.com/cd/E11882_01/server.112/e41360/toc.htm)



See in particular section In-Memory Parallel Execution

Oracle® Database VLDB and Partitioning Guide, 11g Release 2 (11.2)

Part Number E16541-08

[https://docs.oracle.com/cd/E18283\\_01/server.112/e16541.pdf](https://docs.oracle.com/cd/E18283_01/server.112/e16541.pdf)

See in particular: Chapter 8: Using Parallel Execution

Oracle® Database Performance Tuning Guide, 11g Release 2 (11.2)

Part Number E16638-04

[https://docs.oracle.com/cd/E28271\\_01/server.1111/e16638.pdf](https://docs.oracle.com/cd/E28271_01/server.1111/e16638.pdf)



## Seeded Batches

Three seeded batches are introduced starting from Release 8.0.6.0.0 and the details of these seeded batches are as follows:

- **<INFODOM>\_PFT\_MANAGEMENTLEDGER\_LOAD\_UNDO:** This Batch can be used to UNDO the data loaded into LEDGER CLASS tables. Parameters passed in the batch are given below:

```
PFTUNDO.sh,##INFODOM##_<Batch Run Id> <Identity Code>-<As of Date> L
```

where:

INFODOM - Information Domain name

BATCH\_RUN\_ID - Any unique identifier to identify the batch execution

Identity Code - Identity code generated by ledger load which needs to be removed

As of Date - MIS DATE of Ledger Load in MM/DD/YYYY format.

- **<INFODOM>\_T2T\_ALL\_INSTRUMENT\_TABLES:** This batch contains the T2T's for the below Instrument Tables:
  - T2T\_ANNUITY\_CONTRACTS
  - T2T\_BORROWINGS
  - T2T\_CASA
  - T2T\_CREDIT\_CARDS
  - T2T\_GUARANTEES
  - T2T\_INVESTMENTS
  - T2T\_LEASES
  - T2T\_LOAN\_CONTRACTS
  - T2T\_MERCHANT\_CARDS
  - T2T\_MM\_CONTRACTS
  - T2T\_MORTGAGES
  - T2T\_MUTUAL\_FUNDS
  - T2T\_OTHER\_SERVICES
  - T2T\_RETIREMENT\_ACCOUNTS
  - T2T\_TERM\_DEPOSITS
  - T2T\_TRUSTS

This batch can be edited to add or remove other Instrument Table T2T's.

- **<INFODOM>\_T2T\_ALL\_TXN\_SUMMARY**: This batch contains the T2Ts for the below Transaction Summary Tables:
  - T2T\_ANNUITY\_TXNS\_SUMMARY
  - T2T\_BORROWINGS\_TXNS\_SUMMARY
  - T2T\_CASA\_TXNS\_SUMMARY
  - T2T\_CREDIT\_CARDS\_TXNS\_SUMMARY
  - T2T\_CREDIT\_LINES\_TXNS\_SUMMARY
  - T2T\_GUARANTEES\_TXNS\_SUMMARY
  - T2T\_INVESTMENTS\_TXNS\_SUMMARY
  - T2T\_LEASES\_TXNS\_SUMMARY
  - T2T\_LOAN\_CONTRACTS\_TXNS\_SUMMARY
  - T2T\_MERCHANT\_CARDS\_TXNS\_SUMMARY
  - T2T\_MM\_CONTRACTS\_TXNS\_SUMMARY
  - T2T\_MORTGAGES\_TXNS\_SUMMARY
  - T2T\_MUTUAL\_FUNDS\_TXNS\_SUMMARY
  - T2T\_OTHER\_SERVICES\_TXNS\_SUMMARY
  - T2T\_RETIREMENT\_ACCTS\_TXNS\_SUMMARY
  - T2T\_TERM\_DEPOSITS\_TXNS\_SUMMARY
  - T2T\_TRUSTS\_TXNS\_SUMMARY

This batch can be edited to add or remove other Transaction Summary Table T2Ts.

---

**APPENDIX K**

# Glossary

**A****Account table**

Stores a detailed set of transaction-level data attributes pertaining to instruments. For example, origination date, outstanding balance, contracted rate, and maturity date. An account table is also known as an instrument table.

**Aggregation**

To summarize detailed instruments to a table in the database which can be used for faster processing in PFT.

**Aggregate Instrument Table**

To avoid the writing of multiple rules that was one rule per table, where the rule logic is essentially identical except for the specification of the Instrument Table.

**Allocation**

Profitability Management engine allocates between two cost centers using the allocation rule.

**Allocation Model**

An Allocation Model consists of a list of individual allocation rules that can be executed as a single unit.

**Allocation Execution History**

This shows the list of executed allocation rules.

**Answers**

Answers is the OBIEE (Oracle Business Intelligence Enterprise Edition), ad-hoc reporting tool.

**As of Date**

Date at which the data is current.

**Assignment date**

Indicates the relevant date for which the associated yield curve has to be referenced. This parameter is used for defining the Spread from Interest Rate Code and Redemption Curve transfer pricing methods. You can choose origination date, last repricing date, or the last day of the associated calendar period as the assignment date.

**Attributed Dimension**

A dimension whose members can have other properties or qualifiers known as *dimension attributes*.

**B****Basis Points**

1/100th of a percent (abbreviated as “bps”)

**Bottom-Up Allocation Methods**

Bottom-up allocation methods are used to construct discrete detailed expenses based on volumes and unit cost. This method is used both in the Management Ledger level profitability and Account Level profitability.

**C**

**Calculation Mode**

Any of the two ways, standard or remaining term, of calculating transfer rates and options costs supported by Oracle Funds Transfer Pricing.

**Charge/Credit Accrual Basis**

The basis on which charge/credit accrues for a business unit and the offsetting treasury unit, similar to the accrual basis used in calculation of interest.

**Circular Allocation Models**

Circular Allocation Model is similar to a Standard Allocation Model, but consists of two lists of allocation rules: a list of "circular" rules and a list of "sweep" rules.

**Constant Allocation Type**

A Constant Allocation rule creates a simple balanced transaction consisting of one debit and one credit. You may optionally specify either one debit or one credit (at a minimum, you must supply at least one debit or one credit).

**Cost Centers**

Cost Centers are subdivided into Overhead Centers, Indirect Support Centers, and Direct Support Centers.

**Credit Risk**

The risk that a loan holder will be unable to repay any portion of the loan.

Customer Account Level Profitability

It is a customer counterparty position level of Profitability Management.

D

**Data Filter Rule**

Data Filters are used to define which data should go into a processing run.

**Data set**

A dimension used for segregating data into different sets according to its use or its source, for example, to separate actuals data, budget data, and encumbrances data. Other uses include separating test data from production data and creating separate data sets for what-if analysis.

**Data Input Helper**

A pop-up dialogue that facilitates the definition of a series of input values (dates or numbers) by allowing several methods for defining structured patterns in the data.

**DDA**

Demand Deposit Account. An example of a DDA is a checking account.

**De-annualize**

To compute the monthly equivalent rate of an annual rate.

**Dimension**

A structure that can be used to categorize business data. A dimension contains members. A dimension can be hierarchical in that you can organize the members into one or more hierarchies, or nonhierarchical.

Dimension Attribute

A property or qualifier that further describes a dimension member. An attribute can be anything such as a date, a number, or a character string. For example, the Geography dimension can have an attribute Population that designates how many people live in that area. Each member of the Geography dimension therefore has an associated population.

**Dimension Based Rule**

A business rule whose definition varies depending on the dimensional values of the data to which it is applied.

**Dimension Identifier**

A numeric string that uniquely identifies each member of a dimension. Dimension identifiers are non-translatable, as they are the same regardless of the language context. Each dimension has its own unique set of columns in the Analytical Applications Infrastructure staging tables that serve as the dimension identifier for that dimension.

**Dimension Member**

The values used to populate dimension columns in account, transaction, or statistical tables. Such values represent the individual organization units, distribution channels, products, and so on of which each dimension is comprised. In a hierarchy, both lowest level and node level values are considered to be dimension members.

**Driver**

A variable that influences the prepayment behavior of an instrument. You can build a custom prepayment model using up to three prepayment drivers. Each driver maps to an attribute of the underlying transaction (age or term, or rate) so the cash flow engine can apply a different prepayment rate based on the specific characteristics of the record.

**Duration**

Duration is the rate of market value change with respect to discount rate changes. It is a measure of market value sensitivity: the lower the value, the less sensitive the market value to changes in interest rates.

**Dynamic Driver Allocation Type**

Dynamic Driver allocation rules aggregate or distribute balances using dynamic data (business resident driver data) such as headcount, square footage, or instrument-level balances.

**E****Entered Currency**

The currency in which business transactions take place. Entered currency might be different from functional Currency. See also: functional currency

**Embedded Rate Risk Profit**

Portion of funding center rate risk result attributed to prior rate bets.

**F****Fact Table**

A table that contains data uniquely differentiated by dimension columns.

**Field Allocation Type**

A Field type allocation is typically used to multiply two columns within a single row in an instrument table update Allocation rule.

**Filter**

A business object that filters the source data that is used as input to a process.

**Funding Center**

Area in a financial institution that receives the transfer pricing charge and credit for funds.

**Functional Currency**

The currency in which an organization keeps its books of accounts. Functional currency is associated with a particular ledger.

**Functional Reporting Lines**

In building Profitability Management view, allocations are constructed to align data into functional reporting lines.

**Funds Transfer Pricing**

A method for valuing all sources and uses of funds for a balance sheet.

**General Ledger**

The main data source that defines an institution's financial reality. The General Ledger reflects all accounting entries.

**Grid**

A logical grouping of cells often surrounded by scroll bars.

**H****Hierarchy**

A structure of dimension members organized by parent-child relationships.

**Historical Term**

The period preceding the assignment date over which the average of daily interest rates from a yield curve is taken. This parameter is used in Moving Averages transfer pricing method.

**I****Instrument**

Synonymous with an account record or an individual contract.

**Instrument Table**

A type of table contained in the OFSAA database used to store account level information.

**Instrument Records**

Rows in the OFSAA database that carry transaction account level information (Example: deposit account by deposit account)

**Interest Rate Code**

User defined code to reference a yield curve or single rate index for historical analysis, transfer pricing, and interest rate forecasting purposes.

**L****Leaf Allocation Type**

Leaf Allocation Type is used to perform an operation between two sets of rows that differ in a single dimension.

**Leaf Fields**



OFSAA database fields used to define hierarchical segmentations of data. They also draw a relationship between the instrument data and the General Ledger data in Ledger/Stat. Also known as key dimensions.

**Leaf Values**

Specific numeric values that make up the Leaf Fields. These are dimension members that can never be used as parent nodes on a hierarchy. The fact data should always be populated with dimension member - leaf values only.

**Ledger Migration**

The process for generating charges or credits, for funds provided or used, for migration to the Management Ledger table based on the transfer rates, adjustments or option costs, obtained from transfer pricing and option cost calculations or propagation processing.

**Ledger Stat**

Table in the OFSAA database that stores all General Ledger as well as statistical information for current and historical periods.

**Level**

A property of hierarchical dimensions that designates a category of like members. For example, in the Geography dimension there might be a level named City and a level named State. Geography members such as Tulsa and Dallas belong in the City level, while Geography members such as Texas and Oklahoma belongs in the State level. The designation of level is the same across all hierarchies within a dimension. In other words, Texas is always a state in all Geography hierarchies.

**Log in**

To access the programs and database of any OFSAA application by providing a valid user name and password.

**Long Run Rate**

One of the user-input parameters of the Vasicek (discrete-time) term structure model; represents the equilibrium value of the one-month annually compounded rate.

**Lookup Driver Table Allocation Type**

Lookup Driver Table allocation rules support multi-factor allocations at the Instrument level.

**M****Management Ledger Table**

Also known as LEDGER\_STAT, the most central fact table in the OFS Analytical Applications Infrastructure. It contains ledger and some statistical data and highly aggregated information, such as cash and other assets, and equity. This table supports Oracle Financial Services analytical applications.

**Monthly Rate**

Yield on a loan contracted at the beginning of a month for a period of one month assuming a continuous compounding basis; the monthly rate is a function of time and scenario; the Rate Generator also computes rates for a different term than a month.

**Multidimensional Profitability**

Multidimensional Profitability supports the profitability analytics by time, profit center, product, and functional reporting lines. Each dimension is completely independent, that is, same reporting lines can be generated for all the Profit Centers, products, and each month.

## N

### **Node**

A dimension value located anywhere in a hierarchy.

### **Node Level Assumption**

An assumption assigned to a dimension value at a level higher than a leaf level. A node level assumption is typically associated with a business rule that uses a hierarchical dimension.

## O

### **Oracle Asset Liability Management**

ALM is an Asset/Liability management simulation tool that generates daily cash flows for each individual transaction record and allows users to model their balance sheet using both deterministic and stochastic methods.

### **Oracle Business Intelligence Enterprise Edition**

OBIEE is Oracle's market leading business intelligence product. OFSAA applications are fully integrated with OBIEE through the OFSAA BI applications, including Oracle Financial Services Asset Liability Management Analytics and Oracle Financial Services Profitability Analytics.

### **Oracle Financial Services Balance Sheet Planning**

Balance Sheet Planning performs distributed budgeting of all balance sheet and income statement accounts.

#### **Oracle Financial Services Funds Transfer Pricing**

Funds Transfer Pricing performs Matched Rate Transfer Pricing for an entire balance sheet.

### **Oracle Financial Services Pricing Management – Transfer Pricing Component**

Pricing Management allows users to call the Transfer Pricing engine in real time, providing the line bankers with on-demand information to support deal pricing.

### **Oracle Financial Services Profitability Management**

Profitability Management assists customers in developing multiple levels of profitability (Example: organizational, product, customer, account profitability).

### **Oracle Financial Services Analytical Applications Infrastructure**

OFSAAI is a set of powerful administrative tools that provide management of the OFSAA environment and access to the database by OFS application users. The Infrastructure is a required component for all OFSAA applications with the exception of Balance Sheet Planning.

### **Oracle Financial Services Advanced Analytical Applications Infrastructure**

The advanced infrastructure provides the modeling framework and stress testing framework as additional complimentary components to supplement to core analytic applications.

### **Organizational Profitability**

Organizational Profitability allocates all income and expense from Cost Centers to Profit Centers to build the profit centers to get the final profit and loss from the Profit Centers perspective.

### **Operating Cost**

Non-interest related cost of running a business.

### **Option Costs**

---

Costs assigned to measure the value of any customer option on an instrument. (Example: prepayments on mortgage loans). These costs are typically added to the base transfer rate to compute an all-in funding rate.

## **P**

### **Portfolio Fields**

Fields in the OFSAA database that are common to multiple instrument tables. The OFSAA Administrator determines the portfolio fields.

### **Product Profitability**

The goal of Product Profitability is to enrich the data by giving product alignment along the profit and loss statement.

### **Profit Centers**

Profit Centers are mainly branches, loan centers, and treasury centers.

### **Process Data**

Data required to produce results.

### **Processing Table**

Instrument table available for, or included in, a Transfer Pricing Process rule run.

### **Product Dimension**

A Product dimension on which Oracle Funds Transfer Pricing product level assumptions are based. The Product dimension should be populated with your product chart of account at a level of detail appropriate for assigning transfer pricing assumptions to your data.

## **R**

### **Risk Adjusted Return on Capital (RAROC)**

RAROC is a risk-based profitability measurement framework for analyzing risk-adjusted financial performance and providing a consistent view of profitability across businesses.

### **Reconciliation**

The process of comparing information from one data source to another.

### **Record**

Usually a single account or transaction, or aggregation of accounts, stored in the database (also called a row).

### **Reference Currency**

Currency in which the instrument data is expressed and designated by the currency code on the record. Within the application, the reference currency must be selected to indicate assumptions that will be applied to corresponding currency designations contained in the account data. Reference currency is also the currency with which an Interest Rate Code is associated. When you create an Interest Rate Code, you select the reference currency.

### **Rule**

A grouping of assumptions, also known as a business rule.

## **S**

### **Simple Dimension**

A dimension that does not have hierarchies or attributes. A simple dimension is just a list of members.

**Single Rate**

A Standard Allocation Model consists of a list of individual allocation rules that run sequentially and that may be executed as a single unit of work.

An interest rate code with only one point defined (Examples: prime rate and 11th District Cost of Funds Index).

**SQL**

Structured Query Language. A direct method of accessing the raw OFSAA database.

**Standard Allocation Models**

A Standard Allocation Model consists of a list of individual allocation rules that run sequentially and that may be executed as a single unit of work.

**Static Driver Allocation Type**

It is used to perform simple factor calculations against a set of source balances.

**Static Driver Table Allocation Type**

Static Driver Table allocation rules offer functionality similar to Dynamic Driver allocation rules but use driver data that is stored in a Profitability Management rule type called Static Table Driver.

**T****Transfer Pricing Rule**

An OFSAA Rule used to specify the method for transfer pricing each balance sheet account.

**Top-Down Allocation Methods**

Top-down allocation methods are used both in Management Ledger level profitability and Account Level profitability

**Transaction summary**

Transaction Summary Data contains information pertaining to Instruments. Generally, Transaction Summary data are the child nodes of Instruments. For example, number of statements received, number of calls received in the call center, number of checks cleared, and so on.

**V****Volatility**

One of the user-input term parameters of all (discrete-time) term structure models; represents a standard deviation of the one-month annually compounded rate.

**Y****Yield Curve**

Term structure of annually compounded zero-coupon bond yield, as recorded in Rate Management > Historical Rates.