Oracle Insurance Allocation Manager for Enterprise Profitability

User Guide

Release 8.1.0.0.0

May 2021

F20120-01

ORACLE
Financial Services

ORACLE

Oracle Insurance Allocation Manager for Enterprise Profitability User Guide

Document Control

Version Number	Revision Date	Change Log
<<TableText>>		

Table of Contents

1 Preface

Topics:

- Foreword
- Intended Audience
- Access to Oracle Support
- Related Documents
- Additional Documents to Read
- Conventions
- Abbreviations

1.1 Foreword

This user guide documents OFS Profitability Management for all versions of release 8.1. Some functional improvements have been introduced in various service packs and point releases within release 8.1.

This section documents the levels at which various functional enhancements to the Oracle Financial Services Profitability Management application were first introduced.

1.2 Intended Audience

Welcome to Release 8 of the Oracle Financial Services Funds Transfer Pricing User Guide.

The Oracle Financial Services Cash Flow Engine Reference Guide provides useful guidance and assistance to the following personas:

- Technical end users
- Functional end users
- Data Administrators
- Consultants
- Systems Analysts
- System Administrators
- Other MIS professionals

See the Related Information Sources for more Oracle Applications product information.

1.3 Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For more 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.

1.4 Related Documents

We strive to keep this document and all other related documents updated regularly. Visit the OHC Documentation Library to download the latest version available. The following are the list of related documents:

- Oracle Insurance Allocation Manager for Enterprise Profitability Pack Installation and Configuration Guide Release 8.1.1.0.0
- Oracle Insurance Allocation Manager for Enterprise Profitability Security Guide Release 8.1.x
- Oracle Financial Services Profitability Management Application Cloning Guide Release 8.1.x

1.5 Additional Documents to Read

Oracle Financial Services Profitability Management Pack is built on the Oracle Financial Services Advanced Analytical Applications Infrastructure (OFS AAI). See the following OFS AAI documents as no separate documents are required at the pack or application level for Oracle Financial Services Profitability Management Applications Pack:

- OFS Analytical Applications Infrastructure (OFS AAAI) Application Pack Installation and Configuration Guide Release 8.1.1.0.0
- OFS Analytical Applications Infrastructure Administration Guide Release 8.1.x
- OFS Analytical Applications Infrastructure Cloning Reference Guide Release 8.1.0.0.0
- OFS Analytical Applications Infrastructure Security Guide Release 8.1.0.0.0
- OFS Analytical Applications Infrastructure User Guide Release 8.1.1.0.0

You can access the common document from the OHC Documentation Library:

- OFS Analytical Applications 8.1.1.0.0 Technology Matrix
- OFS Data Model Utilities Guide
- OFS Cash Flow Engine Reference Guide
- OFS Asset Liability Management User Guide

1.6 Conventions

The following text conventions are used in this document:

Table 1: Document Conventions

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, file names, text that appears on the screen, or text that you enter.
Hyperlink	Hyperlink type indicates the links to external websites and internal document links.

1.7 Abbreviations

The following table lists the abbreviations used in this document:

Table 2: Abbreviations

Abbreviation	Meaning
AIX	Advanced Interactive eXecutive
BDP	Big Data Processing
DBA	Database Administrator
DDL	Data Definition Language
DEFQ	Data Entry Forms and Queries
DML	Data Manipulation Language
EAR	Enterprise Archive
EJB	Enterprise JavaBean
ERM	Enterprise Resource Management
FTP	File Transfer Protocol
HDFS	Hadoop Distributed File System
HTTPS	Hypertext Transfer Protocol Secure
J2C	J2EE Connector
J2EE	Java 2 Enterprise Edition
JCE	Java Cryptography Extension
JDBC	Java Database Connectivity
JDK	Java Development Kit
JNDI	Java Naming and Directory Interface
JRE	Java Runtime Environment
JVM	Java Virtual Machine
LDAP	Lightweight Directory Access Protocol
LHS	Left Hand Side

Abbreviation	Meaning
MFA	Multi-Factor Authentication
MOS	My Oracle Support
OFSAA	Oracle Financial Services Analytical Applications
OFSAAI	Oracle Financial Services Analytical Application Infrastructure
OFSAAAI	Oracle Financial Services Advanced Analytical Applications Infrastructure Application Pack
OHC	Oracle Help Center
OLAP	On-Line Analytical Processing
OLH	Oracle Loader for Hadoop
ORAAH	Oracle R Advanced Analytics for Hadoop
OS	Operating System
RAM	Random Access Memory
RDBMS	Relational Database Management System
RHEL	Red Hat Enterprise Linux
SFTP	Secure File Transfer Protocol
SID	System Identifier
SSL	Secure Sockets Layer
TNS	Transparent Network Substrate
URL	Uniform Resource Locator
VM	Virtual Machine
WAR	Web Archive
XML	Extensible Markup Language

2 Oracle Insurance Allocation Manager for Enterprise Profitability

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.

2.1 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.

2.2 Oracle Insurance Allocation Manager for Enterprise Profitability

Oracle Insurance Allocation Manager for Enterprise Profitability (OFS PFT INS) 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 the profitability, senior management must give them the power to identify the products that are profitable in each market segment, for each customer.

OFS PFT INS 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, OFS PFT INS delivers complete, accurate, and inclusive profiles of profitability.

2.3 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 OFS Analytical Applications Infrastructure User Guide.

3 OFSAAI Overview

Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) is a general-purpose Analytics Applications infrastructure that provides the tooling platform necessary to rapidly configure and develop analytic applications for the financial services domain. It is built with Open-Systems Compliant architecture providing interfaces to support business definitions at various levels of granularity.

Applications are built using OFSAAI by assembling business definitions or business metadata starting from data-model to lower grain objects like Dimensions, Metrics, Security Maps, and User Profile to higher-order objects like Rules, Models, and Analytic Query Templates which are assembled using the lower grain ones. In addition to application definition tools, it provides the entire gamut of services required for Application Management including Security Service, Workflow Service, Metadata Management, Operations, Life-cycle Management, public API's and Web Services that are exposed to extend and enrich the tooling capabilities within the applications.

Oracle Financial Services Analytical Applications Infrastructure is the complete end-to-end Business Intelligence solution that is easily accessible via your desktop. A single interface lets you tap your company's vast store of operational data to track and respond to business trends. It also facilitates the analysis of the processed data. Using OFSAAI you can query and analyze data that is complete, correct, and consistently stored in a single place. It has the prowess to filter data that you are viewing and using for analysis.

It allows you to personalize information access to the users based on their role within the organization. It also provides a complete view of your enterprise along with the following benefits:

- Track enterprise performance across information data stores.
- Use one interface to access all enterprise databases.
- Create consistent business dimensions and measures across business applications.
- Automate the creation of coordinated data marts.
- Use your business language to get fast and accurate answers from all your databases.
- Deploy an open XML and web-based solution against all major relational or multi-dimensional databases on Microsoft Windows and UNIX servers.

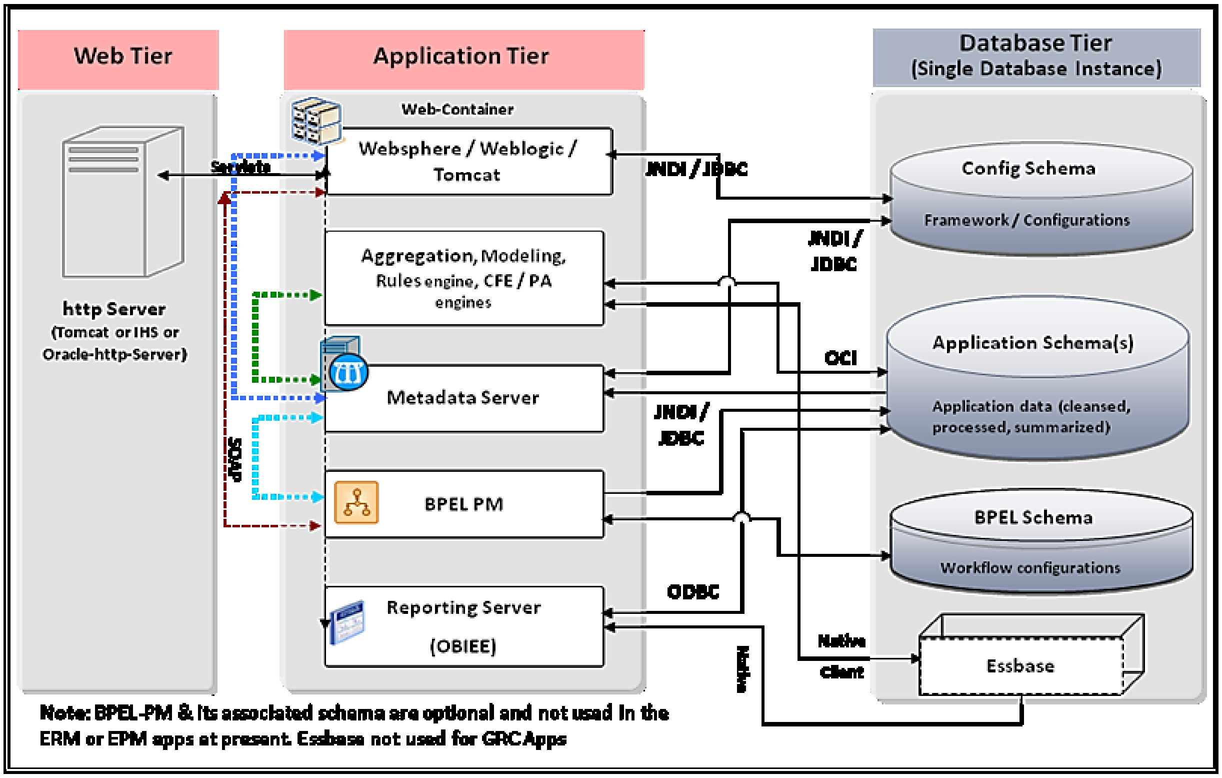
This chapter provides an overview of Infrastructure, its components, and explains how these components are organized in the Splash window with the user login process.

3.1 Components of OFSAAI

The OFSAA Infrastructure consists of the following components/modules that are used to deploy an analytical solution.

All components are encapsulated within a common Security and Operational framework as shown in the following figure.

Figure 1: Security and Operational framework



Infrastructure also supports many business analytical solutions (s) like Operational Risk, PFT, and Basel, which are licensed separately to the organization. This manual provides an overview of only the technological components.

For a detailed overview of OFSAAI modules, see Modules in OFSAAI section.

3.2 Accessing OFSAA Applications

OFSAA can be accessed through your web browser as soon as the System Administrator (SA) installs and configures Oracle Financial Services Analytical Applications.

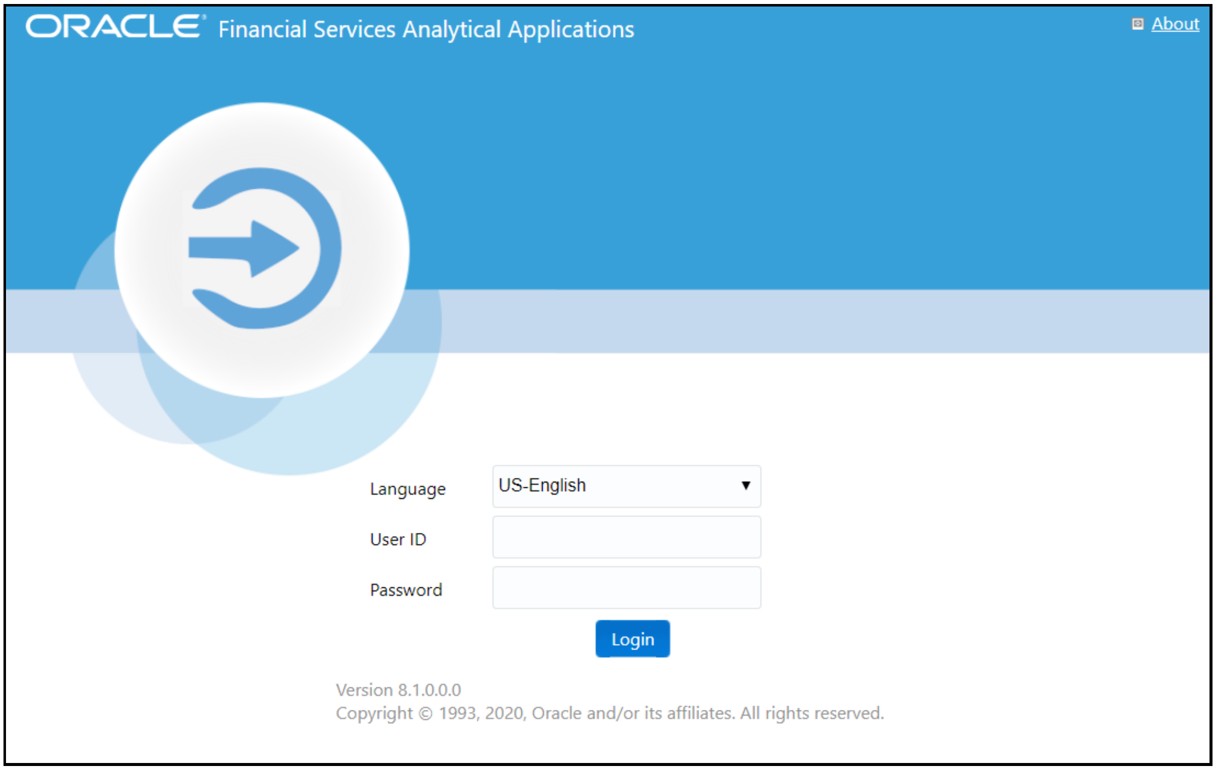
The SA will provide you with a link through which you can access Oracle Financial Services Analytical Applications. You can access the login window through your web-browser using the URL http(s): <IP Address of the Web Server > :<servlet port>/<context name>/login.jsp.

You can also log in to the application with the host name instead of the IP address.

3.3 OFSAA Login Page

On entering the URL (<IP Address/hostname of the Web Server>:<servlet port>/<context name>/login.jsp) in your browser window, the *OFSAA Login Page* is displayed:

Figure 2: OFSSA login Page



You can select the required language from the **Language** drop-down list. The language options displayed in the drop-down list are based on the language packs installed for the OFSAA infrastructure. Based on the selected Language, the appropriate language login window is displayed.

Enter the **User ID** and **Password** provided by the System Administrator and click **Login**. You will be prompted to change your password on your first login. For details on how to change a password, see the Changing Password section.

In case the OFSAA setup has been configured for OFSAA native Security Management System (SMS) Authentication, the password to be entered will be as per the password restrictions set in the OFSAA SMS repository.

3.3.1 Log in as System Administrator

Post-installation, the first login into Infrastructure is possible only for a System Administrator through user-id "sysadmn". This ID is created at the time of installation with the default password as "password0".

Enter User ID as "sysadmn" and password as "password0". Click **Login**.

3.3.2 Log in as System Authorizer

System Authorizer ID is also created at the time of installation with the default password "password0". This ID is required to authorize the users created by the system administrator.

Enter login id as "**sysauth**" and password as "password0". Click **Login**.

3.3.3 Log in as Business User

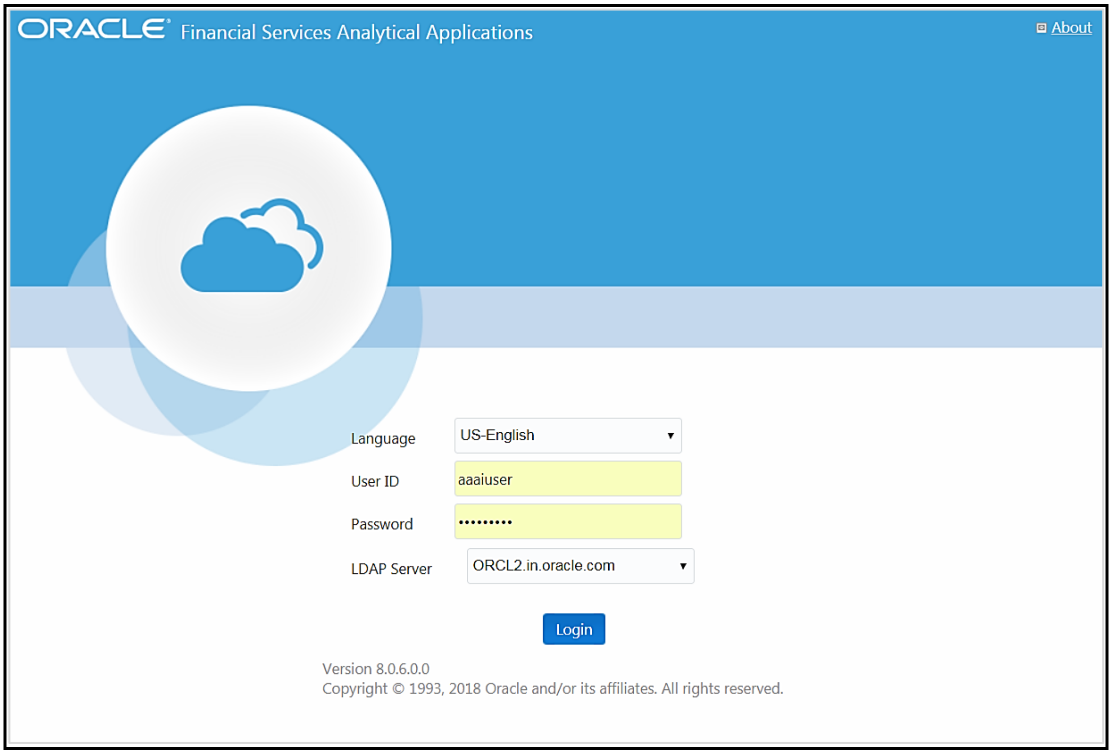
The Business users will be created by System Administrator and will be authorized by the System Authorizer.

Enter User ID and Password provided by the System Administrator and click **Login**.

3.3.3.1.1 OFSAA Login if LDAP Servers are configured

If the OFSAA setup has been configured for LDAP Authentication, the login page is displayed as shown:

Figure 3: OFSSA Login Page



1. Enter your **User ID** and **Password (as in LDAP store)** in the respective fields.
2. Select the appropriate **LDAP Server** from the drop-down list, against which you want to get authenticated. This is optional. If you do not select any server, you will be authenticated against the appropriate LDAP server.

NOTE	For SYSADMN/ SYSAUTH/ GUEST users, no need to select any LDAP server as they are always authenticated against the SMS store. Additionally, in case a specific user has been marked as “SMS Auth Only” in the *User Maintenance* window even though the OFSAA instance is configured for LDAP authentication, then that user will also be authenticated against SMS store instead of LDAP store. The user has to enter the password as per the SMS store.

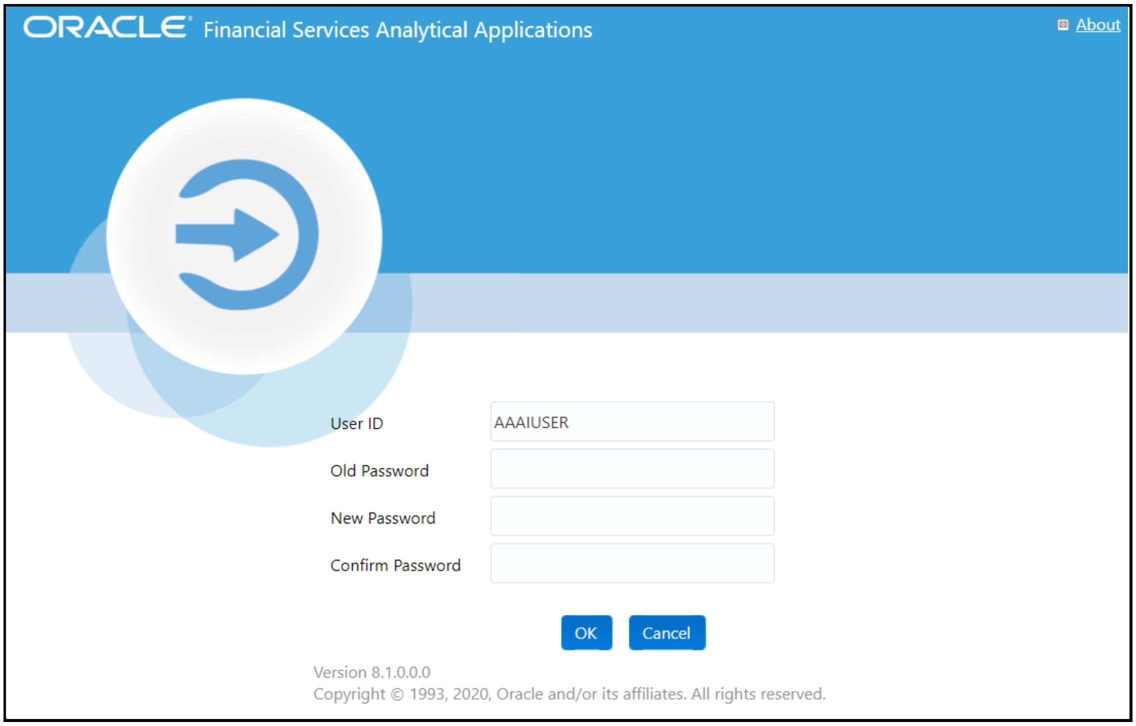
3.4 Changing Password

You can choose to change your password any time by clicking your username appearing on the right top corner and selecting **Change Password**.

Note that this option is available:

- If **SMS Authentication & Authorization is configured as Authentication Type** from the *Configuration* window**.**
- If **LDAP Authentication & SMS Authorization** is **configured as Authentication Type** from the Configuration window and the **SMS Auth Only** checkbox is selected for the user in the *User Maintenance* window**.**
- If **SSO Authentication & SMS Authorization** is **configured as Authentication Type** from the *Configuration* window and the **SMS Auth Only** checkbox is selected for the user in the *User Maintenance* window**.**

Figure 4: OFSSA Change Password window



In the *Change Password* window, enter a new password, confirm it, and click **OK** to view the *OFSAA Login* window. See the following guidelines for Password Creation:

- Passwords are displayed as asterisks (stars) while you enter. This is to ensure that the password is not revealed to other users.
- Ensure that the entered password is at least six characters long.
- The password must be alphanumeric with a combination of numbers and characters.
- The password should not contain spaces.
- Passwords are case sensitive and ensure that the Caps Lock is not turned ON.

- By default, the currently used password is checked for validity if password history is not set.
- The new password should be different from previously used passwords based on the password history, which can be configured. For more information, see the Configuration section in the System Configuration chapter.

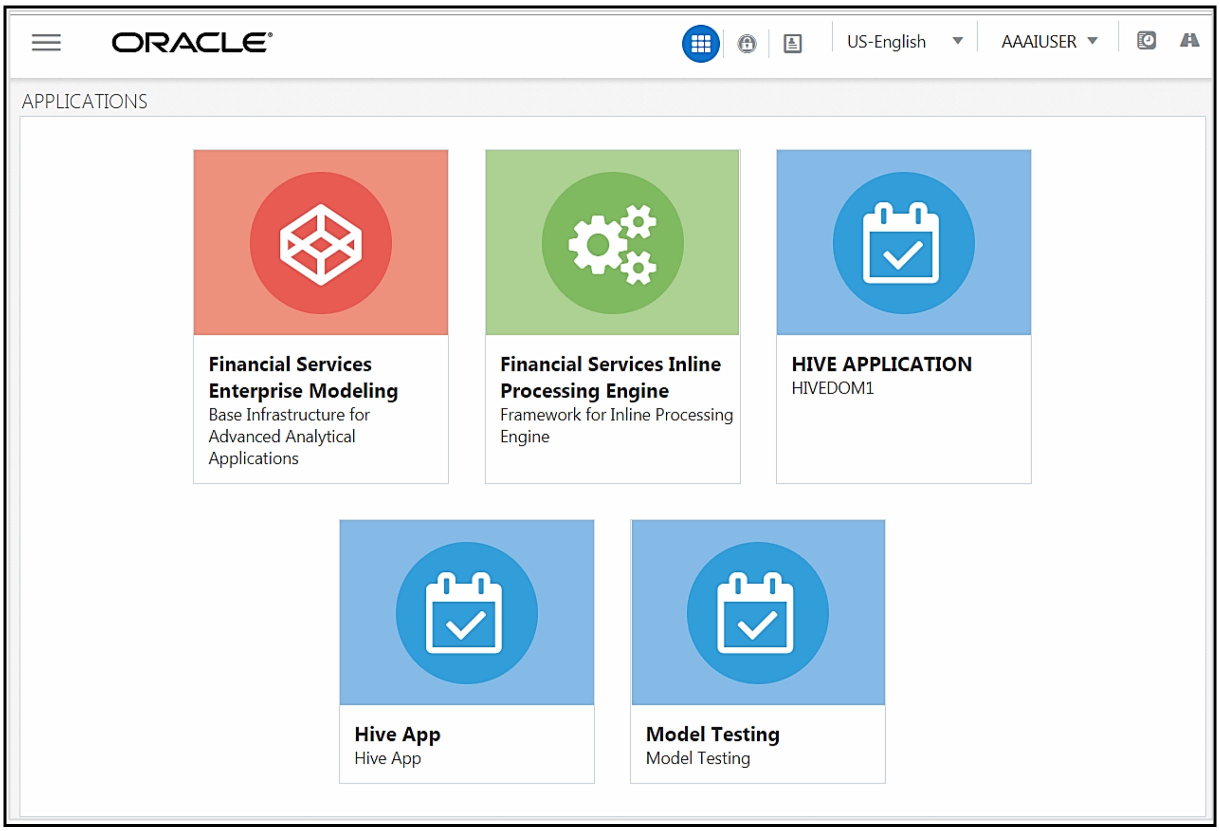
If you encounter any of the following problems, contact the System Administrator:

- Your user ID and password are not recognized.
- Your user ID is locked after three consecutive unsuccessful attempts.
- Your user ID has been disabled.
- The guest user cannot change the password.

3.5 OFSAA Landing Page

On successful login, the *OFSAA Landing* Page is displayed.

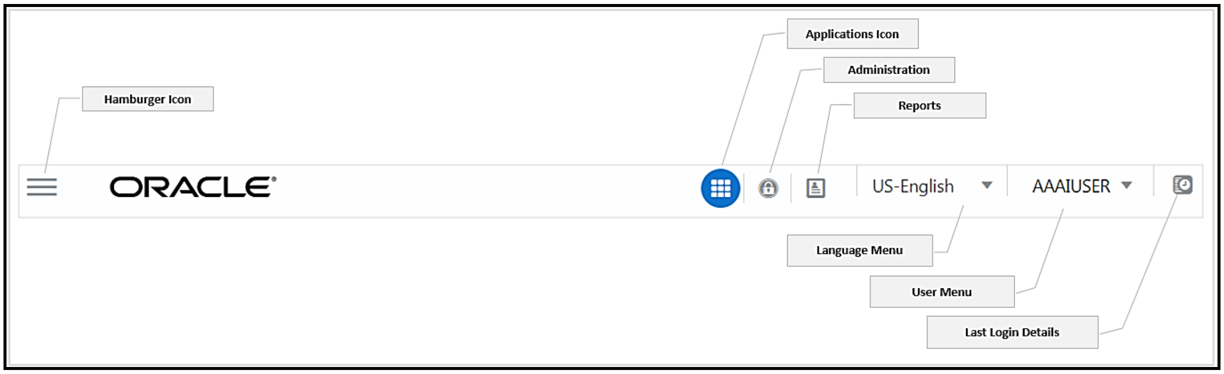
Figure 5: *OFSAA Landing Page*



OFSAA Landing screen shows the available Applications as tiles, for which a user has access. Clicking the respective Application tile launches that particular Application. You can change the landing page based on your preference. For more information, see the Preferences section.

3.5.1 Header

Figure 6: OFSSA Header



Hamburger/Navigation Menu Icon- This icon is used to trigger the Application Navigation Drawer.

Application Icon- This icon is used to show the available Applications installed in your environment at any time.

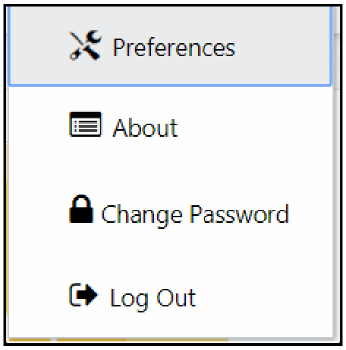
Administration Icon- This icon is used to go to the *Administration* window. The *Administration* window displays modules like System Configuration, Identity Management, Database Details, manage OFSAA Product Licenses, Create New Application, Information Domain, Translation Tools, and process Modelling Framework as Tiles.

Reports Icon- This icon is used to launch various User Reports such as user Status Reports, User Attribute Reports, User Admin Activity reports, User Access Reports, and Audit Trial reports.

Language Menu- It displays the language you selected in the OFSAA Login Screen. The language options displayed in the Language Menu are based on the language packs installed in your OFSAA instance. Using this menu, you can change the language at any point of time.

User Menu- Clicking this icon displays the following menu:

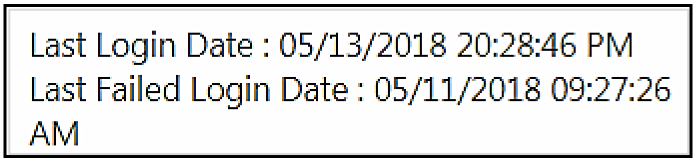
Figure 7: User Menu



- **Preferences**- To set the OFSAA Landing Page.
- **Change Password**- To change your password. For more information, see the Change Password section. This option is available only if SMS Authorization is configured.
- **Log Out**- To log out from OFSAA applications.

Last Login Details - This displays the last login details as shown:

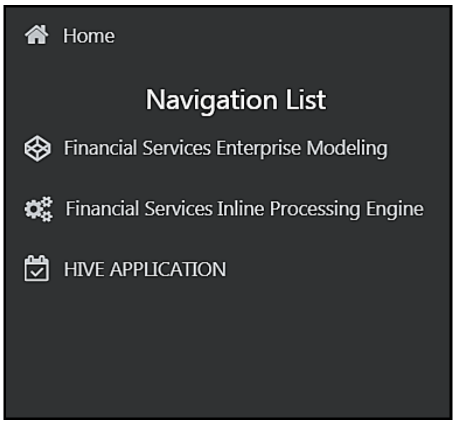
Figure 8: Last Login Details



3.5.2 Navigation Drawer

Click **Hamburger Icon** to launch the Navigation Drawer as shown:

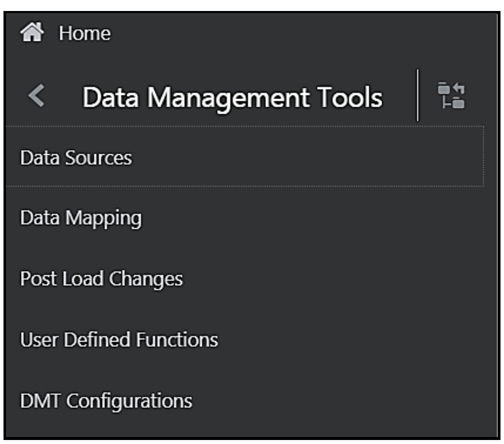
Figure 9: Navigation List drawer



Here the navigation items appear as a list. The First Level menu shows the installed applications. Clicking an application displays the second-level menu with the application name and Common tasks menu. The arrangement of the menu depends on your installed application.

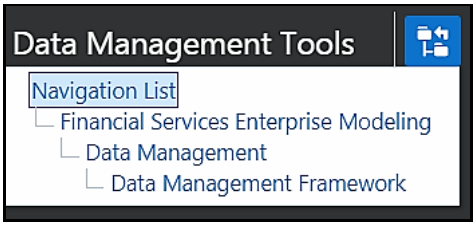
Clicking an item in the menu displays the next level sub-menu and so on. For example, to display Data Sources, click Financial Services Enterprise Modeling > Data Management > Data Management Framework > Data Management Tools > Data Sources.

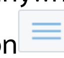
Figure 10: Data Management Tools Menu



Click  **Hierarchical Menu** to display the navigation path of the current sub-menu as shown:

Figure 11: Data Management Tools Hierarchical Menu



The RHS Content Area shows the Summary page of Data Sources. Click anywhere in the Content Area to hide the Navigation Drawer. To launch it back, click the Hamburger icon .

Click **Home** to display the OFSAA Landing Screen.

3.6 Modules in OFSAAI

- **Data Model Management** is intended for uploading the warehouse data from the operational systems to database schema using ERwin XML file.
- **Data Management Framework** is a comprehensive data integration platform that facilitates all the data integration requirements from high-volume and high-performance batch loads to event-driven integration processes and SOA-enabled data services. This module is used for managing Data movement. This includes sub-modules like Data Sources, Data Mapping, Post Load Changes, and Data Quality Framework.
- **Data Entry Forms and Queries** module facilitates you to design web-based user-friendly Data Entry windows with a choice of layouts for easy data view and data manipulation. This module has sub-modules like Forms Designer, Data Entry, and Excel Upload.
- **Data Maintenance Interface** module helps in the design and creation of forms that are in a user-specified format. Authorized users with the required privileges can use these forms to view and update existing data in the database.
- **Unified Analytical Metadata** is intended for the Information and Business Analysts who are instrumental in supporting and affecting analytical decisions. This module is used to define and maintain analytical metadata definitions. This module has sub-modules like Alias, Derived Entity, Dataset, Dimension Management, Business Measure, Business Processor, Build Hierarchy, Business Dimension, Essbase Cube, Filters, Expression, Map Maintenance, and Cube Migration.
- **Rule Run Framework** facilitates you to define a set of rules, reporting objects, and processes that are required to transform data in a warehouse. This module has sub-modules like Rule, Process, Run, and Manage Run Execution.
- **Metadata Browser** module provides extensive browsing capabilities of metadata, helps in tracking the impact of changes to metadata, and traces through to the source of originating data. The metadata in the *Metadata Browser* window is organized into different categories like Data Foundation Metadata, Business Metadata, and Process Metadata.
- **Operations** module facilitates you in the administration and processing of business data to create the highest level of efficiency within the system and to derive results based on a specified rule. It includes sections like Batch Maintenance, Batch Execution, Batch Scheduler, Batch Monitor, Batch Processing Report, Batch Cancellation, and View Log.

- **Questionnaire** module is an assessment tool, which presents a set of questions to users, and collects the answers for analysis and conclusion. It can be interfaced or plugged into OFSAA application packs.
- **System Configuration & Identity Management** module facilitates System Administrators to provide security and operational framework required for Infrastructure. The *Administration* window has a Tiles menu with Tiles like System Configuration, Identity Management, Database Details, Manage OFSAA Product Licenses, Create New Application, Information Domain, Translation Tools, and Process Modelling Framework.
- **Object Administration** facilitates System Administrators to define the security framework with the capacity to restrict access to the data and metadata in the warehouse, based on a flexible, fine-grained access control mechanism. These activities are mainly done at the initial stage and then on a need basis. It includes sections like Object Security, Object Migration, and Utilities (consisting of Metadata Difference, Metadata Authorization, Save Metadata, Write-Protected Batch, Component Registration, Transfer Document Ownership, and Patch Information).

3.7 Logging in OFSAA

Logging in OFSAA is done using Log4J. The log files are available in the following locations:

- **UI/Web Logs**: `<DEPLOYED LOCATION>/<Context>.ear/<Context>.war/logs`
- **Application Logs**: `$FIC_HOME/logs`
- **Execution Logs**: `/ftpshare/logs/<MISDATE>/<INFODOM>/<COMPONENT NAME>/<LOG FILE NAME>.log`

3.7.1 Purging of Logs

Configure the logger-related attributes in the `RevLog4jConfig.xml` file available in the `$FIC_HOME/conf/` folder. Each log file will have appenders in this file and attributes pertaining to this particular appender can be changed.

The default size of the log files is set to 5000 KB and the number of maximum backup log files retained is set to 5, both of which are configurable. Increasing these parameters to a higher value should depend on the server hardware configurations and may reduce the performance.

To configure the Logs file size, follow these steps:

1. Navigate to `$FIC_HOME/conf` folder or `<DeployedLocation>/<context.war>/<context>/` and locate `RevLog4jConfig.xml` file.
2. Configure the logger-related attributes in the `RevLog4jConfig.xml` file. This file will have Appenders for each log file.

   Sample Appender for UMM log file is shown:

   ```
   <RollingFile name="UMMAPPENDER"
   fileName="/scratch/ofsaaweb/weblogic/user_projects/domains/cdb/applicat
   ions/cdb.ear/cdb.war/logs/UMMService.log"

   filePattern="/scratch/ofsaaweb/weblogic/user_projects/domains/cdb/appli
   cations/cdb.ear/cdb.war/logs/UMMService-%i.log" >
   ```

```
<PatternLayout>
  <Pattern> [%d{dd-MM-yy HH:mm:ss,SSS zzz aa}{GMT}] [%-5level] [WEB]
%m%n </Pattern>
  </PatternLayout>
  <Policies>
       <SizeBasedTriggeringPolicy size="5000 KB" />
  </Policies>
   <DefaultRolloverStrategy max="5"> <!-- number of backup files -->
       </DefaultRolloverStrategy>
  </RollingFile>
```

3. To change the log file size, modify the value set for `SizeBasedTriggeringPolicy size`.
4. To change the number of backup files to be retained, modify the value set for `DefaultRolloverStrategy max`.

3.7.2 Log File Format

In OFSAA, log format is standardized and can be read by any standard log analysis tool. The standard log format is as follows:

```
[GMT TIMESTAMP] [LOGGER LEVEL] [LOGGER LOCATION]  [MODULE/COMPONENT]
[LOGGED IN USER] [JAVA CLASS] <LOG MESSAGE>
```

Sample:

```
[25-04-18 10:08:41,066 GMT AM] [INFO    ] [WEB]  [UMM]  [UMMUSER]
[BUSINESSMETADATA] Inside createImplicitObjectsForAllInfodom

[25-04-18 10:08:41,069 GMT AM] [INFO    ] [WEB]  [UMM]  [UMMUSER]
[BUSINESSMETADATA] Call createImplicitObjectsForMapper for infodom =
TESTCHEF

[25-04-18 10:08:42,142 GMT AM] [DEBUG] [WEB]  [UMM]  [UMMUSER]
[BUSINESSMETADATA] Source created successfully for infodom TESTCHEF

[25-04-18 10:08:42,142 GMT AM] [INFO    ] [WEB]  [UMM]  [UMMUSER]
[BUSINESSMETADATA] Start - code added to create user group hierarchy for
this infodom

[25-04-18 10:08:42,142 GMT AM] [INFO    ] [WEB]  [UMM]  [UMMUSER]
[BUSINESSMETADATA] Inside createUserGroupHierarchyForInfodom
```

4 Data Entries Forms and Queries

Data entry Forms and Queries (DEFQ) within the Infrastructure system facilitates you to design web-based user-friendly *Data Entry* windows with a choice of layouts for easy data view and data manipulation. An authorized user can enter new data and update the existing data in the shared database. Data entry Forms are primarily focused to create data entry systems that access the database and load the generated input data.

To access Data Entries Forms and Queries:

1. Login to OFSAA.
2. Click from the header to display the applications in a Tiles menu.
3. Select the **Financial Services Enterprise Modeling** application from the Tiles menu. The Navigation list to the left is displayed.
4. Click **Common Tasks** to expand the list.
5. Click **Data Entries Forms and Queries** to expand the list further. The following links are displayed on the Navigation list:
    a. Excel Upload (Atomic)
    b. Forms Designer
    c. Forms Authorization
    d. Data Entry

4.1 Excel Upload (Atomic)

The *Atomic Schema Upload* window consists of Excel Utilities such as Excel-Entity Mappings and Excel Upload. The Excel Entity Mappings and Upload utilities have restricted access depending on the following function roles mapped:

- Users with XLADMIN and XLUSER function roles can perform both mapping and upload operations.
- Users with the XLADMIN function role can only define mapping and authorize, but cannot upload the file.
- User with the XLUSER function can only retrieve mapping definition (pre-defined by XLADMIN user) and can upload the file based on the retrieved mapping.

Click on the below links to view the section in detail.

- Excel-Entity Mappings
- Excel Upload

4.1.1 Navigating to Excel Upload (Atomic)

You can access the *Excel Upload* window by expanding **Data Entries Forms and Queries** from the Navigation list to the left and clicking **Excel Upload (Atomic)**.

4.1.2 Excel-Entity Mappings

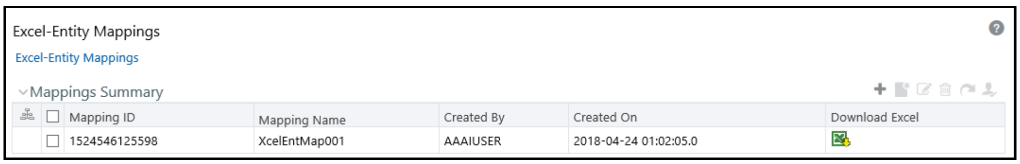
Excel-Entity Mapping helps you to map Excel Data to the destination table in the database. Excel-Entity Mapping supports excel files created in Microsoft 2007 and earlier versions along with the option to map and upload multiple sheets created within a single excel file. You need to have the XLADMIN function role mapped to define the mapping.

4.1.3 Adding Excel-Entity Mappings

To define the mapping in the *Excel-Entity Mappings* window:

1. From the LHS menu of *DEFQ- Excel Upload* window, click **Excel-Entity Mappings**. The *Excel-Entity Mappings* window is displayed.

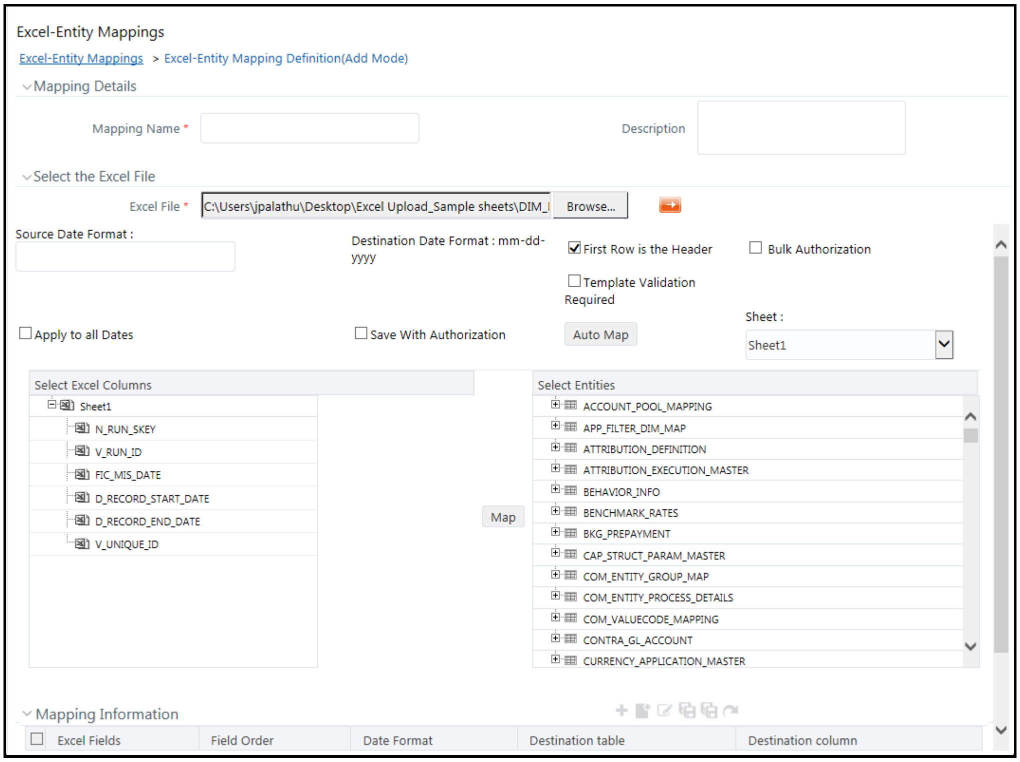
Figure 12: Excel-Entity Mappings window



2. Click the button in the Mappings Summary toolbar. The *Add Excel-Entity Mappings* window is displayed.
3. Enter the **Mapping Name** and a brief **Description**.
4. Click **Browse**. The Choose File to Upload dialog is displayed.
5. Select the required Excel file to be used as the template and click the button.

   The columns in the selected Excel template are listed in the Select Excel Columns grid and the database tables are listed in the Select Entities grid.

Figure 13: Excel-Entity Mappings window



6. Enter the format in which the dates are stored in the excel sheet in the **Source Date Format** field.
7. Select the **Apply to all Dates** checkbox if you want to apply the source date format to all date fields in the excel sheet.
8. Select the **First Row is the Header** checkbox if your Excel template has a header row.
9. Select the **Template Validation Required** checkbox to validate whether the Excel template you use is the same as the Excel sheet you use during the Excel Upload window. The validation is done when you upload the excel sheet. The error will be displayed if there is any mismatch between the Excel template you use to map and the actual Excel sheet you upload.
   This field is displayed only if you have selected the **First Row as the Header** checkbox.
10. Select the **Bulk Authorization** checkbox to assign the “Excel_Name” across the selected column. For example, the selected column “v_fic_description” will have the Excel Name assigned.

NOTE	Ensure that the selected “**Varchar2**” column contains the required length to hold the Excel Name. To select Bulk Authorization, you need to have **Save with Authorization** checkbox selected.

11. Select **Save with Authorization** checkbox to authorize the data upon successful data load. The three mandatory fields namely Maker ID, System Date, and Authorization Status are displayed in the Select Excel Columns grid.
    You need to map these fields to the corresponding columns in the Select Entities grid. The value

for the Maker ID column is updated with the User ID of the user who is performing the Excel Upload. The value for Maker Date is updated with the current System Date during which the upload is performed and the value for Authorization Status is updated with the flag 'U'. See Save with Authorization to create a Form where the uploaded data can be authorized.

12. Select a column from the Select Excel Columns grid and select an attribute or column from the required table from the Select Entities grid. Click **Map**.

13. Click **Automap**. The respective columns with similar names in the Excel sheet and the database are mapped. You need to manually map the other columns. The mapping details are displayed in the Mapping Information grid which facilitates you to edit the details as required.

14. Click **Save Mapping**. The *Excel-Entity Mapping* window displays the excel-database table mapping details.

    In the *Excel-Entity Mappings* window, you can also do the following:

    - Click the View button in the Mappings Summary toolbar to **View** the mapping details.
    - Click the Edit button in the Mappings Summary toolbar to **Edit** the mapping details.
    - Click the Delete button in the Mappings Summary toolbar to **Delete** the mapping details.
    - Click the Download Excel button to download the Excel template used in the mapping.

4.1.4 Excel Upload

Excel Upload helps you to upload Excel Data to the destination table in the database. You need to have the “XLUSER” function role mapped to access the *Excel Upload* window and retrieve mapping definition (pre-defined by XLADMIN user) to upload excel data. Excel Upload supports excel files created in Microsoft 2007 and earlier versions along with the option to map and upload multiple sheets created within a single excel file. You need to ensure that the excel data contains the dates in the format as defined in Add Excel-Entity Mapping definition.

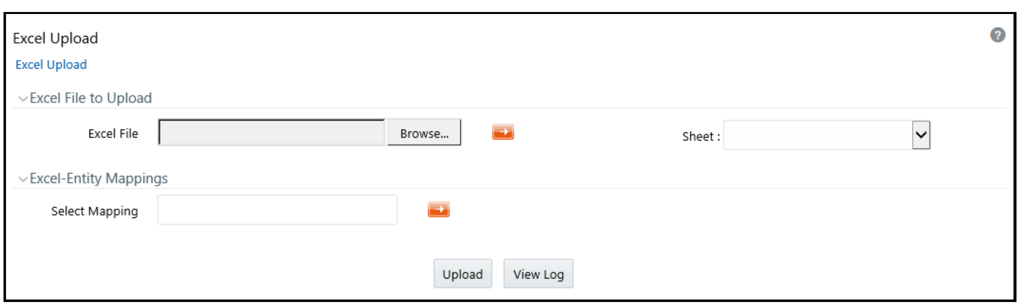
To upload excel data in the *Excel Upload* window:

1. Click **Browse** in the Excel File to Upload grid. The Choose File to Upload dialog is displayed.

2. Select the required Excel file and click the Move button.

   Select the required sheet in the Excel file from the **Sheet** drop-down list and the Preview grid displays the data of the selected sheet of the Excel file.

Figure 14: Excel Upload window



3. Click the Move button in the Excel-Entity Mappings grid. The Mapping Selector dialog is displayed with the pre-defined mapping details.

4. Select the checkbox adjacent to the required mapping definition and click **OK**.

> **NOTE** You can download the Excel template used in the mapping by clicking the Download Excel button.

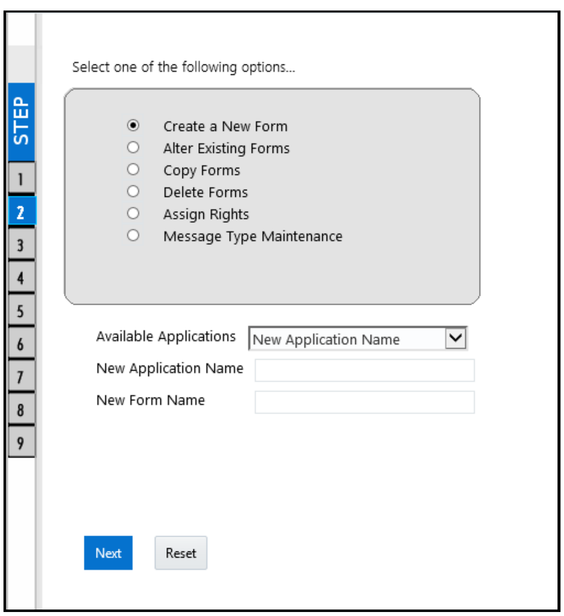
5. Click **Upload**. A confirmation dialog is displayed on successful upload and the excel data is uploaded to the database table. You can click on **View Log** to view the log file for errors and upload status.

4.2 Forms Designer

> **NOTE**
> 1. This functionality doesn't work when CSRF is enabled. To disable CSRF, see the section Update General Details.
> 2. This functionality displays only on the Microsoft Internet Explorer™ browser.

Forms Designer within the Data Entry Forms and Queries section facilitates you to design web-based user-friendly Forms using the pre-defined layouts. You can access DEFQ - Forms Designer by expanding **Data Management Framework** and **Data Entry Forms and Queries** within the tree structure of the LHS menu and selecting **Forms Designer**.

Figure 15: Forms Designer window



The *DEFQ - Forms Designer* window displays a list of pre-defined options to create, modify, and delete Forms. You can also assign rights and define messages. By default, the option to Create a New Form is selected and the left pane indicates the total steps involved in the process. The available options are as indicated below. Click on the links to view the section in detail.

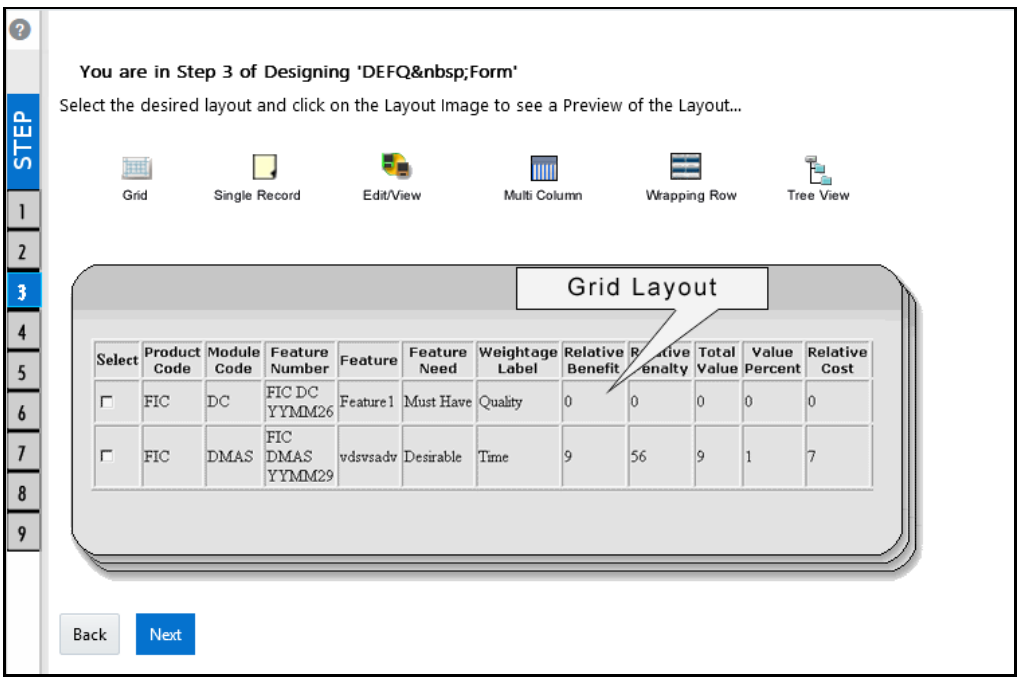
- Creating a New Form
- Altering Existing Forms
- Copying Forms
- Deleting Forms
- Assigning Rights
- Message Type Maintenance

4.2.1 Creating a New Form

To design a new Form in the *DEFQ - Forms Designer* window:

1. Ensure that **Create a New Form** option is selected and do the following:
   - Specify the application name by either entering the **New Application Name** or selecting **Available Applications** from the drop-down list.
   - Enter the **New Form Name**.
2. Click **Next**. The *DEFQ - Layout* Window is displayed with a range of pre-defined layouts for you to choose from.

Figure 16: DEFQ – Layout window (Step 3 of Designing Form)



The following table describes the layouts in the DEFQ – Layout window.

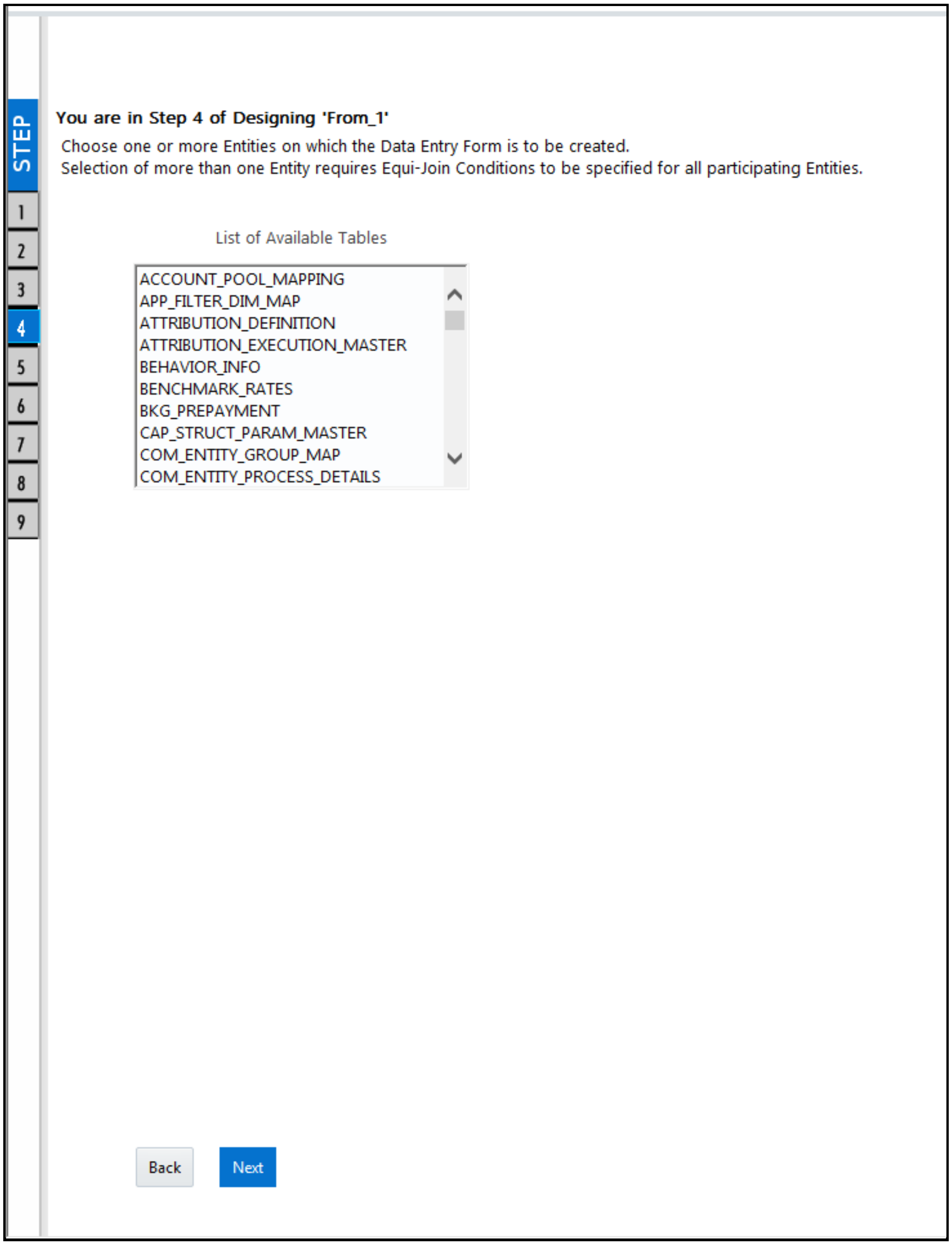
Table 3: Layouts in the DEFQ – Layout window and their Description

Layout	Description
Grid Layout	It is the default layout that displays the records in the Form of a table/grid with multiple rows of data.

Layout	Description
Single Record Layout	It displays a single record at a time.
Edit/View Layout	It is a combination of the Single Record and Grid layout. By selecting a record in the grid, the record is displayed in a single record format, which is editable. By default, the first record will be displayed in the editable grid. **Note**: The column names are editable only during altering the created Form.
Multi-Column Layout	It 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.
Wrapping Row Layout	It 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.
Tree View Layout	It displays the Hierarchical dimensional table with the selected dimension details. You can select the following options: Dimensional Table Tree Parent-Child Tree **Note**: The process to create a Form using the Tree View Layout differs from the procedure explained below. For more information, refer Create Tree View Form in the References section.

3. Select the required layout and click **Next**. The List of Available Tables is displayed.
4. Select the required Table from the list on which the Form is to be created.

Figure 17: DEFQ – List of Available Tables Selection window (Step 4 of Designing Form)



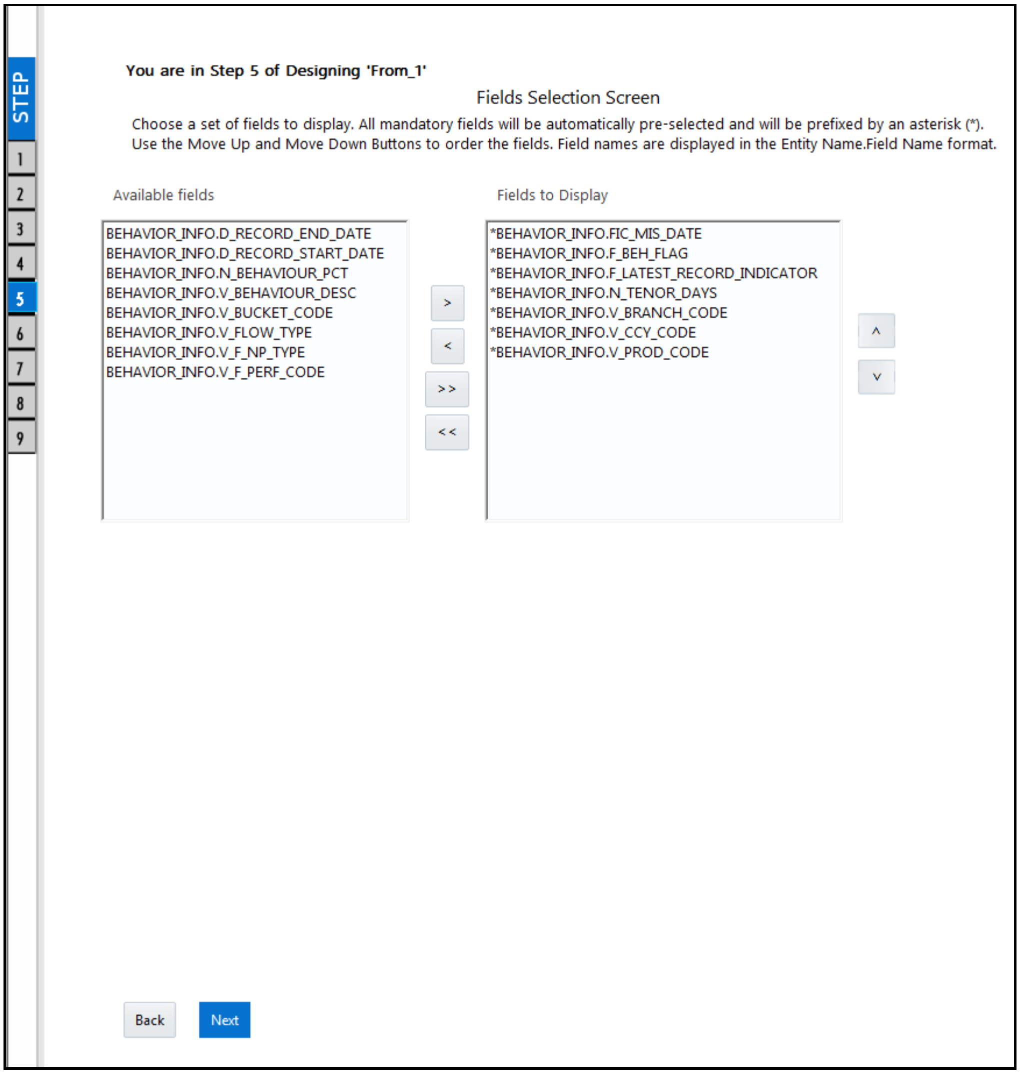
NOTE	You should use tables with names not longer than 25 characters. This is a limitation.

For multiple selections, you can either press the **Ctrl** key for nonadjacent selection or the **SHIFT** key for adjacent selections. Click **Next**, the *Fields Selection* window is displayed.

> **NOTE** If multiple tables are selected, you need to specify Join Conditions. Select the Table from the drop-down list and select the Available Fields. Specify the **Join Condition**. Click **Next**, the join conditions are validated and the *Fields Selection* window is displayed.

5. Select the fields to be joined from the **Available Fields** list and click the Move icon. You can press the **Ctrl** key for multiple selections and also click Move All icon to select all the listed fields. All mandatory fields are auto-selected and are indicated on the window with an asterisk (*****).

Figure 18: DEFQ – Fields Selection window (Step 5 of Designing Form)



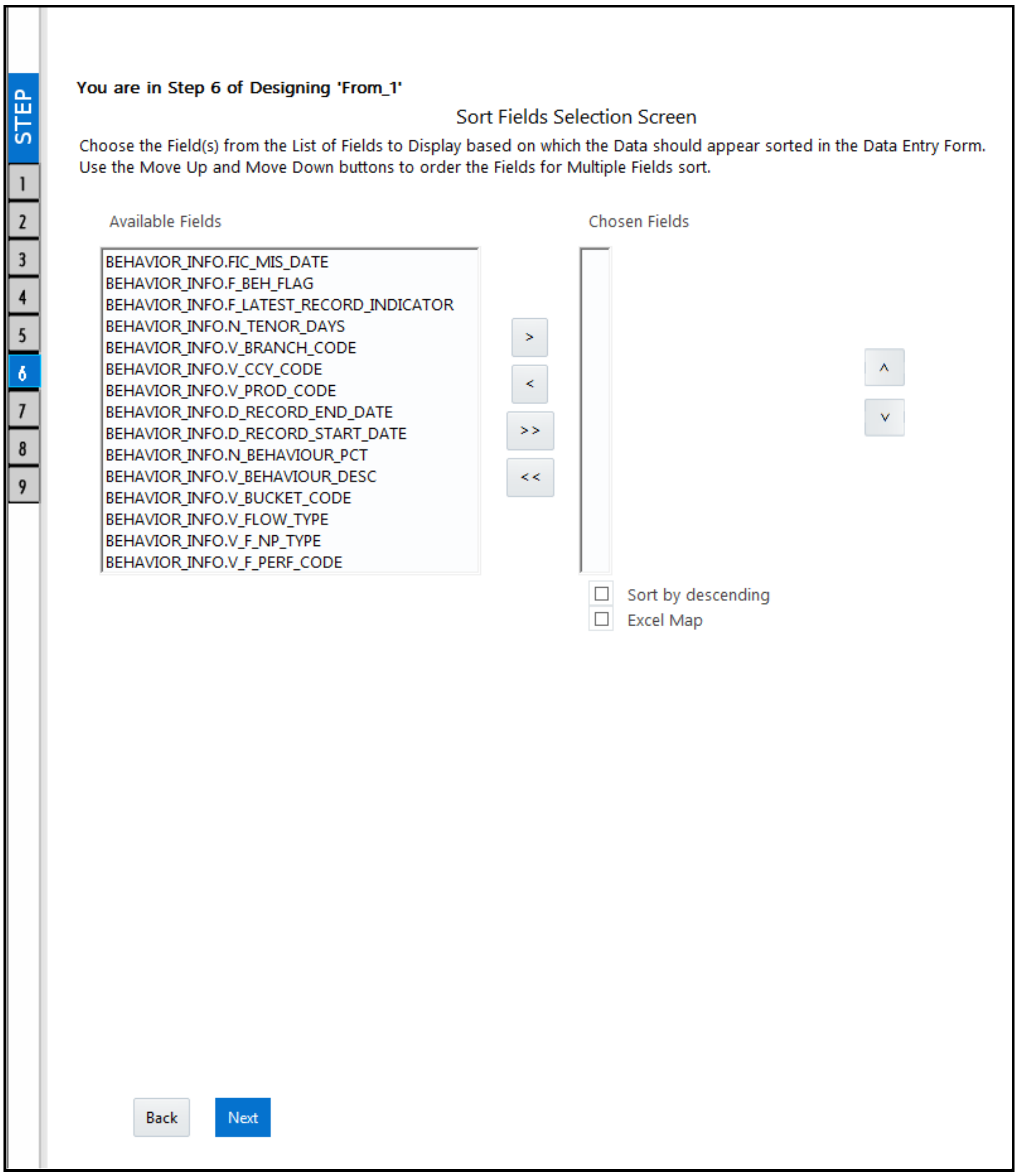
NOTE

You can click or buttons to arrange the fields in the required order as intended to display in the Data Entry Form. The fields order need not be similar to the arrangement in the underlying table.

Ensure the fields selected are not of CLOB data type since it is not supported in DEFQ.

6. Click **Next**. The *Sort Fields Selection* window is displayed.

Figure 19: DEFQ – Sort Fields Selection window (Step 6 of Designing Form)



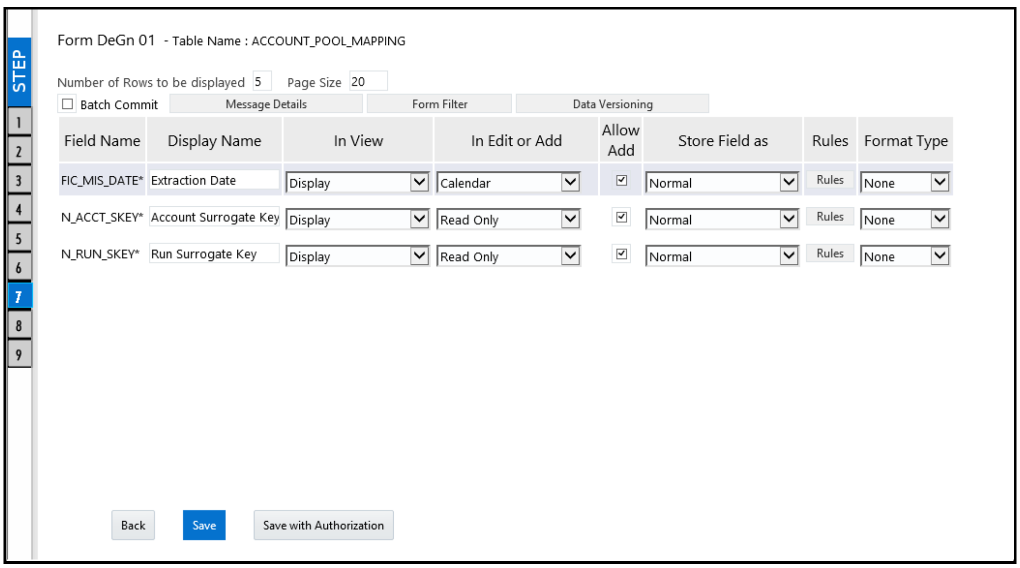
You can sort the fields in the required order as intended to display in the Data Entry Form. Also, the mandatory fields which need user inputs are indicated in the '*****' symbol and are auto-selected in the Selected Fields pane.

- Select the field from the **Available Fields** list and click ›. You can press the **Ctrl** key for multiple selections and also click » to select all the listed fields.
- (Optional) To arrange multiple fields, select the **Sort by Descending** checkbox.
- (Optional) Select the **Excel Map** checkbox to enable Bulk Authorization.

NOTE In case you have selected the **Excel Map** checkbox, you need to select "Excel Name" from the **Store Field As** a list in the *DEFQ Field Properties* window. Only on selection, the "SelectExcelSheetName" list is displayed for the authorizer in the *DEFQ - Data Entry* window.

7. Click **Next**. The *DEFQ Field Properties* window is displayed with the Form details such as Field Name, Display Name, In View, In Edit/Add, Allow Add, Store Field as, Rules, and Format Type.

Figure 20: DEFQ – Field Properties window (Step 7)



Specify the parameters for each field as tabulated.

The following table describes the fields in the DEFQ – Field Properties window.

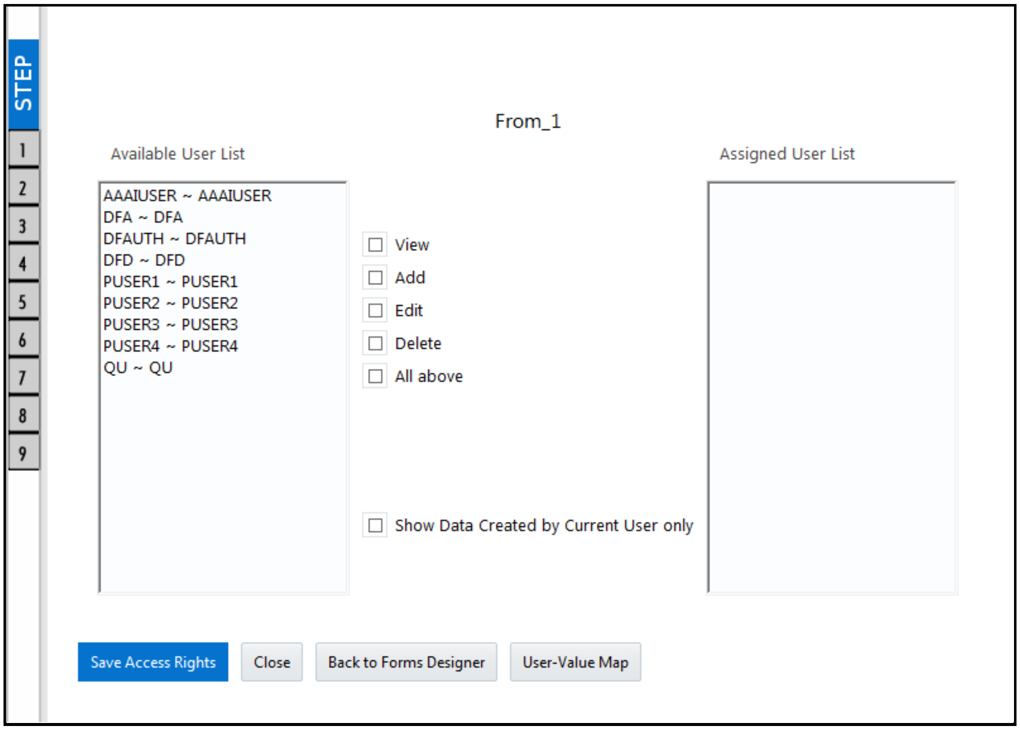
Table 4: Fields in the DEFQ – Field Properties window and their Description

Field	Description
Display Name	Edit the default Display Name if required.
In View	Select either Display or Do not Display to display the field in the Form. If the field is a foreign key field or if more than one table is selected, then the following options are available in the drop-down list; Same Field Alternate Display Field Do not Display options

Field	Description
In Edit/Add	Specify the edit parameters by selecting from the drop-down list. The available options depend on the type of field selected. For normal fields, you can select Text Field, Text Area, Select List, Protected Field, Read-Only, and Do Not Show. For foreign key fields s you can select Read-Only, Select List, and Do Not Show. For primary key fields, you can select Read Only and Do Not Show. For calendar fields, you can select Calendar and Do Not Show. **Note**: If you choose the **Select List** option, you need to define the values. For more information, refer to Define List of Values.
Allow Add	Select the checkbox to permit users to add a new record. **Note**: An alert message is displayed if you are trying to save a Form with add option disabled for the mandatory fields.
Store field as	Select the required option from the drop-down list. You can select 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, Maker Remarks, and Excel Name (If Excel Map is selected in *Sort Fields Selection* window).
Rules	Click **Rules** and specify Rules and Expressions for the selected field in the *Specifying Rules and Expressions for Data - Validations* window. For more information, refer Applying Rules section in the References.
Format Type	Select the required Format type from the drop-down list depending on the field type selected. CLOB data type is not supported.
Batch Commit	Select the checkbox to group all the sets of table Forms into a batch. All the Form tables are executed along with the batch execution and if in case, a Form in the table fails to execute, the entire set of Forms are returned.
Message Details	Click **Message Details** to define the message type for Creator and Authorizer in the *Messaging Details for a Form* window. For more information, refer to Define Message Details.
Form Filter	Click Form Filter to define an expression for Form-level filter condition in the *Filter for Form* window.
Data Versioning	Click **Data Versioning** to perform data versioning on an authorized Form. For more information, refer to Form Data Versioning.

8. Click either **Save** to only save the Form details or click **Save for Authorization** to save the changes with authorization. For more details, refer Save for Authorization section.

Figure 21: DEFQ – Save window



NOTE	Sometimes, on clicking **Save**, the form does not get saved. This is because the Java heap size setting for the OFSAAI service is set too high and the webserver memory setting is too low. Contact System Administrator to modify it to the appropriate setting by viewing the log file created in the path: $FIC_APP_HOME/common/FICServer/logs/.

While saving, the *User for Mapping - DEFQ* window is displayed which facilitates you to assign user rights to the Form. For more information, refer to Assign Rights.

4.2.2 Altering Existing Forms

To alter the field details of an existing Form in the *DEFQ - Forms Designer* window:

1. Select **Alter Existing Forms** from the available options and do the following:
   - Select the **Available Applications** from the drop-down list.
   - Select the **Available Forms** from the drop-down list. The listed Forms are dependent on the DSN (Data Source Name) that you have specified.
2. Click **Next**. The *Fields Selection* Window is displayed.

   Add or **remove** the selected fields as required to be displayed in the Form. You can choose a field from the **Available Fields** list and click > to add, or choose the selected field from the

Fields to the Display list and click ‹ to de-select. You can press the **Ctrl** key for multiple selections and also click » or « buttons to select/deselect all the listed fields.

3. Click **Next**. The *Sort Fields Selection* Window is displayed.
   - Sort the fields in required order as intended to display in the Form. You can choose a field from the list and click › or ‹ buttons to select/deselect. You can also click » or « buttons to select/deselect all the listed fields.
   - Select a field and click ˄ or ˅ buttons to arrange fields in the required order.
   - (Optional) To arrange multiple fields, select the **Sort by Descending** checkbox.
   - (Optional) Select the **Excel Map** checkbox to enable Bulk Authorization.

NOTE	In case you have selected the **Excel Map** checkbox, you need to select "Excel Name" from the **Store Field As** a list in the *DEFQ Field Properties* window. Only on selection, the "SelectExcelSheetName" list is displayed for the authorizer in the *DEFQ - Data Entry* window.

4. Click **Next**. The *DEFQ Field Properties* window is displayed.

   Modify the parameters for each field as required. Refer to DEFQ Field Properties details.

5. Click either **Save** to save the Form details or click **Save for Authorization** to save the changes with authorization.

   While saving, the *User for Mapping - DEFQ* window is displayed which facilitates you to assign user rights to the Form. For more information, refer to Assign Rights.

4.2.3 Copying Forms

You can duplicate and recreate a Form with the required variants from an existing Form. You can also change user rights or display options and other subtle variations for the selected layout.

To Copy a Form in the *DEFQ - Forms Designer* window:

1. Select **Copy Forms** from the available options and do the following:
   - Select the application from the **From Application** drop-down list which consists of the required Form which you want to copy.
   - Select the application from the **To Application** drop-down list for which you want to copy the Form.
   - Select the required Form from the **Save Form** drop-down list.
   - Enter a name for the Form in the **As Form** field.
2. Click **Next**. The specified Form is duplicated as a new Form and a confirmation dialog is displayed with the status.

4.2.4 Deleting Forms

You can remove the forms which are not required in the system by deleting them from the *DEFQ - Forms Designer* window.

1. Select **Delete Forms** from the available options and do the following:
   - Select the application from the **Available Application** drop-down list which consists of the required Form which you want to delete.
   - Select the Form from the **Available Forms** drop-down list which you want to delete.
2. Click **Next**. An information dialog is displayed for confirmation. Click **OK**.

4.2.5 Assigning Rights

You can assign user permissions to view, add, edit, and delete the Form details in the User for *Mapping - DEFQ* window.

1. Select **Assign Rights** from the available options and do the following:
   - Select the required application from the **Available Applications** drop-down list.
   - Select the required form for which you want to assign rights to a user from the **Available Forms** drop-down list.
2. Click **Next**. The *DEFQ- Assign Rights* window is displayed.
3. Select the required user from the **Available User List**. You can also click < or > buttons to reload the previous/next set of users in the list.
4. Select the checkbox corresponding to the user permissions such as **View**, **Add**, **Edit**, **Delete**, or **All Above**. You must give View permission to allow users to Edit or Delete a Form.
5. Select **Authorize** or **Auto-Authorize** checkbox as required.

   The **Authorize** and **Auto-Authorize** options are applicable for all the forms that have been saved with the Authorize option. The **Auto-Authorize** feature for records is applicable in scenarios where the Creator and Authorizer are the same. If a user has **Add** and **Auto-Authorize** permissions, the data entered by the user is auto authorized and the data will be in **Authorized** status. In the case of normal Authorization, the Record added by the creator has to be authorized by a different user who has **Authorize** permissions.

   > **NOTE** The **Auto-Authorize** feature in Forms Designer is applicable only for data entered through the *Data Entry* window and not through the *Excel Upload* window.

6. Select the **Show Data Created by Current Users Only** checkbox if you want the current user to view data created by him only.
7. Click **User Value Map** to map users to the form based on data filter.
8. Click **Save Access Rights**. A confirmation dialog is displayed after saving and the user is added to the **Assigned User List**.

User Value Map

This feature allows you to create a data filter based on any field/column of the table you selected for designing the Form. When a user tries to access the form in the *Data Entry* window, data will be filtered and displayed based on the selected field, to the users associated with that column.

NOTE	The data type of field/column you select to define the filter should be NUMBER or VARCHAR. The users mapped to the DEFQ form whose assign rights are authorized through "Forms Authorization" can save the filter.

There are two types of filters, Global Data Filter and Custom Data Filter.

Global Data Filter: In this filter, the value will be fetched from the DEFQ_GLOBAL_VALUES table of the Atomic schema, which is automatically created during information domain creation. The table needs to be populated manually through excel upload. The table contains all the entities and the users mapped to them.

Custom Data Filter: This filter enables the user to provide a custom filter for the form you design. In this filter, you should enter values for all the users mapped to the form manually.

To set a Data Filter:

9. Click **User Value Map** in the *DEFQ- Assign Rights* window.

   The *User Value Map* window is displayed.

10. Select the **Global Data Filter** option to filter the data globally.

    - Select the field based on which the data should be filtered and displayed for the user, from the Fields to the Display section.

NOTE	Normally the user can access all the data from the table whenever the DEFQ form is created. Based on this filter, the user will be displayed only the data which is mapped to him.

11. Select the **Custom Data Filter** to provide a custom filter for a specific DEFQ Form.

    - Select **User ID** from the drop-down list and enter **Values** for that user. It is mandatory

12. Click **Save.**

4.2.6 Message Type Maintenance

You can manage the Message Type details which alert the Creator of the Form or to an Authorizer in the *DEFQ Message Type Maintenance* window. Message Type details can be defined while creating a Form. For more information, refer to Define Messaging Details.

In the *DEFQ - Forms Designer* window, do the following:

1. Select **Message Type Maintenance** from the available options and click **Next**.

   The *DEFQ - Message Type Maintenance* window is displayed.

2. Select the message category from the **Message Type** drop-down list.
3. Edit the message details by doing the following:
   - The defined **Message Subject** and **Message Content** are auto-populated. Edit the details as required.
   - Add or remove the defined recipients. Double-click on the required member to toggle between the **Available** and **Mapped Recipients** list.

NOTE	Selecting Authorizer alerts all the selected authorizers for authorization.

4. Click **Save**. A confirmation is displayed on updating the Message Type details.

4.3 Forms Authorization

NOTE	**3.** This functionality doesn't work when CSRF is enabled. To disable CSRF, see the section Update General Details. **4.** This functionality displays only on the Microsoft Internet Explorer™ browser.

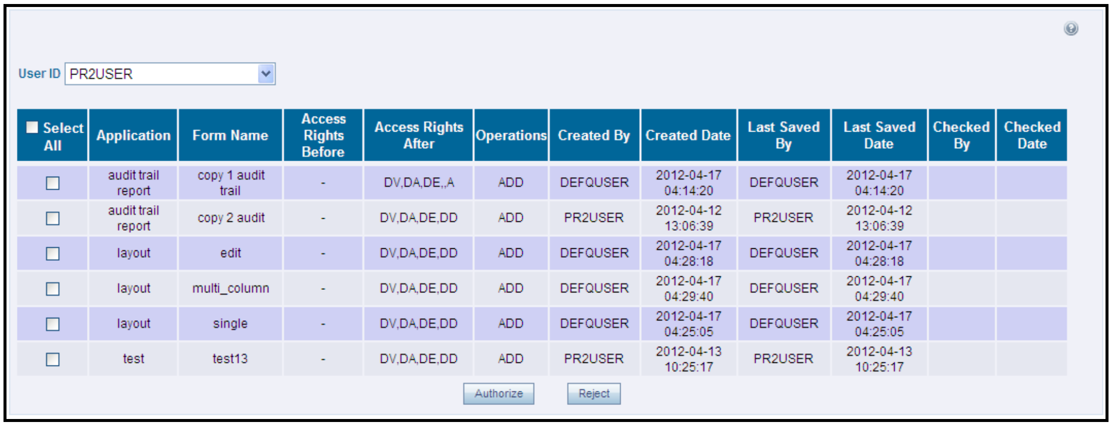
Forms Authorization within the Data Entry Forms and Queries section of the Infrastructure system facilitates you to view and authorize/approve any changes that are made to the privileges assigned to a user in a particular Form.

You need to have the **FRMAUTH** function role mapped to access the *Forms Authorization* window.

NOTE	You cannot authorize or reject a right request created by you, even if you have the **FRMAUTH** function role mapped.

You can access the *Forms Authorization* window from the left-hand side (LHS) menu of the Infrastructure home page. Click "+" and expand the Data Model Management and select **Data Entry Forms and Queries**.

Figure 22: Forms Authorization window



The *Forms Authorization* window displays the list of privileges assigned to a user in different Forms. These privileges include create, view, modify, delete, authorize, and auto-authorize records. The *Forms Authorization* window allows you to select a user from the drop-down list adjacent to the **User ID** field. This field displays the User IDs associated with the selected Information Domain.

On selecting a user from the **User ID** field, the columns in the *Forms Authorization* window lists the grants requested for that user on different Forms as listed below.

The following table describes the columns in the Forms Authorization window.

Table 5: Column Names in the Forms Authorization window and their Description

Column Name	Description
Application	Lists the specific application to which the Form has been assigned.
Form Name	Displays the Form Name.
Access Rights Before	Displays the available Right Requests for the selected user in the Form. **Note**: For the new Form, the column remains blank.
Access Rights After	Displays the Right Requests raised for authorization. **DV** - DEFQ VIEW **DA** - DEFQ ADD **DE** - DEFQ EDIT **DD** - DEFQ DELETE **A** - AUTHORIZE **DU** - AUTO AUTHORIZE **S** - SHOW DATA CREATED BY CURRENT USER ONLY
Operations	Displays the operation carried out in the Form. For example, "**ADD**" indicates a new form is created and specific roles are assigned.
Created By	Displays the USER ID from which the Right Request has been created.
Created Date	Displays the date on which the Right Request has been created.
Last Saved By	Displays the USER ID from which the previous Right Request change has been saved.

Column Name	Description
Last Saved Date	Displays the date on which the previous Right Request change has been saved.
Checked By	Displays the USER ID from which the Right Request has been authorized.
Checked Date	Displays the date on which the Right Request has been authorized.

To authorize or Reject a form in the *Forms Authorization* window:

1. Select the **User ID** from the drop-down box. 4B43BThe Right Requests submitted on various forms are displayed.
2. Select the checkbox(s) adjacent to the requests to authorize / reject.

   You can also select all the requests at once for a user, by clicking the **Select All** checkbox.
3. Click **Authorize / Reject** to authorize or reject the selected Right Requests.

   Once Form action privileges are authorized for a user, those actions can be performed on the Form. For an existing Form with certain rights, the rights remain the same until the changes are authorized/rejected by an authorizer.

NOTE Special chars are not allowed in DEFQ definitions except underscore (_).

4.4 Data Entry

NOTE This functionality doesn't work when CSRF is enabled. To disable CSRF, see the section Update General Details.

This functionality displays only on the Microsoft Internet Explorer™ browser.

Data Entry within the Data Entry Forms and Queries section of the Infrastructure system facilitates you to view, add, edit, copy, and delete data using the various layout formats and Authorize/Re-authorize data records based on the permissions defined during the Form creation.

You can use the Search option to query the records for specific data and also export the data in Microsoft Excel format for the reference. You can launch multiple instances of the *Data Entry* window using the URL to search and update records simultaneously.

You can access DEFQ - Data Entry by expanding the Data Entry Forms and Queries section of the Data Model Management module within the tree structure of the LHS menu.

NOTE An alert message is displayed if you are not mapped to any Forms in the system.

Figure 23: DEFA – Data Entry window

The *DEFQ - Data Entry* window displays the list of Data Entry Forms and Query Forms mapped to the logged-in user in the LHS menu. You can select the required Form to view the details. In the *DEFQ - Data Entry* window, you can do the following:

- Viewing Form Details
- Editing Form Details
- Adding Form Data
- Authorizing Records
- Exporting Form Data
- Copying Form Data
- Deleting Form Details

4.4.1 Viewing Form Details

The *DEFQ - Data Entry* window displays the selected Form Data in the View mode by default. The Forms are displayed based on the application names in the LHS menu. There are various layouts available to customize the view and by default, the Form details are displayed in the layout in which it was designed.

In the *DEFQ - Data Entry* window, the following layout types are available. You can click on any of the following layouts to view the Form details. The buttons i.e. **Previous Page**, **Back**, **Next**, and **Next Page** helps you to navigate through the records. However, the customized header sorting does not apply when you have navigated to Previous or Next pages.

NOTE The **Roll Back** option can be used only for authorized records i.e. after the records are edited and saved, you can roll back/undo the changes in view mode.

The following table describes the Layouts in the DEFQ – Data Entry window.

Table 6: Layouts in the DEFQ – Data Entry window and their Description

Layout	Description
Single Record	To view the details of a single record at any given point. You can use the navigation buttons to view the next record in the table.
Editable View	To view and edit a single record. A list of five rows/records is displayed by default, and the same can be changed by entering the required number in **Display Rows**. You need to select the required record from the list to view/edit and click **Save** to update the changes.
Grid (Default)	To view all the records in a list. A list of five rows/records is displayed by default, and the same can be changed by entering the required number in **Display Rows**. You can click on the column header to alphabetically sort the list of records in the table.
Multi-column	To view all the columns of a selected record. This layout enables you to view a record without having to scroll or with minimum scrolling to view all the columns.
Wrapped rows	To view all the rows of a selected record. This layout enables you to view a wrapping row easily without having to scroll horizontally to view the columns.

4.4.2 Searching Records

In the *DEFQ - Data Entry* window, you can search for a record in the View, Edit, and Authorize modes. You can perform a quick **Search** to find a specific record or run an **Advanced Search** to further query the record for the required details.

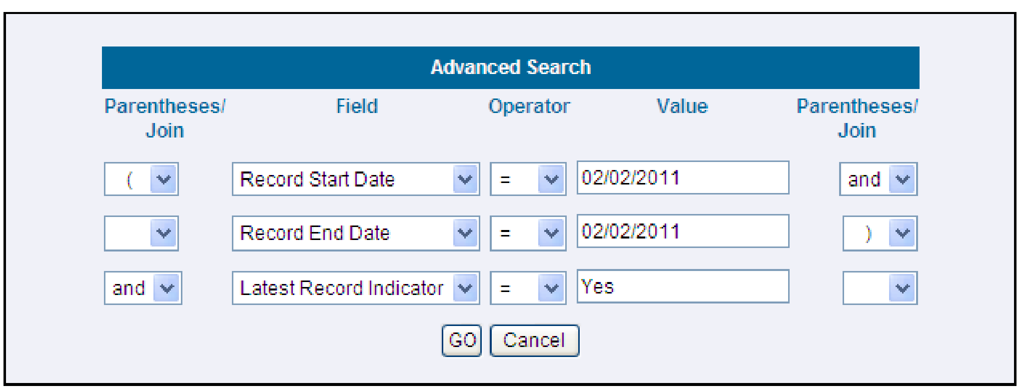
To search for a record in the *DEFQ - Data Entry* window:

1. Click Search. The search fields are displayed.
2. Select **Field Name** from the drop-down list.
3. Enter the **value/data** in the Search field.
4. Click **Go**. The search results are displayed in the list.

To perform an **Advanced search** in the *DEFQ - Data Entry* window:

5. Click within the Search fields. The *Advanced Search* window is displayed.

Figure 24: Advanced Search window



6. Select the required Parentheses/Join, Field, Operator from the drop-down list and enter the **Value** as required to query the Form data.
7. Click **GO**. The results are displayed with the field names containing the searched data.

4.4.3 Editing Form Details

You can edit the permitted Form field values in the *DEFQ - Data Entry* window. However, you cannot modify the primary key fields which are displayed in a non-editable format.

To edit Form Details in the *DEFQ - Data Entry* window:

1. Open the required Form in view mode and click . The editable fields are enabled.
2. Enter/update the required details.
3. Click **Save** and update the changes.
4. If required, you can click **Reset** to undo the changes and return to original field values.

   If you have edited an Authorized record, the same is again marked for authorization. Once the record is updated, a modified status flag is set, and only these record changes can be rolled back. The Roll Back option is supported in view mode only for authorized records, i.e. records that are updated and saved.

4.4.4 Adding Form Data

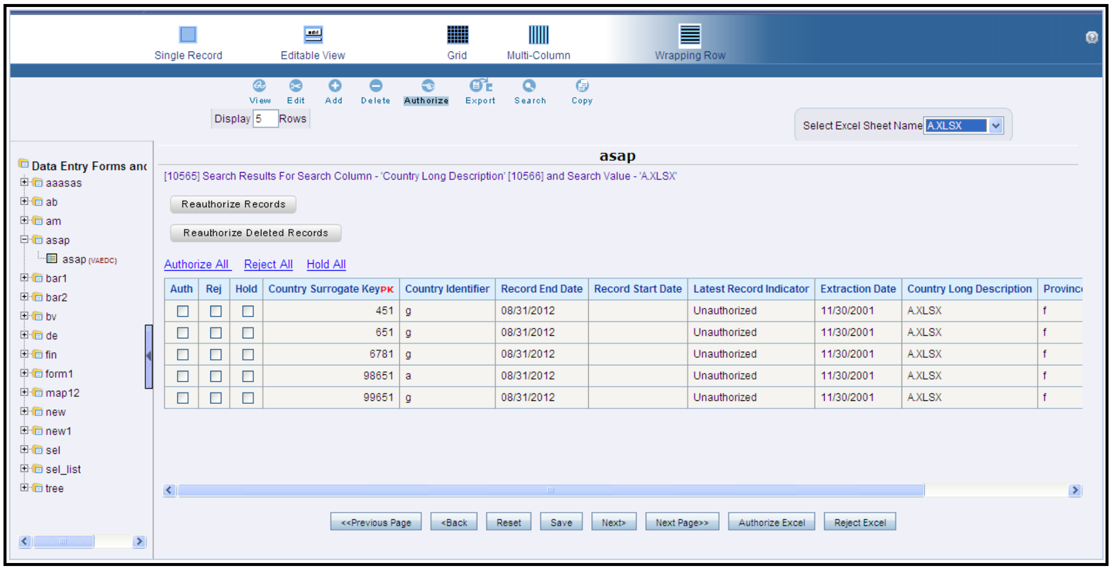
You can add a row to the required table and enter the field details. To Add Form Data in the *DEFQ - Data Entry* window:

1. Open the required Form in view mode and click .
2. By default, five rows are displayed. You can modify by specifying the number of required rows in the **Display Rows** field and clicking **Reset**.
3. Enter the required numeric data in the new fields. If you want to view the numeric data separated by commas, enter the details accordingly.
4. Click **Save** and update the data to the selected table.

4.4.5 Authorizing Record

You need to have DEFQMAN and SYSAUTH function roles mapped to access and authorize Forms in the DEFQ framework. You can Authorize a single record or all the records of a selected Form within the *DEFQ - Data Entry* window. You can Authorize a record in a table that has a primary key field. A primary key field in the record is indicated by “PK”. You need to have the authorization rights defined by the user who has created the record. You can also Reject or Hold inappropriate records in the table.

Figure 25: DEFQ - Data Entry Authorization window



The status of each record in the table is indicated with an "AuthFlag" as indicated below:

- **Unauthorized** records are displayed with the status flag "**U**"
- **Authorized** records are displayed with the status flag "**A**".
- **Rejected** records are displayed with the status flag "**R**".
- **Modified** records are displayed with the status flag "**M**".
- **Deleted** records are displayed with the status flag "**D**".
- If an **Unauthorized** record is on **Hold**, the status flag is displayed as "**H**".
- If a **Modified** record is on **Hold**, the status flag is displayed as "**X**".
- If a **Deleted** record is on **Hold**, the status flag is displayed as "**Z**".

To Authorize Data in the *DEFQ - Data Entry* window:

5. Open the required Form in view mode and click .

   The list of available records for Authorization is displayed. If there are "no records" for Authorization in the selected Information Domain, an alert message is displayed.

6. Select the "Auth" checkbox adjacent to the required record with the status flag "**Unauthorized / Put On Hold**" and click **Save**. A confirmation dialog is displayed. Click **OK**.

   You can also do a **Bulk Authorization** if Excel Map is selected in the *Sort Fields Selection* window. Select the mapped Excel Name from the "SelectExcelSheetName" drop-down list. The *DEFQ - Data Entry* window displays only those records which are uploaded through the selected Excel sheet. Click **Authorize Excel**. A confirmation dialog is displayed. Click **OK**.

You can Reject / Hold a record by doing the following:

- To **Reject** a record, select the checkbox in the "**Rej**" column adjacent to the required record and click **Save**. A confirmation dialog is displayed. Click **OK**.

You can also Reject records in Bulk Mode if Excel Map is selected in the *Sort Fields Selection* window. Select the mapped Excel Name from the "SelectExcelSheetName" drop-down list. The *DEFQ - Data Entry* window displays only those records which are uploaded through the selected Excel sheet. Click **Reject Excel**. A confirmation dialog is displayed. Click **OK**.

- To **Hold** a record and to authorize or reject at a later point, select the checkbox in the "**Hold**" column adjacent to the required record and click **Save**.

In the *DEFQ - Data Entry* window, you can also do the following:

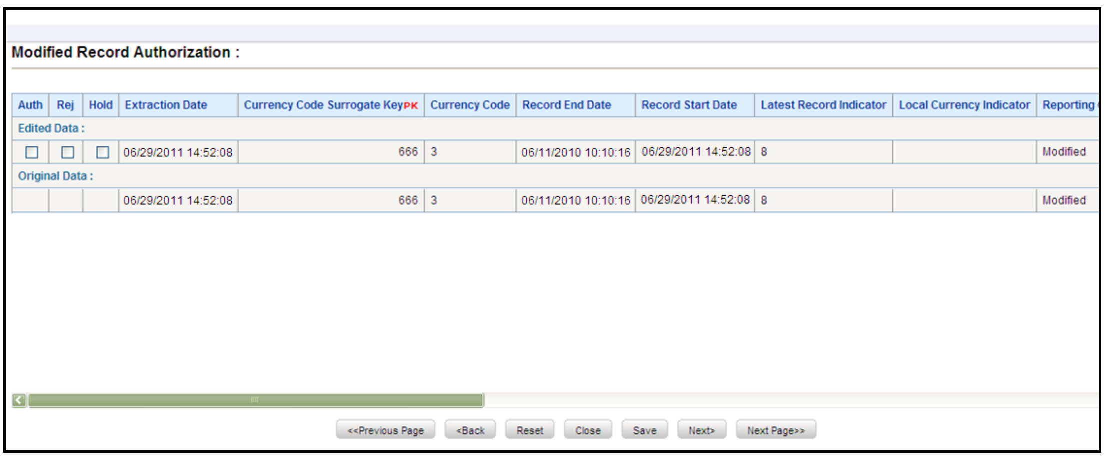
- Click **Authorize All** and click on **Save** to authorize all the records displayed on the current page.
- Click **Reject All** and click on **Save** to reject all the records displayed on the current page.
- Click **Hold All** and click on **Save** to hold all the records displayed on the current page.

If you have enabled the option to send alerts to the Creator of the Form in the *Message Type Maintenance* window, a message is sent indicating that the records are authorized/rejected/put on hold.

4.4.5.1 Re-authorizing Records

You can re-authorize an authorized record that has been updated by other users. When an authorized record is updated, the status flag (AuthFlag) is set to "M" indicating that the record has been modified and needs re-authorization.

Figure 26: DEFQ - Data Entry Re-Authorize window



To re-authorize modified records in the *DEFQ - Data Entry* window:

1. Open the required Form in view mode and click 
.

   The list of available records with the Authorization status is displayed. If there are "no records" for Authorization in the selected Information Domain, an alert message is displayed.

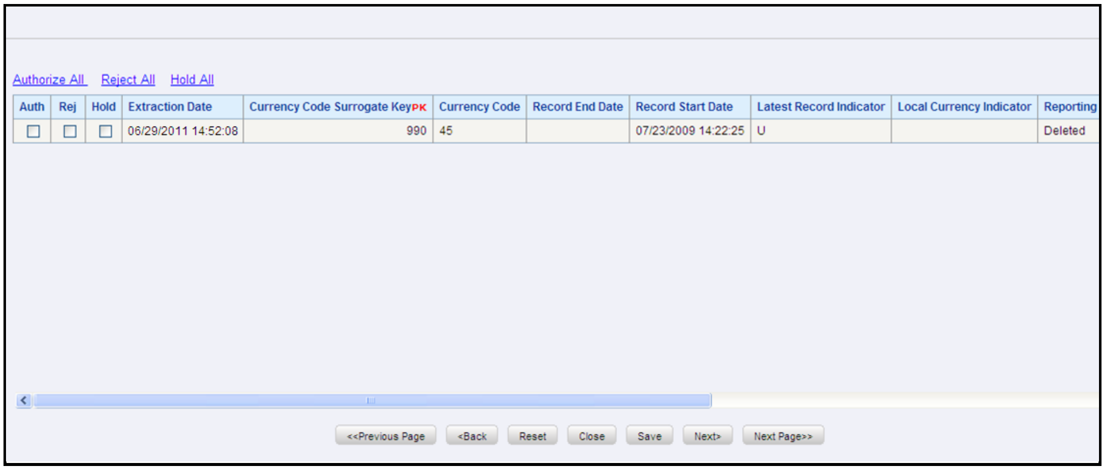
2. Click **Reauthorize Records**. The *DEFQ Authorization* Window is displayed.
3. Select the "Auth" checkbox adjacent to the required record.
4. Click **Save**. On re-authorization, a confirmation message is displayed.

You can also select the checkbox adjacent to “Rej” to reject the record, or “Hold” to re-authorize or reject at a later point. A message is sent to the Form creator indicating that records are authorized/rejected/put on hold.

4.4.5.2 Re-authorizing Deleted Records

You can re-authorize the delete action when an authorized record has been deleted by other users. When an authorized record is deleted, the status flag (AuthFlag) is set to “D” indicating that the record has been deleted and needs re-authorization.

Figure 27: DEFQ - Data Entry Re-Authorize Deleted Records window



To re-authorize deleted records in the *DEFQ - Data Entry* window:

1. Open the required Form in view mode and click 
.

   The list of available records with the Authorization status is displayed. If there are “no records” for Authorization in the selected Information Domain, an alert message is displayed.

2. Click **Reauthorize Deleted Records**. The *DEFQ Authorization* Window is displayed.
3. Select the “Auth” checkbox adjacent to the required record.
4. Click **Save**. On re-authorization, a confirmation message is displayed.

   You can also select the checkbox adjacent to “Rej” to reject the record, or “Hold” to re-authorize or reject at a later point. A message is sent to the Form creator indicating that records are authorized/rejected/put on hold.

4.4.6 Exporting Form Data

You can export the required record(s) to a selected location in CSV format. To Export Form Data in the *DEFQ - Data Entry* window:

1. In the View mode, select the checkbox adjacent to the record(s) which you want to export.
2. Click 
. The File Download dialog is displayed.
3. Click **Save**. The Save As dialog is displayed.
4. Select the location and click **Save**. The selected record is exported.

4.4.7 Copying Form Data

You can copy the existing fields and create new fields in a record. When you copy a field, the primary key values are incremented from the pre-defined value to the next acceptable value. However, the other fields can be modified as required.

To copy fields in the *DEFQ - Data Entry* window:

1. Open the required Form in view mode and click .

   The list of available records is displayed. All the primary field data (indicated by *) is incremented by default.

2. Click **Save**. The field values are added to the record.

   You can click **Edit** to modify the values or click **Next** to copy the next set of fields.

4.4.8 Deleting Form Details

You can remove Form details that are no longer required by deleting them from the *DEFQ - Data Entry* window.

1. In the View mode, select the checkbox adjacent to the record which you want to delete.
2. Click . An information dialog is displayed.
3. Click **OK** to confirm and delete the record.

4.4.9 References

This section of the document consists of information related to intermediate actions that need to be performed while completing a task. The procedures are common to all the sections and are referenced where ever required. You can refer to the following sections based on your need.

4.4.9.1 Creating Tree View Form

The process to create a Form using the Tree View Layout differs from the procedure as explained for other layouts. You can create a Form using the Tree View Layout, by selecting either Dimensional Table Tree or Parent-Child Tree.

4.4.9.2 Dimensional Table Tree

If you want to create a Form using the Dimension table Tree, select **Tree view** > **Dimension Table Tree** option in the *DEFQ - Layout* window. On clicking **Next**, you need to provide the required details in the following windows:

1. **Dimension Table Selection**: Enter the **Root Name** and select the Table. Click **Next**.
2. **Fields Selection**: Select required Fields to Display from Available fields and click **Next**.
3. **Dimension Node Selection**: Select Field Nodes from Available fields and click **Next**.
4. Select Dimensional Tree Nodes for the selected fields and click **Next**.

5. **DEFQ Field Properties** window: Specify the required details. For more information, refer to DEFQ Field Properties.

4.4.9.3 Parent-Child Tree

If you want to create a Form using the Parent-Child Tree, select **Tree view** > **Parent-Child Tree** option in the *DEFQ - Layout* window. On clicking **Next**, you need to provide the required details in the following windows:

1. **Hierarchy Table** Selection: Enter the **Root Name** and select the Table. Click **Next**.
2. **Parent-Child Node Selection**: Select Parent Node, Child Node, and Node Description from the drop-down list.
3. **Fields Selection**: Select required Fields to Display from Available fields and click **Next**.
4. **DEFQ Field Properties window**: Specify the required details. For more information, refer to DEFQ Field Properties.

4.4.9.4 Applying Rules

You can apply rules to Validate Form Data to specific fields such as Text Field, Text Area, or Protected Field. To specify rules for a field in the *DEFQ - Forms Designer DEFQ Field Properties* window:

1. Click **Rule** adjacent to the required field. The *Specifying Rules and Expressions for Data Validations* window is displayed.
2. Select the required Fields, Operators, and Functions from the list.
3. Enter the Rule Expression in the Expression Viewer field.
4. Depending on the data type of the selected field, the following column constraints are displayed. Select the required checkbox.
   - No Spaces
   - Characters Only
   - Alpha Numeric
   - Not Null
   - Non Negative
5. Select the **Alignment** type from the drop-down list.
6. Click **OK** and save the details.

4.4.9.5 Defining List of Values

While creating a Form, if you choose the **Select List** field parameter option in the In Edit/Add column in the *DEFQ Field Properties* window, you need to define the list of values in the *Select List* window. However, you do not need to define the values for foreign key fields and primary key fields.

In the *Select List* Window, select the required Field Type from the following options:

- **Comma Separated Values**: Supports only the user-specified values while creating a Form.
- **Dynamic List of Values**: Supports field name from a table and stores it in the database. The same can be used during Data Entry.

If **Comma Separated Values** is selected:

1. Enter the **List of Values** to be displayed.
2. Specify **Alternate Display Values** to be displayed.
3. Click **OK** and save the specified list of values.

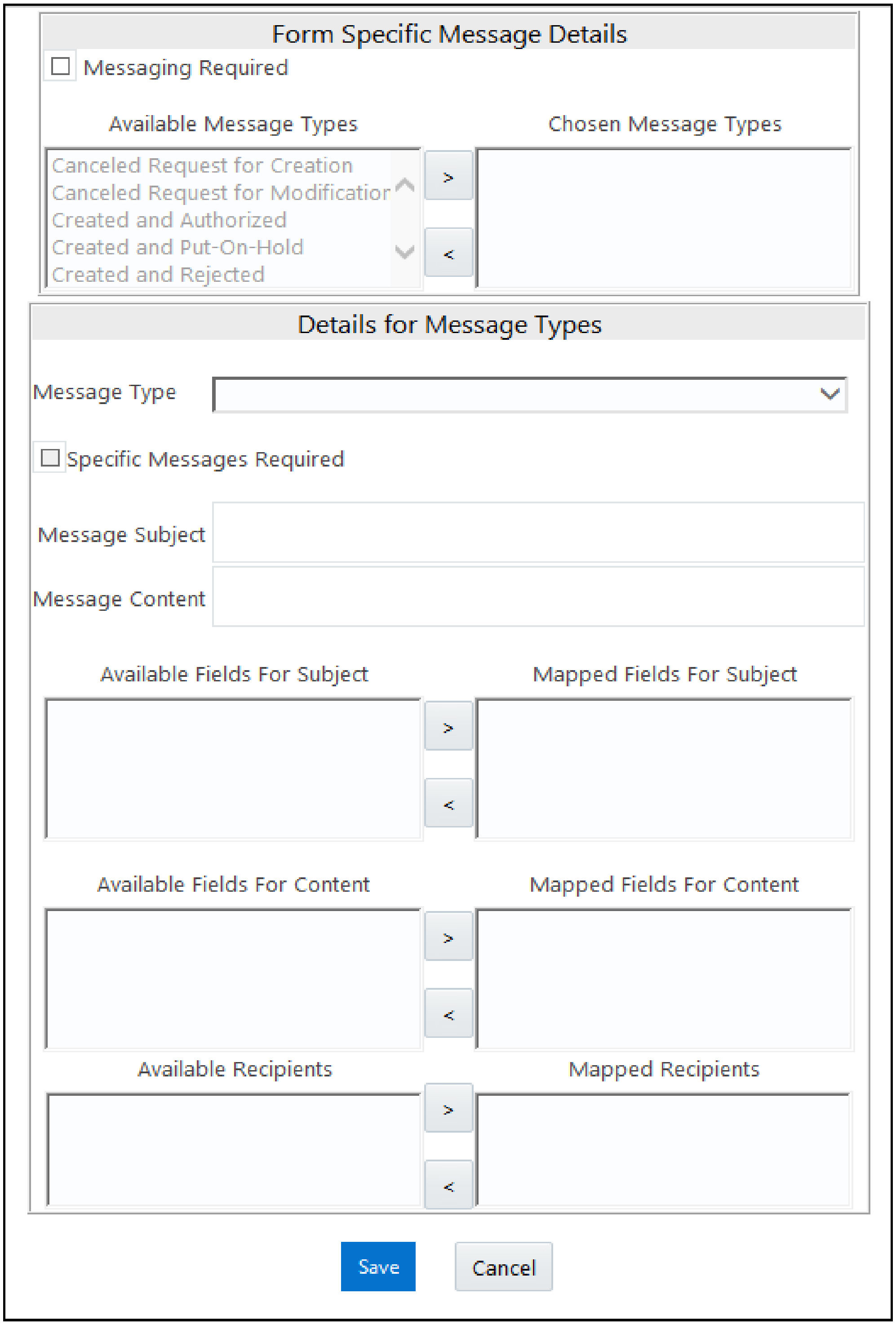
If **Dynamic List of Values** is selected:

4. Select Table Value, List Value, and Display Value field.
5. Select the Field, Operator, and Functions from the list.
6. Define a filter condition for the selected values.
7. Click **OK** and save the specified list of values.

4.4.9.6 Defining Messaging Details

While creating a Form, you can click **Message Details** in the *DEFQ Field Properties* window to define the messaging details. You can specify an alert message which is sent to the Creator of the Form or an Authorizer.

Figure 28: Messaging Details window



In the *Messaging Details for a Form* window:

1. Select **Messaging Required** checkbox to activate the Messenger feature.

NOTE	If the option is not selected, a single mail is sent for the entire batch. Message details such as recipients, subject, and contents are fetched from the metadata

2. Select the required **Available Message Types** from the list and click .
3. Select the **Message Type** from the drop-down list based on a specific action.
4. Select **Specific Messages Required** to add a specific message.
5. Select Available Fields for **Subject**, **Content**, & **Recipients** from the list and click .
6. Click **Save** and save the messaging details. You also need to select **Save with Authorization** in the *DEFQ Field Properties* window for the messages to be functional.

4.4.9.7 Form Data Versioning

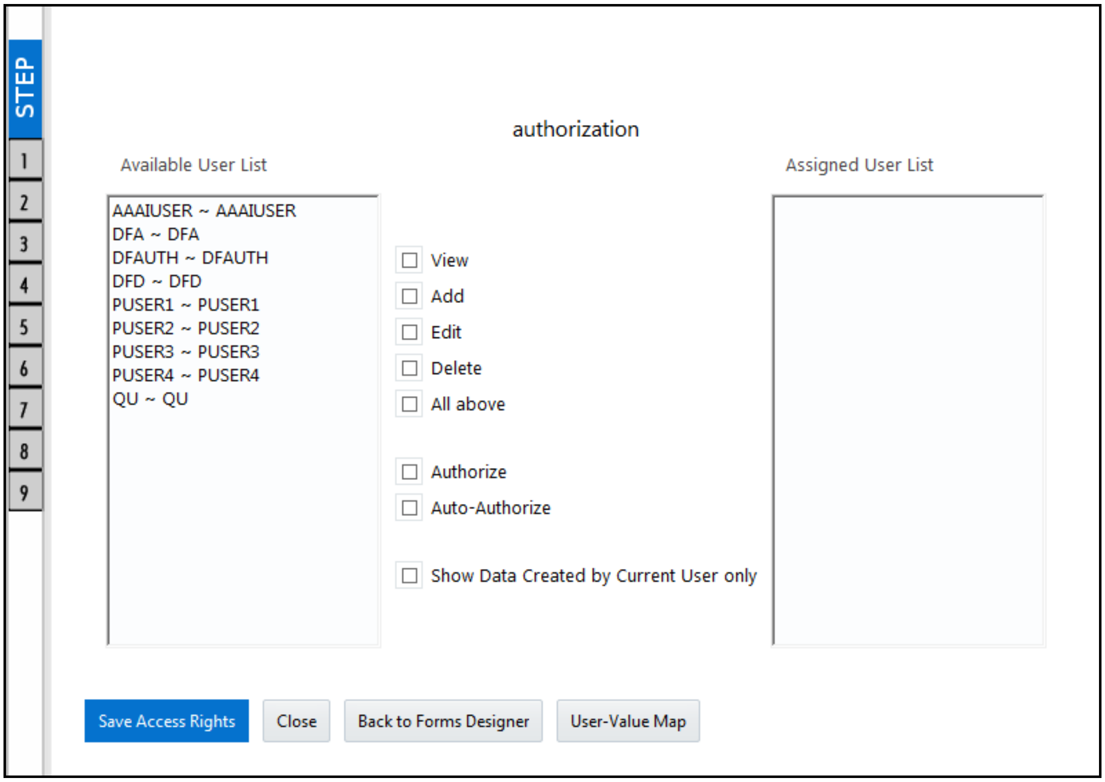
You can perform data versioning on an authorized Form. The modifications made to the particular Form are tracked and displayed as per date versioning. In the *Data Versioning for Form* window, do the following:

1. Select **Enable Data Versioning** checkbox to ensure that the version is tracked.
2. Select the **Table** and **Version Identifier** from the drop-down list.
3. Click **OK** and save the versioning details.

4.4.9.8 Save with Authorization

The **Save with Authorization** feature in Forms Designer (*Sort Fields Selection* window) allows you to authorize the uploaded data. Authorization serves as a checkpoint for the validation of uploaded data.

Figure 29: DEFQ – Data Entry Save Authorization window



To authorize the uploaded data, you need to create a Form in DEFQ with the **Save with Authorization** checkbox selected.

4. Before any DEFQ Form is created to authorize the data, the underlying table in the data model needs to have the below columns added to its table structure. You need to perform a data model upload to have the new structures reflected in the application.

   Columns required:

   ```
   V_MAKER_ID VARCHAR2(20),
   V_CHECKER_ID VARCHAR2(20),
   D_MAKER_DATE DATE,
   D_CHECKER_DATE DATE,
   F_AUTHFLAG VARCHAR2(1),
   V_MAKER_REMARKS VARCHAR2(1000),
   V_CHECKER_REMARKS VARCHAR2(1000)
   ```

5. Navigate to Create a New Form in the Forms Designer section and complete the design steps up to Step 6. From the *DEFQ Field Properties* window explained in step 7, select the appropriate values as listed below for **Store Field As** depending on the columns selected:

   ```
   V_MAKER_ID - MakerID
   V_CHECKER_ID - CheckerID
   D_MAKER_DATE - Maker Date
   D_CHECKER_DATE - Checker Date
   ```

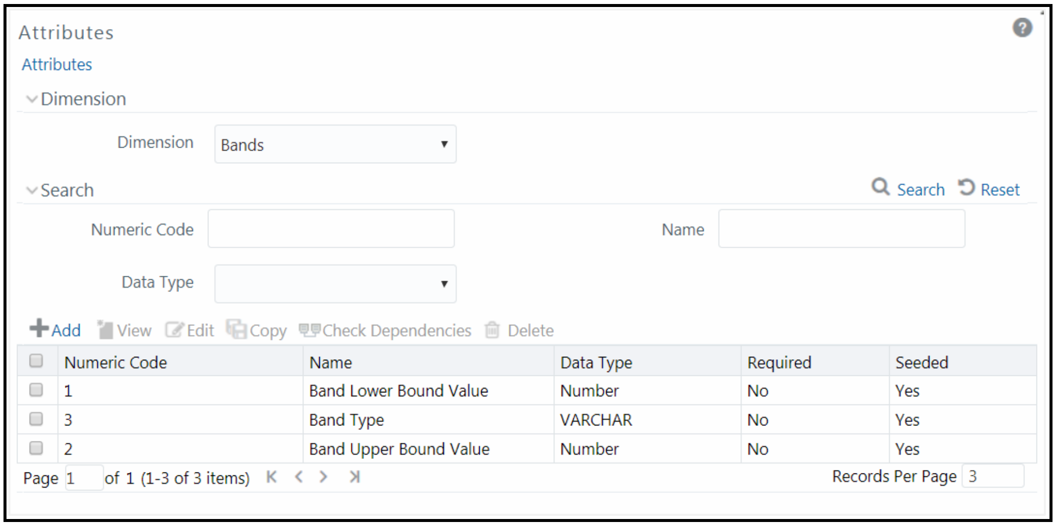
```
F_AUTHFLAG - AuthFlag
V_MAKER_REMARKS - Maker Remarks
V_CHECKER_REMARKS - Checker Remarks
```

6. Click **Save with Authorization**. Once data is loaded into the table, you can log in as 'Authorizer' and navigate to the *Data Entry* window. Select the Form to open and authorize the records loaded.

5 OFSAA Attributes

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

Figure 30: Attributes window



The *Attributes* window displays the list of pre-defined Dimension Attributes with the other details such as the Numeric Code, Name, Data Type, Required, and Seeded. You can search for a specific Attribute based on Numeric Code, Name, or Data Type and view the list of existing definitions within the system.

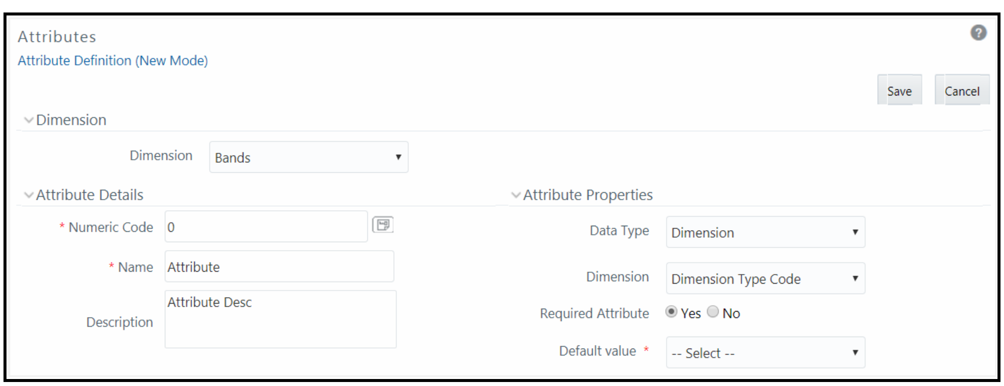
5.1 Adding Attribute Definition

Attributes facilitate you to define the properties or qualifiers for the Dimension members. The Write role should be mapped to your user group, from the *User Group Role Map* window.

To create an Attribute definition in the *Attributes* window:

1. From the *Attributes* window, click ✚ **Add**. The *Attribute Definition (New Mode)* window is displayed.

Figure 31: Attributes window



2. In the Dimension section, select the required dimension from the drop-down list.
3. Click the button in the **Numeric Code** field. A unique code is auto-generated.

   You can also manually enter the code in the **Numeric Code** field.
4. Enter the **Name** and required **Description** for the Attribute.

NOTE	Name: The characters & ’ ” are restricted in the name field. Description: The characters ~&+' "@ are restricted in the description field.

5. Enter the Attribute window is as tabulated:

The following table describes the fields in the Attribute window.

Table 7: Fields in the Attributes window and their Descriptions

Field	Description
Fields marked in red asterisk (*) are mandatory.	
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 the default value. Enter a Scale value >= 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 <= the scale defined for NUMBER_ASSIGN_VALUE in the dimension's underlying attribute table. See the OFS Data Model Utilities Guide for further details on the attribute table.
Required Attribute	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.

Field	Description
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. 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 the button to select a valid date as the Default Value from the calendar. If **STRING** is selected as the **Data Type**: Enter an alphanumeric value in the **Default Value** field. The Maximum number of characters allowed in the Default value field for String Data Type is 1000.

6. Click **Save**. The entries are validated and the defined Attribute is captured.

5.1.1 Viewing Attribute Definition

You can view individual Attribute Definition details at any given point. The Read-only role should be mapped to your user group.

To view the existing Attribute Definition details in the *Attribute* window:

1. Select the checkbox adjacent to the Numeric Code of the Attribute, whose details are to be viewed.
2. Click **View** button in the Dimension Attributes tool bar.

   The *View – Attributes* window is displayed with the details such as Dimension, Numeric Code, Name, Description, and Attribute Properties.

5.1.2 Modifying Attribute Definition

You can modify the Name, Description, or Default Value fields of an attribute definition. The Write role should be mapped to your user group.

To modify an existing Attribute Definition in the *Attributes* window:

1. Select the checkbox adjacent to the Numeric Code of the Attribute, whose details are to be updated.

2. Click **Edit** button in the Dimension Attribute tool bar. The **Edit** button is disabled if you have selected multiple Attributes. The *Edit - Attributes* window is displayed.

3. Edit the Attribute details such as Name, Description, or Default value. For more information, see Add Attribute Definition.

4. Click **Save** to save the changes.

5.1.2.1 Copying Attribute Definition

The Copy Attribute Definition facilitates you to quickly create a new Attribute Definition based on the existing attributes or by updating the values of the required attributes. The Write role should be mapped to your user group.

To copy an existing Attribute Definition in the *Attributes* window:

1. Select the checkbox adjacent to the Numeric Code of the Attribute, whose details are to be duplicated.

2. Click **Copy** button in the Dimension Attributes toolbar to copy a selected Attribute definition. **Copy** button is disabled if you have selected multiple Attributes.

3. In the *Copy – Attributes* window you can:

   - Create a new attribute definition with existing variables. Specify new **Numeric Code** and **Attribute Name**. Click **Save**.

   - Create a new attribute definition by updating the required variables. Specify new **Numeric Code** and **Attribute Name**. Update the required details. For more information, see Add Attribute Definition. Click **Save**.

     The new attribute definition details are displayed in the *Attributes* window.

5.1.3 Attribute Definition Dependencies

You can view the dependencies of Attributes. The Read-only role should be mapped to your user group.

To view the dependency of an attribute in the *Attributes* window:

1. Select the checkbox adjacent to the Numeric Code of the Attribute whose dependency is to be checked.

2. Click the button in the Dimension Attributes toolbar. The **Check Dependencies** button is disabled if you have selected multiple attributes. The *Attributes Dependency Information* window is displayed with the dependency details.

5.1.4 Deleting Attribute Definition

You can remove the Attribute Definitions which are not required in the system by deleting them from the *Attributes* window. The Write role should be mapped to your user group.

1. Select the checkbox adjacent to the Numeric Code(s) of the Attributes whose details are to be removed.

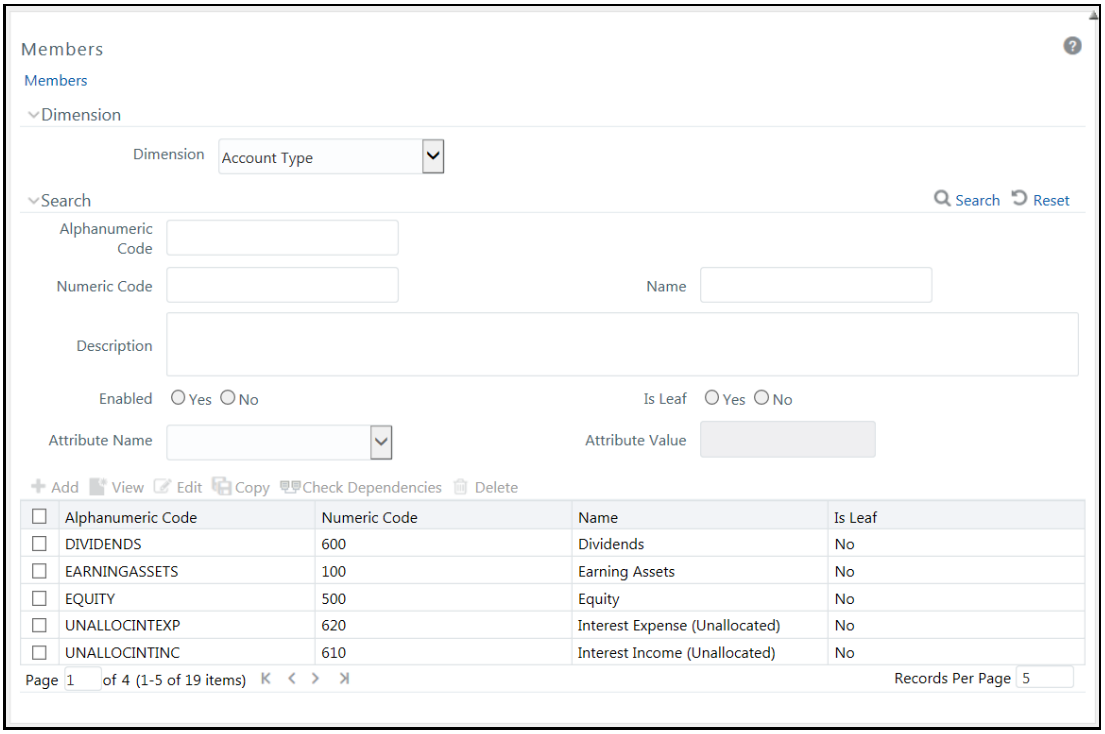
2. Click **Delete** button in the Dimension Attributes tool bar.
3. Click **OK** in the information dialog to confirm the deletion.

6 OFSAA Dimension Members

Dimension Members refer to the individual items that constitute a dimension when data is categorized into a single object. Example, Product, Organization, Time, and so on. Members are available within the Dimension Management section of the Infrastructure system.

For more information on how to set up alphanumeric and numeric codes, see Configurations to use Alphanumeric and Numeric Codes for Dimension Members section in OFSAAI Administration Guide.

Figure 32: Members window



The *Members* window displays the list of pre-defined Dimension Members with the other details such as the Alphanumeric Code, Numeric Code, Name, and Is Leaf. You can also search for a specific Member based on Alphanumeric / Numeric Code (irrespective of whether the dimension is configured to be numeric or alphanumeric), Name, Description, Enabled status, Is Leaf status, Attribute Name, or Attribute Value and view the list of existing definitions within the system.

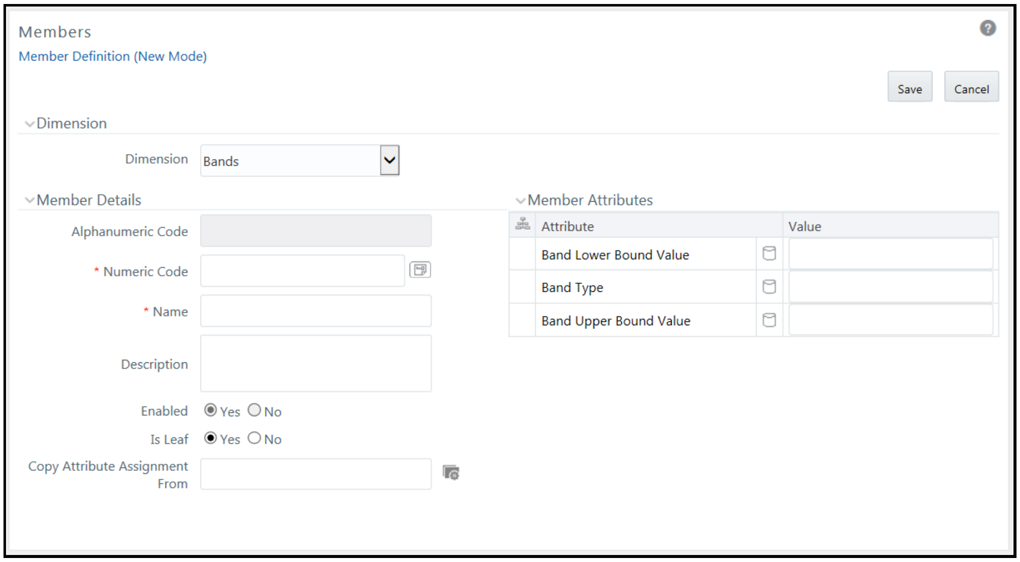
6.1 Adding Member Definition

This option allows you to add member definitions. The Write role should be mapped to your user group.

To create an Attribute definition in the *Attributes* window:

1. Click **Add** from the toolbar. The *Member Definition (New Mode)* window is displayed.

Figure 33: Members Add window



2. In the Dimensions section, select the required **Dimension** from the drop-down list.
3. Enter the Member Details as tabulated:

The following table describes the fields in the Member Add window.

Table 8: Fields in the Members Add window Field and their Descriptions

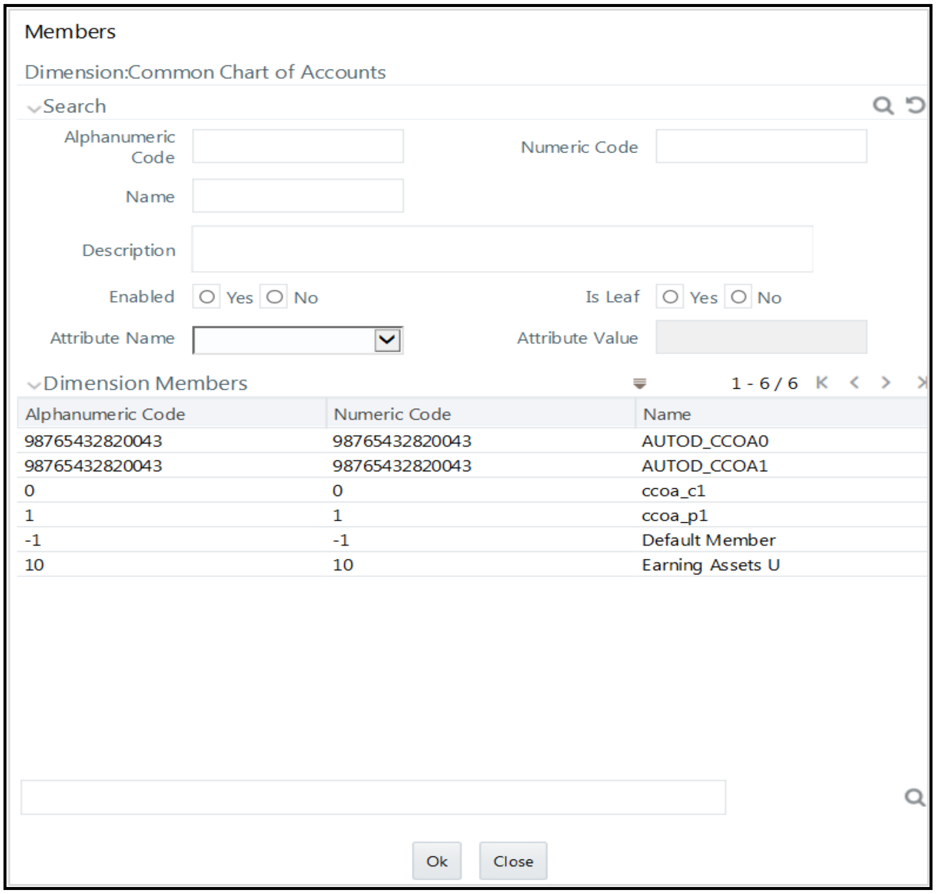
Field	Description
Fields marked in red asterisk (*) are mandatory.	
Alphanumeric Code	The **Alphanumeric Code** field is editable only if the selected Dimension accepts Alphanumeric Code. For example, **Billing Method** Dimension. Else, the field is Read Only and the value is fetched from the **Numeric 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 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.

Field	Description
Enabled	This field is set to **Yes** by default and is editable only in the *Edit* window. **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 the 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.

NOTE If the Dimension is selected as "Common Chart of Accounts", proceed further. Else, jump to step 5.

4. Click the button in **Copy Attribute Assignment From** field. The *Member Browser Properties* window is displayed. This field can be left blank so that the Member Attributes panel can be filled in without considering the values already assigned.

Figure 34:Member Browser Properties window



- Select the required Member from the Dimension Members list.

  Click button in the Search grid to search for a specific Member based on Alphanumeric Code, Numeric Code, Name, Description, Enabled status, Is Leaf status, Attribute Name, or Attribute Value. You can also click button to find a member present in the Dimension Members grid using keywords.

- Click **OK**. The selected Member is displayed in the **Copy Attribute Assignment From** field in the *New – Member Details* window and the details of the selected Attribute are displayed in the Member Attributes section. You can edit the Attribute details as indicated:

  Edit Attribute based on a date by clicking the (Calendar) icon.

  Edit Attribute based on a Dimension Value by selecting from the drop-down list.

  Edit Attribute based on a Number Value by entering the valid numerical value.

  Edit Attribute based on a String Value by specifying an alphanumerical value.

5. Click **Save** and the defined Member Definition is captured after validating the entries.

6.1.1 Viewing Member Definition

You can view individual Member Definition details at any given point. To view the existing Member Definition details in the *Members* window:

1. Select the checkbox adjacent to the Alphanumeric Code of the Member, whose details are to be viewed.

2. Click **View** button in the toolbar.

   The *View – Member Details* window is displayed with the details such as Dimension, Member Details, and Member Attributes details.

6.1.2 Modifying Member Definition

To modify an existing Member Definition in the *Members* window:

1. Select the checkbox adjacent to the Alphanumeric Code of the Member, whose details are to be updated.

2. Click **Edit** button in the toolbar. The **Edit** button is disabled if you have selected multiple Members. The *Edit – Member Details* window is displayed.

3. Edit the Member details as required. For more information, see Add Member Definition.

4. Click **Save** to save the changes.

6.1.3 Copying Member Definition

The Copy Member Definition facilitates you to quickly create a new Member Definition based on the existing attributes or by updating the values of the required members.

To copy an existing Member Definition in the *Members* window:

1. Select the checkbox adjacent to the Alphanumeric Code of the Member, whose details are to be duplicated.

2. Click the **Copy** button in the toolbar to copy a selected Member definition. **Copy** button is disabled if you have selected multiple Members.

3. In the *Copy – Member Details* window you can:

   - Create a new Member with existing variables. Specify the **Numeric Code** and new **Member Name**.

   - Create a new Member definition by updating the required variables. Specify the **Numeric Code** and new **Member Name**. Update the required details. For more information, see Add Member Definition. Click **Save**.

     The new member definition details are displayed in the *Members* window.

6.1.4 Member Definition Dependencies

You can view the dependencies of Members. To view the dependency of member in the *Members* window:

1. Select the checkbox adjacent to the Alphanumeric Code of the Member, whose dependency is to be viewed.

2. Click the **Check Dependencies** button in the toolbar. The **Check Dependencies** button is disabled if you have selected multiple members. The *Members Dependency Information* window is displayed with the dependency details.

6.1.5 Deleting Member Definition

You cannot delete predefined members or the members that are the Nodes for a hierarchy.

To delete a Member in the *Members* window.

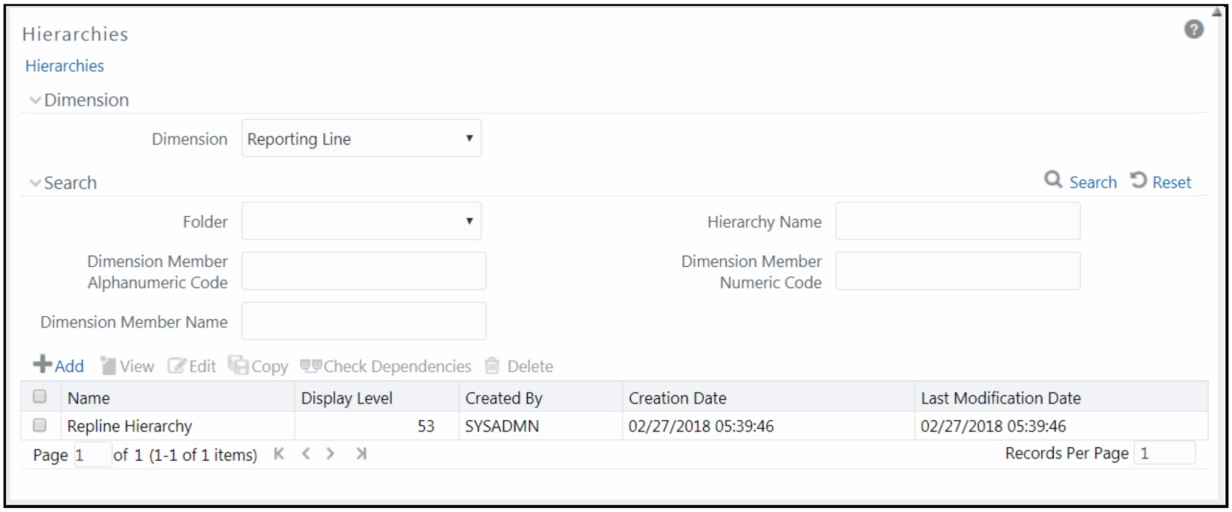
1. Select the checkbox adjacent to the Alphanumeric Code(s) of the Members, whose details are to be removed.
2. Click **Delete** button in the Dimension Members tool bar.
3. Click **OK** in the information dialog to confirm the deletion.

7 OFSAA Hierarchies

Hierarchies refer 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 are available within the Dimension Management section of the Infrastructure system.

You can access the *Hierarchies* window by expanding **Unified Analytical Metadata** and **Dimension Management** within the tree structure of the LHS menu and selecting **Hierarchy Maintenance**.

Figure 35: Hierarchies window



The *Hierarchies* window displays the list of Hierarchies created in all public folders, shared folders to which you are mapped, and private folders for which you are the owner, along with other details such as the Name, Display level, Created By, Creation Date, and Last Modification Date. For more information on how object access is restricted, see the *Object Security in the AMHM module* section in the OFS Analytical Applications Infrastructure User Guide.

You can also search for a specific Hierarchy definition based on Folder, Hierarchy Name, Dimension Member Alphanumeric Code, Dimension Member Numeric Code, or Dimension Member Name and view the existing definitions within the system.

7.1 Adding Hierarchy Definition

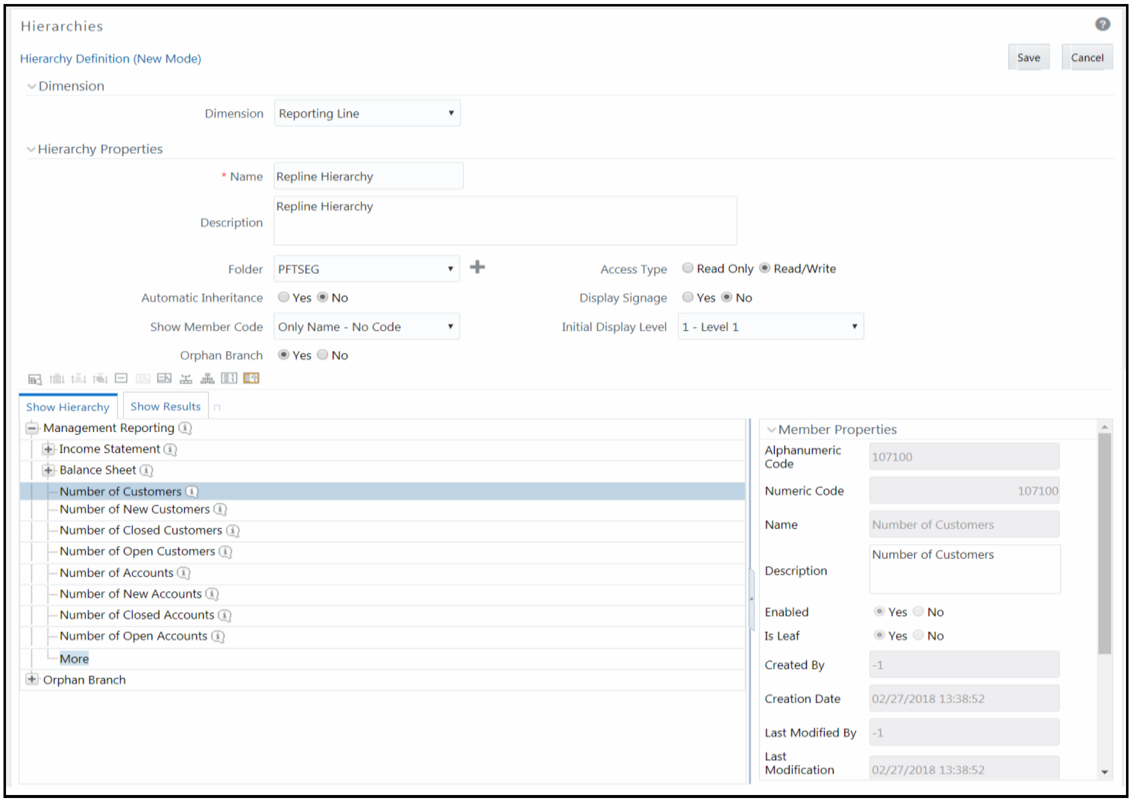
In the *Hierarchies* window, you can create Hierarchy Definition up to 15 levels by default. The maximum permissible levels are up to 58 Hierarchies. To create a hierarchy, the Write role should be mapped to your user group.

NOTE	When an AMHM hierarchy is created, implicitly a UAM Business hierarchy also gets created and will be listed in the *Summary* window of Business Hierarchy.

To create a Hierarchy definition in the *Hierarchies* window:

1. Click + **Add** button in the Hierarchies toolbar. The *New – Hierarchy Details* window is displayed.

Figure 36: Hierarchies window



2. Select **Dimension** from the drop-down list. The selected Dimension from the *New – Hierarchy Details* window is displayed as the default dimension for which the member has to be defined.

The following table describes the fields in the Hierarchy Properties window.

Table 9: Fields in the Hierarchies window and their Description

Field	Description
Fields marked in red asterisk (*) are mandatory.	
Name	Enter the **Name** of the Hierarchy. **Note**: The characters &' " are restricted.
Description	Enter the required **Description** for the Hierarchy. **Note**: The characters ~&+' " @ are restricted.
Folder	Select the folder where the hierarchy is to be stored from the drop-down list. The *Folder selector* window behavior is explained in the User Scope section see *Appendix A* in the OFS Analytical Applications Infrastructure User Guide. Click to create a new private folder. The *Segment Maintenance* window is displayed. For more information, see the *Segment Maintenance* section in the OFS Analytical Applications Infrastructure User Guide. **Note**: You can select **Segment/Folder Type** as Private and the **Owner Code** as your user code only.

Field	Description
Access Type	Select the Access Type as **Read Only** or **Read/Write**. **Read-Only**: Select this option to give other users access to only view the hierarchy definition. **Note**: A user with Phantom and Write role can modify or delete the hierarchy even though the access type is selected as Read-only. **Read/Write**: Select this option to give all users access to view, modify (including Access Type), and delete the hierarchy definition.
Automatic Inheritance	Click **Yes** to inherit the hierarchy properties of the parent to the child. Click **No** if you want to define a new hierarchy.
Display Signage	Click **Yes** to display the Signage to the right-hand side of the member in the Show hierarchy panel. Else, click **No**.
Show Member Code	Select from the drop-down list as one of the following: **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.
Initial Display Level	Select the Initial Display level from the drop-down list.
Orphan Branch	Click **Yes** to display the Orphan Branch in the Show Hierarchy panel. Else, click **No**.

3. To add Child under the **Show Hierarchy** tab:
   a. Right-click in the Show Hierarchy tab.
   b. Select **Add Child** option and the *Add Member* window is displayed.
   c. Select the required Member and click . The Member is displayed in the **Selected Members** panel. Click to select all Members which are shown in the Show Members pane. Click to select all nodes/ members in the server.

      You can click to deselect a Member or click to deselect all the Members.

      You can click to search for the required member using Alphanumeric code, Numeric Code, Name, Description, Attribute Name, or Attribute Value.

      You can also click button to toggle the display of Numeric Code left, right, or name and click button to display Alphanumeric Code left, right, or name.
   d. Click **OK**. The selected Member is displayed as Child under the **Show Hierarchy** panel in the *New – Hierarchy Details* window.

4. To add Sibling:
   a. Right-click on the Child and select the option **Add Sibling**. The *Add Member* window is displayed.
   b. Select the required Member and click . The Member is displayed in the **Selected Members** panel. You can click to select all Members which are shown in the Show Members pane. Click to select all nodes/ members in the server.
   c. You can click to deselect a Member or click to deselect all the Members. You can also Click to search for the required member.
   d. Click **Apply**. The selected Member is displayed as **Sibling** below the **Parent** under the Show Hierarchy panel in the *New – Hierarchy Details* window.
5. To add Leaf under a Parent, Child, or Sibling:
   a. Right-click the Parent or Child and select **Add Leaf**. The *Add Member* window is displayed.
   b. Select the required Member and click . The Member is displayed in the **Selected Members** panel. You can click to select all Members which are shown in the Show Members pane. Click to select all nodes/ members in the server.

      You can click to deselect a Member or click to deselect all the Members. You can also Click to search for the required member.
   c. Click **Apply**. The selected Member is displayed as Leaf below the Parent or Sibling under the **Show Hierarchy** panel in the *New – Hierarchy Details* window.
6. To define Level Properties:
   a. Select **Level Properties** from the options under Parent, Child, Sibling, or Leaf, and the *Level Properties* window is displayed.
   b. Enter the valid **Name** and **Description** in the respective fields.
   c. Click **OK** and the Levels defined are displayed in the drop-down in the **Initial Level Display** field in the **Hierarchy Properties** grid in the *New – Hierarchy Details* window.
7. To cut and paste Child or Sibling:
   a. Right-click on any node and select **Cut**.
   b. Right-click on any node and **Paste as Child** or **Paste as Sibling**.
8. To **Delete** and Undelete:
   a. Right-click on the node to be deleted and select **Delete Node**.

      The node deleted is stroked out.
   b. Right-click and select **UnDelete** to cancel deletion of the node.
9. To add Child / Sibling / leaf:
   a. Right-click on any node and select **Create and add Child**. The *New - Member Details* window is displayed. For more information, see Add Member Definition.

b. Right-click on any node and select **Create and add Sibling**.

c. Right-click on any node and select **Create and add leaf**.

10. To view the Member Properties and Member Attributes of a node in the Show Hierarchy panel:

a. Click **<** button and the Member Property grid is displayed.

b. Click on a Member. The properties such as Alphanumeric code, Numeric Code, Name, Description, Enabled, Is Leaf, Created By, Creation Date, Last Modified By, Last Modification Date, Attribute, and Value of the selected Member are displayed in the Member Properties and Member Attributes grids.

In the *Hierarchies* window you can also:

— Click to collapse the members under a node.

— Click or to expand a branch or collapse a branch.

— Click or to focus or defocus a selected node except the root node.

— Click or to view the name of members right or left.

— Click or to view the Numeric code values of members right or left.

— Click or to show code or show the name of the members.

— Click button to view the Advanced Properties of the nodes.

11. Click **Save** in the New – *Hierarchy Details* window to validate and capture the entries.

The Audit Trail section at the bottom of the window displays the metadata about the Hierarchy with the option to add additional information as comments. The User Comments section facilitates you to add or update additional information as comments.

7.1.1 Viewing Hierarchy Definition

You can view individual Hierarchy Definition details at any given point. To view the existing hierarchy Definition details in the *Hierarchies* window:

1. Select the checkbox adjacent to the Hierarchy Name.

2. Click **View** button in the Hierarchies tool bar. The **View** button is disabled if you have selected multiple Hierarchies.

The *View – Hierarchy Details* window is displayed with all the hierarchy details.

In the *View – Hierarchy Details* window you can click button to search for a member using the Alphanumeric Code, Numeric Code, or Member Name in the Search dialog.

NOTE	The search functionality of this button will not return any values if you search for a node in the Orphan Branch of the hierarchy.

7.1.2 Modifying Hierarchy Definition

You can modify the Name, Description, Folder, Access Type, Automatic inheritance, Display Signage, Show Member Code, Initial Display level, Orphan branch, Show hierarchy details in *Edit – Hierarchy Details* window.

> **NOTE** When you modify a Hierarchy, the implicitly created UAM Business Hierarchy will also get updated.

1. Select the checkbox adjacent to the Hierarchy Name whose details are to be updated.
2. Click **Edit** button in the Hierarchies tool bar. The **Edit** button is disabled if you have selected multiple Members. The *Edit – Hierarchy Details* window is displayed.

   In the *Edit – Hierarchy Details* window you can click button to search for a member using the Alphanumeric Code, Numeric Code, or Member Name in the Search dialog. Edit the Hierarchy details as required. For more information, see Add Hierarchy Definition.
3. Click **Save** and save the changes.

7.1.3 Copying Hierarchy Definition

The Copy Hierarchy Definition facilitates you to quickly create a new Hierarchy Definition based on the existing attributes or by updating the values of the required hierarchies.

To copy an existing Hierarchy Definition in the *Hierarchies* window:

1. Select the checkbox adjacent to the Hierarchy name whose details are to be duplicated.
2. Click **Copy** button in the Hierarchies toolbar to copy a selected Hierarchy definition. **Copy** button is disabled if you have selected multiple Hierarchies. The *Copy – Hierarchy Details* window is displayed.

   In the *Copy – Hierarchy Details* window you can click button to search for a member using the Alphanumeric Code, Numeric Code, or Member Name in the Search dialog.
3. In the *Copy – Hierarchy Details* window you can:
   - Create a new hierarchy definition with existing variables. Specify a new **Hierarchy Name**. Click **Save**.
   - Create a new hierarchy definition by updating the required variables. Specify a new Hierarchy Name and update the required details. For more information, see Add Hierarchy Definition. Click **Save**.

   The new Hierarchy definition details are displayed in the *Hierarchy* window.

7.1.4 Hierarchy Definition Dependencies

You can view the dependencies of Hierarchies. To view the dependency of hierarchy in the *Hierarchies* window:

1. Select the checkbox adjacent to the Hierarchy Name.

2. Click the button in the Hierarchies toolbar. The **Check Dependencies** button is disabled if you have selected Hierarchy definitions. The *Hierarchies Dependency Information* window is displayed.

7.1.5 Deleting Hierarchy Definition

You can remove the Hierarchy Definitions which are not required in the system by deleting them from the *Hierarchy* window.

> **NOTE** When you delete an AMHM Hierarchy, the implicitly created UAM Business Hierarchy will also get deleted, if it is not used in higher objects.

1. Select the checkbox adjacent to Hierarchy Name(s) whose details are to be removed.
2. Click **Delete** button in the Hierarchies tool bar.
3. Click **OK** in the information dialog to confirm the deletion.

To delete an existing Business Hierarchy in the *Business Hierarchy* window:

4. Select the checkbox adjacent to the required Business Hierarchy code.
5. Click the button from the Business Hierarchy tool bar. A confirmation dialog is displayed.
6. Click **OK**. The Business Hierarchy details are marked for delete authorization.

8 OFSAA Filters

Filters in the Infrastructure system allow you to filter metadata using the defined expressions.

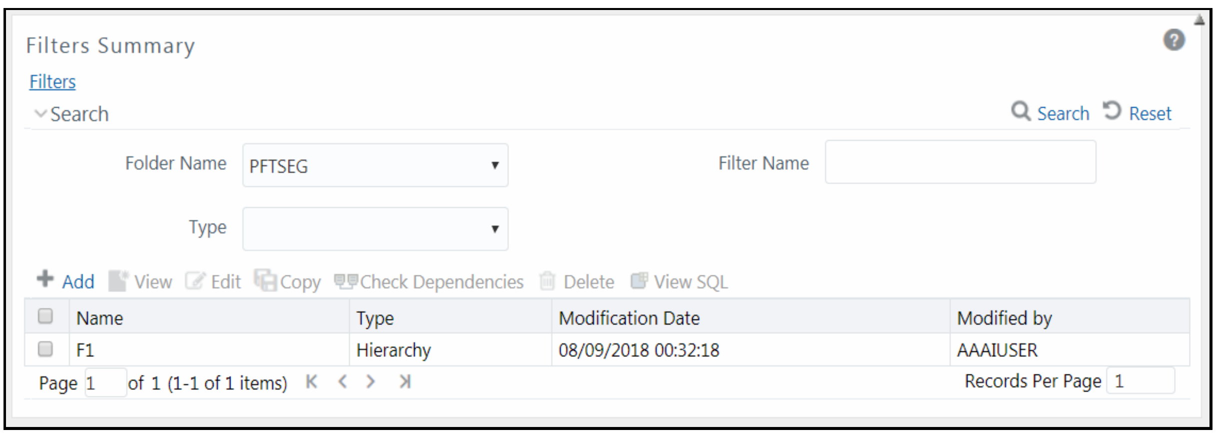
8.1 Navigating to Filters

You can access Filters by expanding the **United Analytical Metadata** section within the tree structure of the LHS menu and selecting **Filter**.

Based on the role that you are mapped to, you can access read, modify or authorize the *Filters* window. For all the roles and descriptions, see *Appendix A* in the OFS Analytical Applications Infrastructure User Guide. The roles mapped to Filters are as follows:

- Filter Access
- Filter Advanced
- Filter Authorize
- Filter Phantom
- Filter Read Only
- Filter Write

Figure 37: Filter Summary window



The *Filters Summary* window displays the list of Filters created in all public folders, shared folders to which you are mapped, and private folders for which you are the owner, along with the other details such as the Name, Type, Modification Date, and Modified By. For more information on how object access is restricted, see the Object Security in Dimension Management module section in the OFS Analytical Applications Infrastructure User Guide.

You can also search for a specific Filter definition based on Folder Name, Filter Name, or Type and view the list of existing definitions within the system. If you have selected Hierarchy from the Type drop-down list, the Dimension drop-down list is also displayed.

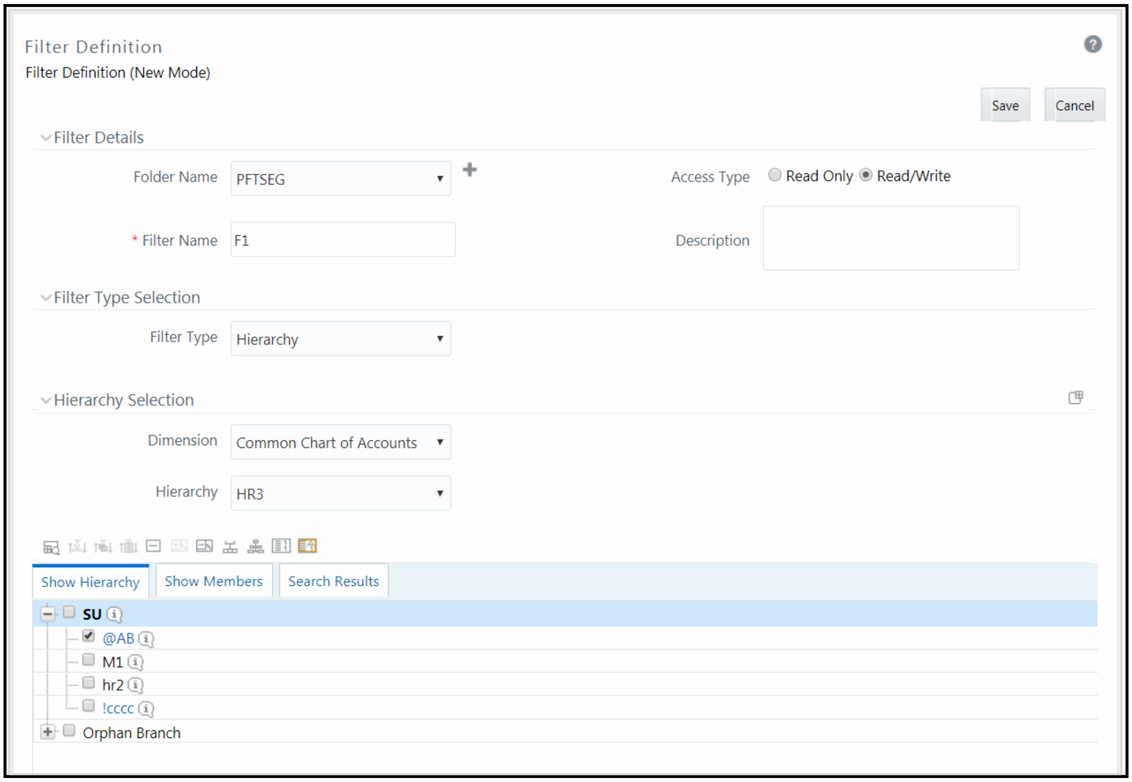
8.2 Adding Filter Definition

This option allows you to add a filter. A Filter can be of 4 types namely, Data Element, Hierarchy, Group, and Attribute. To create a filter definition, the Write role should be mapped to your user group.

To create a new filter from the *Filters Summary* window:

1. Click **Add** button in the Filters toolbar. The *Filter Definition* window is displayed.

Figure 38: Filter Definition New window



2. Enter the Filter Details section details as tabulated:

The following table describes the fields in the Filter Definition window.

Table 10: Fields in the Filter Definition window and their Description

Field	Description
Filter Details	
Folder Name	Select the Folder Name where the Filter is to be stored from the drop-down list. The *Folder selector* window behavior is explained in the *User Scope* section in the OFS Analytical Applications Infrastructure User Guide. Click to create a new private folder. The *Segment Maintenance* window is displayed. For more information, see the *Segment Maintenance* section in the OFS Analytical Applications Infrastructure User Guide. **Note**: You can select **Segment/Folder Type** as Private and the **Owner Code** as your user code only.

Access Type	Select the Access Type as **Read Only** or **Read/Write.** **Read-Only**: Select this option to give other users access to the 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 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.

3. From the Filter Type Selection pane, select the **Filter Type** from the drop-down list.

   There are four different Filter Types available in the Filter Type Selection grid as tabulated. Click the links to navigate to the appropriate sections.

The following table describes the fields in the Filter Type pane.

Table 11: Fields in the Filter Type pane and their Description

Filter	Description
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. **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 (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. **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.

Filter	Description
Group	**Group Filters** can be used to combine multiple Data Element Filters with a logical "AND". **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	**Attribute Filters** are created using defined Attributes. Attribute filters facilitate you to filter on one or more Dimension Type Attributes. For each attribute, you can select one or more values. **Example**: 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 (example. Mortgages, Ledger, and so on.), which meet the specified member attribute criteria.

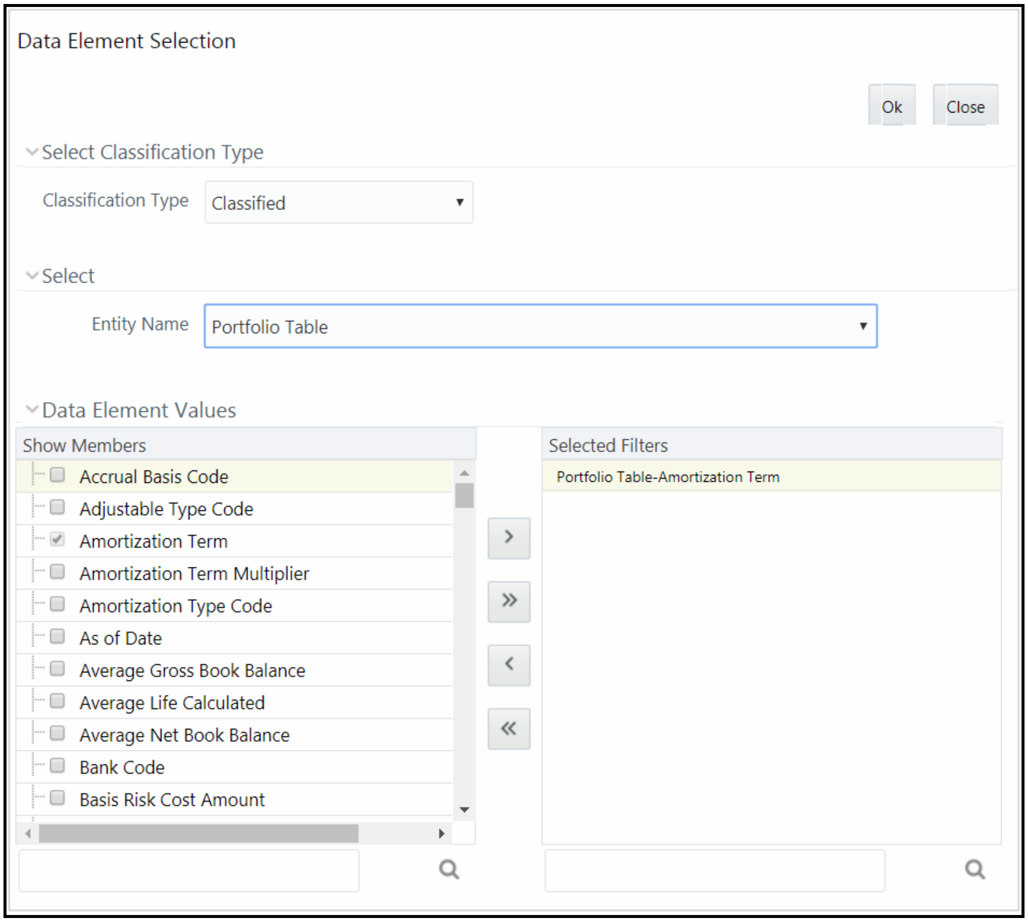
Once the required filter conditions are defined, save the Filter definition.

8.2.1 Define Data Element Filter

When you have selected the Filter Type as Data Element, define the Filter conditions by doing the following in the Data Element Selection section:

1. In the Data Element Selection section, click the  button. *The Data Element Selection* window is displayed.

Figure 39: Data Element Selection window



- 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.
- Select the required database table from the **Entity Name** drop-down list. The associated members are displayed in the Show Members section.
- Select the required member and click . The member is listed in the Selected Members panel. Click to move all Members.

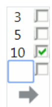
  You can click to deselect a Member or click to deselect all Members.
- Click **OK**. The selected Data Elements are displayed in the **Data Element Selection** field.

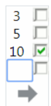
2. Select the **Filter Method** from the drop-down list.

For each column you wish to include in your Data Filter definition, you must specify one of the following Filter Method:

The following table describes the fields in the Data Filter Definition.

Table 12: Fields in the Data Filter Definition window and their Description

Filter	Description
Specific Values	**Specific Values** are used to match a selected database column to a specific value or values that you provide. You may either include or exclude Specific Values. You can add additional values by clicking the **Add** button. Click adjacent to **Add** button to add 3, 5, 10 rows by selecting the checkbox adjacent to 3, 5, or 10 respectively. You can add a custom number of rows by specifying the number in the text box provided, as shown, and click .  To remove a row, select the checkbox and click **Delete**. 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. Select **Include Values** or **Exclude Values** to include or exclude the selected values.

Filter	Description
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 can add additional values by clicking the **Add** button. Click adjacent to **Add** button to add 3, 5, 10 rows by selecting the checkbox adjacent to 3, 5, or 10 respectively. You can add custom number of rows by specifying the number in the text box provided, as shown and click .  To remove a row, select the checkbox and click **Delete**. 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 to select the numeric member ids. If OFSA Datatype is CODE, provide either numeric values or click 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
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). 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 OFSAAI Expression rule. You may any of the following operators when choosing the Expression Filter Method: =, <> (meaning "not equal to"), <, >, <=, or >=.

- 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 or buttons to move a selected Filter Condition up or down.
- Click the button to delete selected individual Filter Conditions records.

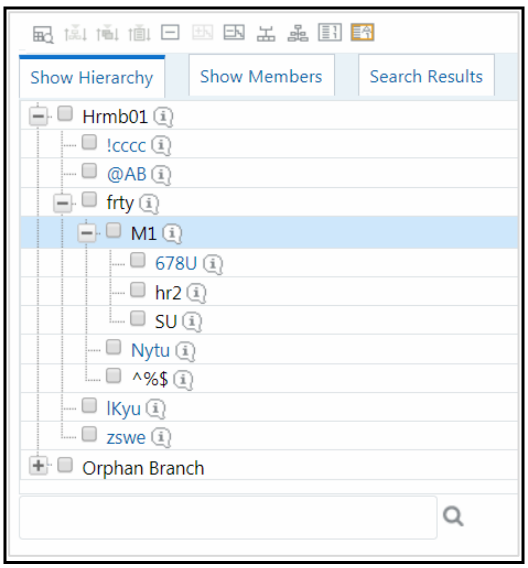
3. Click **Add** or **Edit** in the *Filter Definition* window if you are creating a new or updating an existing Filter definition.
4. Click **Save** to validate the entries and save the filter details.

8.2.2 Define Hierarchy Filter

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.
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.

Figure 40: Show Hierarchy tab



The Show Hierarchy tab displays the leaves in each node in ascending order of Members.

To sort the nodes alphabetically, HIERARCHY_IN_FILTER_SORT-$INFODOM$-$DIMENSION_ID$=$VALUE$ in the AMHMConfig.properties file present in the deployed location should be set as Y. You should add such entry for all the required Dimension IDs for the sort functionality to work for those dimensions.

For example:
HIERARCHY_IN_FILTER_SORT-OFSAAINFO-4345=Y

Restart servers after making any change in AMHMConfig.properties file for the change to take effect.

From this pane, you can:

- Click button to search for a hierarchy member using Dimension Member Alphanumeric Code, Dimension Member Numeric Code, Dimension Member Name, or Attribute and by keying in Matching Values in the Search dialog. The search results are also displayed in the ascending order of Member Names.

- Click to collapse the members under a node.
- Click or to expand a branch or collapse a branch.
- Click or to view the name of members right or left.
- Click or to view the Numeric code values of members right or left.
- Click or to show code or show the name of the members.
- Click or to focus or defocus a selected node except the root node.

  You can also click the button to find a member present in the nodes list using keywords. For a large tree (nodes>5000), this search will not return any value if the tree is not expanded.

4. Click **Save** to validate the entries and save the filter details.

8.2.3 Define Group Filter

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 checkbox(s) adjacent to the required member names in the Available Filters section and click . The selected members are displayed in the Selected Filters section. Click to select all the Members.

   You can click to deselect a Member or click to deselect all the Members.

   You can also click 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.

8.2.4 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 the button in the Attribute Values grid. The *Attribute Values* window is displayed.

   In the *Attribute Values* window, the **Dimension** field under the 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.

4. 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 the button to delete it.

You can use the Attribute Values present in the Attribute Values grid to generate conditions.

5. Click **Add** button in the Attribute Values grid. The Filter Conditions grid is populated with the filter condition using all the Attribute values.

   You cannot define two conditions using the same attributes. Because conditions are joined with a logical 'AND' and this will make the query invalid.

   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 button to view the SQL statement in the *View SQL* window. Click the button to view a long filter condition in the View Condition dialog.

6. Click **Save**. The Attribute Filter definition is saved.

8.3 Viewing Filter Definition

You can view individual Filter details at any given point.

To view the existing Filter Definition details in the *Filters Summary* window:

1. Select the checkbox adjacent to the Filter Name.
2. Click **View** button in the Filter tool bar.

   The *View – Filter Details* window is displayed with the filter details.

8.4 Modifying Filter Definition

This option allows you to modify the details of Filters.

1. Select the checkbox adjacent to the Filter Name whose details are to be updated.
2. Click **Edit** button and the *Edit – Filter Details* window is displayed. Modify the required changes. For more information, see Add Filter Definition.
3. Click **Save** to save the changes.

8.5 Copying Filter Definition

The Copy Filter Definition facilitates you to quickly create a new Filter Definition based on the existing parameters or by updating the values.

To copy an existing Filter Definition in the *Filters* window:

1. Select the checkbox adjacent to the Filter Name which you want to create a copy.
2. Click **Copy** button in the Filters tool bar. **Copy** button is disabled if you have selected multiple checkboxes. The *Copy – Filter Details* window is displayed.
3. In the *Copy – Filter Details* window you can:

- Create a new filter definition with existing variables. Specify a new **Filter Name** and click **Save**.
- Create a new filter definition by updating the required variables. Specify a new Filter Name and update the required details. For more information, see Add Filter Definition. Click **Save.**

The new filter definition details are displayed in the *Filters Summary* window.

8.6 Checking Dependencies

You can view the dependencies of a defined Filter. You can use a filter in a Run definition. However, the Run definitions are not shown as dependent objects when you check dependency for a filter. This is a limitation.

To check the dependencies of a filter from the *Filters Summary* window:

1. Select the checkbox adjacent to the Filter Name.
2. Click the button in the Filters tool bar. The **Check Dependencies** button is disabled if you have selected multiple members.

   The *Dependent Objects* window is displayed with Object ID, Name, and ID Type of the dependent Objects.

8.7 Viewing SQL of Filter

You can view the corresponding SQL of a defined filter.

To view the SQL of a filter from the *Filters Summary* window:

1. Select the checkbox adjacent to the filter to view the SQL.
2. Click View SQL button. The SQL equivalent of the selected filter is displayed in the View SQL window.

8.8 Deleting Filter Definition

You can remove the Filter Definitions which are not required in the system by deleting them from the *Filters Summary* window.

NOTE	A filter definition with dependency cannot be deleted. However, if the dependent object is a Run Definition, you can delete the filter definition. This is a limitation.

1. Select the checkbox adjacent to the Filter Name whose details are to be removed.
2. Click **Delete** in the Filters tool bar.
3. Click **OK** in the information dialog to confirm the deletion.

9 OFSAA Expressions

An Expression is a user-defined tool that supplements other IDs and enables to manipulate data flexibly. Expression has three different uses:

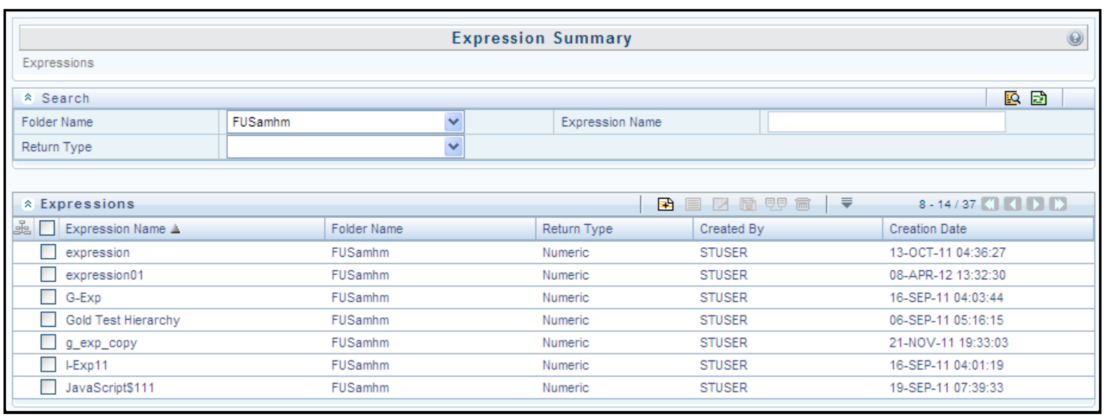
- To specify a calculated column that the Oracle Financial Services Analytical Application derivatives from other columns in the database.
- To calculate assignments in data correction.
- To create calculated conditions in data and relationship filters.

Example: 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.

Based on the role that you are mapped to, you can access read, modify or authorize *Expression* window. For all the roles and descriptions, see *Appendix A* in the OFS Analytical Applications Infrastructure User Guide. The roles mapped to Expression are as follows:

- Expression Access
- Expression Advanced
- Expression Authorize
- Expression Phantom
- Expression Read Only
- Expression Write

Figure 41: Expression Summary window



The *Expression Summary* window displays the list of pre-defined Expressions with other details such as the Expression Name, Folder Name, Return Type, Created By, and Creation Date. For more information on how object access is restricted, see the *Object Security in Dimension Management module* section in the OFS Analytical Applications Infrastructure User Guide.

You can also search for a specific Expression definition based on Folder Name, Expression Name, or Return Type and view the list of existing definitions within the system.

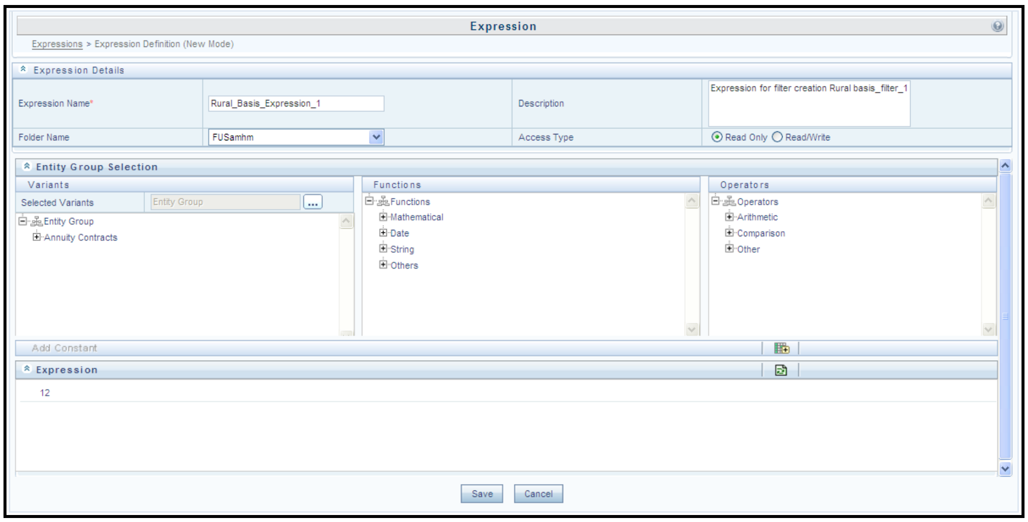
9.1 Adding Expression Definition

This option allows you to add an expression definition using variables, functions, and operators. The Write role should be mapped to your user group.

To create a new Expression from the *Expressions Summary* window:

1. Click **Add** button in the Expressions Toolbar. The *New - Expression* window is displayed.

Figure 42: Expression Summary New window



2. In the Expression Details grid:

   - Enter the **Expression Name** and the required **Description**.

NOTE	Expression Name: The characters &' " are restricted in the name field. Description: The characters ~&+' "@ are restricted in the description field.

   - Select the **Folder Name** from the drop-down list.
     - The *Folder selector* window behavior is explained in the *User Scope* section in the OFS Analytical Applications Infrastructure User Guide.
     - Click to create a new private folder. The *Segment Maintenance* window is displayed. For more information, see the *Segment Maintenance* section in the OFS Analytical Applications Infrastructure User Guide.

NOTE	You can select **Segment/Folder Type** as Private and the **Owner Code** as your user code only.

   - 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 expression.

NOTE	A user with Phantom and Write role can modify or delete the expression even though the access type is selected as Read-only.

- **Read/Write**: Select this option to give all users access to view, modify (including Access Type) and delete the expression.

3. In the Entity Group Selection grid:
   - In the Variants section, click the button The *Variant Selection* window is displayed.
     — Select the **Entity Type** and **Entity Name** from the drop-down lists.
     — Select the required member and click . The member is displayed Selected Members list. Click to select all the Members.

     You can also click to deselect a Member or click to deselect all Members.

     — Click **OK**. The selected Entity Name and Members are displayed in the Variants section in the *New Expression* window.
   - In the Variant's section, click "**+**" to expand Entity Group and double-click to select the required Entity. The selected Entity is displayed in the Expression grid.
   - In the Function section, click "**+**" to expand Functions and select a function such as Mathematical, Date, String, or Others options. The selected Function is displayed in the Expression grid. For more information see the *Function Types and Functions* section in the OFS Analytical Applications Infrastructure User Guide.
   - In the Operators section, click "**+**" to expand Operators and select an operator such as Arithmetic, Comparison, or Others. The selected Operator is displayed in the Expression grid. For more information see the *Operator Types* section in the OFS Analytical Applications Infrastructure User Guide.
     — You can click button from the Add Constant grid to specify a Constant Value. Enter the numerical value and click .
   - In the Expression grid, you can right-click on 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 button in the Expression grid to clear the Expression.

4. Click **Save** to validate the entries and save the new Expression.

9.2 Viewing Expression

You can view individual Expression details at any given point. To view the existing Expression details the *Expression Summary* window:

1. Select the checkbox adjacent to the Expression Name.
2. Click **View** button in the Expressions tool bar.

The *View Expression* window is displayed with the Expression details.

9.3 Modifying Expression

You can modify the Expression details as required in the Edit – Expression screen.

1. Select the checkbox adjacent to the Expression Name whose details are to be updated.
2. Click **Edit** button and the Edit – Expression window is displayed. Modify the required changes. For more information, see Add Expression Definition.
3. Click **Save** and upload the changes.

9.4 Copying Expression

The Copy Expression facilitates you to quickly create a new Expression based on the existing parameters or by updating the values. To copy an existing Expression in the *Expression Summary* window:

1. Select the checkbox adjacent to the Expression Name which you want to create a copy.
2. Click **Copy** button in the Expressions tool bar. **Copy** button is disabled if you have selected multiple checkboxes. The *Copy – Expression* window is displayed.
3. In the *Copy – Expression* window you can:
   - Create a new Expression with existing variables. Specify a new **Filter Name** and click **Save**.
   - Create a new Expression by updating the required variables. Specify a new Expression Name and update the required details. For more information, see Add Expression Definition. Click **Save.**

The new Expression details are displayed in the *Expression Summary* window.

9.5 Checking Dependencies

You can view the dependencies of a defined Expression in the Expression Summary screen:

1. Select the checkbox adjacent to the required Expression Name.
2. Click the button in the Expressions tool bar. The **Check Dependencies** button is disabled if you have selected multiple expressions.

The *Dependent Objects* window is displayed with Object id, Name, and id type of the dependent Objects.

9.6 Deleting Expression

You can delete an expression that has a Read/Write Access Type. To delete an expression from the *Expression Summary* window:

1. Select the checkbox adjacent to the Expression Name(s) whose details are to be removed.
2. Click **Delete** in the Expressions tool bar.
3. Click **OK** in the information dialog to confirm the deletion.

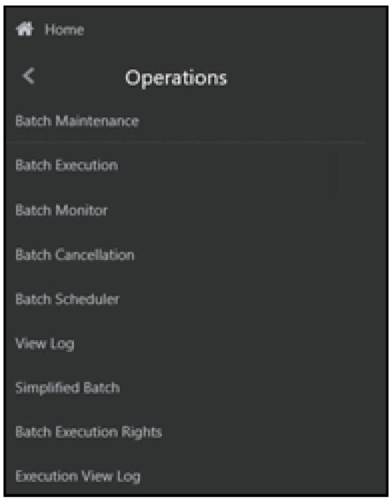
10 Simplified Batches

Simplified Batches provide a streamlined version of the functionality available in the broader standard Oracle Financial Services Analytical Applications (OFSAA) batch module. The Simplified Batch feature is designed to facilitate the creation, maintenance, and execution of batches for the EPM application engines and various other OFSAA components. You can use its drag-and-drop approach to task selection especially helpful in the management of large batches.

10.1 Standard OFSAA Infrastructure Batching Functionality

You can view, execute, schedule, and monitor Simplified Batches within the OFSAA Operations batch framework. To view the Simplified Batch summary page, navigate to Common Object Maintenance, select Operations, and then select Simplified Batch.

Figure 43: Accessing the Simplified Batch Window



10.1.1 Batch Maintenance

The **Batch Maintenance** window is used to view the Simplified Batches, but you cannot edit them. Create and modify them within the **Simplified Batch** window.

10.1.1.1 Batch Execution

You can execute the Simplified Batches from the **Simplified Batch** summary page or the OFSAA Infrastructure UI by selecting **Operations**, and then **Batch Execution**.

10.1.1.2 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 in the **Batch Maintenance** window.

For more information, see the [OFS Analytical Applications Infrastructure User Guide].

This chapter describes the creation, maintenance, and execution of Simplified Batches.

Topics:

- Simplified Batches
- Simplified Batch details
- Creating a Simplified Batch
- Running a Simplified Batch
- Viewing the Task Logs
- Additional Batch Options

10.1.2 Simplified Batches

To view the **Simplified Batch** summary page, navigate to **Common Object Maintenance**, select **Operations**, and then select **Simplified Batch**. The **Simplified Batch** summary page displays a list of 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 **Simplified Batch** details window is displayed.

The **Simplified Batch** summary page has a Search pane and a summary table. When you first navigate to the **Simplified Batch** summary page, the batches stored within your preferred folder are presented on the summary page. This preferred folder is determined by the set of Application Preferences that were last saved in which you selected the Active for Master Maintenance check box. For more information, see the Application Preferences section.

The summary table presents all the batches that meet the search criteria. The **Simplified Batch** summary page offers several icons that allow you to create a new batch or perform different functions when a batch is selected.

Click the column header (the column name) to sort on a column in the **Simplified Batch** summary page.

Figure 44: Simplified Batch summary Page

10.1.2.1 Searching for a Simplified Batch

The Search page allows you to restrict the batches visible within the summary table by any combination of **Name**, **Folder**, or **Status**. Search by Name is a wild card character search, that is, searching for Name-like pricing will find all batches that include pricing within their names.

To search for a Simplified Batch Rule, follow these steps:

1. Enter the desired search criteria and click **Search**.

   Click **Reset** to clear any search criteria and refresh the **Simplified Batch** summary page.

2. The search results are displayed in a tabular format containing all of the Simplified Batch Rule curves that meet the search criteria. The **Simplified Batch Rules** summary page offers several icons to perform the different functions when a Simplified Batch Rule is selected.

   - **Add**: Click the Add icon to begin the process of building a new Simplified Batch. The Add icon is disabled if any rows in the summary table are selected.
   - **View**: Select a single row in the summary table to enable the View icon. Click View to view the detailed definition of a Simplified Batch on a read-only basis. The View icon is only enabled when a single batch is selected.
   - **Edit**: Select a single row in the summary table that enables the Edit icon. Click Edit to modify an existing Simplified Batch. The Edit icon is only enabled when a single batch is selected.
   - **Copy**: Select a single row in the summary table to enable the Copy icon. Click Copy to create a copy of an existing Simplified Batch. The Copy icon is only enabled when a single Simplified Batch is selected. You can also duplicate a batch using the Save As from the Simplified Batch Detail page.
   - **Delete**: Select one or more rows in the summary table to enable the Delete icon. Click Delete to deletes the batch or batches are selected.
   - **Run**: Select a single row in the summary table to enable the Run icon. Click Run to execute the selected Simplified Batch rule. The Run icon is only enabled when a single Simplified Batch rule is selected. The As-of-Date and other parameters applicable to each task within a Simplified Batch will vary by task type. For more information, see the *Specifying Task Parameters* section in the OFS Analytical Applications Infrastructure User Guide.
   - **Refresh**: Click Refresh to refresh the Simplified Batch summary page, including updates to the Last Execution Date and Status.

3. To select a Simplified Batch Rule, select a check box in the first column of the table. More than one Simplified Batch Rules can be selected at a time, but this disables some of the action icons.

   You can select or deselect all the Simplified Batch Rules in the summary table by selecting the check box in the upper left-hand corner of the summary table directly to the left of the Name column header.

   The following list provides the Simplified Batch Rule details based on the search criteria.

   - **Name**: Displays the Simplified Batch rule's short name. Hover on a batch name displays the rule's system identifier, long name, and description.

> **NOTE** 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 the 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

10.2 Simplified Batch Details

When you Add, Edit, or View a batch, the Simplified Batch details window is displayed. The Simplified Batch details window presents the following panes for defining and maintaining your batch:

- Simplified Batch - header level information, such as Rule Name, and so on
- Batch Execution Type
- Search Task Details
- Task Details
- Audit Trail

When you **Add**, **Edit**, or **View** a Simplified Batch Rule, the **Simplified Batch Rule** details window is displayed. The **Simplified Batch Rule** details window includes a Simplified Batch Rule details pane, three Simplified Batch Rule panes, and an **Audit Trail** pane.

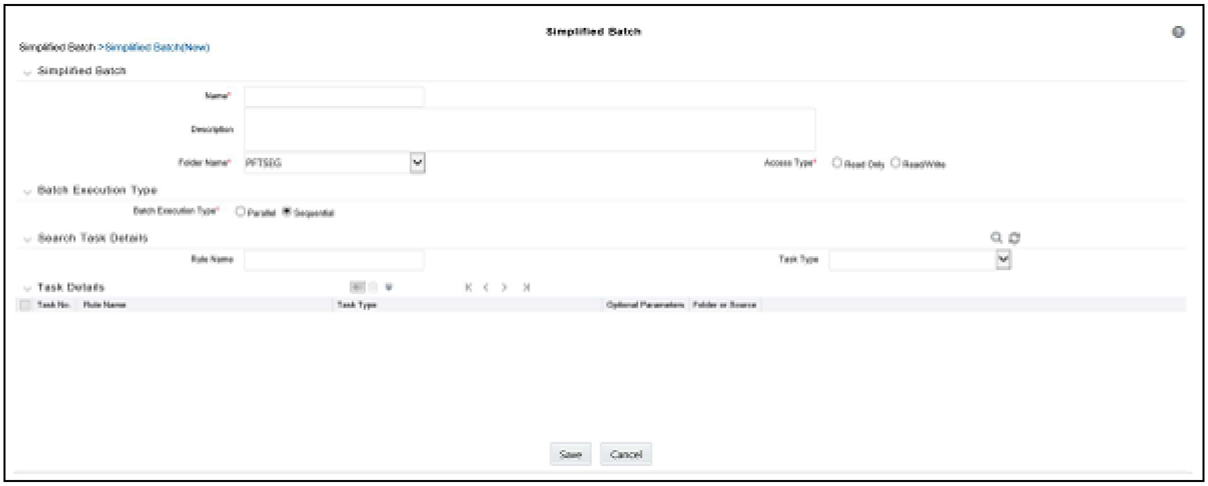
The Audit Trail pane is a standard footer pane for every OFSAA rule type. It displays Created By, Creation Date, Last Modified By, and Modification Date. The **User Comments** tab can be used to add comments to any rule, subject to a maximum of 4000 characters.

10.3 Creating a Simplified Batch

To create a Simplified Batch Rule, follow these steps:

1. Click Add (+) from the Simplified Batch Rule summary page.

Figure 45: Simplified Batch details window to create a new Simplified Batch Rule



2. Enter the following information in the Simplified Batch Rule details window:
   - **Name * and Description**: Simplified Batch Name and Description.

     You cannot edit the Name after you have created the rule.

     The maximum length for the Name is based on (name and Infodom), subject to the total length not exceeding 42 characters. Underscore (_) is the only special character allowed for a Simplified Batch name.

     The maximum length for the Description is 300 characters. Allowed special characters for a Simplified Batch description are underscore (-), hyphen (-), Colon (:), Dot (.), and Space.
   - **Folder Name***: Select a folder in which the batch is to be stored.
   - **Access Type***: Specify whether you want the batch rule to be Read, Write, or Read-Only.

*** indicates Mandatory UI element.**

3. Click **Save**. To build out your Simplified Batch Rule, enter the data within the **Simplified Batch Rule** details page. For more information, see the following sections:
   - Batch Execution Type
   - Searching Task Name
   - Task Details
   - Task Selection

10.3.1 Batch Execution Type

Select the tasks in the batch to run in Parallel or Sequentially. To run some tasks sequentially and some in parallel, utilize the OFSAA Infrastructure Operations batching framework to construct the batch. The Simplified Batching Framework provides a simpler and faster pathway to specify many of the batches that require, but the OFSAA Infrastructure Operations framework offers more flexibility.

Figure 46: Batch Execution Type on Simplified Batch details Window

10.3.2 Searching Task Name

You can construct batches containing dozens or even hundreds of tasks. In such cases, it can become difficult to isolate a specific task. The Search Task details pane allows you to focus on the Task Details and display a specific task name or task type. The Rule Name search is a wild card character search. Enter **Task Type**, **Folder**, or **Task Name**, and then click **Search**. Click **Refresh** to remove the search criteria and refresh the Task Details pane to display all tasks within the batch.

Figure 47: Searching for Task Details

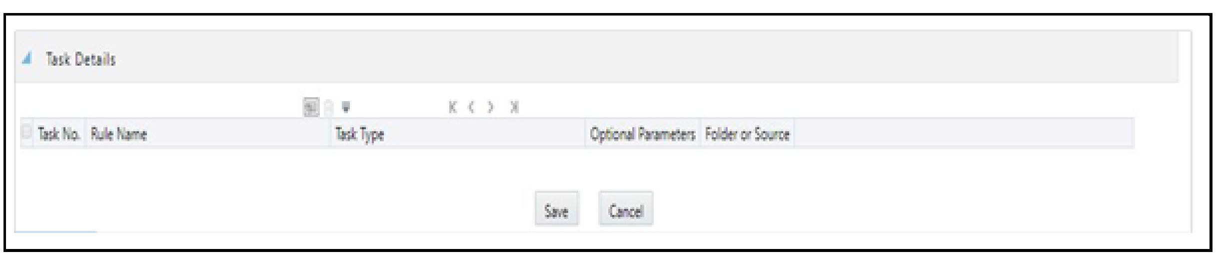
10.3.3 Task Details

For a new batch rule, the **Task Details** pane will initially be empty. Use the Task Selector to populate the Task Details pane with the processes to run in the batch. For a Sequential batch, the Task Details pane presents the tasks in the order in which they will be run. For a Parallel batch, all tasks are initiated in parallel.

The **Task Details** pane also provides a **Delete** icon for removing tasks from the batch.

- **Delete**: Selecting one or more rows in the Task Details page enables the Delete icon. Click Delete to remove the task or tasks that you have selected. You can also remove tasks through the Task Selection window.
- **Task Selector**: The Task Selector icon is always enabled. Click the Task Selector to open the Task Selection window.

Figure 48: Task Details pane of Simplified Batch details Window



This screen displays the following fields:

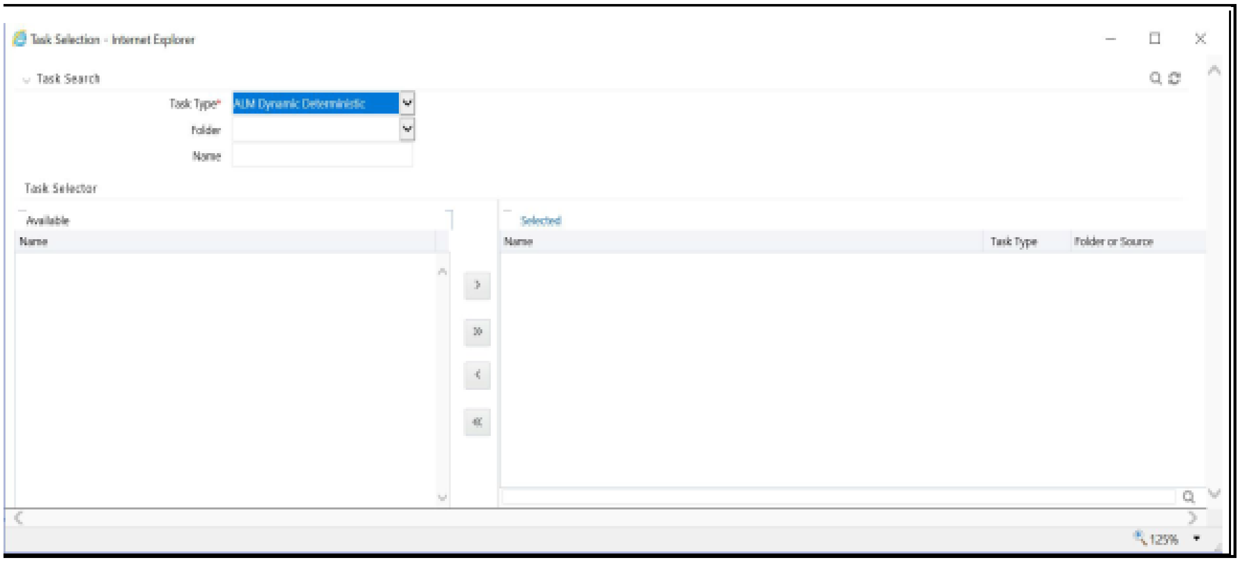
- **Task Number**: Each task is arranged sequentially by the task number.
- **Rule Name**: Displays the task's short name, such as an Allocation rule name. The complete or long name under which the task is submitted for execution (becoming the task's Batch Run ID) is a concatenation of several identifiers, as noted in the View Log. For more information, see the Viewing the Task Logs section.
- **Task Type**: Use Simplified Batches to submit the different tasks for each of the application engines and various other components. For ALM, the following task types can be submitted within a Simplified Batch:
  - 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 the installed Application pack (or packs).
- **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 more information about the task parameters, see the *Specifying Task Parameters* section in the OFS 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.

10.3.4 Task Selection

Click **Task Selector** from the **Task Details** pane to open the **Task Selector** window to support the maintenance of tasks in a batch.

Figure 49: Task Selection window



Use the **Task Selection** window to perform the following:

- Choose tasks to include in your batch
- Remove tasks from your batch
- Reorder tasks within your batch

To select the task from the **Task Selector** window, follow these steps:

1. After opening the **Task Selection** window, the list of **Available Tasks** on the left-hand side of the **Task Selector** window is blank.
2. To populate the list of available tasks, search the task in the **Task Search** pane. You must select a specific type of task from the Task Type drop-down list, and optionally specify search criteria for Folder (or Source) and Name, where applicable to the task type, and click Search.
3. Select the Task Type.

   For ALM, the following are the available task types:

   - 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 the installed Application pack (or packs).
4. Select the **Source** or **Folder**.

- Extract Data Task

Select the Extract Data task to select a data source before executing your search.

- Transform Data Task

After selecting the Transform Data task, do not select a Folder before executing the search.

- All Other Task Types

Select any other Task Type to constrain the search to a particular folder. Alternatively, search for the tasks in all Folders by selecting blank (all folders).

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 window).

You can further refine the search by entering a **Task Name**. Search using **Name** is wild card character searches, that is, searching for Name-like rate will find all tasks that include rate within their Name.

5. 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 and click more to display the names on the next page. You can control the number of tasks listed per page by changing the setting in the `SIMPLE_BATCH_PAGE_SIZE` parameter in the `FusionApps.properties` file in the `WEB-INF/props` directory under context root. The recommended default value is 50 records per page. As you increase the value over a certain amount, you will start to see a slower response in the moving tasks between the Available and Selected lists.

6. Move the tasks between the Available and Selected lists.

Use the **Select** (>) or **Select All** (>>) to move rules from the displayed page of the Available list to the Selected list or vice versa. The **Select** (>) 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 remove rules from the **Selected** list. You can also use the drag-and-drop feature to place the rules in a specific position in the **Selected** list to specify the execution order for a Sequential Type batch.

Tasks within the **Available** list of tasks are disabled (greyed-out) after they are selected (that is, included in the **Selected** list of tasks).

The **Task Selector** also supports drag-and-drop operations for choosing Available tasks, removing Selected tasks, or rearranging the sequence of the Selected tasks.

- Select an **Available** task and then drag-and-drop it anywhere within the Selected list, where the targeted placement is displayed in the 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 the first task in the range to be selected, press the Shift key, click the last task in the range to be selected, and then drag-and-drop the set of tasks.
- To select or deselect a set of tasks that are not in sequence, click each task individually while pressing the Control key, after clicking on the last task to drag-and-drop the set of tasks.

7. Find a Selected task.

To search for any of the selected tasks, enter the Task Name or partial name in the field under the Selected list, and then click **Search**.

10.4 Running a Simplified Batch

To run a Simplified Batch from the Simplified Batch summary page, follow these steps:

1. Select a Simplified Batch from the **Simplified Batch** summary page.
2. Click **Run** to execute the batch.
3. A confirmation message is displayed. Click **Yes** to proceed, or **No** to return to the **Simplified Batch** summary page.
4. After clicking **Yes**, a **Date Selection** window is displayed. If any **Extract Data** or **Transformation Data** tasks are included in the 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. To change the Date, either directly enter a date in the format displayed, or use the **Calendar** tool.
5. Click **OK**.
6. A confirmation message is displayed. Click **OK**.
7. You can monitor the status from the **Simplified Batch** summary page or the **Simplified Batch** details page.
    c. To monitor the status from the **Simplified Batch** summary page, click **Refresh**. The summary page 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).
    d. To monitor the status from the Simplified Batch details page, see the Viewing the Task Logs section.

NOTE This currently applies to all task types except Extract Data and Transform Data.

10.5 Viewing the Task Logs

To view the processing log for each task in the batch, navigate to the log details from the **Simplified Batch** details page:

1. Select the batch from the **Simplified Batch** summary page.
2. Click **View** to navigate to the **Simplified Batch** details page.
3. For a specific task in the **Task Details** pane, navigate to the processing log through **View Log**.
4. In the **View Log** window, view the log details through the **Task ID** link.

    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:

- The 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).

10.6 Additional Batch Options

You can also view, execute, schedule, monitor, cancel, and view task logs for Simplified Batches within the OFSAA Infrastructure Operations batching framework.

11 OFSAA Rate Management

OFSAA Rate Management is a comprehensive utility enabling you to manage currencies, yield curves, interest rates, and currency exchange rate data with a high degree of security and. OFSAA Rate Management also allows you to maintain economic forecasts such as GDP growth, inflation rates, or unemployment projections that can be linked to your models for interest rates, exchange rates, or new business growth.

Historical rate data obtained from OFSAA Rate Management is utilized by the Enterprise Performance Management (EPM) applications (OFS Funds Transfer Pricing, OFS Profitability Management, OFS Asset Liability Management, and OFS Balance Sheet Planning).

- Interest Rates
- Currency
- Currency Rates
- Economic Indicators

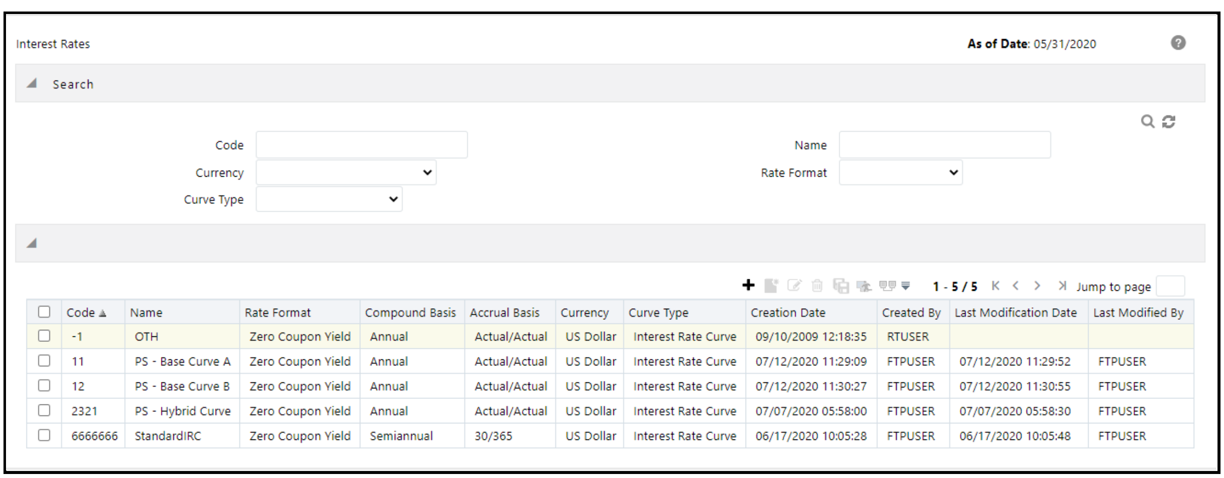
11.1 Interest Rates

The quality and availability of interest rate information vary 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 in 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 creation and maintenance of historical rate data for each yield curve you define.

Historical interest rate data from OFSAA Rate Management is utilized in OFSAA Transfer Pricing to generate transfer rates and option costs. Historical interest rate data is also utilized in OFSAA Asset Liability Management and OFSAA Balance Sheet Planning to generate forecasted interest rate scenarios.

To view the Interest Rate Code Summary page, navigate to **Common Object Maintenance** and select **Rate Management**, and then select **Interest Rates**.

Figure 50: Interest Rate Summary Page



11.1.1 Searching for Interest Rate Curve

A Search pane is provided to search for interest rate curves using Code, Name, Currency, Rate Format, or Curve Type parameters.

To search the Interest Rate, follow these steps:

1. Enter the search criteria and click **Search**.
2. Click **Reset** to remove any **Code**, **Name**, **Currency**, **Rate Format**, or **Curve Type** constraint specified and refresh the window.

   The search results are displayed in a table containing all of the interest rate curves that meet the search criteria. The **Interest Rates Summary** page offers several icons that allow you to perform different functions when an interest rate curve is selected.

   - **Add**: Click **Add** to build a new interest rate curve. The **Add** icon is disabled if any row in the pane is selected.
   - **View**: Select a single row in the table to enable the **View** icon. Click **View** to view the contents of an Interest rate curve in read-only format. The **View** icon is enabled only when a single **Interest rate curve** is selected in the table.
   - **Edit**: Select a single row in the table to enable the **Edit** icon. Clicking the **Edit** icon allows you to modify a previously saved Interest rate curve. The **Edit** icon is only enabled when a single row is selected.

     You can control the number of rows to display on the window by selecting the Pagination Options icon from the action bar.
   - **Delete**: Select one or more rows out of the table to enable the **Delete** icon. Clicking on the Delete deletes the Interest rate curve you have selected.
   - **Copy**: Selecting a single row in the table enables the **Copy** icon. Click the **Copy** icon to create a copy of an existing Interest rate curve. The **Copy** icon is only enabled when a single Interest rate curve is selected.
   - **Check Dependencies**: Select an interest rate curve and then click the **Check Dependencies** icon to generate a report on all rules that utilize your selected interest rate curve.

     The **Check Dependencies** icon is only enabled when a single interest rate curve is 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 the OFS Data Model Utilities User Guide.

     `FusionApps.properties` file must be present under Web Server `<deployed path>/WEB-INF/props and $FIC_WEB_HOME/webroot/WEB-INF/props` directories.

     Update the entry details of the source name. It must be `infodom_SOURCE_NAME=Actual_Source_Name`.

     For example, if Infodom Name is `DEV6INFO` and the source Name is `DEVETLSRC`, then the entry will be `DEV6INFO_SOURCE_NAME=DEVETLSRC`.

     Launch the Historical Interest Rates Data Loader from the Interest Rates Summary page after making this change.

For more information, see Doc ID 2233513.1.

To execute a historical Interest Rate data load, execute the **Historical Interest Rates Data Loader**. A warning message will appear *Upload all available Interest Rates and Parameters?* Click **Ok** to load all historical rates and parameters.

Click **Data Loader** to execute all the interest rates and parameters.

3. To select an **Interest rate curve**, select a check box in the first column of the table. More than one Interest rate curve can be selected at a time, but this disables some of the icons.

   You can select or deselect all the Interest rate curves in the Summary page by selecting the check box in the upper left-hand corner of the Summary table directly to the left of the Code column header.

   The following list provides the Interest Rate details based on the search criteria.

   - **Code**: Displays the interest rate curve's code. The code is a unique number in the range of 1 to 9999999. Hover on a row in the pane to display 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 (that is, 30/360, Actual/Actual, and so on).
   - **Currency**: Displays the Currency (Reference Currency) to which an Interest rate curve is applicable.
   - **Curve Type**: Displays the curve type as an 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 the 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 the Interest rate curve.

11.2 Interest Rates Details

When you **Add**, **Edit**, or **View** an interest rate curve, the **Interest Rate Code Details** window is displayed. The **Interest Rate Code Detail**s window includes an **Interest Rate Code Details** pane, five **Interest Rate Code** tabs, and an **Audit Trail** pane.

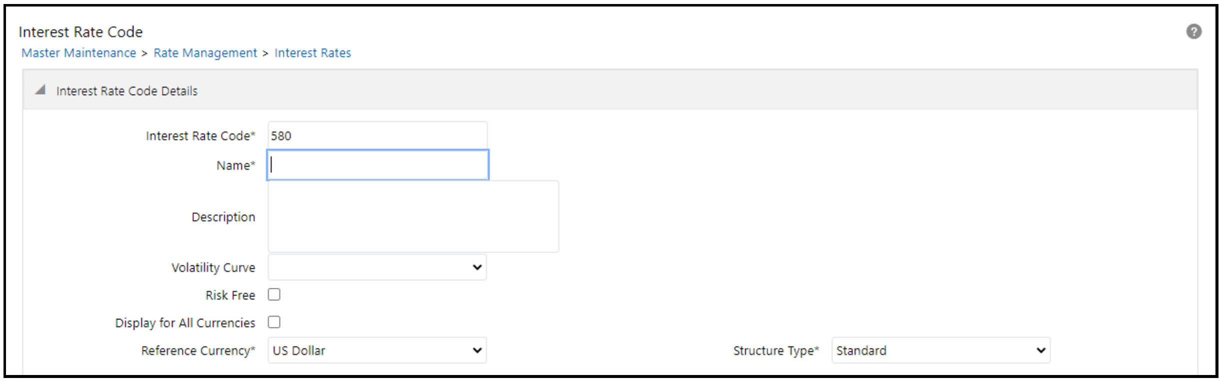
The **Audit Trail** pane is a standard footer pane for every OFSAA rule type. The **Audit Trail** pane displays **Created By**, **Creation Date**, **Last Modified By**, and **Modification Date**. The **User Comments** tab can be used to add comments to any rule, subject to a maximum of 4000 characters.

11.2.1 Creating an Interest Rate Code

To create an Interest Rate Code, follow these steps:

1. Click Add from the Interest Rate Code Summary page.

Figure 51: Interest Rate Code Details window



2. Enter the following information in the Interest Rate Code window.

   - **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 Management or other OFS Analytical Applications, you reference yield curves by Name, not by Interest Rate Codes. Interest Rate Codes are embedded within your instrument data (for example, the INTEREST_RATE_CD and T_RATE_INT_RATE_CD columns within the instrument data are populated with Interest Rate Codes). After you have saved a yield curve, you cannot renumber its Interest Rate Code.
   - **Name and Reference Currency**: You must provide a Name and Reference Currency for your yield curve. Unlike Interest Rate Codes, you can rename or change the Reference Currency for previously saved yield curves. While you can 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 (for example, US Dollar LIBOR, Euro LIBOR, and so on). The Reference Currencies drop-down list displays only Active currencies. For more information on Active and Inactive currencies, see the Currency section.
   - **Description**: You can optionally describe or modify your yield curve's description at any time.
   - **Structure Type**: This attribute is required for each yield curve. 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 more information, see Hybrid Term Structure Tab under Interest Rate Code Tabs. After you have saved the yield curve, you cannot change the selected Structure Type.
   - **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. For FTP Volatility,

both the Terms tab and Historical Rates tab are available with this option. For ALM Volatility, only the Historical Rates tab is available. After you have saved the yield curve, you cannot change the selected Volatility Curve.

Volatility curves are used in FTP to calculate the Rate Lock Option Costs. ALM Volatility is used to evaluate embedded options for the Black 76 market valuation.

- **Risk-Free**: (Optional) This flag is for tagging IRCs as risk-free. That is editable 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 to make them available for assumption mapping to any currency. Assumption rules 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 appears in assumption rules for all currencies.

3. Click **Save**. To build out the Interest Rate Code, you must enter data within the Interest Rate Code tabs. For more information, see the Interest Rate Code Tabs section.

11.2.2 Interest Rate Code Tabs

Interest Rate Code tabs are used to define the 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

11.2.2.1 Navigating Between Interest Rate Code Tabs

For new yield curves, you must begin with the Attributes tab. After you have selected attributes for a yield curve, you cannot edit them. After assigning the attributes, navigate to the Terms tab to define a term structure for your yield curve or volatility curve, that is, an overnight rate, a one-month rate, a three-month rate, and so on. To navigate to the Terms tab, either click Apply on the Attribute tab or click the Terms tab.

NOTE	You must specify an Interest Rate Code, Name, and Reference Currency in the Interest Rate Code Details window before navigate to the Terms tab.

The first time you navigate to the Terms tab, an initial 1-month term point is provided, but even if this is the only term point you want for the curve, you must click Apply to finish term structure specification. In future revisions to the curve's definition, navigate directly to the Historical Rates tab, but if you modify the term structure, you must always click Apply on the Term tab before navigating to the Historical Rates tab.

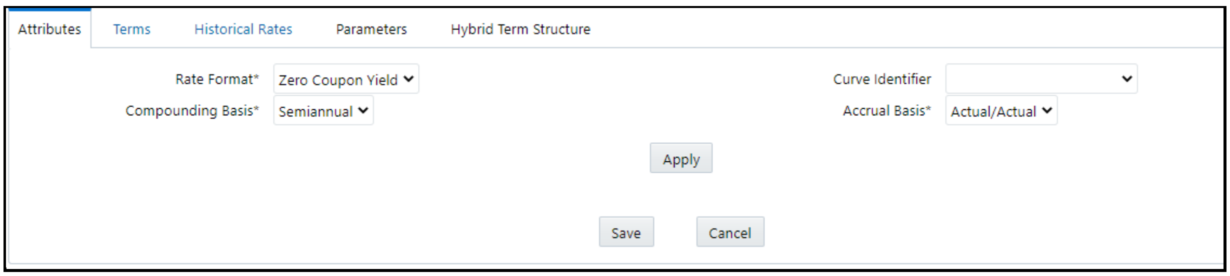
The Historical Rates tab is used to input interest rate or volatility data. This tab is used for maintaining the interest rates database. To navigate to the Historical Rates tab, either click Apply on the Terms tab or select the Historical Rates tab if you have already defined your term structure.

> **NOTE** You must specify the following before navigating to the Historical Rates tab:
>
> - an Interest Rate Code, Name, and Reference Currency in the Interest Rate Code Details window
> - a term structure in the Terms tab

11.2.2.1.1 Attributes Tab

Yield curve attributes include Rate Format, Compounding Basis, Accrual Basis, and Curve Identifier.

Figure 52: Attribute Tab of Interest Rate Code screen



All attributes are disabled if the Volatility Curve is selected. The following list describes the fields in the Attribute Tab of Interest Rate Code screen.

- **Rate Format**: To define the yield curve, you must select either the Zero Coupon or Yield to Maturity Rate Format. Rates entered into Rate Management (in the Historical Rates tab) are always entered in the nominal form, such as 5.125% or 6.875%, not as discount factors. For more information on how the two rate formats affect internal cash flow engine calculations, see the OFS Cash Flow Engine Reference Guide.
- **Compounding Basis**: Select a Compounding Basis for the 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 more information on Compounding Basis and how different compounding bases affect cash flow calculations in OFSAA, see the OFS Cash Flow Engine Reference Guide.
- **Accrual Basis**: Select an Accrual Basis for the yield curve. The Accrual Basis list depends on the 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 more information on Accrual Basis and how different accrual bases affect cash flow calculations in OFSAA, see the [OFS Cash Flow Engine Reference Guide].

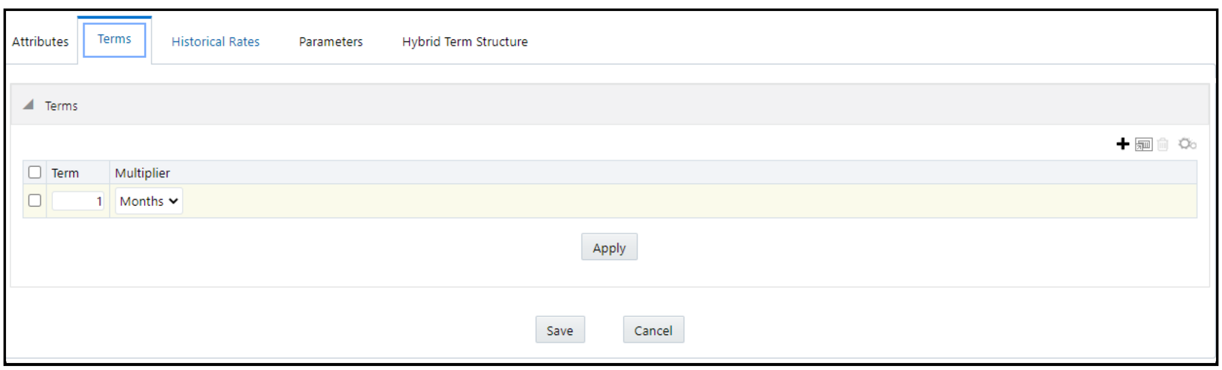
- **Curve Identifier**: The curve identifiers can be mapped to the Adjustable Rate Mortgage (ARM) indices used for ADCO prepayment model processing. This is a drop-down list that contains the six Curve Identifier types, that is, Par Treasury, Zero-coupon Treasury, Par LIBOR/Swap, Zero-coupon LIBOR/Swap, Prime Lending Rate, and Cost of Funds Index. The Reference Currency for these indices is always US Dollar.

  The curve identifier is optional when the IRC is not used for ADCO prepayment model processing. This is applicable only in ALM and HM applications.

11.2.2.1.2 Terms Tab

Use the Terms tab to construct the yield curve's term structure. You can specify as many yield curve terms from the 1 day to 100 years range.

Figure 53: Terms Tab on Interest Rate Code window



> **NOTE** When constructing a volatility curve, the term types available are Contract Term and Expiration Term. You must add the relevant terms for each of these dimensions. Click **Apply** after defining terms for each dimension to save the assumptions before proceeding.

- **Adding New Term Points**: Click **Add** to add new term points by selecting a Term value and a Multiplier (such as 7 days, 2 months, 5 years, and so on). You can also add term points using the Data Input Helper option. For more information, see the Data Input Helper section.

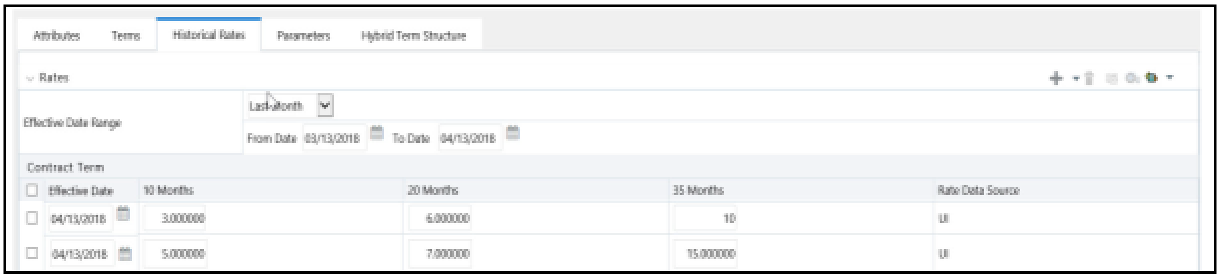
- **Deleting Existing Term Points**: To delete an existing term, select the term point (or terms), and click **Delete.** A confirmation message is displayed. Click **Ok**.

11.2.2.1.3 Historical Rates Tab

Use the Historical Rates tab to define, modify, or view interest rate data. Enter data in simple percentages (such as 5.125, 4.875, and so on). If you are entering historical rates for a volatility curve, then enter volatilities for the combination of Contract Term and Expiration Term. Select the desired Expiration Term from the drop-down list to complete the Volatility Matrix. Effective dates must be entered separately for each Expiration term in the list.

11.2.2.1.4 Historical Rates Tab on Interest Rate Code window

Figure 54: Interest Rate Code window - Historical Rates Tab



> **NOTE** FTP Volatility curves are only applicable to FTP Rate lock option cost calculations and ALM Volatility only applies to embedded option market valuation.

To enter historical rates for an FTP volatility curve, enter volatilities for the Contract Term. Select the desired Expiration Term from the drop-down list to complete the Volatility Matrix. Effective Dates must be entered for each Expiration Term.

To enter historical rates for an ALM Volatility Surface, enter volatilities based on the following two dimensions:

- Strike Rate (Vertical Axis)
- Expiration Date (Horizontal Axis)

For a new ALM Volatility Surface, enter an As of Date and the number of breakpoints for Strike Rate and Expiration Date, then click Generate. The size of the matrix cannot exceed 20 x 20. For Strike Rate, enter values in ascending order. For Expiration Date, enter values in ascending order with values greater than the currently specified As of Date. At every intersection of Strike Rate and Expiration Date, enter a volatility amount in percent (that is, 25 = 25%).

The Rate Data Source parameter shows from where the rates are taken from, either they are entered through the User Interface, loaded through the Data Loader, or generated using the Generate Rates of Hybrid IRC.

You can perform the following tasks:

- Add Historical Rates

- Rate Lookup Behavior Between Term Points
- Rate Lookup Behavior Beyond Term Points
- Rate Lookup Behavior Between Effective Dates
- Generate Graph
- Excel Import or Export
- Deletion of Historical Rates
- Data Input Helper

11.2.2.1.5 Add Historical Rates

By default, the Historical Rates tab displays interest rate data for the past month (for example, for the 30 days leading up to the current date). Click the Effective Date Range drop-down list to expand your view to the last 3 months, 6 months, one year, 3 years, 6 years, or all rate data.

11.2.2.1.6 Rate Lookup Behavior between Term Points

The OFS Cash Flow Engine is common to OFS FTP, OFS PMTPC, OFS ALM, and OFS BSP applications. To lookup rates from Rate Management, the Cash Flow Engine performs 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 must 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 performs interpolation to determine the implied three-month rate. While each of the applications supports simple linear interpolation, OFS FTP and OFS ALM also support Cubic and Quartic Spline interpolation methods. These more advanced methods will be supported for all the OFS Analytical Applications in a future release.

11.2.2.1.7 Rate Lookup Behavior Beyond Term Points

If the Cash Flow Engine must determine a rate from a yield curve for a term point smaller than the first term point of the yield curve, then the Engine utilizes the first term point of the yield curve. For example, if the Engine must determine an overnight rate from a yield curve whose first term point is one month, the Engine utilizes the one-month rate.

If the Cash Flow Engine must determine a rate from a yield curve for a term point greater than the longest term point on the yield curve, the Engine utilizes the last term point of the yield curve. For example, if the Engine must determine a 30-year rate from a yield curve whose last term point is 10 years, the Engine utilizes the 10-year rate.

11.2.2.1.8 Rate Lookup Behavior between Effective Dates

When you are looking up rates from Rate Management for a business date, the Cash Flow Engine helps to find if 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 before the business date of interest (May 31, 2010, in this case).

11.2.2.1.9 Deletion of Historical Rates

To delete historical rates entered, select one or more rows and then click Delete ( ).

11.2.2.1.10 Generate Graph

The Generate Graph option allows you to generate the graph for selected Historical rates.

To generate a graph, follow these steps:

1. Select the Effective Date Range on the Historical Rates tab.

   The From Date and To Date fields will be automatically updated after selecting the Effective Date Range.

2. Select the Effective Date using the corresponding **Calendar** icon.
3. Enter the term points in respective fields and click **Apply**.
4. Select the **Terms** using the corresponding check boxes.
5. Click **Generate Graph**. The graph is generated. Here, you can view the graph for Interest Rate vs Effective Date or Interest Rate vs Term Point.

11.2.2.1.11 Excel Import or Export

NOTE	Starting from the 8.0.4 release, the Excel Import feature will work only on the Internet Explorer browser. Excel Export feature is supported by all browsers such as Chrome or Firefox.

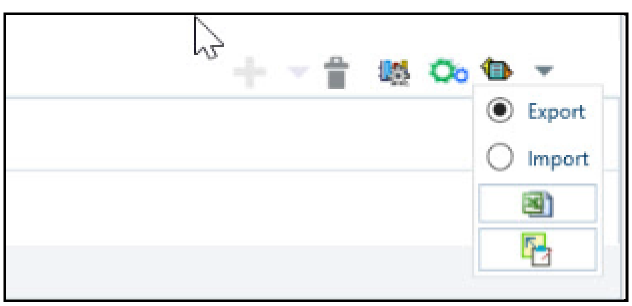
To aid in data entry, use the Excel Import or Export functionality to add or edit rate data to historical rates. This is an optional step.

Excel Export:

To export the data, follow these steps:

1. On the Interest Rates toolbar, click the **Excel** icon.
2. Click **Export** to export data for the chosen selected effective date range. Within the same block, select Export to Excel, which launches the Excel application and displays the data window including headers.

Figure 55: Export and Import options for Historical Rates data



Excel Import:

You can add rows to the pane in the same format. After complete data is entered (or existing data is edited), copy the range of the pane you want to append and go back to the **Historical Rates** pane. In the same block, click the **Import**. The data copied from excel will appear in the **Historical Rates** pane.

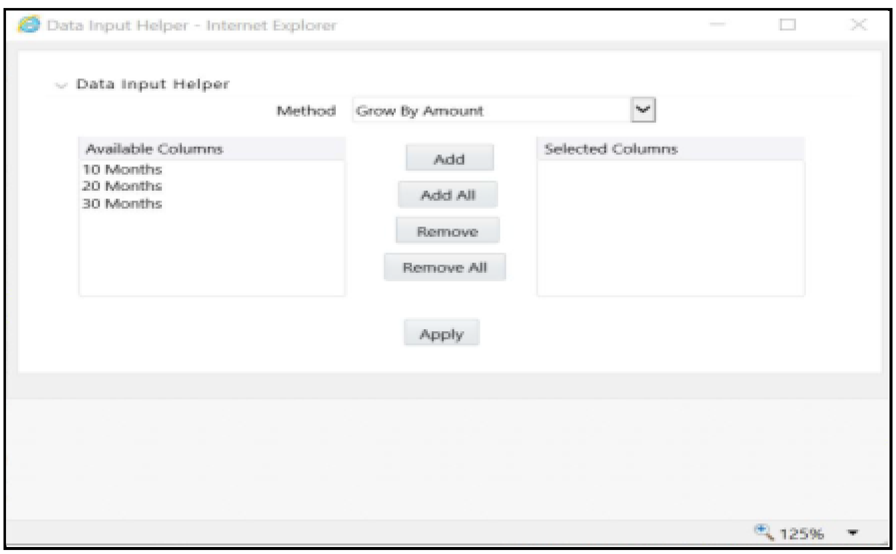
3. If appending data that pre-existed for the same effective date, the import will overwrite existing data.
4. In some cases, some fields will be output to the pane that is not editable, such as bucket start and end dates (when defining forecast assumptions). Add or edit data in the columns that would be editable only in the UI itself.

Data Input Helper

Use the Data Input Helper feature to copy from a row where you have already defined the interest rate definition or apply a fixed value down the page. This is an optional step. To use data input helper, follow these steps:

1. Select the check box next to the rows that you want to work with or use the Select All option by selecting the check box on the header row.
2. Select **Data Input Helper** ().
3. From the **Data Input Helper** window, select method - **Grow by Amount** or some other appropriate method.
4. Select the term points from the **Available Columns** list and click **Add** or **Add All** to move the data to the **Selected Columns** list.

Figure 56: Data Input Helper window



Based on the 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 the Effective Date option is displayed in the Available Columns list.

5. Click **Apply** to modify the Start date and enter the Incremental Value.

11.2.2.2 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 the following ways:

- System-generated calculations through Parameter Estimation
- Direct input into the UI
- Excel Import
- UI entry through Data Input Helper methods
- Data Loader

The following interest rate models are available:

- Extended Vasicek
- Ho and Lee
- Merton
- Vasicek

11.2.2.2.1 Parameter Estimation

This section explains the procedure to calculate the estimated parameters.

Prerequisites

Installation of R and Oracle R Enterprise (ORE) is required to use the Term Structure Parameter Estimation functionality under Rate Management - Interest Rates, for computing term structure parameters.

For more information, see the OFS Advanced Analytical Applications Infrastructure Installation and Configuration Guide.

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 look back term and a sufficient number of observations (available historical rates) for the IRC. If you rerun with a different look-back term, it will overwrite the existing parameters for the selected Term Structure Model on that IRC's Effective Date.

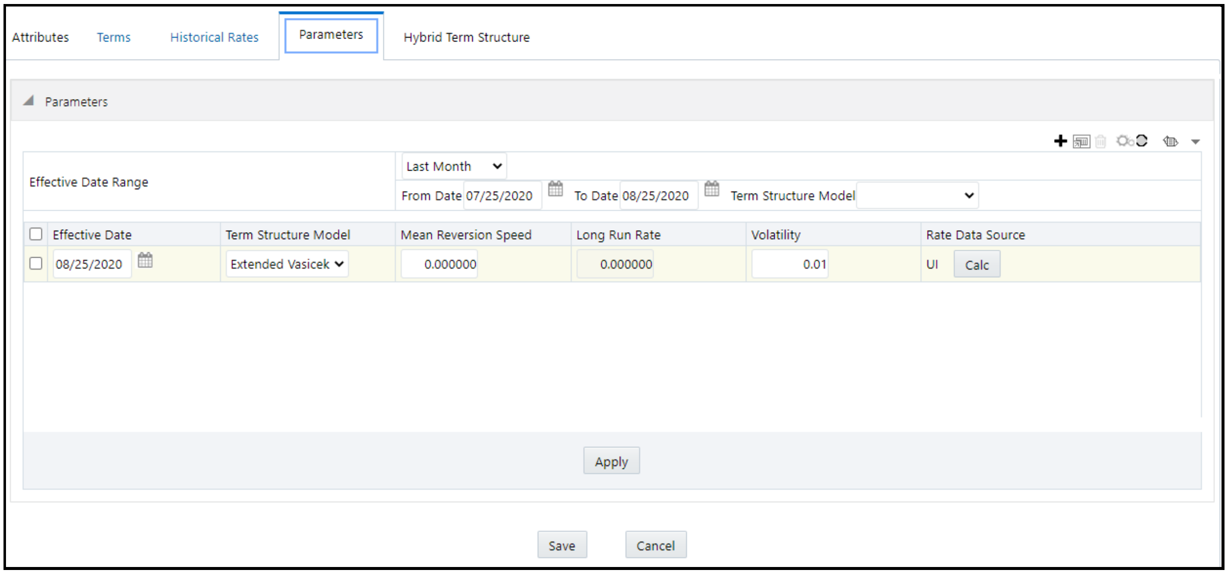
- **Term point**: Underlying historical rates must be available for a 30-Day or 1-Month term point.
- **Minimum Number of historical rates**: A total of at least 10 historical rates (observations) are required, on appropriate look-back dates.
- **Lookback Dates**: Historical rates must be available on dates looking back from the Parameter's Effective Date (the End Date), in 30-day intervals moving backward 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 must 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 before the End Date, and so on.

The minimum relative term for all lookbacks must be at least 300 Days (that is, 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 to identify the Interest Rate Code, and the parameter estimation Engine will exit.

To define the Parameter Estimation, follow these steps:

Figure 57: Parameters Tab on Interest Rate Code window



To edit or recalculate the existing Parameters, enter the Effective Date Range filter.

1. The following icons are available here:
   - Add
   - Delete
   - Data Input Helper
   - Refresh
   - Excel Export or Import
2. Default parameters for the **Extended Vasicek Model** are displayed for one Effective Date (the System Date on which the Interest Rate Code was created). You can edit these parameters or add new parameters using the **Add**. Use the **Refresh** to return to existing Parameters.

> **NOTE** Steps 3 and 4 are applicable only if you are Adding or Editing Parameters.

To add or edit the parameters, use the **Data Input Helper** or **Excel Import or Export**. For more information, see the Excel Import or Export and Data Loader sections.

3. Enter the **Effective Date**. Note that the **Effective Date** cannot be greater than the **Current System Date**.
4. Select the Model from the **Term Structure Model** drop-down list. Effective Date and Term Structure Model combination must be unique within this IRC.
5. The following term structure models are utilized in stochastic modeling of interest rates in OFS FTP and OFS ALM:
   - Extended Vasicek
   - Ho and Lee
   - Merton
   - Vasicek
6. The following parameters needed by the models:

Table 13: List of supported parameters for Models Term structure models in Interest Rate

Model	Parameter 1	Parameter 2	Parameter 3
Extended Vasicek	Volatility	Mean Reversion Speed	
Ho and Lee	Volatility		
Merton	Volatility		
Vasicek	Volatility	Mean Reversion Speed	Long Run Rate

7. Values for Long Run Rate and Volatility are in percentages.

   For example, a Long Run Rate of 5% is displayed as 5.000. To maintain the integrity of data, Rate Management restricts the accepted input values. The valid range and the default setting for each parameter.

Table 14: Valid Range and Default Values of Interest Rate Parameters

Parameter	Valid Range	Default Value
Volatility	0% to 500%	0.01
Mean reversion speed	0.00 to 500	0.0
Long run rate	0.00% to 500%	0.0

8. The **Rate Data Source** indicates if you have directly entered Parameters through the UI. You can calculate the Parameters to display system-generated values.
9. Select Calculate to view the Term Structures Parameter Estimation window.
10. If you are directly entering the Parameters, then enter values appropriate to the Model, then click **Apply** else click **Calculate**.
11. The End Date is auto-populated with the Effective Date. Enter the Relative Term with Multiplier to define the rate look-back period.
12. The Term or Multiplier value must be at least 300 Days.

13. The Start Date is automatically updated after entering a Relative Term.
14. Click **Calculate Number of Observations** to confirm the number of rates found for appropriate dates within the Relative Term. If there are at least 10 observations, then the **Estimate** option will become active.
15. Click **Estimate** to calculate the parameters and store them in the Historical Parameters table (FSI_IRC_TS_PARAM_HIST). A confirmation message is displayed. Click **Ok**. The calculation will complete and you will be directed back to the Parameters tab.
16. 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 displays an error message.

Delete parameters by selecting one or more rows and then clicking **Delete**. For more information on term structure models and stochastic processing, see the OFS Funds Transfer Pricing User Guide, OFS Asset Liability Management User Guide, and OFS Cash Flow Engine Reference Guide.

11.2.2.3 Hybrid Term Structure Tab

Hybrid Term Structures allows you to specify the following three types of Hybrid yield curves:

- Spread
- Moving Average
- Merge

NOTE	Minimum and Maximum Hybrid IRC types from the Hybrid IRC type selection is intended for future enhancement and must be ignored in the current release. The Parameters and Hybrid Term Structure tabs are disabled if the 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 must be updated. After defining, the Hybrid yield curves can be used like any other interest rate curve in the system. You can reference these curves within the OFSAA application business rules that allow the selection of an Interest Rate Code.

Hybrid Curve tab on Interest Rate Code window

Hybrid Curve Type: Spread: A Spread hybrid yield curve is defined as the difference between two standard yield curves. The Spread type of hybrid yield curve is 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 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, specify the percentage weighting applied to each of the underlying standard hybrid yield curves.

11.2.2.3.1 Define a Hybrid Curve

To define a hybrid curve, follow these steps:

1. Select the **Source Type** as **Hybrid** in the **Hybrid Term Structure** tab, and then select the **Hybrid Curve Type** (Spread, Moving Average, or Merge).
2. Select the **Interest Rate Codes** for the hybrid type and click **Apply**.

11.2.2.3.2 Generate Historical Rates

After a hybrid curve is defined, generate the 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 select the Generate Frequency as monthly, it generates month-end values only. If you select daily, it generates the maximum number of historical values.

To generate the rates, follow these steps:

1. Select the interpolation method as **Linear**, **Cubic**, or **Quartic**.
2. Select the **Generate Frequency** (Daily, Weekly, Bi-Weekly, or Monthly) and enter the **Specific Date Range** (From Date and To Date).
3. Click **Generate**. The rates will be populated and you will be directed to the Historical Rates tab to view the results.

   If **Hybrid Curve Type** is selected as **Moving Average,** then select the **Moving Average Term** in Days, Months, or Years.

11.2.2.3.3 Automate (Schedule) the Generate Rates Action in Hybrid IRCs

The generation of the Hybrid IRC rates is available within the Hybrid IRC of Rate Management UI. As a manual process, you can additionally schedule a Run through an ICC Batch process or a Simplified Batch process.

If you are running Generate Rates through the UI, then provide the following three parameters:

- **Interest Rate Code**: For example, you are in edit mode for a particular Hybrid IRC, 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. Or, a list of IRCs can be specified with comma-separated values within quotes.

'IRC_Code1, IRC_Code2, IRC_Code3' and so on.

NOTE	Use ALL in the Parameter list for all IRCs.

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, 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.

Topics:

- Dependent Tables
- Prerequisites to running - Hybrid IRC batches
- Batch Creation and Running Batches
- Viewing Log Messages

Dependent Tables

The following list describes the details of the 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 must have a seeded date for IRC_HYBRID_SCHEDULER DT.
- **DATE_TASK_STEP_PRECEDENCE**: Similar to DATE_TASK_MASTER, this table must also have an entry for IRC_HYBRID_SCHEDULER DT.

Prerequisites to Running Hybrid IRC Batches

- The function FN_FSI_IRC_HYBRID_SCHEDULER must be in Compiled status in the Atomic schema. Sometimes, the functions created in the Database can be in an invalid state due to some issues (by the installer), so use the Compile option in the Database to get them in a valid state.

- You 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 must already exist for all parent and dependent IRCs for the relevant effective date range.

NOTE	Seeded data related to Batch information must be present in the DATE_TASK_MASTER and DATE_TASK_STEP_PRECEDENCE tables in the Config schema.

11.2.3 Batch Creation

There are two methods for creating and running the batch processes, ICC Batch and Simplified Batch. The following section describes how to set up and run the Hybrid Rate Generation using both approaches.

For more information, see the OFS Analytical Applications Infrastructure User Guide.

Running Hybrid Rate Generation using ICC Batch:

1. Navigate to **Common Object Maintenance**, select **Operations**, and then select **Batch Maintenance**. Click **Create Batch**.
2. Enter the following details in the Batch Maintenance window:

   Component = "Transform Data"

   Rule Name = "IRC_HYBRID_SCHEDULER"

   Parameter List = 'IRC_Code'

NOTE	Use ALL in the Parameter list to use all IRCs.

3. Save the batch.
4. Navigate to **Common Object Maintenance**, select **Operations**, and then select **Batch Execution**.
5. Search the **Batch** created under **Batch Maintenance** and select it.
6. Enter the **Information Date** for the batch.
7. Click **Execute Batch**.

   To schedule a future Hybrid Rate Generation Batch Process, or to schedule the Hybrid Rate Generation Batch Process on a recurring basis, do the following:

8. Navigate to Common Object Maintenance, select Operations, and then select Batch Scheduler.
9. Search the Batch to schedule and select it.
10. Click New Schedule or Existing Schedule:

NOTE	An Existing Schedule can be selected only if there are existing scheduled batches to view.

11. If New Schedule is selected, the New Schedule details appear. Enter the Scheduled Task Name.

    To schedule the Hybrid Rate Generation on a recurring basis, select Daily, Weekly, Monthly, or Adhoc.

12. In the Schedule Task pane, enter **Start Date** and **End Date** in the Date field.
13. In the **Run Time** field, enter the time for the next validation to be run.
14. 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 that is displayed in the View Log.

    For more information, see the OFS Analytical Applications Infrastructure User Guide.

11.2.3.1 Running Hybrid Rate Generation using Simplified Batch

To run Hybrid Rate Generation using Simplified Batch, follow these steps:

1. Navigate to **Common Object Maintenance**, select **Operations**, and then select **Simplified Batch**.
2. Click **Add**.
3. From the **Task Details** pane, click **Select Task**. In the Task Selection window, select **Task Type** as **Transform Data**. Click **Search**. In the Task Selector, select the **IRC_HYBRID_SCHEDULER** and click **Ok**.
4. Enter the Parameters and click **Save**.
5. In the **Simplified Batch Summary** page, search the **Hybrid Rate Generation** batch, and select it. Click the **Run** icon.
6. Click **Yes** to confirm you want to continue, enter a date (this will be your TO DATE), and click **Ok** to continue. A confirmation message is displayed.
7. Click **Ok**. When the batch is complete, navigate to **Operations** and select **View Log** to view the processing log.

Any error messages or warnings are accessible from the View Log window.

NOTE	Simplified Batch does not provide access to logs for Transform Data tasks.

For more information, see the OFS Analytical Applications Infrastructure User Guide.

11.2.3.2 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. Navigate to the Operations menu and select View Log.
2. Enter search criteria as Data Transformation and related Batch ID
3. Select the Task ID to view the log information

   You can additionally query the FSI_MESSAGE_LOG table directly to view the error log details.

   For more information, see the OFS Analytical Applications Infrastructure User Guide.

11.3 Currency

Financial institutions 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 can also define and add your user-defined currencies.

To view the Interest Rate Code Summary page, navigate to Common Object Maintenance and select Rate Management, and then select Currency. This page displays a comprehensive list of more than 170 seeded ISO currency codes.

Currency Summary page

11.3.1 Searching for Currency

A Search pane is provided to search for currencies by Name, Currency (by ISO currency code), Status, or Reporting Currency.

To search the currency, follow these steps:

1. Enter the search criteria and click **Search**.

2. Click the **Reset** to remove the search criteria you have specified and refresh the window.
3. The search results are displayed in a tabular format containing all of the currencies that meet to search criteria. The Currency summary pane offers several icons as follows to perform different functions when a currency is selected.
   - **Add**: Click **Add** to begin the process of adding a new currency. The **Add** icon is disabled if any rows in the pane are selected.
   - **Edit**: Select a single row out of the pane to enable the **Edit**. Click the **Edit** icon to modify an existing currency. The **Edit** icon is only enabled when a single currency has been selected.

     You can control the number of rows to display on the page by selecting the Pagination Options icon from the action bar.
   - **Delete**: Select one or more rows out of the pane to enable the **Delete** icon. Click the Delete icon to delete the currency or currencies you have selected.
   - **Check Dependencies**: Select a currency and then click the **Check Dependencies** icon to generate a report on all rules that utilize your selected currency.

     The **Check Dependencies** icon is only enabled when a single currency has been selected.
4. The **Currencies** window contains all of the currencies that meet the search criteria. The Currencies Summary page has several icons to perform different functions when a currency is selected.

   To select a currency, select the check box in the first column of the pane. More than one currency can be selected at a time but this will cause some of the icons to become disabled. Select a check box a second time deselects the currency.

   Select or deselect all of the currencies in the Summary page by selecting the check box in the upper left-hand corner of the Summary page directly to the left of the Code column header.

   The following list provides the currency details based on the search criteria:
   - **Code**: For seeded currencies, these are ISO Currency Codes. For user-defined currencies, these can 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 can 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 currency is associated with forecasting purposes. Define multiple yield curves each of which has the same Reference Currency, but a currency can only have one Reference Interest Rate Code.
   - **Reporting Currency**: A reporting currency is an active currency to which balances in other currencies can be consolidated to facilitate reporting. Balances in reporting currencies can 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 can be either Active or Inactive. You must Activate a currency before doing the following:

Define that currency as a Reference Currency for a yield curve (For more information, see the **Reference Currency** section in the Interest Rates Details window.)

Enter Exchange Rate data for a currency (For more information, see the Currency Rates section.)

Define Forecast Rates for that currency within OFS ALM (For more information, see the Forecast Rates section).

11.3.2 Editing Currencies

To edit a currency, select a currency, click **Edit**. You cannot modify any currency Code value. You can modify the Currency Name, Reference Interest Rate Code, Reporting Currency value (Yes or No), and the currency's Status (Active or Inactive).

You cannot 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

After completing the edits, click **Save**.

11.3.3 Adding Currencies

To add a currency, follow these steps:

The first row in the Currencies window is an empty row that is ready for you to edit as a new currency. Select this empty row, click Edit, and generate the new currency.

or

Generate a new currency by clicking **Add**. This generates another empty row at the top of the Currencies window for editing purposes.

11.3.4 Deleting Currencies

Select one or more currencies and then click **Delete**. You cannot delete any currencies that are utilized elsewhere in the system.

11.4 Currency Rates

OFSAA Rate Management's Currency Rates module uses the currencies defined and activated in the Currency module to support the creation and maintenance of historical exchange rates. Currency exchange rates are utilized in:

- OFS Funds Transfer Pricing Ledger Migration processes
- OFS Asset Liability Management Currency Consolidation process (see the OFS Asset Liability Management User Guide)
- OFS Profitability Management multi-currency allocations.

- OFS Hedge Management and IFRS Valuations (see the Oracle Hedge Management and IFRS Valuations User Guide)

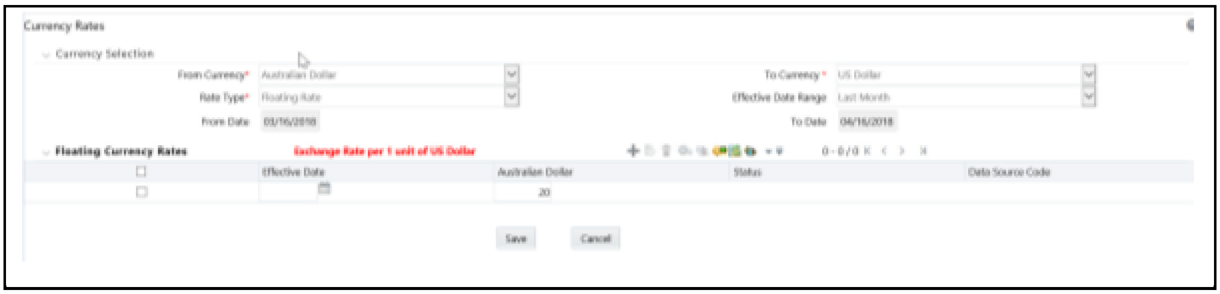
To view the Currency Rate, navigate to **Common Object Maintenance**, select **Rate Management**, and then select **Currency Rates**. In this window, you can manage historical exchange rates between currencies.

For ALM, FTP, and PFT, From Currency defaults to the Initial Currency selection from the Assumption Management defaults in the active Application Preferences window. You can select another From Currency from the drop-down list that displays all Active currencies.

11.4.1 Preparing to Work with Exchange Rate Data

To view, edit, or delete the exchange rate data, you must enter the **To Currency** value. With the From Currency, the To Currency drop-down list displays only Active currencies.

Currency Rates Summary page



After selecting a To Currency value, select a value for Rate Type as Floating Rate or Fixed Rate (the default selection is Floating Rate). After selecting a To Currency value, a second pane is displayed as 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).

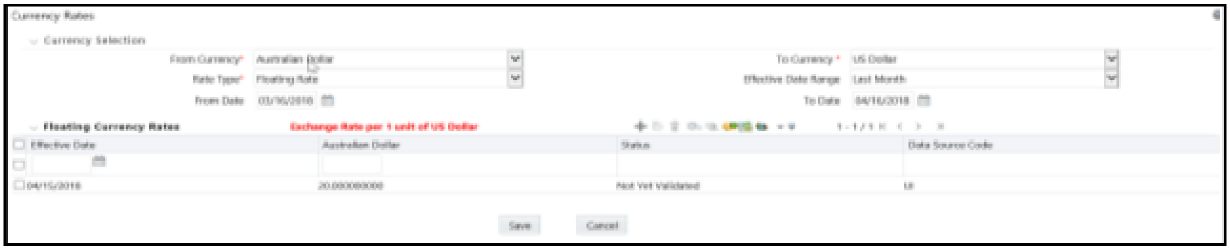
- **Floating Rates**: Floating exchange rates, such as those between the US Dollar (USD), the Pound Sterling (GBP), the Japanese Yen (JPY), and the Euro (EUR), are market-driven and can change from day-to-day, hour-to-hour, or minute-to-minute.
- **Fixed Rates**: Some countries, especially smaller countries or countries that have experienced significant inflation in the recent past, can wish to "peg" their currency to a larger, more stable currency such as the US Dollar, Japanese Yen, or Euro.

11.4.2 Adding Exchange Rate Data

After you have specified a value for the **To Currency**, the Floating Currency Rates section appears. To define a Fixed Rate relationship, select the Fixed Rate Type and replace the Floating Currency Rates with the Fixed Currency Rates.

Both Currency Rates panes 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 data into the blank row, and click **Save**.

Defining a Currency Rate



The following list describes the columns in the Currency Rate window:

- **Effective Date**: Directly enter a date or select the Calendar ( ) icon to choose an effective date for your new exchange rate data point.

  Rate Management stores the historical exchange rate data. You cannot 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 the Forecast Rate Scenarios. If you have gaps in the 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, then the application will utilize the exchange rate for February 11, 2010.

- **Status**: Status is a read-only display that 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 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. See Table 12 for example.

Table 15: Example of one Currency to another Currency Conversion

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

11.4.3 Adding Multiple Exchange Rates

Click **Add** to add additional blank rows to enter the additional Effective Dates and Exchange Rates. After adding the multiple new exchange rates, click Save.

11.4.4 Editing Exchange Rate Data

Select the check box on the left-hand side of any row to enable the **Edit** icon. After clicking Edit, the row becomes active to edit the **Effective Date** and (or) the **Exchange Rate**. Click **Save** to save the changes.

11.4.5 Viewing Exchange Rate Data

By default, both the Floating Currency Rates pane and the Fixed Currency Rates pane 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 in the Currency Selection window.

Click the **View** icon to a specific range of effective dates by modifying the From Date, To Date, or both dates within the Currency Rates window.

11.4.6 Deleting Exchange Rate Data

Select one or more check boxes on the left-hand side of any row to enable the **Delete** icon. After clicking **Delete**, a confirmation message is displayed. Click **Ok**.

11.4.7 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 the OFS Data Model Utilities User Guide.

To execute a data load, click the **Data Loader** icon. A warning message is displayed Upload all available Currency Rates?

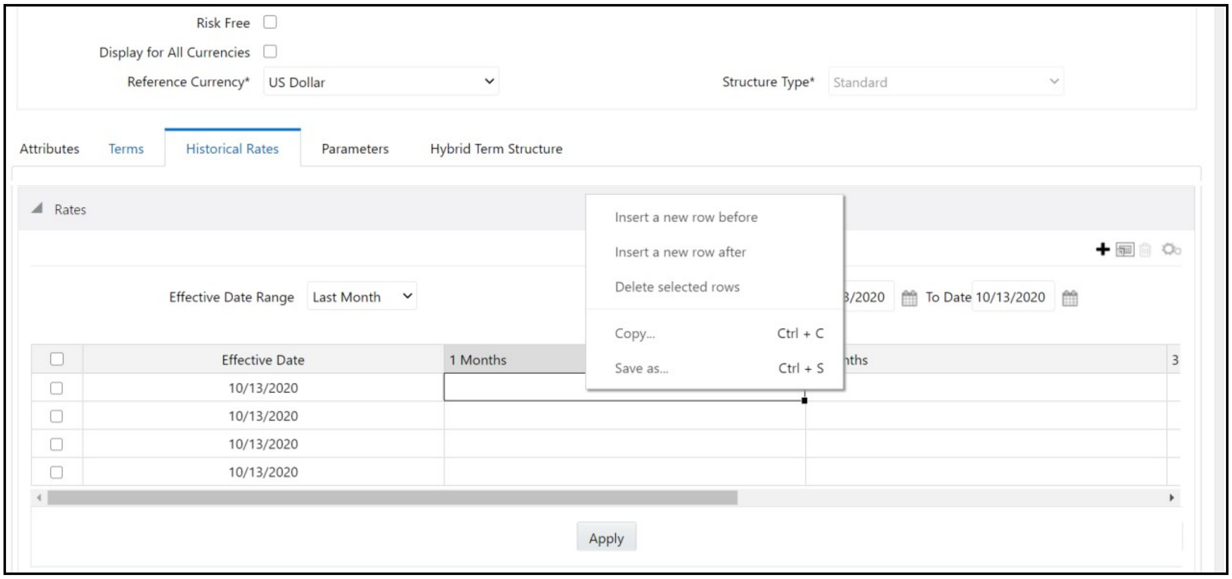
Click the **Data Loader** icon to execute all the currency rates.

11.4.8 Excel Compatibility

This functionality is used for adding or editing historical exchange rates from an Excel sheet. You can right-click on any cell in the table and perform the following actions:

- Delete or Insert multiple rows
- Apply a Formula on a particular column
- Export the data to excel using the **Save As** option.
- You can copy data from excel and paste the same using standard copy, paste shortcuts **Ctrl+C** and **Ctrl+V**.

Figure 58: Historical Rates Tab



11.5 Currency Exchange Rate Validation

Exchange Rate Validation has 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.

11.5.1 Features of Exchange Rate Validation

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 the OFSAA applications. Some of these rates can 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 an exchange that needs to be triangulated. Also, 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.

11.5.2 Currency Rate Validation

Load currency rates through the Rate Management or using the Historical Rate Data Loader. The initial status is Not Yet Validated.

1. Execute the Currency Rate Validation engine.
2. Navigate to **Operations**, select **Batch Maintenance**, and then click **Create Batch**.

Component = "Transform Data"

Rule Name = "Rate_Validation"

Parameter List (Required) = FROM_DATE and TO_DATE, for example, '20110101', '20110131'

After completion of the validation process, the status in the Currency Rates window changes to Valid or Invalid (in case there is an issue).

3. View both input and calculated results in the FSI_EXCHNG_RATE_DIRECT_ACCESS table. The OFSAA processing engine reads this table for sourcing historical exchange rates.

NOTE	Executing Rate Validation is a required step when a multi-currency setup is enabled. All of the OFSAA processing engines require exchange rates to be validated through this process 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.

11.5.3 Validating Exchange Rate Relationships

You must run the exchange rate validation process after adding or modifying exchange rate data. Run the process immediately or schedule one or more to be run in the future.

Each exchange rate has one of the following statuses:

- **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 Currency Rates window of the Rate Management.

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. For more information, see the Viewing Log Messages.

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 period.

- A Child currency within a fixed relationship must not be a Child currency in any other fixed relationship during the same 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 the same period.

- 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 period. If it does, then the following message is displayed: Invalid fixed relationship creates a circular relationship with other fixed exchange rates.
- 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 or To or Both currency or currencies in the new exchange rate already exist in a fixed relationship for the same period.
- If any exchange rate is equal to 0, then a warning message is displayed. 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.

11.5.4 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 Currency Rates window of the Rate Management.

11.5.4.1 Running a Validation Immediately

Execute the exchange rate validation using the Currency Rates Validation option.

Currency Rate Validation window

To execute the exchange rate validation, follow these steps:

1. Select a check box on the left-hand side of any row to enable the **Currency Rates Validation** () icon.
2. Click **Currency Rates Validation**.

3. To execute exchange rate validation from the Currency Rates window, the following two options are available:
   - **Specify Dates**: After selecting this option, a Select Dates pane is displayed to enter or verify the FROM_DATE and TO_DATE parameters. These dates will be passed to the batch for execution.
   - **FROM_DATE**: This defaults to the date of last rate validation.
   - **TO_DATE**: This defaults to the current date.
   - **Validate For All Dates**: Select this option to validate all the 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 of history available to be processed.

   You can execute rate validation using a Simplified Batch or the ICC Batch window. You can also launch it from the Currency Rates UI (Currency Rates Validation toolbar).

To run the validation using the Simplified Batch, follow these steps:

1. Click **Common Object Maintenance**, select **Operations**, and then select **Simplified Batch**, and then click **Add**.
2. From the **Task Details** pane, click **Select Task**. In the **Task Selection** window, choose **Task Type** as **Transform Data**. Click **Search**. In the Task Selector window, select the **Rate_Validation** and 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.

   From_Date and **To_Date** must 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. 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 and select View Log to view the processing log.

   Any error messages or warnings are displayed in View Log. For more information, see Viewing the Messages.

   Simplified Batch does not yet provide access to logs for Transform Data tasks.

   For more information, see the OFS Analytical Applications Infrastructure User Guide.

To run the validation using the ICC Batch Framework, follow these steps:

8. Navigate to **Operations**, select **Batch Maintenance**, and then click **Add** to create a new batch.

9. Search for the above batch, and select it.
10. In the **Task Details** pane (toolbar), click **Add**. In the **Task Selection** window, choose the **Task Type** as **Transform Data**. Click **Search**. In the **Task Selector**, select the **Rate_Validation** task and click **Ok**.
11. Back in the **Task Definition** window, select the **Rule Name** as **Rate_Validation** and enter the optional parameters to specify the From and To dates, using the format YYYYMMDD, YYYYMMDD.

    From_Date and **To_Date** must 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.
12. Click **Save**.
13. Navigate to Operations, and select **Batch Execution**. Search your batch and select it. Enter Information Date and click **Execute Batch**.
14. Click **Ok**. A message is displayed stating Batch triggered successfully, also displaying the Batch Run ID.
15. Click **Ok**.
16. To view rates, query the database table FSI_EXCHNG_RATE_DIRECT_ACCESS after the run is complete.

    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, 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:

Table 16: Column details of 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)

17. When the batch is complete, navigate to Operations and select View Log to view the processing log. Any error messages or warnings are displayed in View Log. For more information, see Viewing the Messages.

For more information, see the OFS Analytical Applications Infrastructure User Guide.

11.5.4.2 Scheduling One or More Validations

To schedule a future validation, or to schedule validations on a recurring basis, follow these steps:

1. Navigate to **Operations** and select **Batch Scheduler**. The **Batch Scheduler** window appears.
2. In the **Batch Name** pane, select the **Scheduled Batch ID**. An unchecked box means that no validation is scheduled to run.
3. Select **New Schedule** or **Existing Schedule**.

   The Existing Schedule can be selected only if there are existing schedule batches to view.

   If New Schedule is selected, the New Schedule pane appears. Enter the Scheduled Task Name.

   To schedule the validation on a recurring basis, select Daily, Weekly, Monthly, or Adhoc.
4. In the **Schedule Task** pane, enter the **Start Date** and **End Date** in the Date field.
5. In the **Run Time** field, enter 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. For more information, see Viewing the Messages.

   For more information, see the OFS Analytical Applications Infrastructure User Guide.

11.5.4.3 Viewing the Messages

Any error messages or warnings generated during the exchange rate validation process are displayed in the Log Information window.

To view the messages, follow these steps:

1. Navigate to **Operations** and select **View Log**.
2. Click the **Task ID**. The Log Information window will display.

These exception messages can also be seen in the FSI_MESSAGE_LOG table with the help of the batch_id which was used during execution.

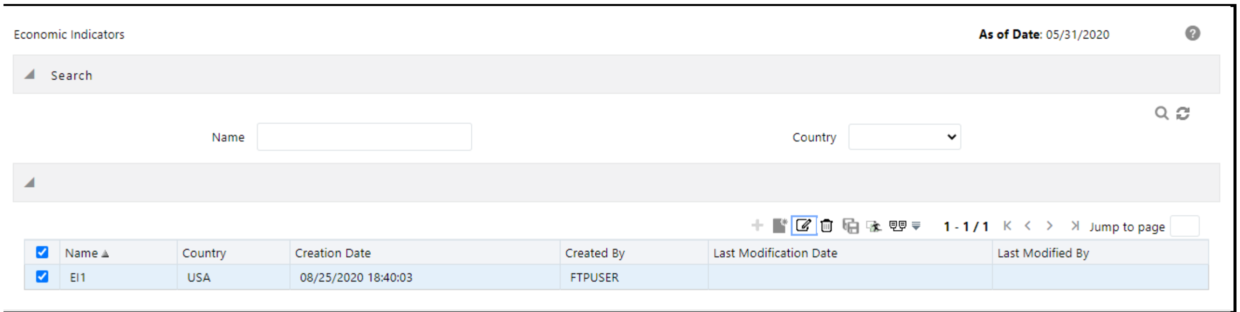
11.6 Economic Indicators Summary Page

An economic indicator is any economic statistic such as the Consumer Price Index (CPI), growth rate of the Gross Domestic Product (GDP), unemployment rate, Purchasing Managers Index, indices of consumer confidence, and so on. 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 can 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.

To view the Economic Indicators, navigate to **Common Object Maintenance** and select the **Rate Management**, and then select **Economic Indicators**, an empty window is displayed. After you have defined one or more Economic Indicators, the Economic Indicators Summary page shows all the Economic Indicators that you have previously built.

Figure 59: Economic Indicators Summary page



11.6.1 Searching for Economic Indicator

A **Search** pane is provided in which you can search for Economic Indicators by **Name** or **Country**. Each Economic Indicators rule is specific to one country.

To search the Economic Indicator, follow these steps:

1. Enter search criteria and click the **Search** icon.
2. Click the **Reset** icon to remove any Name or Country search criteria and refreshes the window.

   The search results are displayed in a tabular format containing all of the Economic Indicators that meet search criteria. The Economic Indicators Summary page offers several icons to perform different functions when an Economic Indicator is selected.

   - **Add**: Click Add to begin the process of building a new Economic Indicator. The Add icon is disabled if any rows in the pane are selected.
   - **View**: Select a single row out of the pane to enable the View. Click the View icon to view the contents of an Economic Indicator on a read-only basis. The View icon is only enabled when a row is selected.
   - **Edit**: Select a single row out of the pane to enable the Edit. Click the Edit icon allows you to modify a previously saved Economic Indicator. The Edit icon is only enabled when a single row is selected.
   - **Delete**: Select one or more rows out of the pane to enable the Delete. Clicking on the Delete deletes the Economic Indicators you have selected.
   - **Copy**: Select a single row out of the pane to enable the Copy. Click the Copy icon to create a copy of an existing economic indicator. The Copy icon is only enabled when a single economic indicator is selected.
   - **Check Dependencies**: Select an Economic Indicator and then click the Check Dependencies icon to generate a report on all rules that utilize your selected economic indicator.

     The Check Dependencies is only enabled when a single economic indicator is selected.

- **Data Loader**: The Data Loader option executes a function to import historical economic indices for all defined Economic Indicators. For more information on setting up the automated process, see the OFS Data Model Utilities User Guide.

  To execute a data load, click on the **Data Loader**.

  A warning message will appear Upload all available Economic Indicators? Click Ok, and all historical indices will be loaded.

To select an Economic Indicator, select a check box in the first column of the pane. More than one Economic Indicators can be selected at a time but this will cause some of the icons to become disabled.

Select or deselect all of the Economic Indicators in the Economic Indicators Summary page by selecting the check box in the upper left-hand corner of the Summary page directly to the left of the Name column header.

The following columns categorize each Economic Indicator on the Summary page:

- **Name**: Displays the Economic Indicator's short name. Performing a mouse-over on a row within the pane 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.

11.6.2 Economic Indicators Details

When you **Add**, **Edit**, or **View** an Economic Indicator, the **Economic Indicator Details** window is displayed. The **Economic Indicator Detail** window includes an **Economic Indicator Details** pane, the **Economic Indicators - Historical Data** pane, and an **Audit Trail** pane.

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

11.6.3 Adding an Economic Indicator

To add an Economic Indicator, follow these steps:

1. Click Add from the Economic Indicator Summary page.

Figure 60: Adding a new Economic Indicator

2. Enter the following information in the Economic Indicator Details window as tabulated.

   - **Name**: The name of your Economic Indicator is how you will subsequently refer to your rule within the OFS Analytical Applications. You cannot rename existing Economic Indicators.
   - **Frequency**: The frequency of your Economic Indicator must match the frequency which with the indicator's data is made public. Unemployment statistics, for example, are normally released on a monthly frequency. Select a frequency from the **Frequency** drop-down list. Available frequencies are **Weekly**, **Monthly**, **Quarterly**, **Semi-Annually**, and **Annually**.
   - **Value Type**: Select a Value Type from the **Value Type** drop-down list. Available Value Types are Numeric, Percentage, and Amount.
     - 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. The value set of Countries is drawn from the seeded Country dimension. OFSAA is seeded with over 70 country values, and you can add user-defined countries.

3. Click **Save**. To build out your historical data, enter data within the **Economic Indicators - Historical Data** pane.

   The Economic Indicators - Historical Data pane displays a single blank row followed by the most recent period's data (if data has previously been stored in the database).

Figure 61: Economic Indicators - Historical Data section on Economic Indicator window

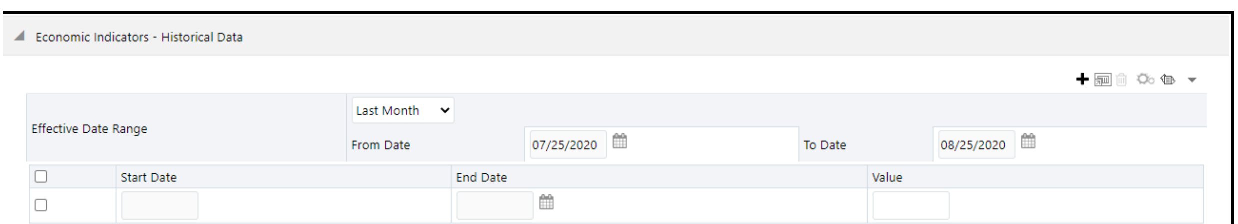


Table 17: Economic Indicators - Historical Data section - Fields and Descriptions

Fields	Description
Start Date and End Date	Select the **Calendar** icon 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 the 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 selected.
Value	Enter the value for your Economic Indicator (such as the unemployment rate).
Adding Multiple Data Points	Click **Add** to add additional blank rows into which you can enter additional Economic Indicator data. When you have finished adding data, click Save.
Editing Economic Indicators - Historical Data	Select a single check box on the left-hand side of any row to enable the Edit. Clicking **Edit** to become the row active. You can edit this row and subsequently save the changes.
Viewing Economic Indicators - Historical Data	By default, the Economic Indicators - Historical Data pane displays the most recent month of historical data. You can restrict the amount of data displayed by selecting a different value from the **Effective Date Range** drop-down list. You can also choose to view a specific date range by modifying the **From Date**, **To Date**, or both dates within the **Economic Indicators - Historical Data** pane.
Deleting Economic Indicators - Historical Data	Select one or more check boxes on the left-hand side of any row to enable the **Delete** icon. After clicking Delete, a confirmation message is displayed to delete the selected rows.
Data Loader	The Data Loader option executes a function to import historical economic indicators for all defined economic indicators. For more information on setting up the automated process, see the OFS Data Model Utilities User Guide. To execute a data load, click the Data Loader. A warning message will appear *Upload all available Economic Indicators?* Click **Data Loader** to execute all the Economic Indicators.
Excel Import and Export	Use Excel Import or Export functionality to add or edit Historical Economic Indicators.

12 Holiday Calendar

This chapter discusses the procedure to create a Holiday Calendar and generate a list of the weekend and holiday dates. Individual applications may consume the Holiday Calendar events in different ways. For more information, see the application-specific User Guides and the OFS Cash Flow Engine Reference Guide.

Topics:

- Overview of Holiday Calendars
- Searching for a Holiday Calendar
- Creating a Holiday Calendar
- Executing Holiday Calendar
- Holiday Exceptions

12.1 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:

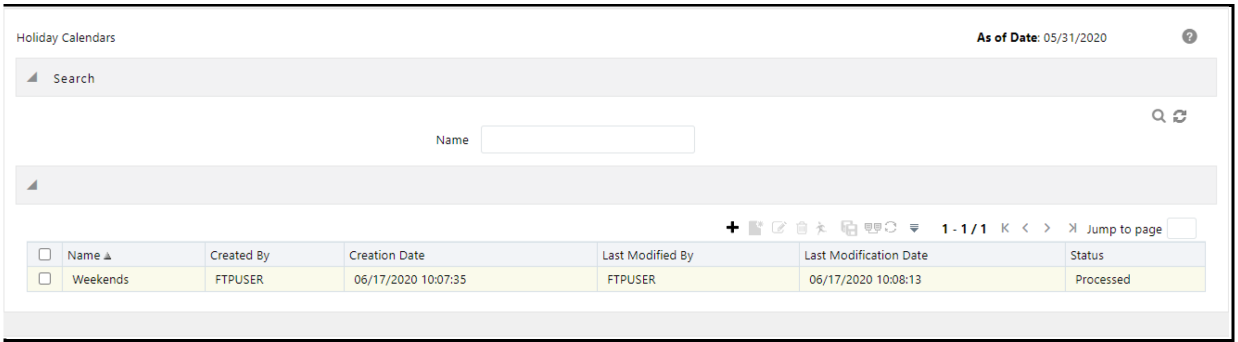
- Searching for a Holiday Calendar.
- Viewing and Updating a Holiday Calendar.
- Copying a Holiday Calendar.
- Deleting a Holiday Calendar.
- Check Dependencies in the Holiday Calendar definitions.
- Refresh the Holiday Calendar summary page.

12.2 Searching for a Holiday Calendar

Search for a Holiday Calendar to perform any of the following tasks:

- View
- Edit
- Copy
- Delete
- Check Dependencies
- Refresh

Figure 62: Holiday Calendars



You must have created a Holiday calendar to be able to search for Holiday Calendar rules.

To search for a new Holiday Calendar rule, follow these steps:

1. From the LHS menu, select **Common Object Maintenance**, and then select **Holiday Calendar** to display the **Holiday Calendars** Summary page. This page holds all Holiday Calendars and related functionality. You can navigate to other pages relating to the Holiday Calendar from this page.
2. Enter the name of the Holiday Calendar and click **Search**.

   Only Holiday Calendars that match the search criteria are displayed.

> **NOTE** You can control the number of rows to display on the screen by selecting the Pagination icons from the action bar.

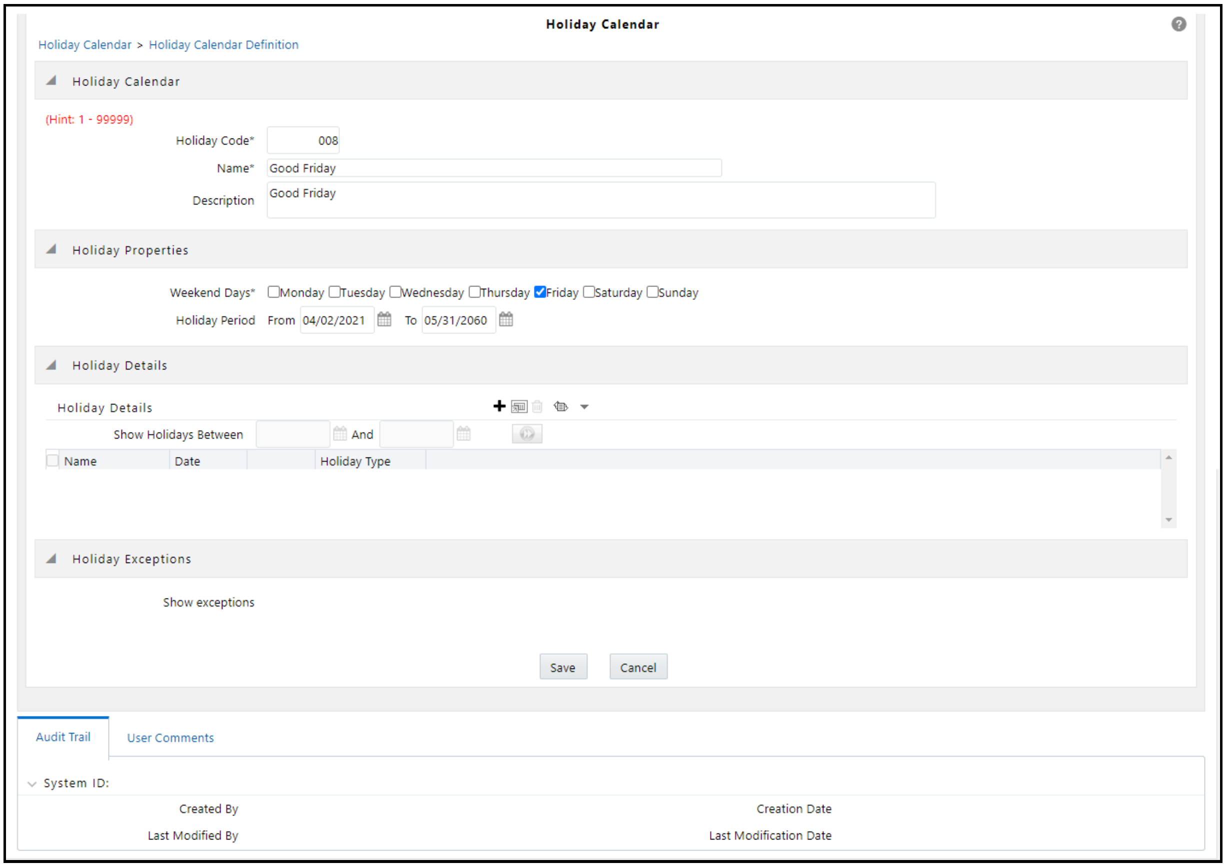
12.3 Creating a Holiday Calendar

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.

To create a new Holiday Calendar rule, follow these steps:

1. From the LHS menu, select **Common Object Maintenance**, and then select **Holiday Calendar** to display the **Holiday Calendars** summary page.
2. Click the **Add** icon. The Holiday Calendar Details page is displayed.

Figure 63: Holiday Calendar Definition



3. Enter a code value for the new Holiday Calendar.
4. 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.
6. The name you assign to the Holiday Calendar must be unique. The name can hold a maximum of 30 characters.
7. 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.
8. In the Holiday Details grid, define the Holiday details for any period within the holiday range defined in step 6. Two types of holidays can be defined: Fixed and Moving.
9. A fixed holiday is deemed as a holiday for every year in the holiday period, for that particular day.

   Example

   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 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. For example, 10th April 2020 is a moving holiday for Good Friday.

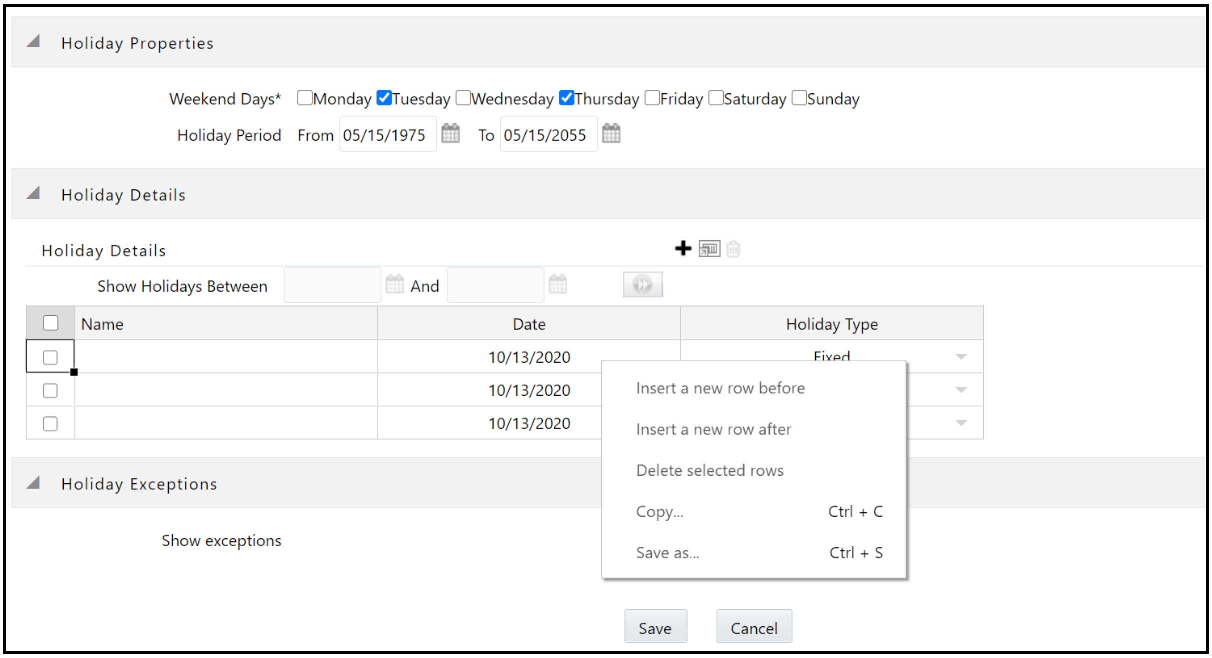
10. Once the Holiday Calendar definition is saved, its status in the Holiday Calendar Summary page is marked as defined.
11. A Holiday Calendar created can also be deleted. Select one or more check boxes of Holiday Calendar definitions and click Delete.

12.3.1 Excel Compatibility

This functionality is used for inserting Holiday Calendar definitions from an Excel sheet. You can right-click on any cell in the table and perform the following actions:

- Delete or Insert multiple rows
- Apply a Formula on a particular column
- Export the data to an Excel sheet using the **Save As** option.
- You can copy data from an Excel sheet and paste the same using standard copy, paste shortcuts Ctrl+C and Ctrl+V.

Figure 64: Holiday Properties Page



12.4 Executing Holiday Calendar

Execute a Holiday Calendar definition to generate calendar dates listing the various types of holidays for a given holiday period.

Prerequisites

Predefined Rules

To execute the Holiday Calendar, follow these steps:

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 are generated displays as Processed in the status column on the summary page.

   If 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

   e. There is a change in the definition of the Holiday Calendar.

   f. There is an update or modification to the Holiday Exceptions defined for that holiday calendar.

In such a case, you 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 a new definition.

12.5 Holiday Exceptions

You can specify exceptions to the holidays. As a prerequisite, a Holiday Calendar must be properly defined and the status of the Holiday Calendar in the summary page must 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 on 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.

1. Click Show Exceptions in the Holiday Exceptions pane. The Holiday Exceptions window opens.
2. The search block in the Exceptions page has the following fields:
   - **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 check box when selected lists all the exceptions, if already defined, for the holidays within the From, To Date range.
3. The search result gives the list of all holidays based on the selection of the above search criteria fields.
4. In the Holiday Exceptions pane, there are two types of exceptions that can be defined: Not a holiday and Shift to.
5. Any holiday can be marked as not a holiday, in which case that day is removed from the Holiday List. If you select Not a Holiday from the Exception Type drop-down, then the Shift to date field is disabled.
6. Spring earlier considered as a holiday in the Holiday Calendar can be marked as Not a Holiday in the Holiday Exceptions window. You can write your comments or remarks in the Notes next to the Exception Type drop-down list.
7. Any holiday can be shifted to another day, in which case the earlier declared holiday is removed from the Holiday List, while the shifted day is included as a holiday.

13 Overview of Oracle Insurance Allocation Manager for Enterprise Profitability

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
- Flexible definition of cost accounting processes with strong accounting controls
- Powerful modeling capabilities

Oracle Insurance Allocation Manager for Enterprise Profitability (PFT INS) addresses these requirements by linking general ledger, account-level, and statistical data together in a unique manner.

13.1 Multiple Dimensions of Oracle Insurance Allocation Manager for Enterprise Profitability

Oracle Insurance Allocation Manager for Enterprise Profitability 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 Oracle Insurance Allocation Manager for Enterprise Profitability.

13.1.1 Organizational View

An Organizational View facilitates the measurement of profitability at any level in your organizational structure, such as Financial Institutions, Insurance institutions, Bank, Region, Company, Legal Entity, or Division levels; or at lower levels such as Branches, 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 Management Ledger data (where your Management Ledger 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 (example, 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 (example, Cost Center plus Company) to populate a compound Organizational Unit dimension in your OFSAA data model.

13.1.2 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, and so on

13.1.3 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, and so on.

13.1.4 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.

13.1.5 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.

13.1.6 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)

13.1.7 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:

- 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 Enterprise Financial Performance Analytics (OFS EFPA) application.

For a complete listing of all seeded Financial Elements, see 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.

13.1.8 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.

13.1.9 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.

13.1.10 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.

13.1.11 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 products with Oracle Insurance Allocation Manager for Enterprise Profitability definitions of the product (see Debit and Credit Conventions).

13.1.12 Product

Nearly every Oracle Insurance Allocation Manager for Enterprise Profitability implementation is interested in a Product view of profitability.

13.1.13 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 OFS Advanced Analytical Applications Infrastructure Installation and 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 Oracle Insurance Allocation Manager for Enterprise Profitability engine can only actively utilize Key Processing Dimensions within allocations rule sources, drivers, or outputs.

13.1.14 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.

13.1.15 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).

13.2 Overview of Oracle Insurance Allocation Manager for Enterprise Profitability Methodologies

13.2.1 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 expenses?
- How much is being spent on salary and benefits expenses?
- 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?

13.2.2 Management Accounting Models

Oracle Insurance Allocation Manager for Enterprise Profitability 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 are 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.

13.2.3 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 begins 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, an so on.

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.

13.2.4 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).

13.3 Initial Loads to the Management Ledger

When you initially load the Management Ledger 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 Management Ledger (that is, 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 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

13.3.1 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)

13.3.2 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 and 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 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 into 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 subtotaling 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, and so on.) 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 (example, 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 (for example Accrued Interest Payable or Receivables, Fee income, and so on.)

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)

13.3.3 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 a Line of Business), if you build out selected balances, those balances could be used in ratio analysis either within the OFS EFPA application or within your own reporting tool. The OFS EFPA application includes predefined cube definitions that generate ratios and formulas that use such balance-sheet balance reporting lines (for example, for risk-adjusted profitability ratios such as return on assets, risk-adjusted return on equity, net income after capital charge, and so on).

14 Application Preferences

In addition to Global Preferences, each of 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 Oracle Insurance Allocation Manager for Enterprise Profitability processes are run.

The procedure for working with and managing Application Preferences includes Updating Application Preferences.

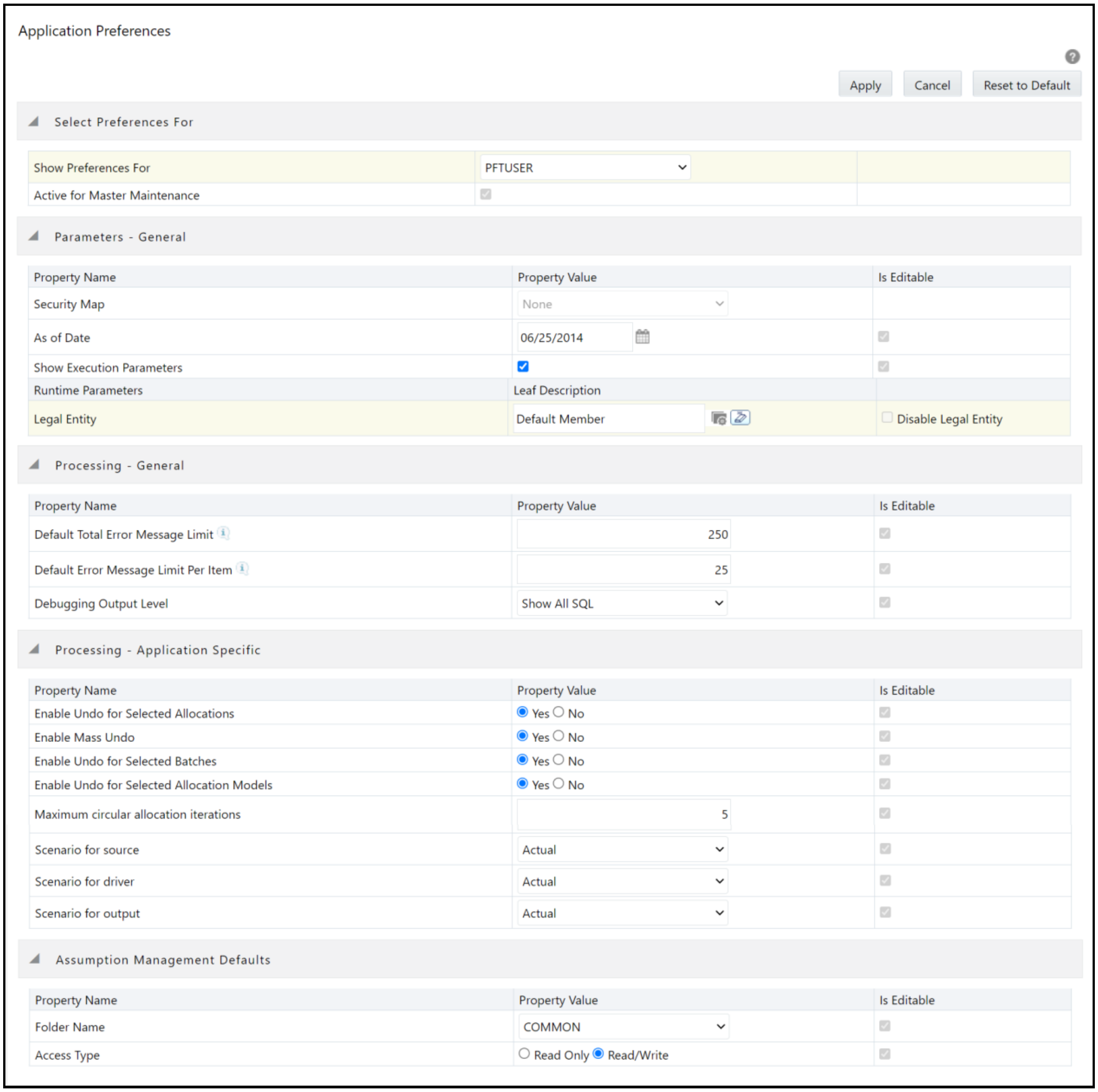
14.1 Updating Application Preferences

Updating Insurance Allocation Manager for Enterprise Profitability Application Preferences is a one-step process. You navigate to the Application Preferences screen and define your Preferences.

To update the Profitability Management application preferences, follow these steps:

1. From the LHS Menu, select **Insurance Allocation Manager**, and then select **Application Preferences** to display the Application Preferences page.

Figure 65: Application Preferences



2. Input values for all the Profitability Management Application Preference items as described in the following table.

Table 18: Fields and Descriptions from the Application Preferences Screen

Term	Description
Select Preference For	

Term	Description
Show Preferences For	There are two modes in which you can access Application Preferences: • **Administrator**: 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. • **End User**: 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.
Active for Master Maintenance	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.
Parameters - General	
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, see the Oracle Financial Services Analytical Applications Infrastructure User Guide.
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).

Term	Description
Legal Entity	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 the 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. **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. No additional parameter is required for the 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 the command line. The default legal entity is a per-user preference setting. To set this, the following steps are required: 1. Create a security map in AAI containing the legal-entity hierarchy as an app-admin. 2. Set the default security map in user preferences. 3. Navigate to the "execution parameters" block in user preferences and choose the default legal-entity member. 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. Default implies -1 code. The default value for the Legal Entity dimension column in all instrument and ledger tables is -1. **NOTE**: 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. Legal Entity is not applicable to allocation rules using the Management Ledger table.
Disable Legal Entity	If the "Disable Legal Entity" checkbox in the application preferences screen is checked, then Legal Entity will no longer be a run-time parameter. On disabling Legal Entity, the 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.
Processing - General	
Default Total Error Message Limit	Oracle Insurance Allocation Manager for Enterprise Profitability 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.

Term	Description
Debugging Output Level	The debugging output level determines the amount of SQL that will be written to the processing log. There are three levels available: • **Do not output SQL to log file**: A log file will not be created. • **Show Significant Calculation SQL**: Log file is created and will contain those SQLs that are tagged as significant. • **Show all SQL**: Log file is created and will contain all the SQL that the engines execute. The Allocation Engine log files can be accessed by the system administrator in the following location on the server: • **$OFSAA_LOG_HOME/<MIS_DATE>/<INFODOM>/Allocation Engine/** folder The file names will be prefixed with the application initials and will also contain the unique batch run id of the execution request. For example: **Allocation Engine_<INFODOM>_Task1_<Allocation SYSID>.log** To UNDO log files can be accessed by the system administrator in the following location on the server: • **$OFSAA_LOG_HOME/<SYS_DATE>/<INFODOM>/Allocation Engine/** folder. The file names will be prefixed with the application initials and will also contain the unique batch run id of the execution request.
Processing - Application Specific	
Enable UNDO for Selected Allocations	Set this value to Yes to enable UNDO functionality for individual allocation rules. For details, see Allocation Execution History.
Enable Mass UNDO	Set this value to Yes to enable Mass UNDO functionality. For details, see Allocation Execution History.
Enable UNDO for Selected Batches	Set this value to Yes to enable UNDO for selected batches. For details, see Allocation Execution History.
Enable UNDO for Selected Allocation Models	Set this value to Yes to enable UNDO for selected allocation models. For details, see Allocation Execution History.
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 the Threshold Source Balance or Threshold Percentage you have selected for your model. For details, see Allocation Models.

Term	Description
Scenario Values	The Application Preference setting for the 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. • **Scenario for Source**: When an allocation rule's Source is configured as Use Application Preferences in the Allocation Source section, the value used by the rule in its Source is the one specified in Application Preferences. • **Scenario for Driver**: When an allocation rule's Driver is configured as Application Preferences in the Allocation Source section, the value used by the allocation rule in its Driver is the one specified in Application Preferences. • **Scenario for Output**: When an allocation rule's Output (applicable to both Debit and/or Credit) is configured as Use Application Preferences in the Dimension section, the value used by the allocation rule is the one specified in Application Preferences.
Assumption Management Defaults	
Folder Name	This parameter allows you to define the default folder selection. The folder selection for all rule types will default 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.
Access Type	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.

3. Click **Apply** to save the changes. Alternatively, click **Reset to Default** if you like to clear all the previously applied inputs and return to the original default state.

15 Insurance Allocation Manager Configuration

The PFT Configuration UIs allow both Administrators and End Users to establish default values and to manage other core configuration parameters.

Topics:

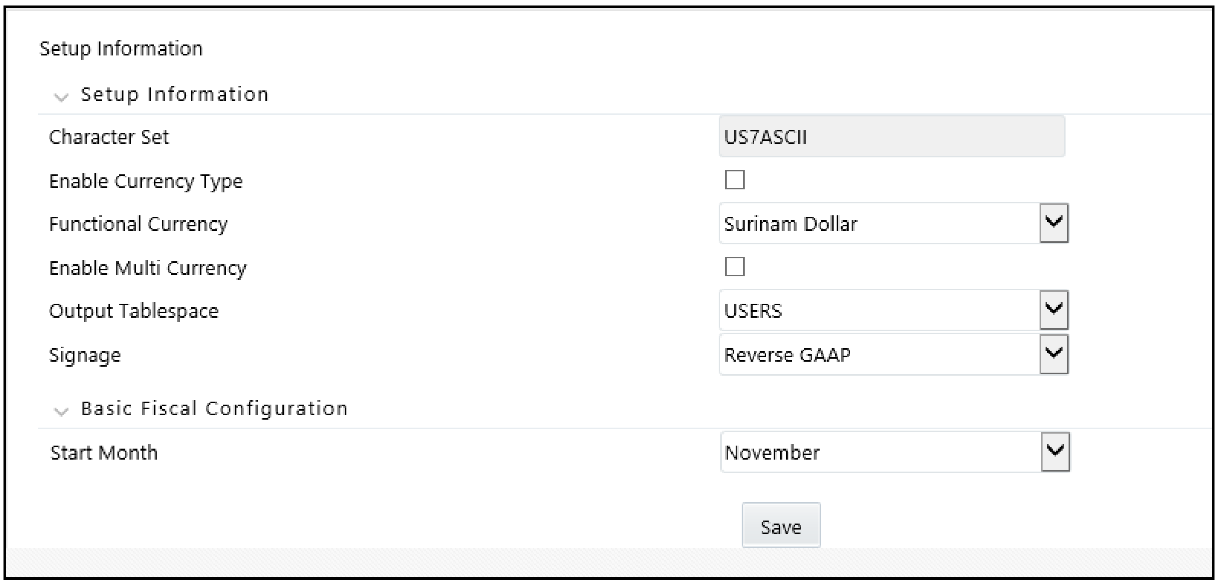
- Setup Information
- Parallel Execution Configuration
- Setup Parameters
- Management Ledger Configuration
- SQL Hint Configuration

15.1 Setup Information

To configure the Setup Information, follow these steps:

1. Select **Insurance Allocation Manager** from the LHS menu, select **Configuration**, and then select **Setup Information**.

Figure 66: Setup Information page



2. Select the **Enable Currency** check box to enable the currency type.
3. Select the relevant **Functional Currency**. Accounts and instruments can be defined across various currencies, but to consolidate the accounts' balances or charges at multiple hierarchy levels, a common functional currency is required which can be set here.
4. Select the **Enable Multi Currency** check box to enable multi-currency.

   The Output Table space does not apply to the PFT application.
5. Select a relevant option for Signage from the following options:

- Natural Signage
- GAAP Signage
- Reverse GAAP Signage

This is an "information only" setting describing how balances are carried.

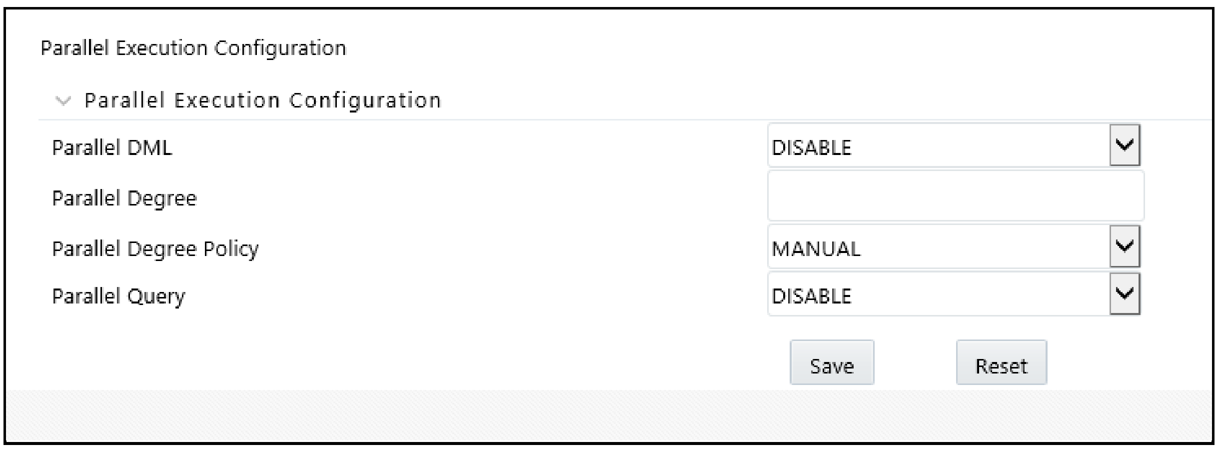
6. Select a **Start Month** of the fiscal year. This helps to define the start of a fiscal year. A fiscal year is a one-year period that customers use for financial reporting and budgeting. A fiscal year is most commonly used for accounting purposes to prepare financial statements. Although a fiscal year normally starts on January 1st and ends on December 31st, you can change it as per your local norms using this field.
7. Click **Save** to save the configuration.

15.2 Parallel Execution Configuration

To configure Parallel Execution, follow these steps:

1. Select **Insurance Allocation Manager** from the LHS menu, select **Configuration**, and then select the **Parallel Execution Configuration**.

Figure 67: Parallel Execution Configuration



2. Select a relevant option for Parallel DML from the following:
   - Disable
   - Enable
   - Force
3. Enter a value for Parallel Degree. This field accepts an integer between 0 and 99.
4. Select a relevant option for Parallel Degree Policy from the following:
   - Manual
   - Limited
   - Auto
5. Select the relevant option for Parallel Query from the following:

- Disable
- Enable
- Force

6. Click **Save** to save the configuration.

For more information, see the Profitability Management Parallel Execution section.

15.3 Setup Parameters

To configure setup parameters, follow these steps:

1. Select **Insurance Allocation Manager** from the LHS menu, select **Configuration**, and then select **Setup Parameters**.

Figure 68: Setup Parameters

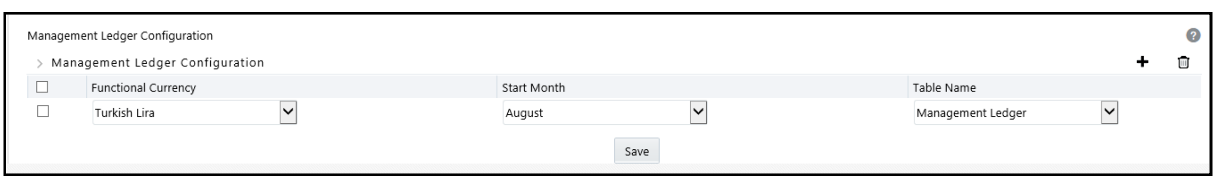
2. Enter the following details:
   - **STRATIFICATION_SCALE_FACTOR**: This parameter is not used by the PFT application.
   - **STATIC_TABLEID_TEMP_TABLE_CROSS_JOIN_LIMIT**: Integers between 0 and 99. The default value is 20 million. If the Cartesian product of the leaf values selected for each dimension exceeds this limit, then the static table driver UI displays only the already defined combinations. Further reducing this parameter value improves the UI performance, since UI only displays the already existing definitions.

15.4 Management Ledger Configuration

To configure the management ledger, follow these steps:

1. Select **Insurance Allocation Manager** from the LHS menu, select **Configuration**, and then select the **Management Ledger Configuration**.

Figure 69: Management Ledger Configuration



2. Select the following details:

- **Functional Currency**: The currency that applies to the Table that you select. This helps to define a functional currency for the ledger table. Accounts and instruments can be defined across various currencies, but to consolidate the accounts at multiple hierarchy levels, across units, a common functional currency is required which you can set here.
- **Start Month**: The start month of the fiscal year. You can choose a ledger table and define the corresponding start month or functional currency.
- **Table Name**: The name of the table.

3. Click **Save** to save the configuration.

For more information on functional currency, see the Functional Currency section.

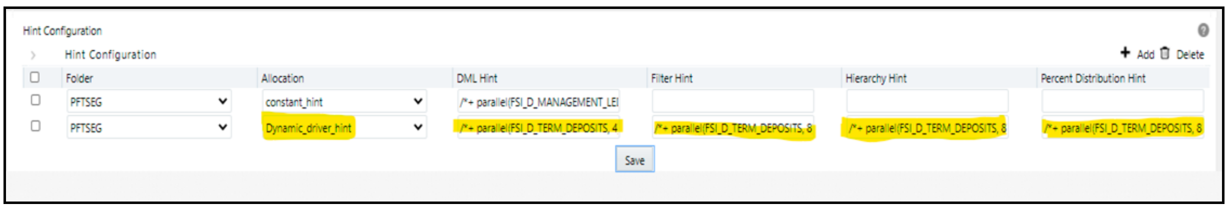
15.5 SQL Hint Configuration

This screen allows you to define SQL hints which will be used by the Allocation engine while forming the SQL. Stores into table FSI_SQL_HINTS_OPTIONS.

To configure the SQL Hits, follow these steps:

1. Select **Insurance Allocation Manager** from the LHS menu, select **Configuration**, and then select **SQL Hint Configuration**.

Figure 70: SQL Hint Configuration



2. Click the **Add** icon to add a new SQL Hint Configuration.
3. Select the following:
   - Folder
   - Allocation
   - DML Hint
   - Filter Hint
   - Hierarchy Hint
4. Click **Save**.

16 Allocation Specification

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

16.1 Summary and 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 detailed screen is displayed.

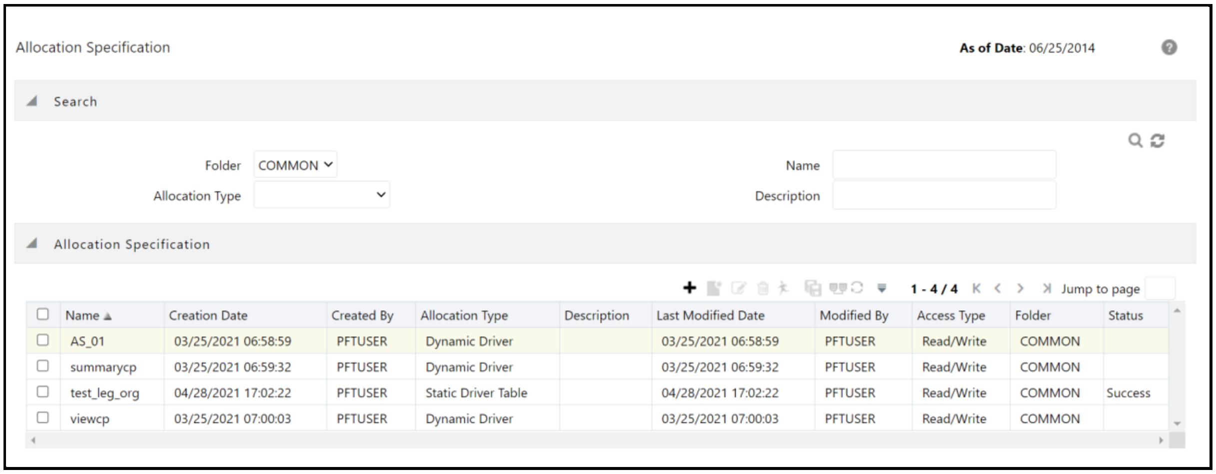
16.2 Navigation within the Summary Screen

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.

16.2.1 Search Container

Your default Folder functions as a search constraint. The value of your default Folder is set in Application Preferences for Oracle Insurance Allocation Manager for Enterprise Profitability. 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.

Figure 71: Allocation Specification Search Screen



- **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.

16.2.2 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 choosing 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 Oracle

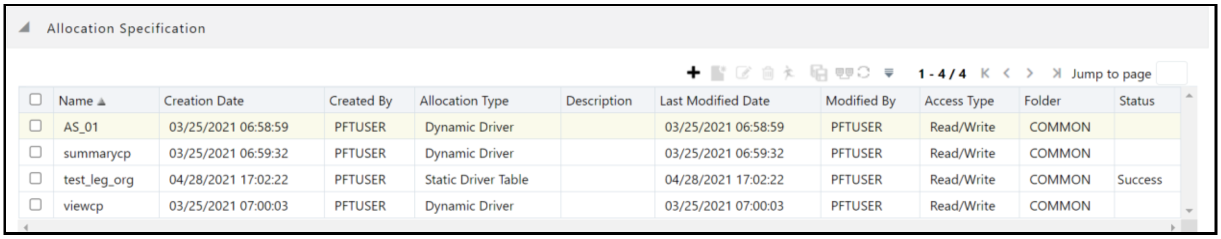
Insurance Allocation Manager for Enterprise Profitability. The Run control is only enabled when a single allocation rule has been selected.

- **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.

16.2.3 Allocation Specification Summary Pane

The following columns categorize each allocation rule in the summary Pane:

Figure 72: Allocation Specification Summary Pane



- **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 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**: Before having executed an Allocation rule for the first time, the 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.

16.3 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 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.

16.3.1 Process Tabs Container

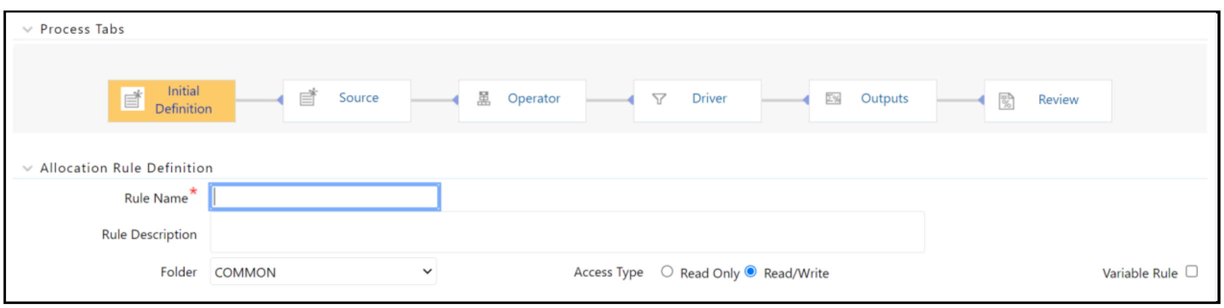
Each of the 6 process tabs is designed to create, edit, or view different components of an allocation's specification. You may navigate from any tab to any other tab at any time. The 6 process tabs are:

- Initial Definition
- Source
- Operator
- Driver
- Outputs
- Review

16.3.2 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 active 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.

Figure 73: Allocation Rule Definition Container



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.

16.4 Initial Definition Process Tab

The Initial Definition process tab allows you to specify Rule Name, Rule Description, Folder, Access Type, and Allocation Type.

16.4.1 Allocation Rule Definition Container

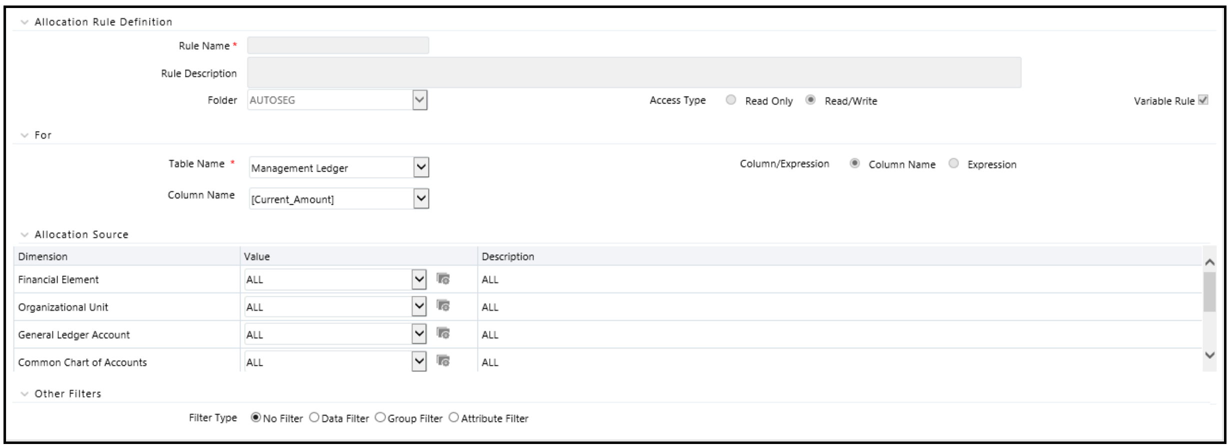
Specify the Allocation Rule Name and Description, select a Folder in which the Allocation rule is to be stored, and specify whether you want the Allocation rule to be “Read/Write” or “Read Only” (Access Type). Naming your Allocation rule is required before it can be saved. Default values for Folder and Access Type are stored in Application Preferences for Oracle Insurance Allocation Manager for Enterprise Profitability.

16.4.2 Variable Rule

Legal Entity is an optional run-time parameter. If the "Disable Legal Entity" checkbox in the application preferences screen is checked, then Legal Entity will no longer be a run-time parameter.

If the 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 the allocation specification screen.

Figure 74: Allocation Rule Definition



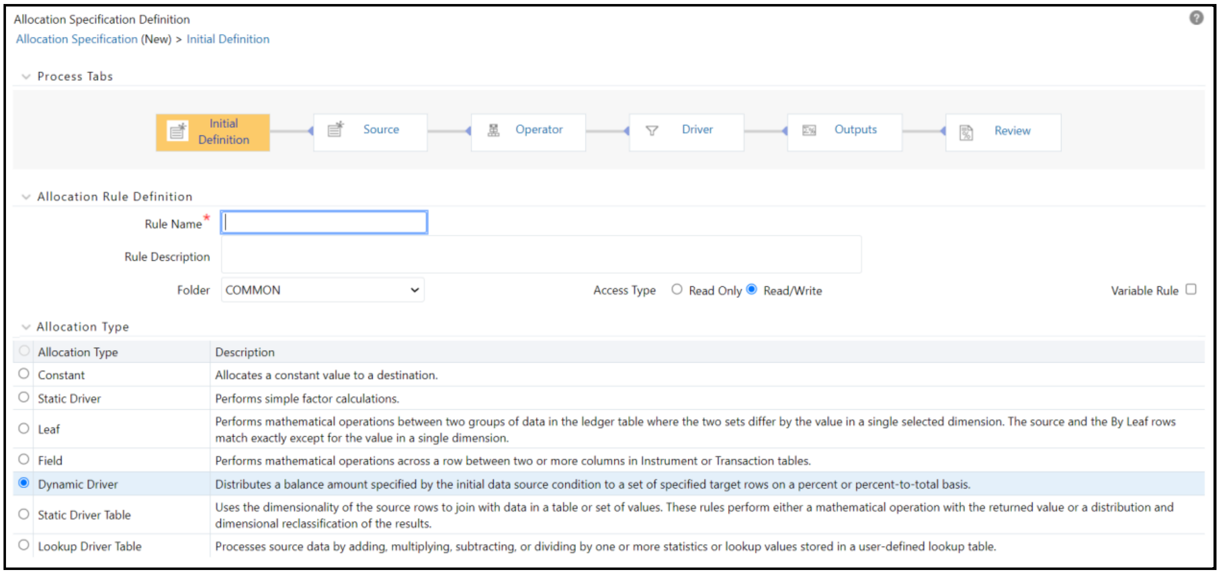
For the already defined Variable Allocation rules, the user should edit the allocations and modify the value of the Legal Entity dimension in all applicable tabs in the allocation specification screen.

Figure 75: Allocation Source Pane

The value of Legal Entity used when your rule executes is specified within your batch definition (for batch processes) or is obtained from your Oracle Insurance Allocation Manager for Enterprise Profitability 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 Management Ledger.

Figure 76: Allocation Specification (New) - Initial Definition



16.4.3 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.

16.4.4 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. Oracle Insurance Allocation Manager for Enterprise Profitability 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 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 to generate partial absorption costing.

16.4.5 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 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 operate 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, Oracle Insurance Allocation Manager for Enterprise Profitability 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 the Oracle Insurance Allocation Manager for Enterprise Profitability 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 the 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, and so on.

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.

16.5 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.

16.5.1 Management Ledger Source

The management ledger level refers to both the Management Ledger table. This section describes the usage of the Management Ledger as the Source in allocation rules. The Management Ledger is a seeded table of Management Ledger class of tables (see OFS 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). For details on the new Management Ledger table, see 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 based on your As-of-Date and Fiscal Year definitions. If your As-of-Date is set to any day in 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 the fiscal year, see Fiscal Year Information.

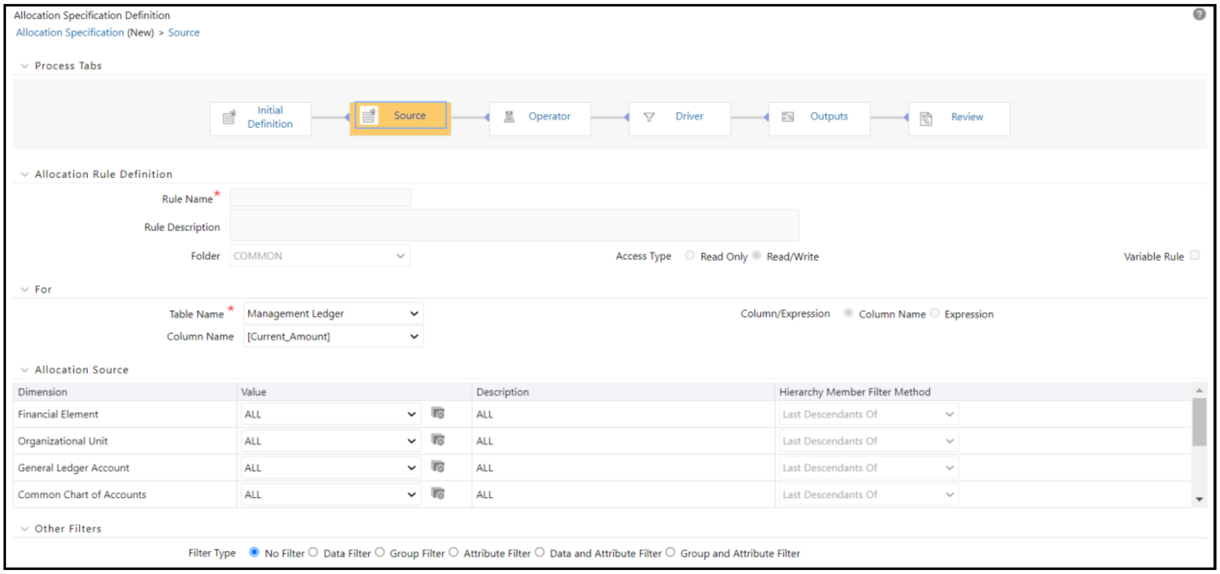
NOTE	Allocation rules that aggregate instrument-level data to the Management Ledger table maintain literal As-of-Dates when posting to the management ledger, but when data is retrieved from the Management Ledger table (i.e., in Allocation Sources or Allocation Drivers), it is consolidated into a month-to-date balance. Similarly, all outputs at the management ledger level are inherently month-to-date balances.

When your Source is the Management Ledger table, Oracle Insurance Allocation Manager for Enterprise Profitability 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 before 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 the fiscal year, see Fiscal Year Information.

Figure 77: Allocation Specification – Source Tab



16.5.2 Management Ledger Source

When your Source is the Management Ledger table, 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 Management Ledger. The <Current Amount> macro selects the current month from Management Ledger based on your As-of-Date and Fiscal Year definitions. If your As-of-Date is set to any day in March, <Current Amount> will generally be interpreted with FISCAL_MONTH as 3. If your fiscal year begins in April, however, your March data is stored in Management Ledger with FISCAL_MONTH as 12 since March is the last month in your fiscal year. For details on the fiscal year, see Fiscal Year Information.

In addition to <Current Amount>, Oracle Insurance Allocation Manager for Enterprise Profitability also offers macros for:

- <YTD Amount>
- <Last Month Amount>
- <Last Month YTD Amount>

16.5.3 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.

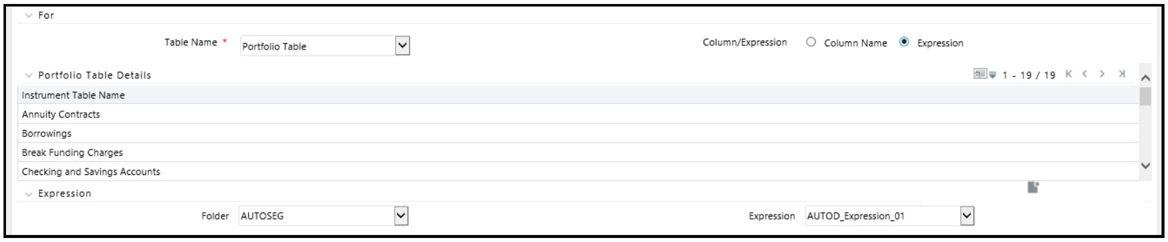
Figure 78: Instrument or Transaction Summary Source

16.5.4 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.

Figure 79: Expression Container



16.5.5 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.

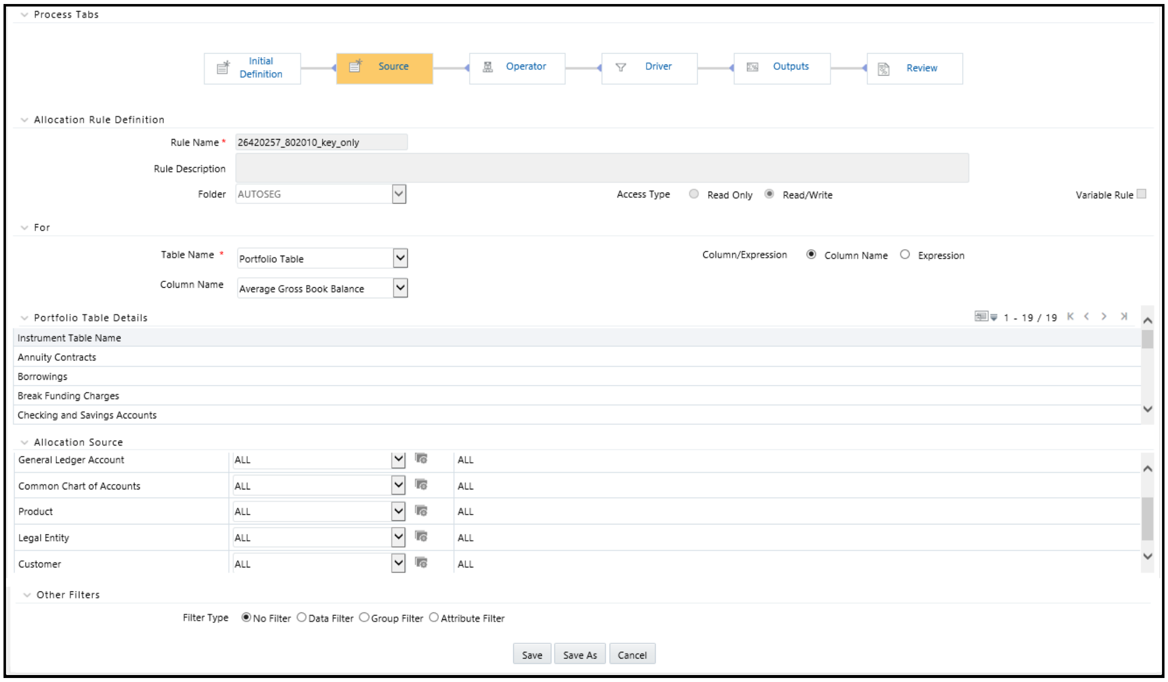
For allocation rules that source data from the Management Ledger-level, you must select a Source scenario from the Allocation Source container. The default for new allocation rules is <Use Application Preferences>. When you use this default value, the Scenario (also called Consolidation Code) is determined by the value that is set in Application Preferences for Oracle Insurance Allocation Manager for Enterprise Profitability 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 the 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 the Scenario is passed as a task parameter, it overrides the value set in Application Preferences. For details on passing task parameters, see the OFS Analytical Applications Infrastructure User Guide.

Figure 80: Source Tab



Some examples of how the Source tab is used to provide input data to an allocation rule include:

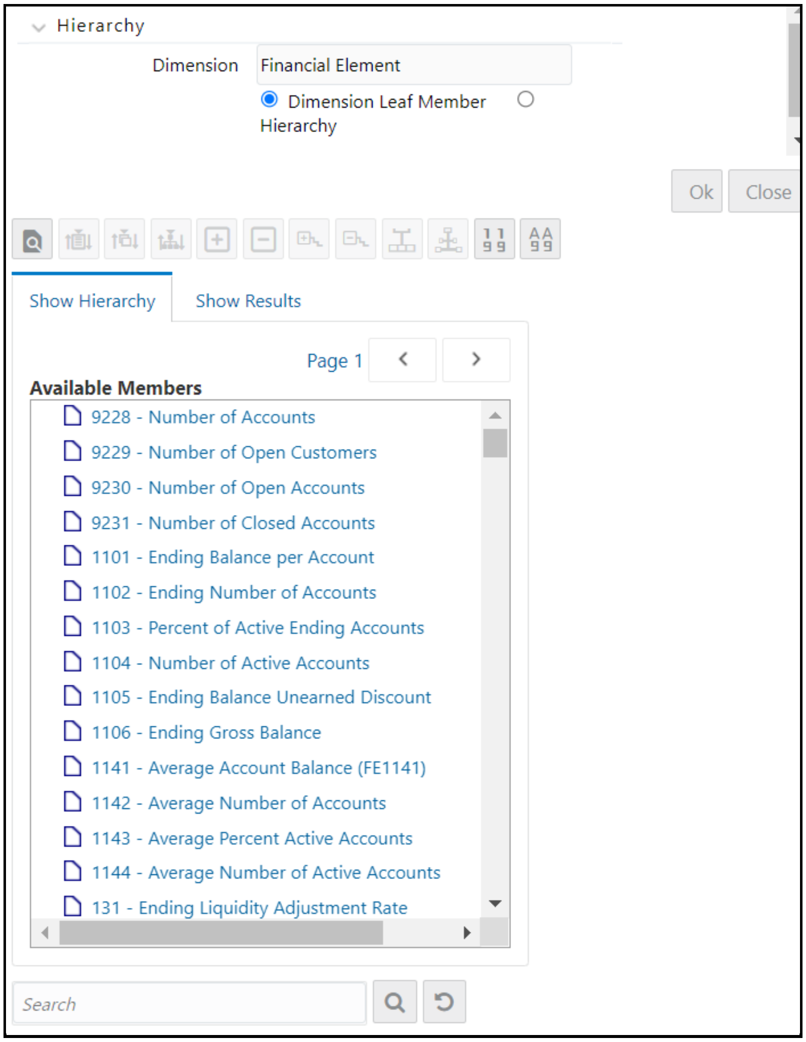
Desired Data	Constraint
Get all expenses for all GL Accounts within a specific cost center	Single leaf constraint on Organizational Unit plus single leaf constraint on Financial Element 457 – Non-Interest Expense
Get all current mortgage balances for adjustable rate products originated in the past year	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 an 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 multi-currency calculations within Oracle Insurance Allocation Manager for Enterprise Profitability, see Multi-Currency across OFSAA Applications.

16.5.6 Hierarchy Browser on the Source Tab

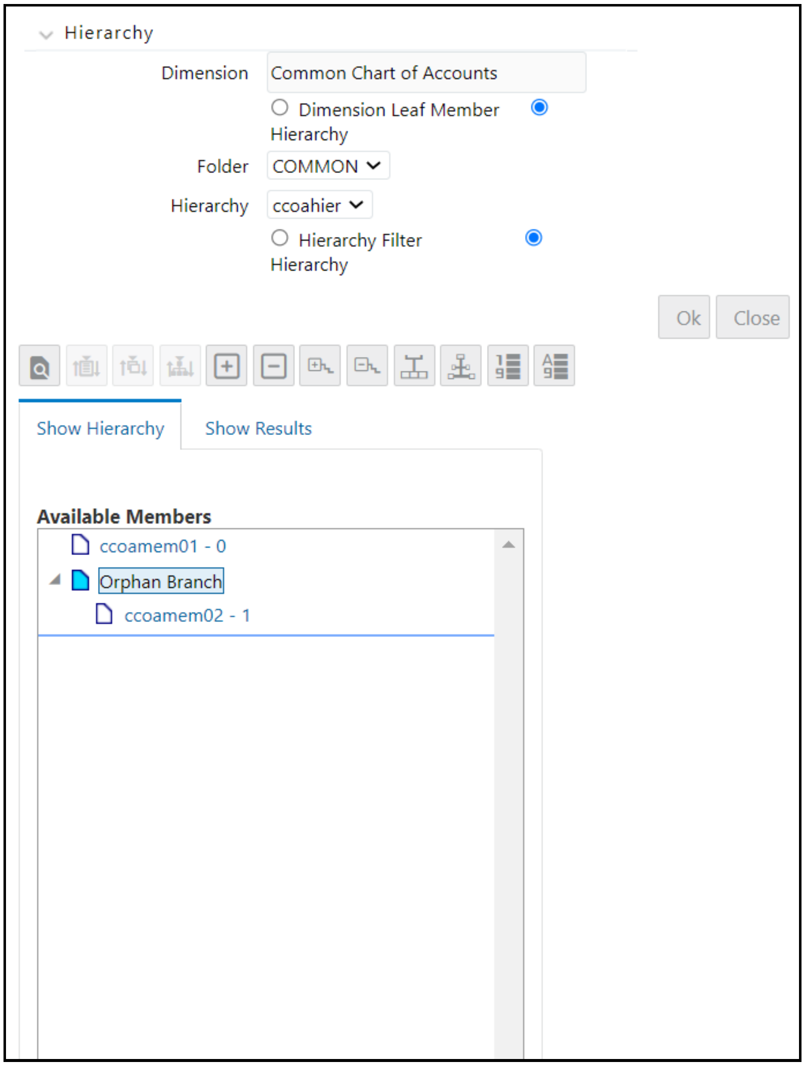
The default for each dimension is <All> (meaning no constraint). To select a constraint, click on the Hierarchy Browser ellipses (…) next to the dimension you wish to constrain.

Figure 81: Hierarchy Browser on Source Tab



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, Name, or Code.

Figure 82: Hierarchy Browser with Additional Search Parameters



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.

Figure 83: Hierarchy

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.

16.5.7 Other Filters Container

You may optionally select a Data Filter, Group Filter, or Attribute filter rule to further constrain your Source data.

16.6 Operator Process Tab

Generally, the Operator process tab allows you to specify how Source data and Driver data will interact to create results.

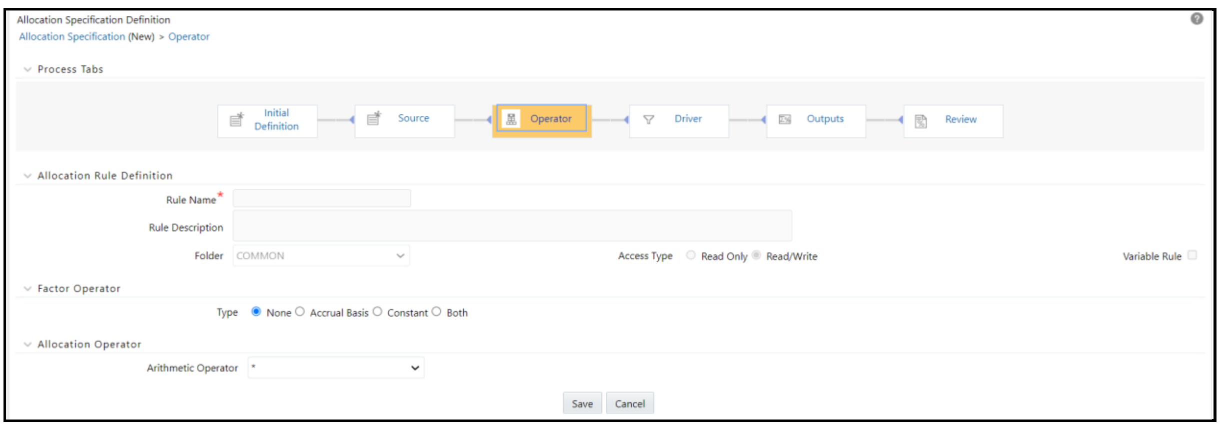
16.6.1 Operator Process Tab for Constant Rules

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.

16.6.2 Operator Process Tab for Static Driver Rules

For Static Driver rules, the Driver process tab is disabled, but the Operator process tab is enabled to allow you to specify a static driver balance. Static amounts are entered into the Factor Operator container.

Figure 84: Operator Process Tab



16.6.3 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.

16.6.3.1 Factor Operator Container

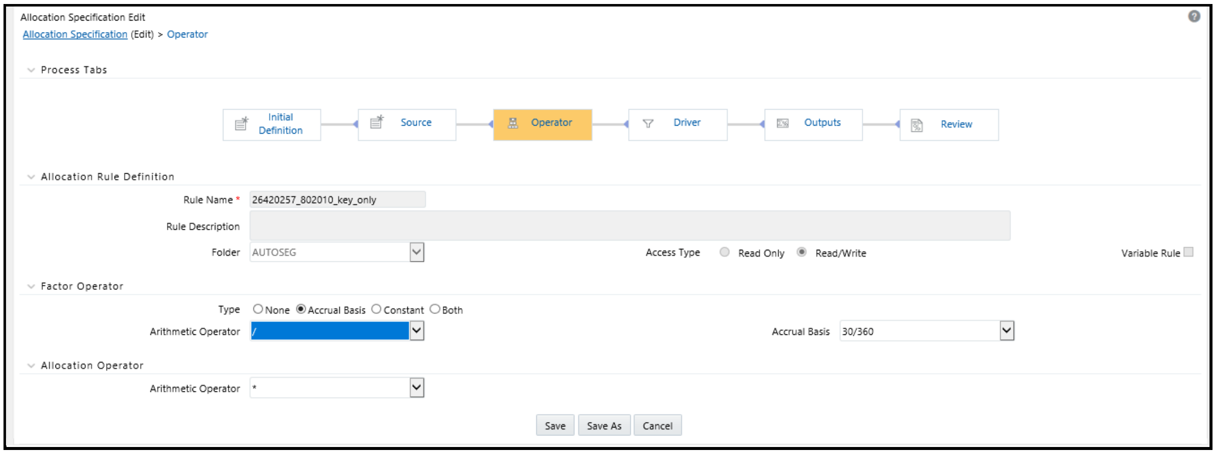
Factor Operators may be used to either

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.

Figure 85: Factor Operator Container



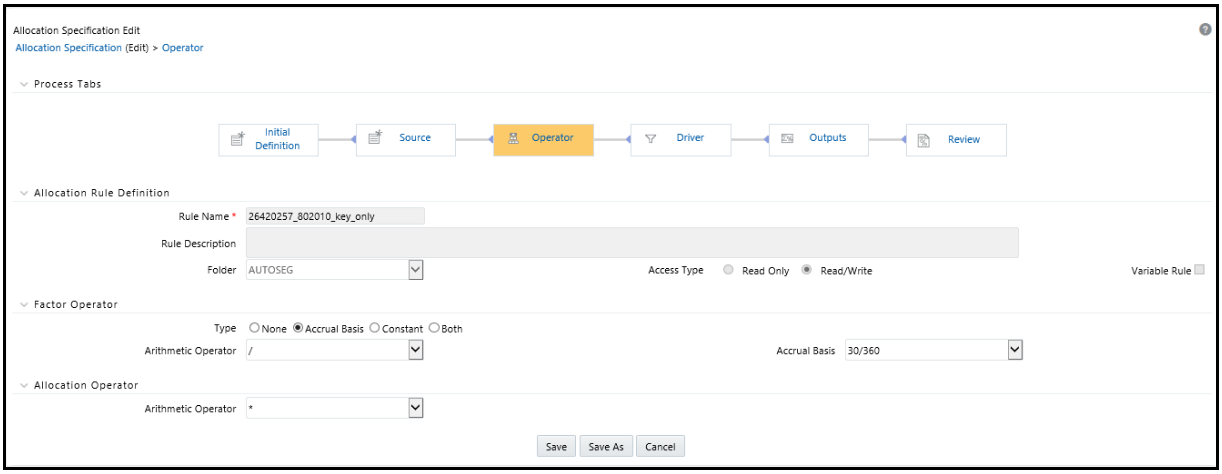
Examples of Usage of the Factor Operator

- 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 the total expense you are choosing to allocate.

16.6.4 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.

Figure 86: Allocation Operator Container



16.7 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 the 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.

16.7.1 For Container

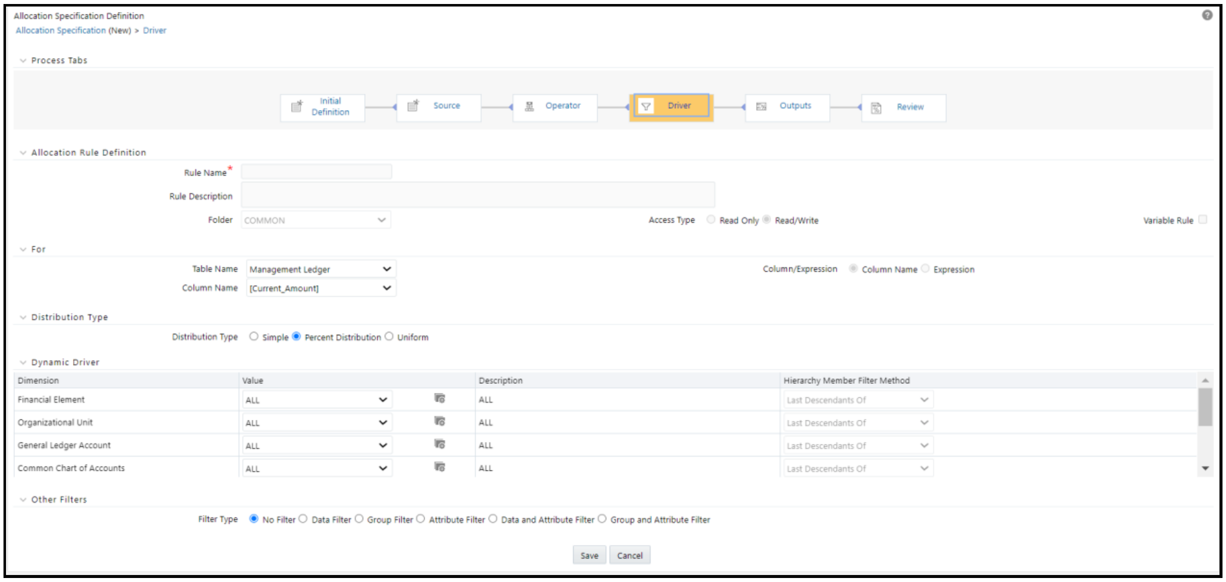
For the Leaf, Field, and Dynamic Driver allocation types, a For container is displayed in which you can choose either a driver table & column or an Expression to serve as the source of your Driver data.

16.7.1.1 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.

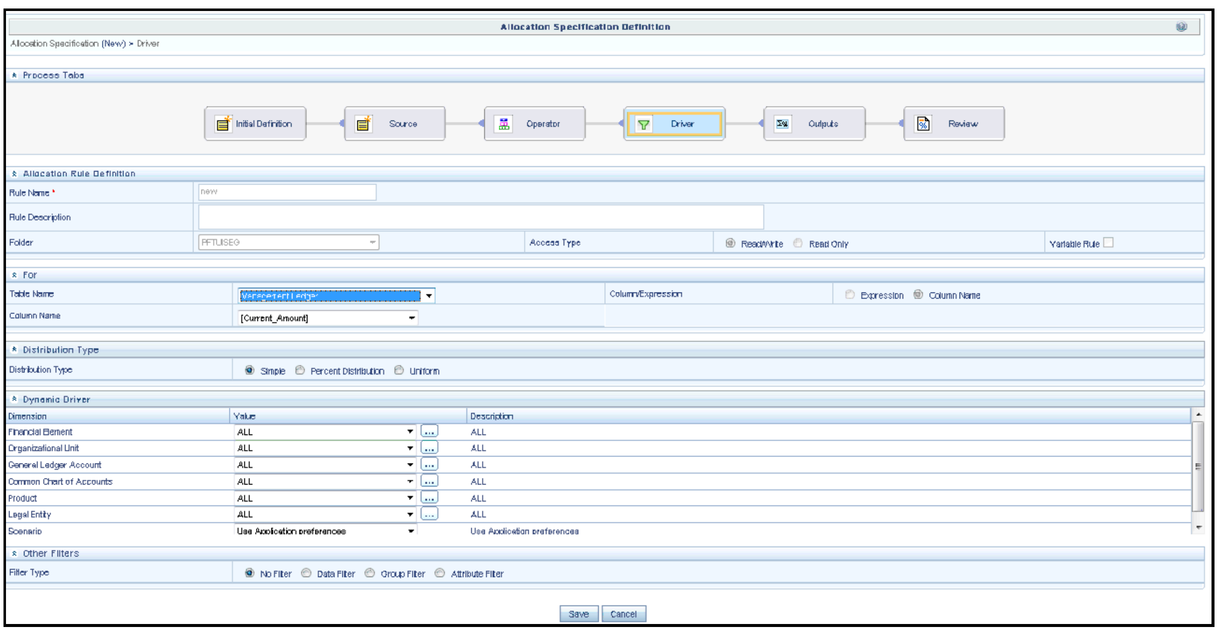
Figure 87: Leaf Driver Container



16.7.2 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.

Figure 88: Dynamic Driver Container



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, the Scenario (also called Consolidation Code) is determined by the value that is set in Application Preferences for Oracle Insurance Allocation Manager for Enterprise Profitability 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 the 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:

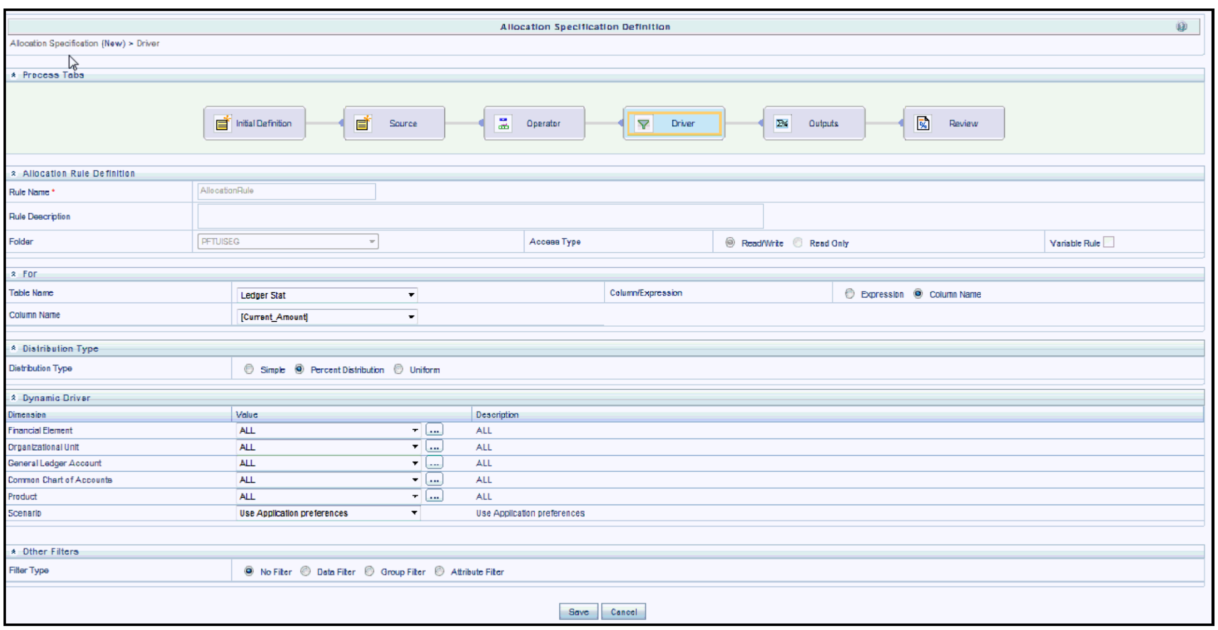
- a Simplified Batch,
- a standard ICC batch, or
- from the command line

When the Source value for the Scenario is passed as a task parameter, it overrides the value set in Application Preferences. For details on passing task parameters, see the OFS Analytical Applications Infrastructure User Guide.

16.7.2.1 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.

Figure 89: Distribution Type Container



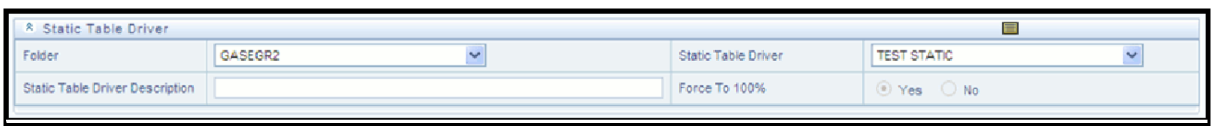
- **Percent Distribution Method**: The most common distribution method is Percent Distribution (sometimes referred to as Force to 100%). Typical use cases include expense allocations as a function of a driver set that has not been normalized, i.e., converted to percentages of the total driver set. For 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.

16.7.3 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 the Static Table Driver chapter.

Figure 90: Static Table driver

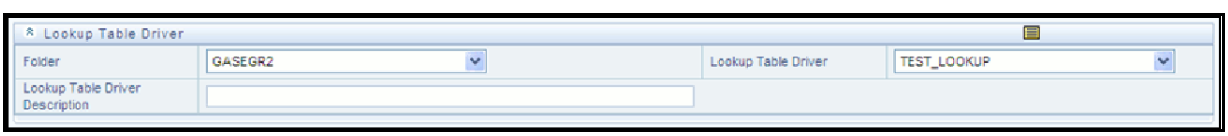


Note that once you have chosen a Static Table Driver rule, a View control is added to the Static Table Driver title bar. Click on this View control to view a read-only version of the Static Table Driver rule you have chosen.

16.7.4 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 Lookup Table Drivers.

Figure 91: Lookup Table Driver



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.

16.7.5 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.

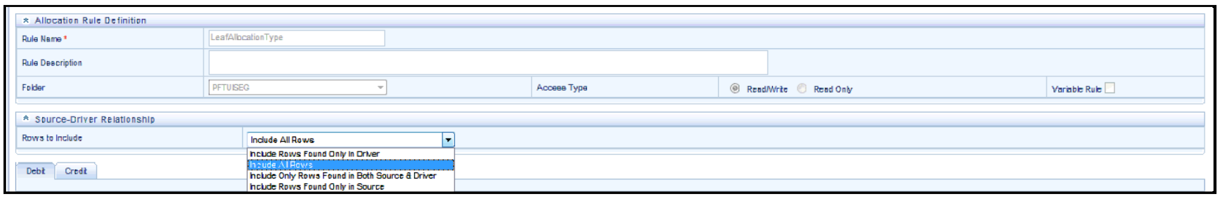
16.8 Outputs Process Tab

The Outputs process tab allows you to specify where the outputs of an Allocation rule are written. When outputting to the Management Ledger 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 the Initial Definition tab.

Figure 92: Outputs Process Tab - Source-Driver Relationship



There are four options 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.

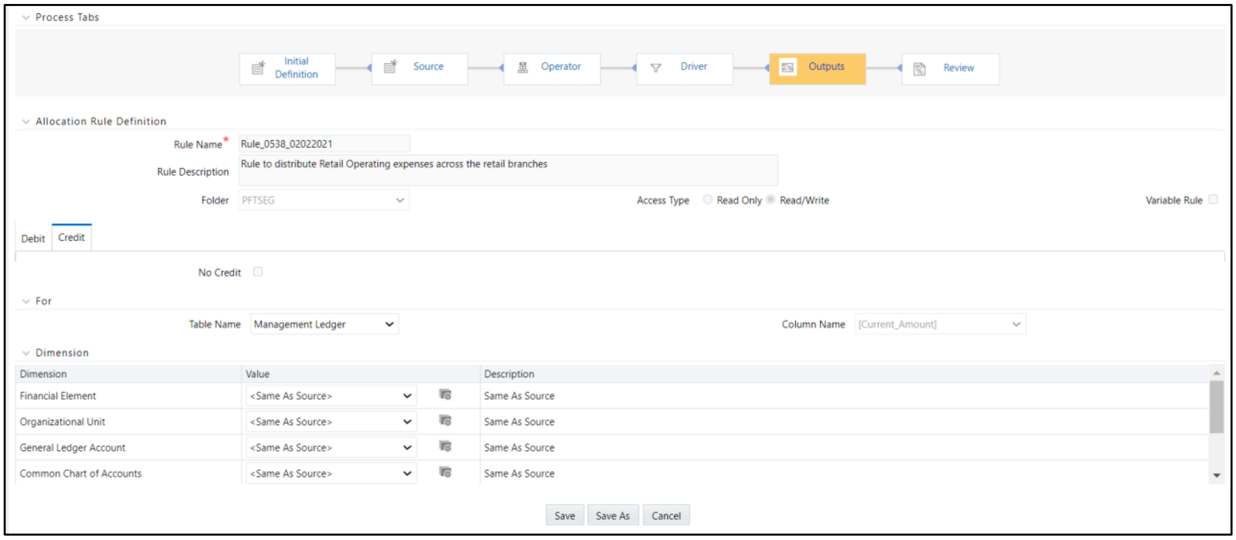
16.8.1 Debit/Credit Tab

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.

16.8.1.1 When Output to the Management Ledger-level

When posting allocation results to the Management Ledger-level, the Oracle Insurance Allocation Manager for Enterprise Profitability 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.

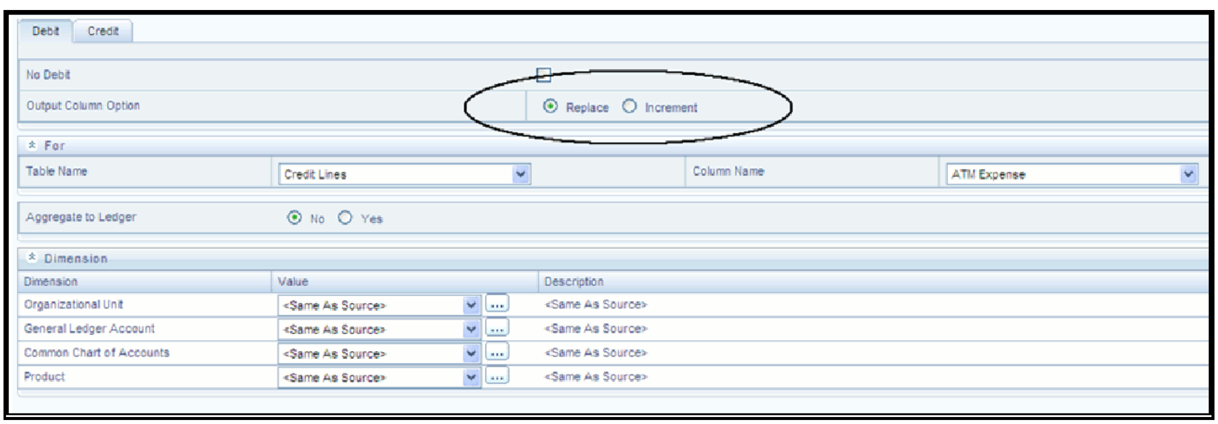
Figure 93: Outputs Tab



When the Output Table is Instrument-level: When using an allocation rule to update an Instrument or Transaction Summary table, the Oracle Insurance Allocation Manager for Enterprise Profitability 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 before release 6.0 is Replace.

Figure 94: Debit Tab



16.8.1.2 For Container

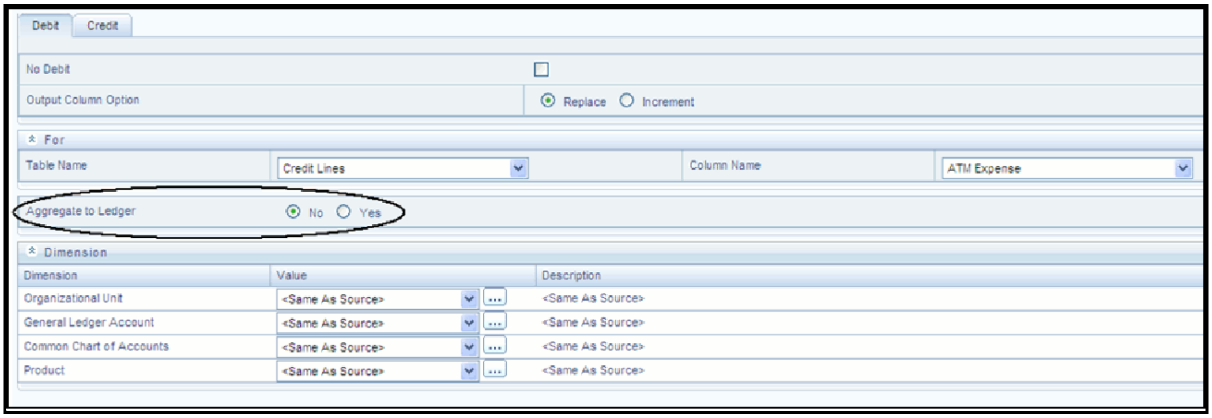
Specify the output table and column for the Allocation rule.

- 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.

16.8.1.3 Aggregate to Ledger

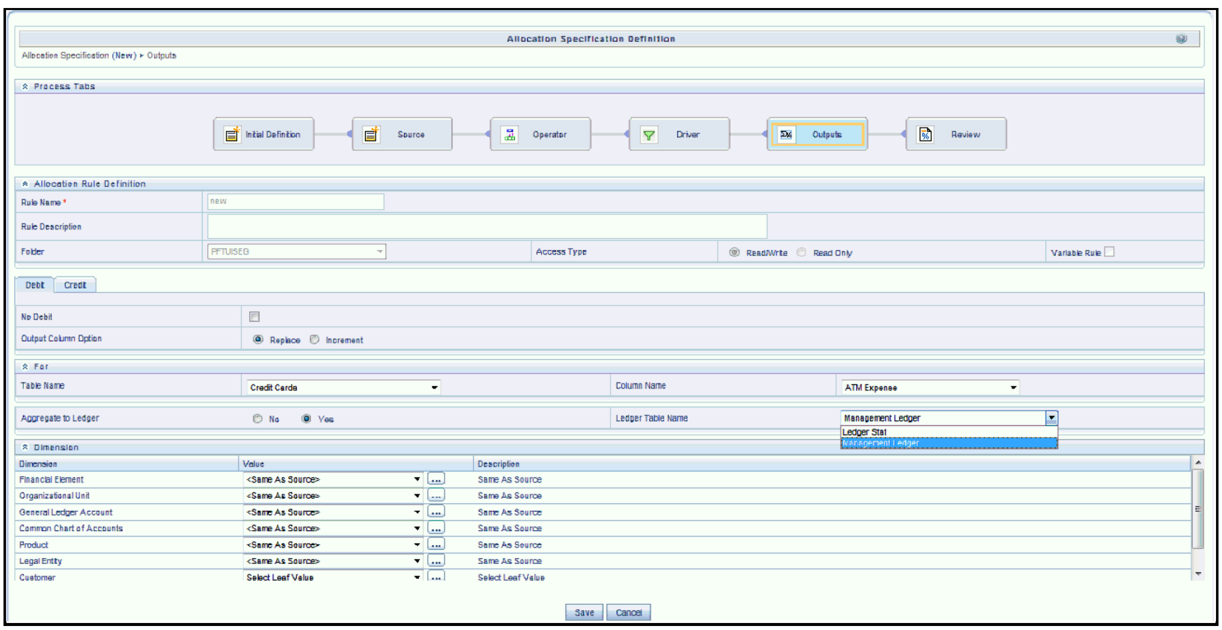
For Allocation rules that update an Instrument or Transaction Summary tables, you have the option of aggregating your results and posting them to the Management Ledger level.

Figure 95: Debit Tab – 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, 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.

Figure 96: Allocation Specification - Outputs



> **NOTE** Lookup Driver Table type allocation rules are capable of sending output to the Management Ledger level.

16.8.1.4 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.

16.8.1.4.1 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, the Scenario (also called Consolidation Code) is determined by the value that is set in Application Preferences for Oracle Insurance Allocation Manager for Enterprise Profitability 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 the 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 the OFS Analytical Applications Infrastructure User Guide.

16.8.1.4.2 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 Oracle Insurance Allocation Manager for Enterprise Profitability 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 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 to 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 and 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.

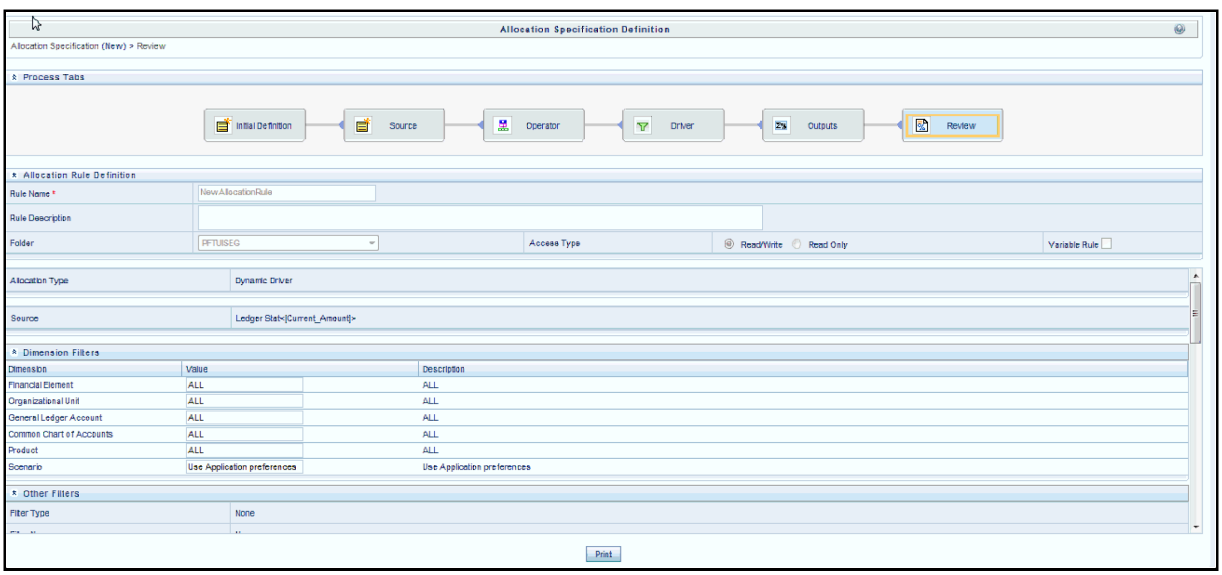
- **Lookup Driver Table Allocation Type**: For Lookup Driver Table allocations, you may choose either a specific leaf value or the <Same as Source> macro for each of your processing dimensions.

16.9 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.

Figure 97: Review/Process Tab



16.10 Allocation Examples

The examples are explained in detail in the following sections.

16.10.1 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.

16.10.2 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 to either enrich your ledger with a 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, and so on. 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 Management Ledger for Financial Element 100 (Ending Balance) using <Same as Source> for every dimension
- Debit Management Ledger 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 offsets. 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.

16.10.3 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 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.

16.10.4 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

16.10.5 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.

16.10.6 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 à Cost process, one Static Driver Table allocation rule could take the place of dozens or hundreds of Static Driver allocation rules.

16.10.7 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 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 that 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) 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) 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.

16.10.8 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 instrument-level balance times a rate times an accrual basis factor 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. If 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. 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.

16.10.9 Examples of Dynamic Allocations

Management Ledger Reclassification Using Instrument-level Driver Data

16.10.9.1 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 19: Summary of the Balances for Example 1

Table	GL Account	Branch	Product	Balance	# of Loans
Management Ledger	Commercial Loan	1	—	$1,000	—
Management Ledger	Commercial Loan	2	—	$2,000	—
Management Ledger	Consumer Loan	1	—	$3,000	—
Management Ledger	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
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, for example, 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.

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 rollup 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 Management Ledger (note: when posting outputs to Management Ledger, you MUST output to the < Current Amount > macro). Set < Same as Source > for each Key Processing Dimension.
5. Set the Debit Output to Management Ledger; 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.

Table 20: Summary of the Management Ledger Rows before and after the Allocation 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
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.

16.10.9.2 Example #2

Note that the 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 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 outages 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 21: Summary of the Instrument Data that does not reconcile to the General Ledger data

Table	GL Account		Branch	Product	Balance	# of Loans
Management Ledger	Commercial Loan		1	—	$1,000	—
Management Ledger	Commercial Loan		2	—	$2,000	—
Management Ledger	Consumer Loan		1	—	$3,000	—
Management Ledger	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
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 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:

Table 22: Summary of data Dynamic Driver allocation creates in Example #2

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

16.10.10 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 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 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 Management Ledger 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 CDs. 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 built 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 the aggregate total allocated costs 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 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.

16.10.11 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 to post to the instrument-level.

In this example, your objective is to distribute Item Processing expenses 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 >.

16.10.12 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.

Table 23: 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

Table 24: 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 to calculate account-level costs for each activity. For example, imagine that you can 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 or your data model or your allocation rules.

16.10.13 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 Management Ledger 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 (Management Ledger 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 Management Ledger 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 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 Management Ledger (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 Static Table Driver.

16.10.14 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.

16.10.14.1 Examples of Static Driver Table Allocations

See Using Static Table Drivers in Static Table Driver.

16.10.14.2 Examples of Lookup Driver Table Allocations

See Lookup Table Driver for examples of using Lookup Driver Table allocation rules.

16.10.14.3 Aggregation to Management Ledger

An allocation rule is written to aggregate one column from an instrument table to a Management Ledger table. Common aggregations include ending or average balances or interest income or transfer charges/credits to specific Financial Elements in the Management Ledger table. Frequently, there are many other non-interest incomes or non-interest expense columns that users might also want to aggregate to the Management Ledger table. For any given column to be aggregated, the operation for CASA is functionally 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.

16.10.14.4 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.

16.10.14.5 Management Ledger to Instrument

Another common rule type allocates from Management Ledger to the instrument-level.

17 Allocation Models

An Allocation Model consists of a list of individual allocation rules that can be executed as a single unit. Oracle Insurance Allocation Manager for Enterprise Profitability supports two types of Allocation 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, and so on. 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 expenses from the HR center.

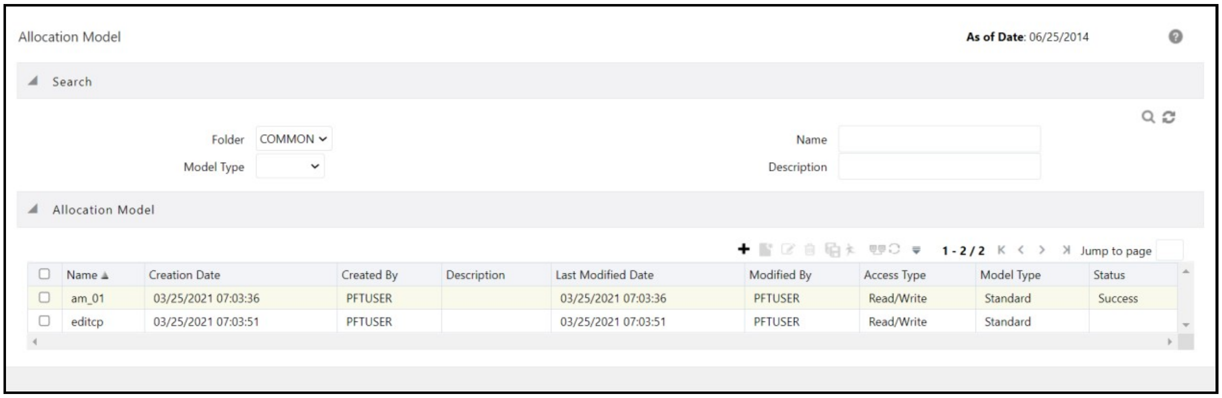
17.1 Summary and Detail Screens

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 detailed screen is displayed.

17.2 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.

Figure 98: Allocation Model Summary Screen



Search Container

Your default Folder functions as a search constraint. The value of your default Folder is set in Application Preferences for Oracle Insurance Allocation Manager for Enterprise Profitability. 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.

17.2.1 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

Oracle Insurance Allocation Manager for Enterprise Profitability. 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.

17.2.2 Allocation Model Specification Summary Grid

The following columns categorize each Allocation Model in the summary grid:

- **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.

17.2.3 Navigation within the Detail Screen

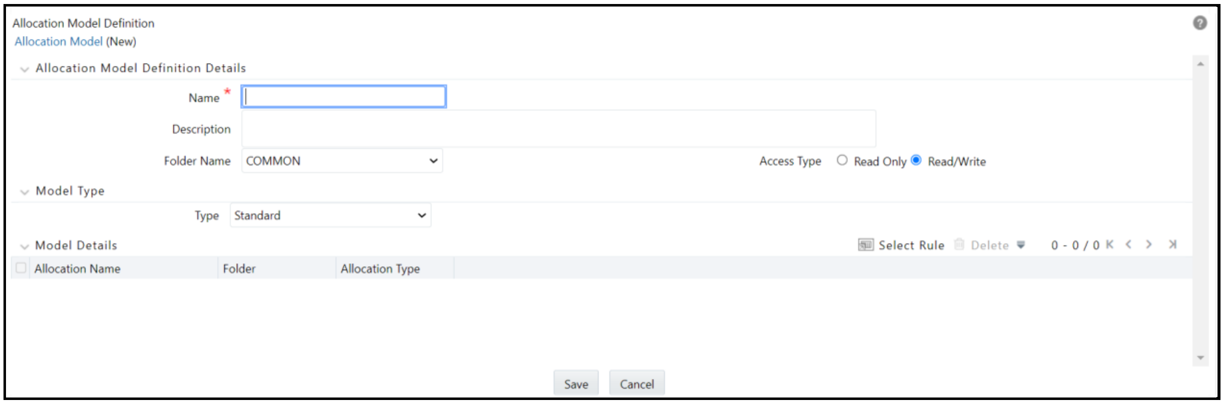
When you Add, Edit, or View an Allocation Model, the Allocation Model Specification Detail Screen is displayed. The detail screen is composed of 4 containers:

- Allocation Model Definition Details
- Model Type
- Model Details
- Audit Trail / User Comments

17.2.3.1 Allocation Model Definition Details Container

To view the Allocation Mode Definition screen, from the LHS menu, select **Insurance Allocation Manager**, select **Rule Specification**, and then select **Allocation Model** to display the **Allocation Model Summary** Screen.

Figure 99: Allocation Model Definition Screen



The Allocation Model Definition Details container allows you to specify a model's Name, Description, Folder, and Access Type.

17.2.3.1.1 Model Type Container

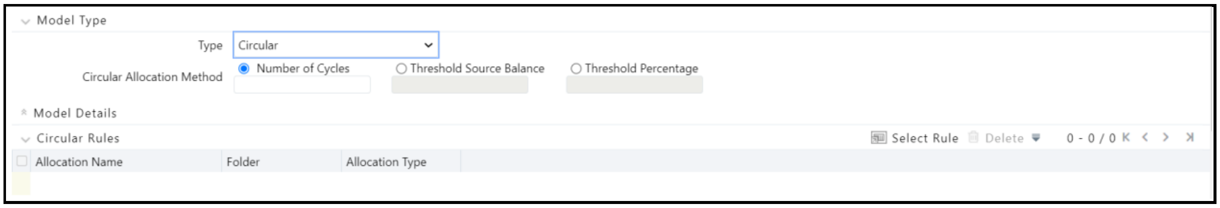
The appearance of the Model Type container depends on whether you are building a Standard Model or a Circular Model.

- **Standard Model**: If you wish to construct a Standard Allocation Model, simply select Standard from the Model Type drop-down list box.
- **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 the 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 the 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.

Figure 100: Circular Model



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**: 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**: 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 Oracle Insurance Allocation Manager for Enterprise Profitability.

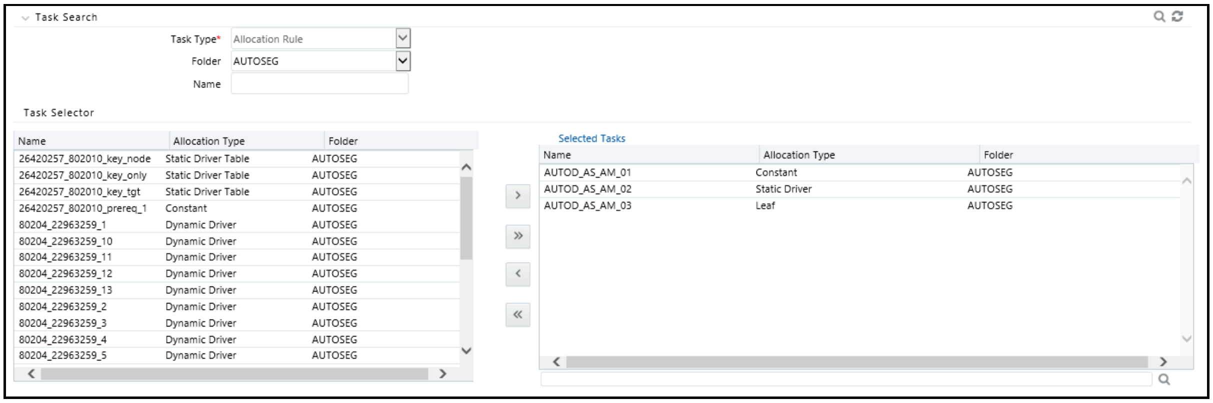
  Within the specification of an Allocation Model, if you attempt to save a model using a value for “Number of Cycles” that either (1) exceeds the value found in Application Preferences or (2) is less than or equal to zero, the user interface will display the following message:
- **Model Details**: If 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”.

17.2.3.1.2 Selecting Allocation Rules to Include

Click the Task Selector to choose the rules you want to include.

Figure 101: Task Search 1



Task selector consists of two containers

- **Task Search**: Task Search contains Task Type, Folder, and Name. The only Task Type presented for Allocation Models is 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” to search for all allocation rules. You may also search for rules by name. After executing a search, qualifying rules are displayed on the left-hand side of the Task Selector.

Figure 102: Task Search 2

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.

18 Static Table Driver

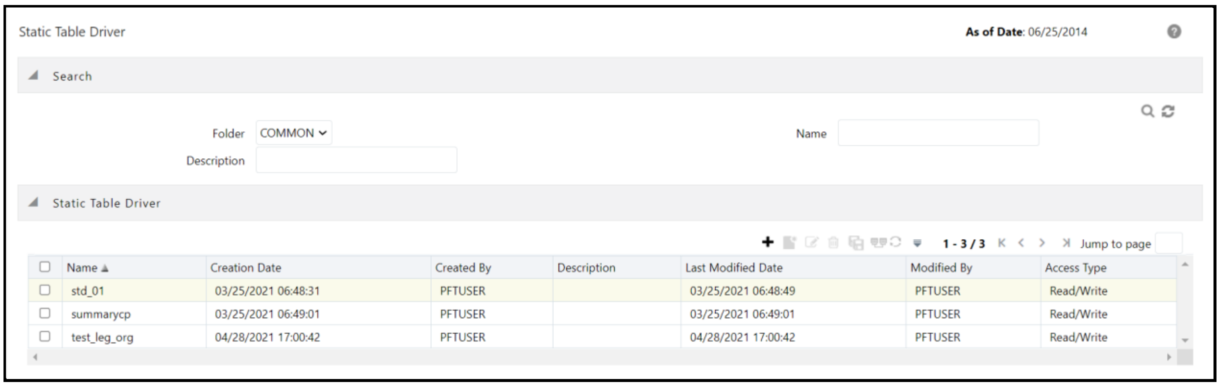
This chapter describes Static Table Driver functionality.

- Static Table Driver Summary & Detail Screens
- Navigation within the Summary Screen
- Navigation within the Detail Screen
- Using Static Table Drivers

18.1 Summary and 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 detailed screen is displayed.

Figure 103: Static Table Driver Summary Screen



18.2 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.

18.2.1 Search Container

Your default Folder functions as a search constraint. The value of your default Folder is set in Application Preferences for Oracle Insurance Allocation Manager for Enterprise Profitability. 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.

18.2.2 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.

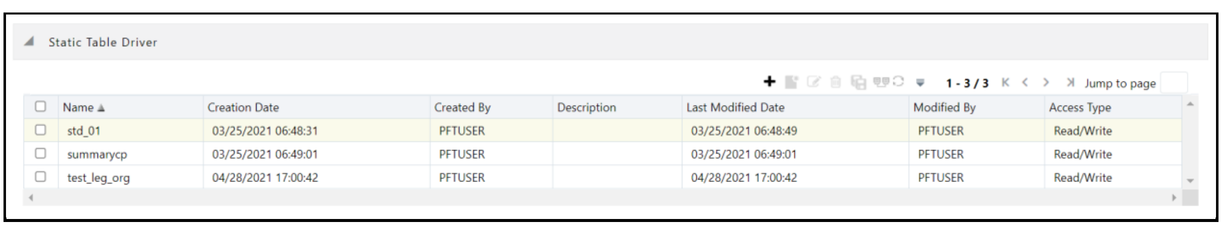
You may select or deselect all of the Static Table Driver rules in the summary grid by clicking on the check box in the upper left-hand corner of the summary grid directly to the left of the Name column header.

- **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 depends 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.

18.2.3 Static Table Driver Summary Pane

The following columns categorize each allocation rule in the summary pane:

Figure 104: Static Table Driver Summary Pane


Static Table Driver

1 - 3 / 3 Jump to page

Name	Creation Date	Created By	Description	Last Modified Date	Modified By	Access Type
std_01	03/25/2021 06:48:31	PFTUSER		03/25/2021 06:48:49	PFTUSER	Read/Write
summarycp	03/25/2021 06:49:01	PFTUSER		03/25/2021 06:49:01	PFTUSER	Read/Write
test_leg_org	04/28/2021 17:00:42	PFTUSER		04/28/2021 17:00:42	PFTUSER	Read/Write

- **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.

18.2.4 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

18.2.5 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 >

18.2.6 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 to update a result column.

18.2.7 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 Oracle Insurance Allocation Manager for Enterprise Profitability.

18.2.8 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.

18.2.9 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. Oracle Insurance Allocation Manager for Enterprise Profitability supports a maximum of 3 Key Leaf dimensions.

18.2.10 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).

18.2.11 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 >.

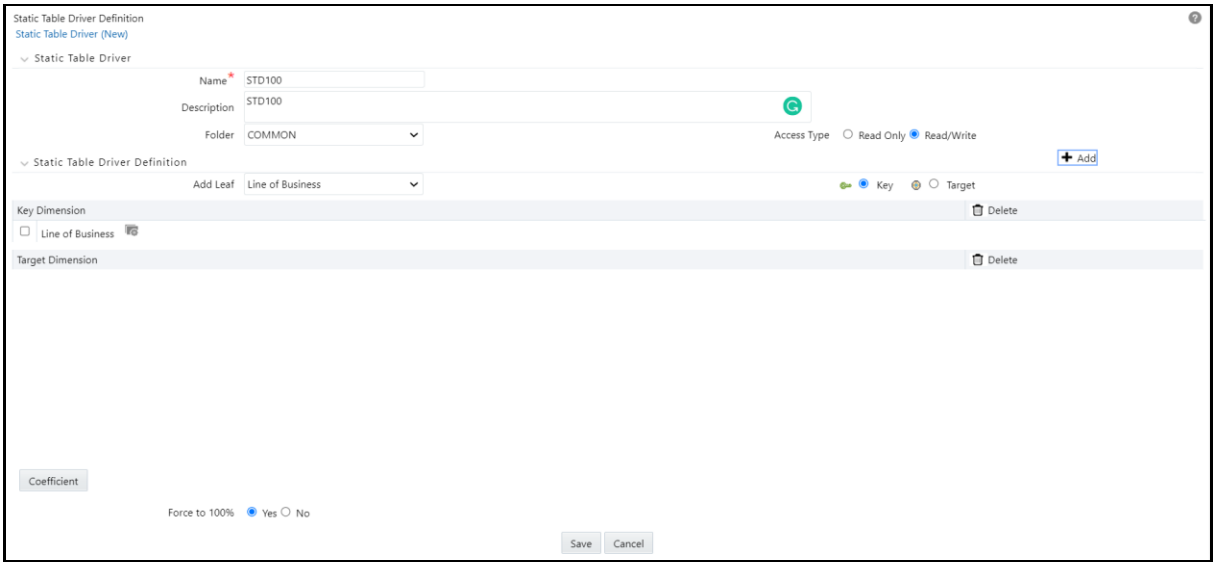
18.3 Creating a New Static Table Driver – Sample Workflow

This section describes a typical workflow for defining a Static Table Driver rule.

To define 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 the Key Dimension section.

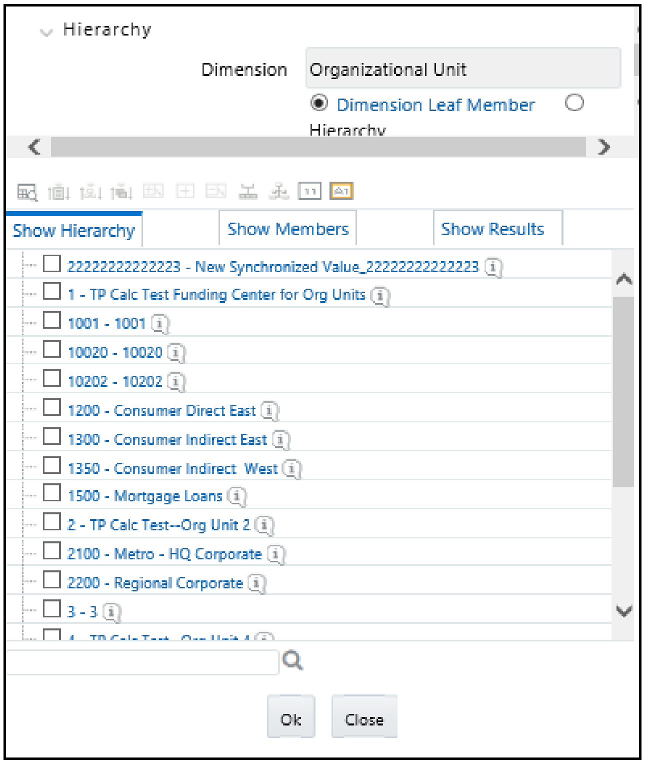
Figure 105: Static Table Driver (New) – Definition Mode



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 member 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, 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**.

Figure 106: Hierarchy Browser



6. After you click **OK** within the Hierarchy Browser window, that window closes and you will be directed back to the 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.

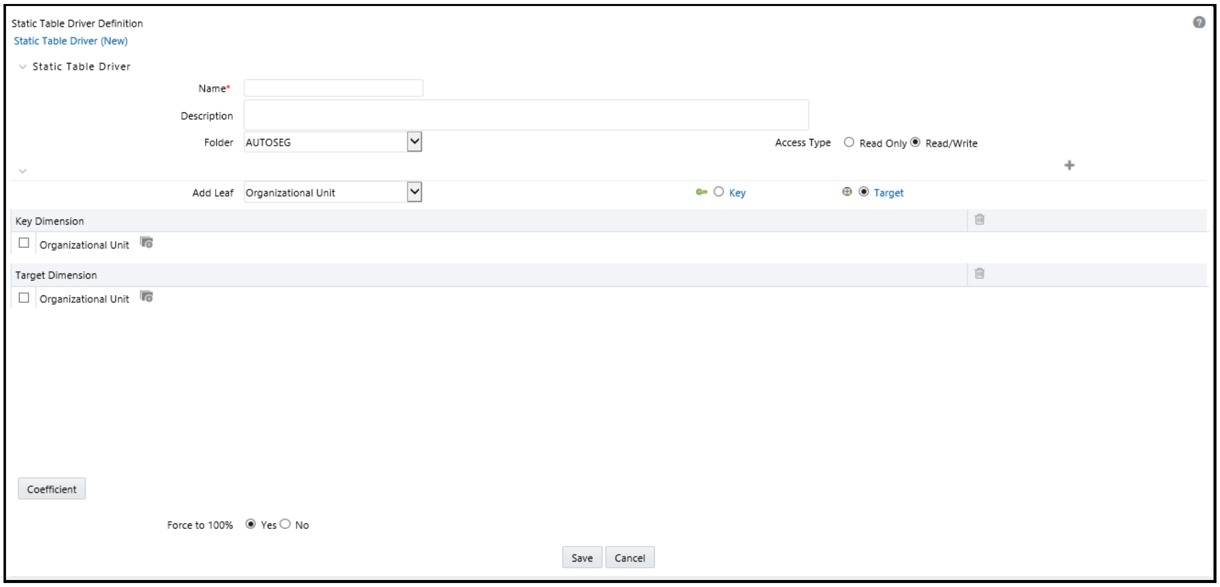
18.3.1 Creating a New Target Leaf

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.

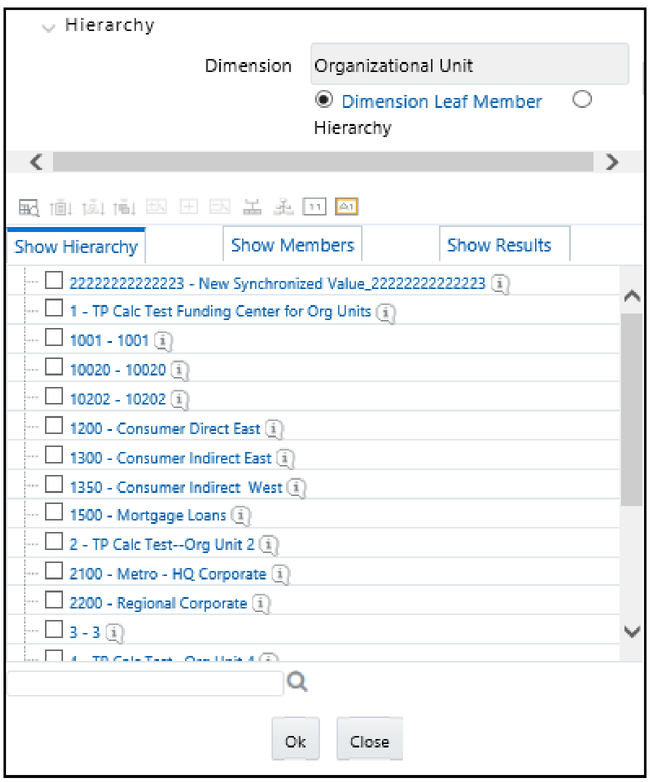
4. The Target Leaf dimension will be added under the Target Dimension section.

Figure 107: Static Table Driver (New) – Definition Screen



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**.

Figure 108: Hierarchy Browser



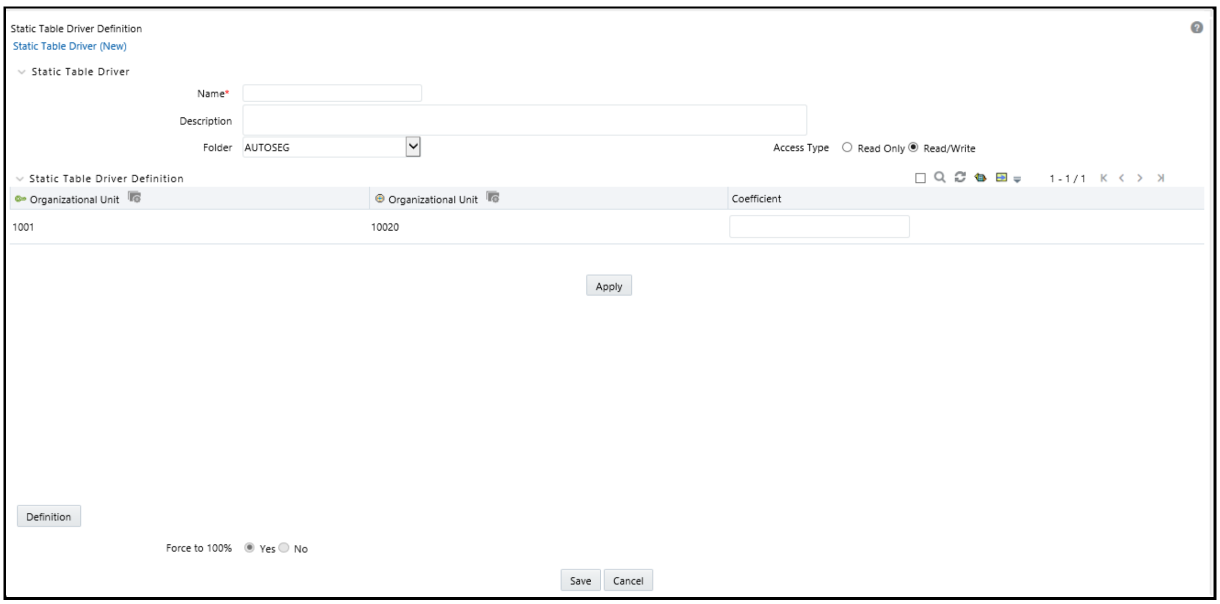
7. After you click on **OK** within the Hierarchy Browser window, that window closes and you will be directed back to the Static Table Driver detail screen However, while the screen still displays your Key Leaf values under the Key Dimension section as rows, it now displays your Target Leaf values under the Target Dimension section as rows.

18.3.2 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.

1. Click on Coefficient to navigate to the Coefficient tab.
2. Enter the coefficient values.

Figure 109: Static Table Driver (New) – Definition Screen



3. Click **Apply**. Click the **Definition** button to view or edit the Key Leaf and Target Leaf dimensions' definition.
4. 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.

18.3.2.1 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”.

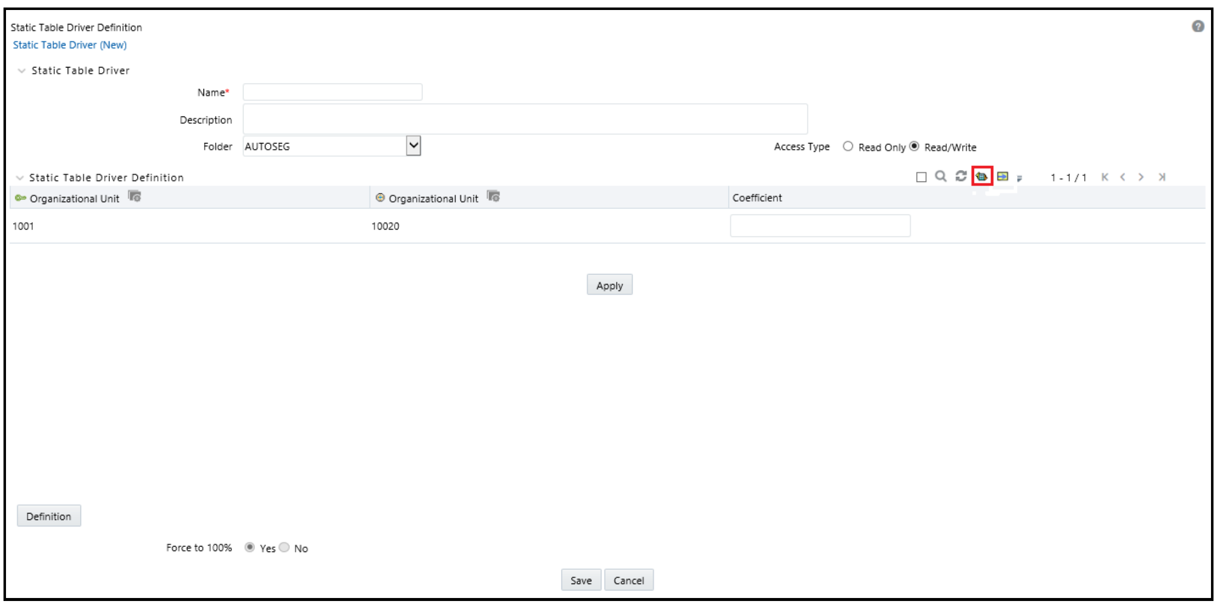
Static Driver Table rules are similar in many ways to Dynamic Driver rules. Select “Yes” for the value of Force to 100% when you want to perform a distribution using non-normalized statistics such as headcount, square footage of occupancy, and so on. If you are using a normalized set of driver statistics (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. Afterward, you can import your spreadsheet back into Oracle Insurance Allocation Manager for Enterprise Profitability.

Figure 110: Static Table Driver (New) – Definition Screen



To export to a spreadsheet:

1. Click on Export.
2. Open using Excel or save the file to the name & location of your choice.

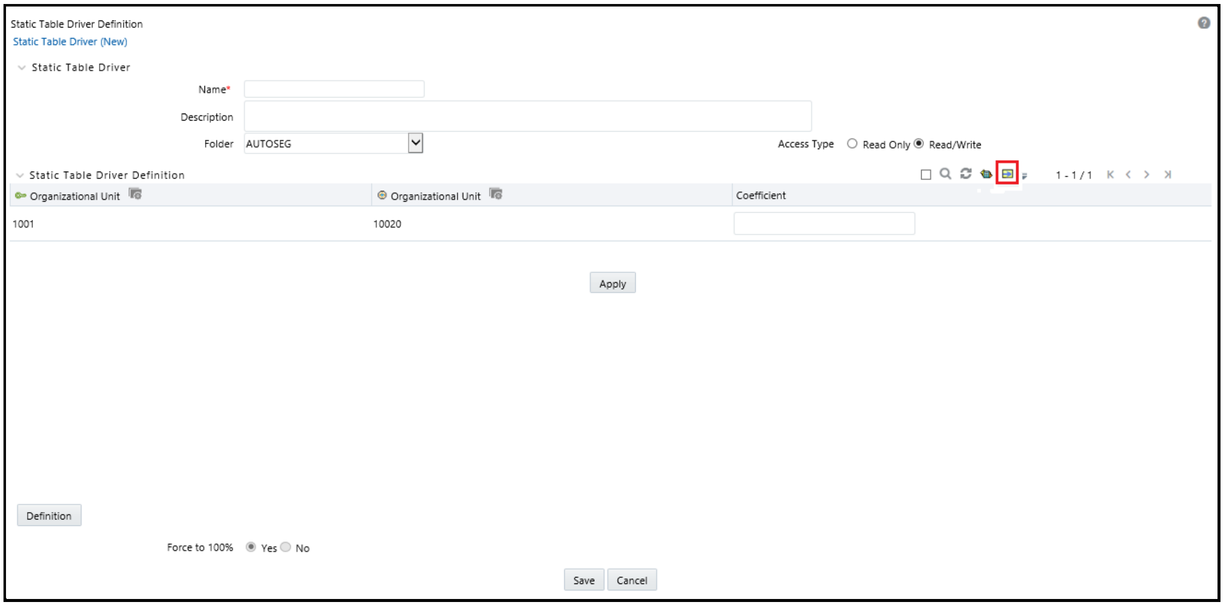
18.3.2.1.1 Excel Import

Excel import functionality is used for adding / editing leaf data in static table drivers.

To import a file, follow these steps:

1. Click Import to trigger an Upload File dialog.

Figure 111: Static Table Driver (New) – Definition Screen



2. **Browse** for the spreadsheet that you want to import.
3. Select the file and click **Open**.
4. Click **Upload File**.

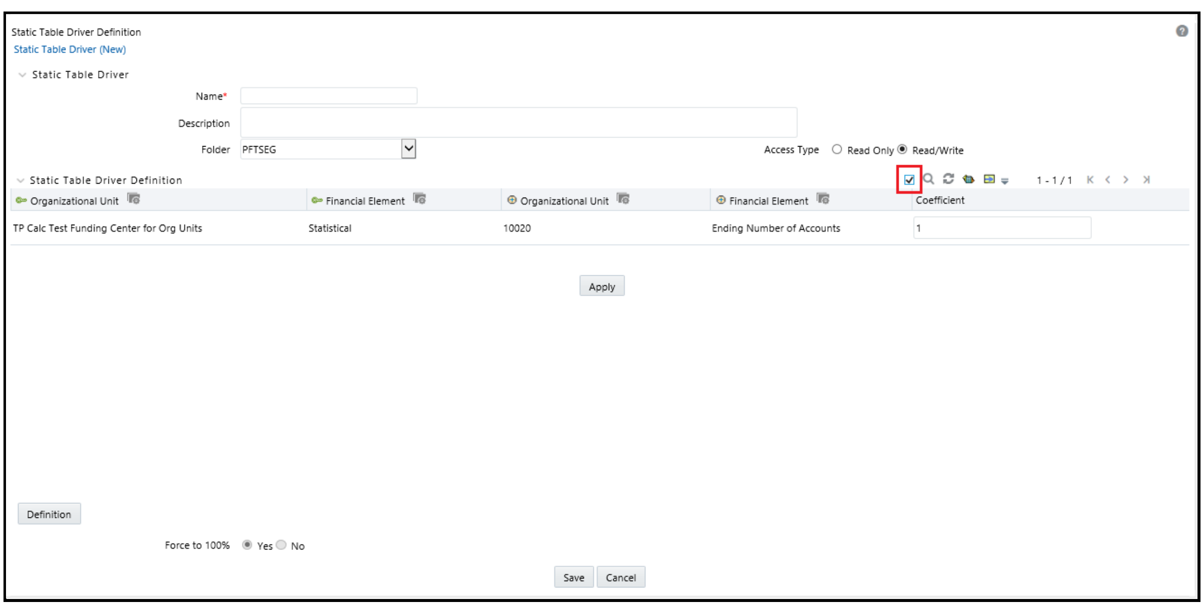
18.3.2.1.2 Search

The Search option allows you to search the Leaf values that have coefficient values defined on the Static Table Driver Definition page. This functionality works only if the **Only Coefficient** option is enabled. You can export this extracted list using the Export functionality.

To perform the search, follow these steps:

1. Enable the **Only Coefficient** using the corresponding check-box.

Figure 112: Static Table Driver (New) – Definition Screen



2. Click **Search**.

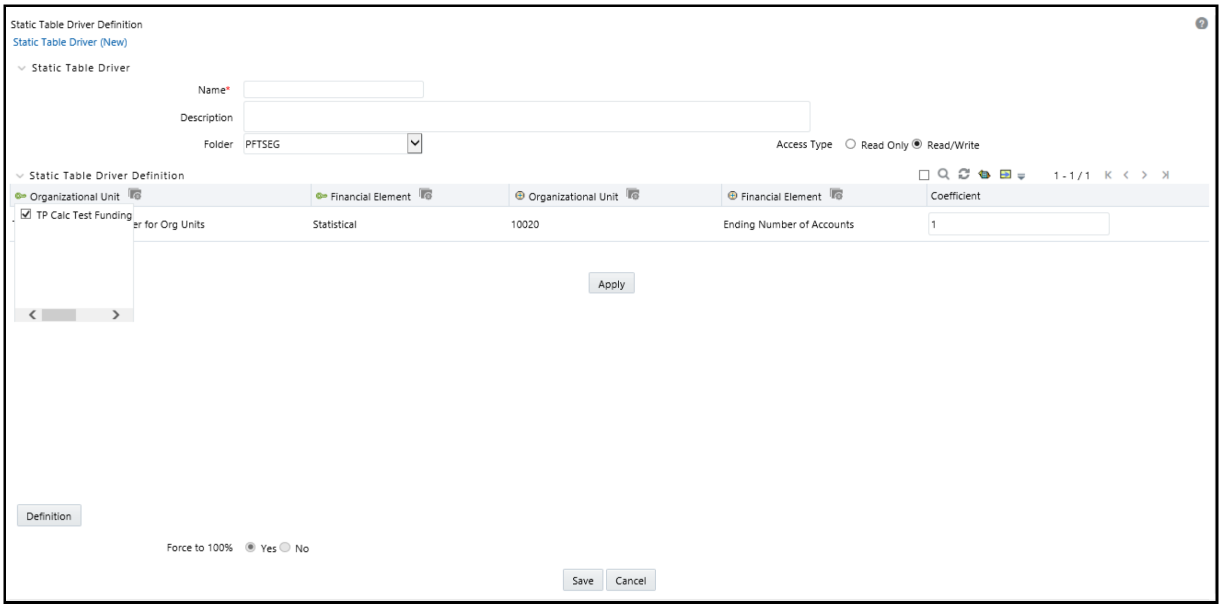
   It shows the list of Leaf values that have coefficient defined. Also, a red dot (.) will be added next to the **Only Coefficient** option. You can also export this list using the Export functionality.

3. Use the **Reset** button to remove the filter.

The Search option allows you to search for specific Leaf.

1. Click ellipses (...) that are 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.

Figure 113: Static Table Driver (New) – Definition Screen



2. Click **Search**. Only selected Leaf values will be displayed. You can also export this list using the Export functionality.

   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.

18.3.2.1.3 Reset

The Reset option on the 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 the 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 a cost center to a 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.

You may toggle back and forth between the specification of valid combinations and the 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.

18.3.3 Working with Exported Static Table Driver Data

At the 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.

Figure 114: Sample Illustration of Exported Static Table Driver Definitions

	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 Managment	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 Managment	8100	Executive	135
8	8400	Facilities Managment	8400	Facilities Managment	160
9	8400	Facilities Managment	8200	Finance Department	185
10	8400	Facilities Managment	8300	Human Resources	210
11	8400	Facilities Managment	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).

18.3.3.1 Updating Exported Static Table Driver Data

You may update the coefficient value for any combination in the spreadsheet (yellow cells in the example above).

18.3.3.2 Adding New Combinations to Exported Data

To add a new combination to your data, follow these steps:

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 Oracle Insurance Allocation Manager for Enterprise Profitability.

Figure 115: Adding New Combinations to Exported Data

	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 Managment	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 Managment	8100	Executive	135
8	8400	Facilities Managment	8400	Facilities Managment	160
9	8400	Facilities Managment	8200	Finance Department	185
10	8400	Facilities Managment	8300	Human Resources	210
11	8400	Facilities Managment	8500	Information Technology Unit	235
12	8200	Finance Department	8100	Executive	260
13	8200		8400		285

18.3.4 Validating the Imported Data

After 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 setup 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.

18.4 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.

18.4.1 Limit on Large Cross Products

The Static Table Driver user interface is designed to allow you to 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.

18.4.2 Managing Large Cross Product Static Table Drivers

18.4.2.1 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.

18.4.2.2 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.

18.4.2.2.1 Adding New Combinations and 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.

18.4.2.2.2 Removing Combinations and 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.

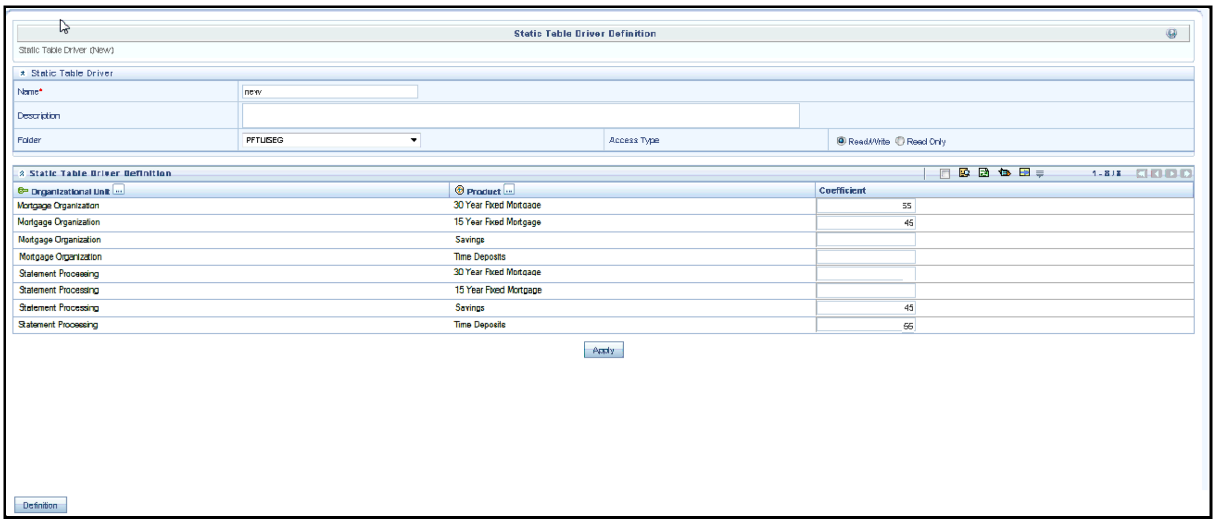
18.5 Using Static Table Drivers

Static Table Drivers are declared as drivers in the Driver process tab for allocation rules of the type Static Driver Table (see 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 General Ledger Account, but not to Product). You need to write allocation rules that distribute expenses within each cost center to the 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).

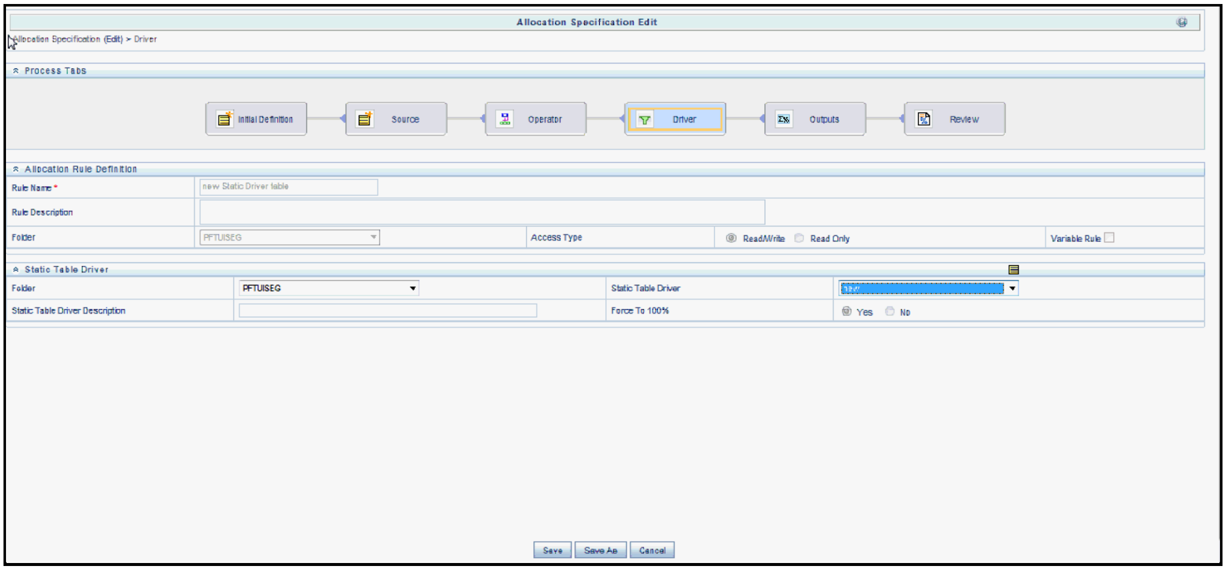
Figure 116: Static Table Driver (New)



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.

Figure 117: Allocation Specification (Edit)

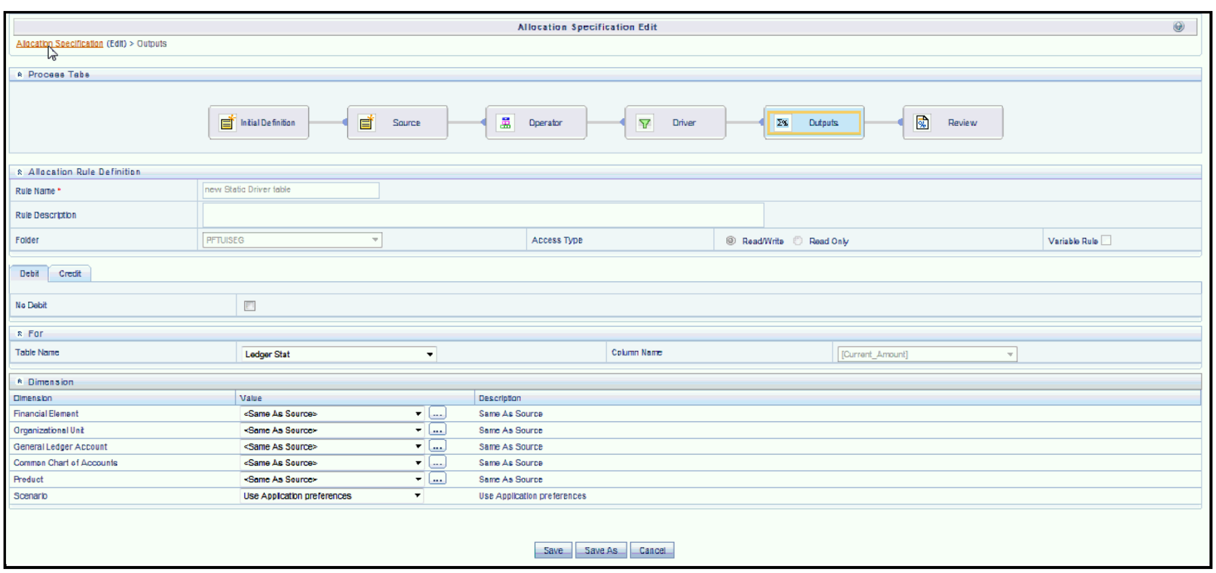


Next, build a new Static Driver Table allocation rule. In the Driver tab, select the Static Table Driver you just built. In the Source tab, constrain the General Ledger Account dimension with a rollup point whose meaning is Total Non-Interest Expense; alternatively, you might specify Financial Element 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 the 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.

Figure 118: Allocation Specification (Edit)



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.

Table 25: Static Table Driver - Output Debits

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 >

19 Lookup Table Driver

This chapter describes Lookup Table Driver functionality.

- Lookup Table Driver Summary and Detail Screens
- Navigation within the Summary Screen
- Navigation within the Detail Screen
- Using Lookup Table Drivers

19.1 Summary and 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 detailed screen is displayed.

19.2 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.

19.2.1 Search Container

Your default Folder functions as a search constraint. The value of your default Folder is set in Application Preferences for Oracle Insurance Allocation Manager for Enterprise Profitability. 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.

19.2.2 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.

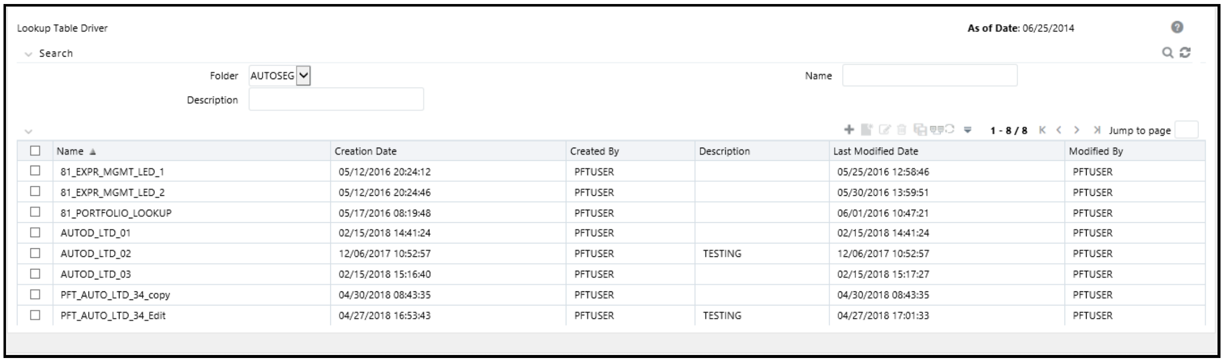
19.2.3 Lookup Table Driver Summary Grid

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

19.2.4 Lookup Table Driver Summary Screen

Figure 119: Lookup Table Driver Summary Screen



19.2.5 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.

19.2.5.1 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.

Figure 120: Lookup Table Driver Definition

19.2.5.2 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 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 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).

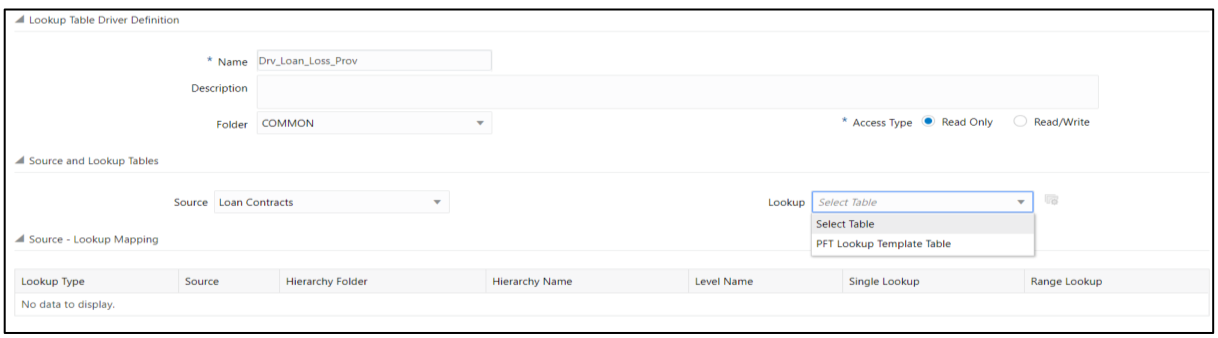
Additional examples of how you might use a Lookup Table Driver rule include the following kinds of assignments:

- 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)

19.2.5.3 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 OFS Data Model Utilities User Guide for details on creating and registering Lookup Tables.)

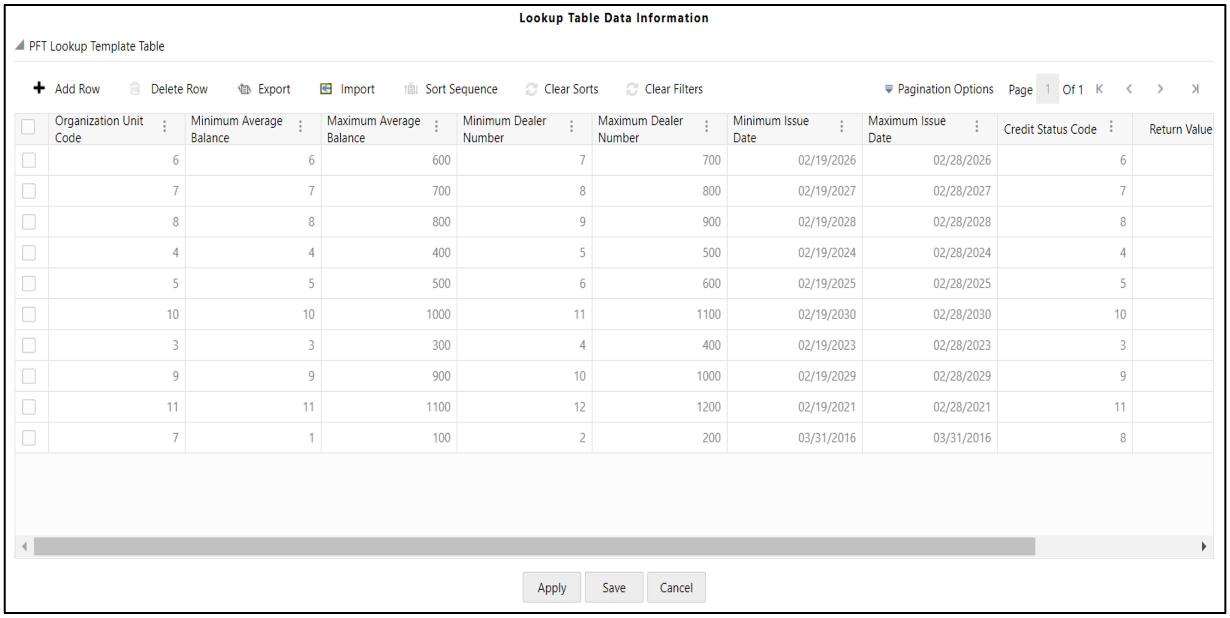
Figure 121: Lookup Table Driver (New)



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 **Lookup Table Data Information** icon adjacent to the Lookup drop-down list to navigate to the Lookup Table Data Information page.

Figure 122: Lookup Table Data Information



The following is an example of the Lookup Table Data Information page. The actual columns will depend on the Lookup Table you selected.

- **Adding a Row**: Click the **Add** icon to add a new row to the selected Lookup table.

  A new row will be added. Enter the values for the row and click **Save**.

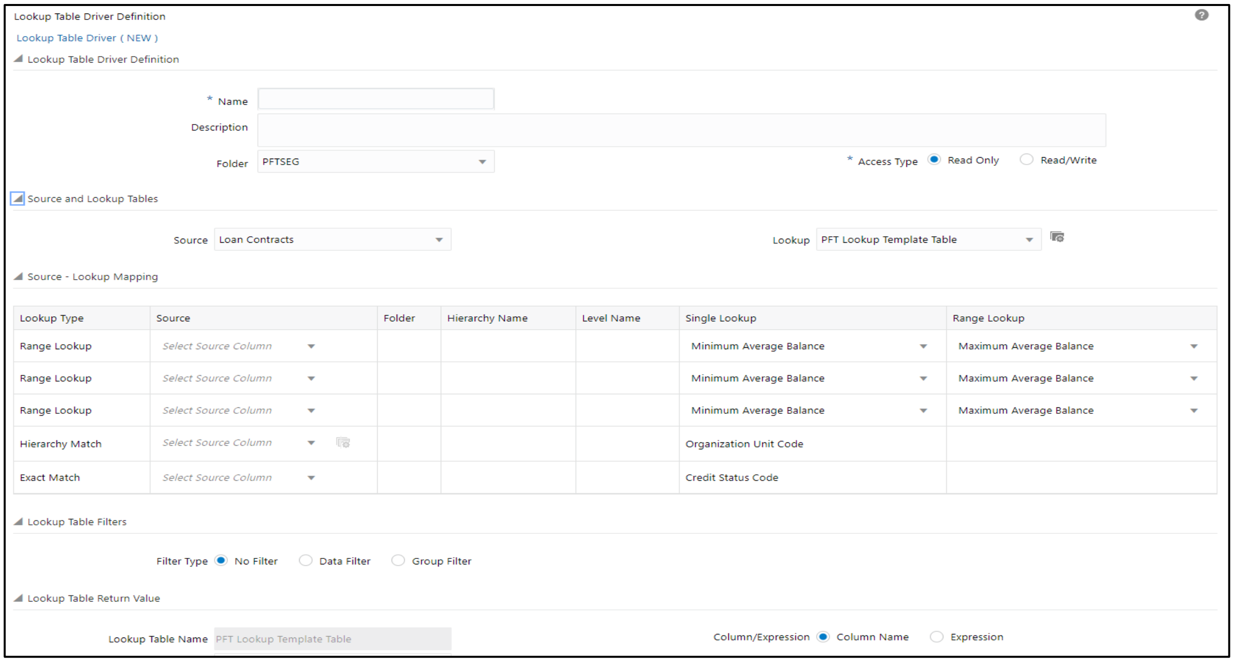
- Deleting Rows: Click the Delete icon to delete the row(s) from the selected Lookup table.
- **Excel Export/Import**: Excel export/import functionality is used for adding/editing Lookup Table Data Information in Lookup Table Drivers. Click the following button to access the Export/Import functionality.

19.2.5.4 Source – Lookup Mapping Container

Lookup Table Driver rules support three types of matching:

- Range Lookup
- Exact Match
- Hierarchy Match

Figure 123: Source Lookup Mapping



19.2.5.4.1 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, example, 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 the 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.

19.2.5.4.2 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).

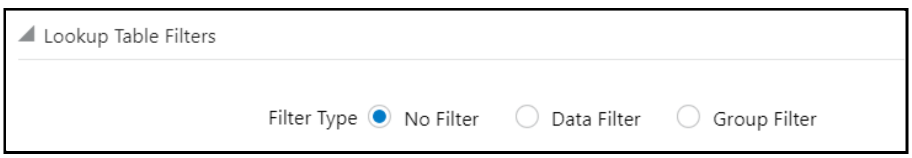
19.2.5.4.3 Hierarchy Match

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 the 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.

19.2.5.4.4 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.

Figure 124: Lookup Table Filters



19.2.5.4.5 Lookup Return Value

Specify the column within your lookup table from which to return a value for each lookup.

Figure 125: Lookup Return Value

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), for example, 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 (example, 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.

20 Allocation Execution History

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

20.1 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 are completed, you may UNDO the results generated by any allocation rule, and you may drill into 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 the 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).

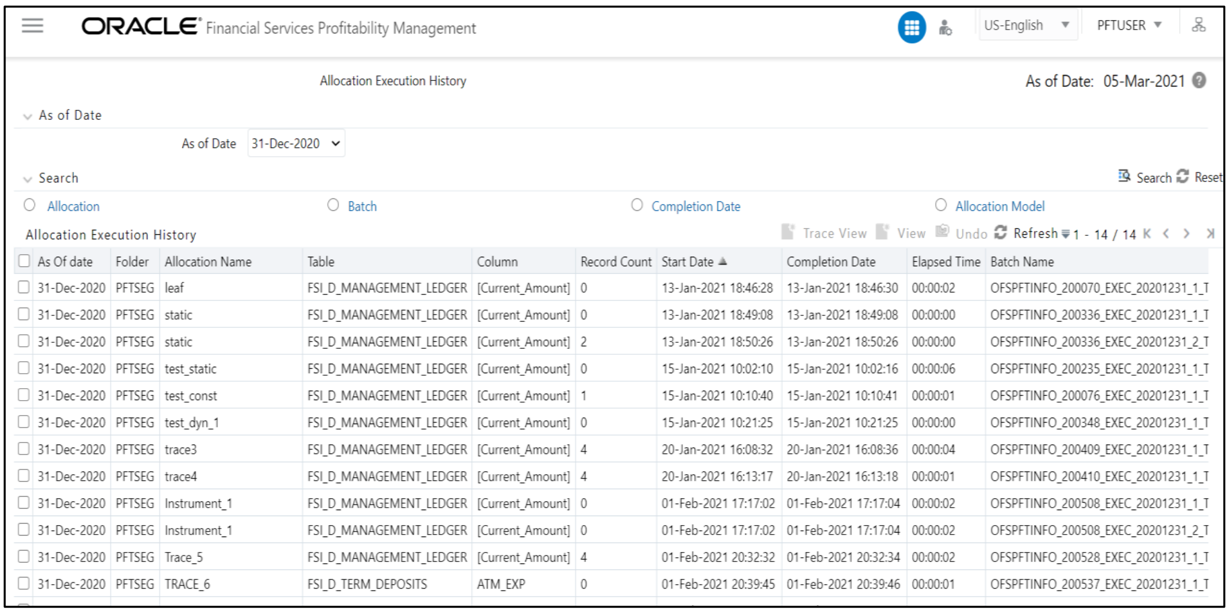
20.1.1 Search Container

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.

Figure 126: Allocation Execution History



- **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 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**: This allows you to view the allocation rules those are grouped under an allocation model. The 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.

20.1.2 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.

  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 is 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.

20.1.3 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**: Displays the As-of-Date for which the allocation rule was executed.
- **Folder**: Displays the name of the Folder to which the allocation rule belongs.
- **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 is 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 is 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 each allocation rule started.

- **Completion Date**: Displays the date and time at which each allocation rule is completed.
- **Elapsed Time**: Displays the elapsed time required for each allocation rule to complete shown in hours, minutes, and 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. The 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.

20.1.3.1 UNDO Functionality

For Allocations that insert or update date 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 action 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.

  Oracle Insurance Allocation Manager for Enterprise profitability supports UNDO of Allocation Models using Batch Execution screen.

  For creating the Batch for UNDO of Allocation Models:

  In the 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 the 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.

20.2 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 Oracle Insurance Allocation Manager for Enterprise Profitability will include rule versioning so that you will always be able to see the structure of any rule at the time it was executed.

20.2.1 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 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 the Management Ledger table is supported.

- **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 date 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 is currently being examined.

20.2.2 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 the 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.

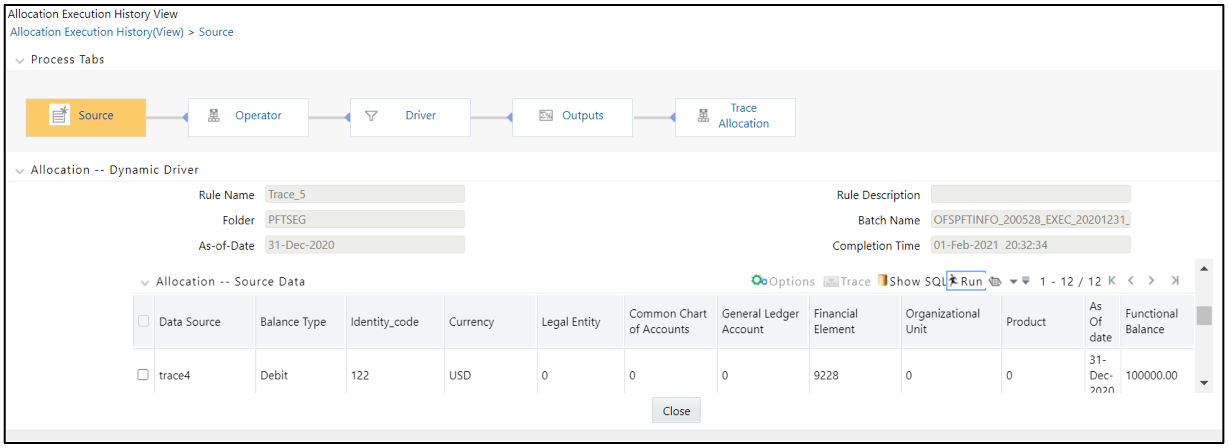
- **Dimension Display**: This container offers you radio buttons with options to display short dimension names, long dimension names, or both within your inline report.
- **Hide/Collapse Dimensions**: This container offers you the option of "collapsing" or "hiding" selected dimensions from your inline 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 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 is currently being examined.

20.2.3 Allocation Driver Data Controls

These controls, including Options, Trace, Show SQL, and Run, operate identically to those described above under Allocation Source Data Controls.

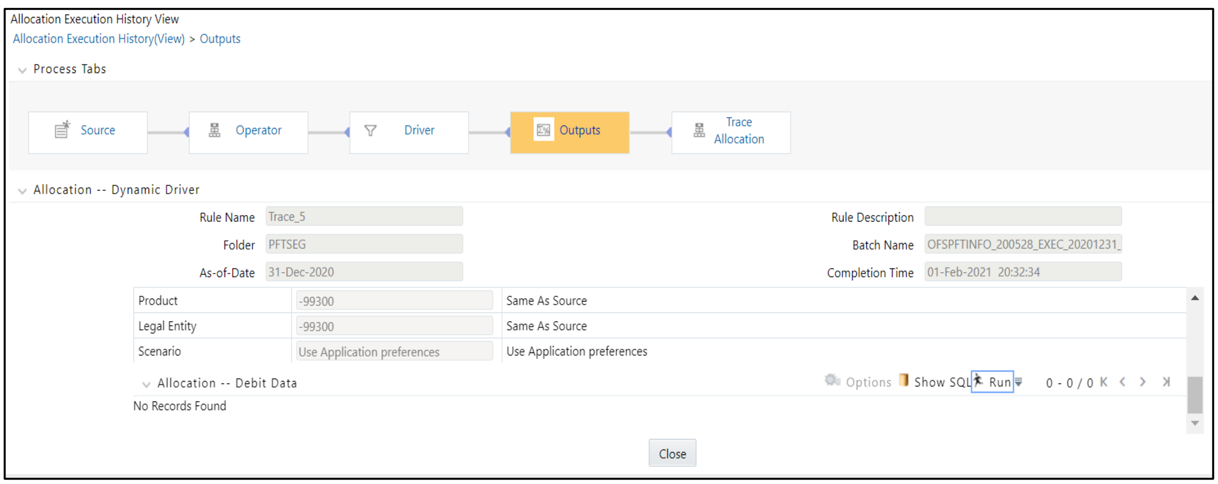
- **Excel Export**: Excel export functionality allows you to export Allocation Execution History source data from the source tab.

Figure 127: Excel Export in Allocation Execution History View Driver Tab



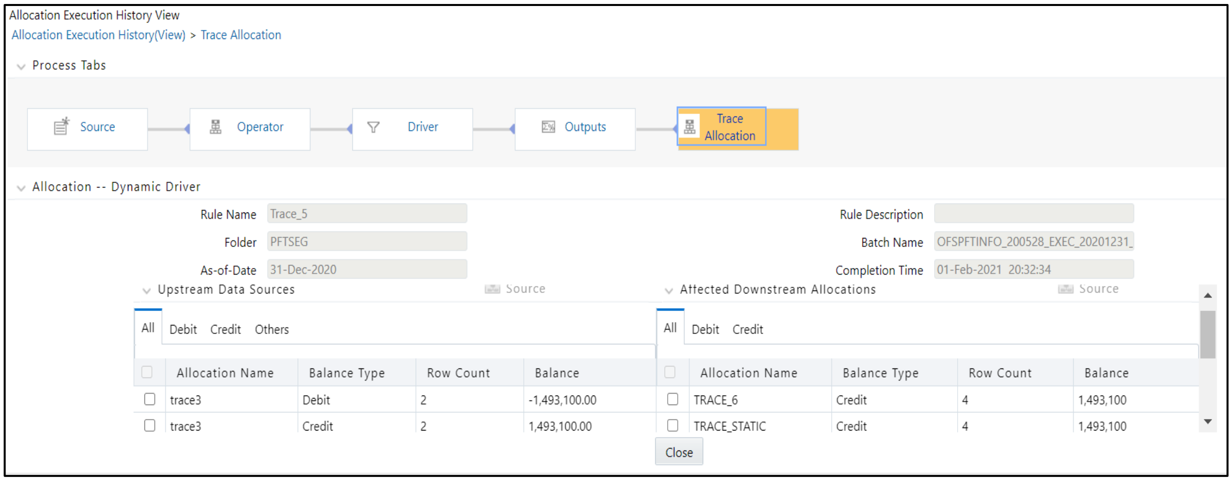
- **Outputs Tab**: The Outputs tab displays the details pertaining to the allocation's Output specification at the time that it was run. This view-only screen also offers an inline report of the Output data generated by the allocation rule at the time that it was run.

Figure 128: Outputs Tab in Allocation Execution History View



- **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, and so on). 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.

Figure 129: Affected Downstream Allocations



- **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.

21 Appendix A: Management Ledger

Topics:

- Statement of Direction
- Frequently Asked Questions

21.1 Statement of Direction

ATTENTION	From Release 8.1, the Management Ledger replaces the Ledger/Stat table as the central repository for management accounting within OFSAA Enterprise Performance Management solutions. Ledger/Stat will continue to be supported for existing or upgrading customers, but it will not be included as part of a new installation. There is a utility available for migrating the allocations defined in the Ledger/Stat table to the FSI_D_MANAGEMENT_LEDGER table. For more details, see the MOS Doc ID 2398698.1.

The new Management Ledger supports the same functionality as historically is 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 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 are directed towards the Management Ledger table class. New enhancements may or may not be extended to apply to the Management Ledger table.

21.2 Frequently Asked Questions

FAQ: 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 supports the same functionality with the Management Ledger as historically has been available with the 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 includes data movement processes for loading its results tables from either or Management Ledger

FAQ: 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.

Future Enhancements:

- All future enhancements to OFSAA applications that utilize ledger-level data will be built to support the Management Ledger structure

FAQ: What are the technical advantages of the new table structure?

The structure of the Management Ledger table eliminates the need to:

- Generate YTD balances for every row at the time of insertion.
- Refresh YTD balances for every row at the time of update.
- Store YTD balances on every row.

YTD balances can 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

FAQ: Can I have multiple Management Ledger tables?

Yes. The Management Ledger table is seeded 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 other data in your Information Domain. For more information, see the *Adding Management Ledger Class Tables* section in the OFS Data Model Utilities User Guide.

FAQ: Will customers need multiple Management Ledger tables?

Most customers will not need multiple Management Ledger tables. The utilization of multiple Management Ledger tables might be useful for some multi-entity or multi-tenant implementations.

FAQ: Can different Management Ledger tables have different dimensions?

Yes, different Management Ledger tables may have different dimensionality.

FAQ: Can I move data between 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.

FAQ: How long will the table be supported?

- Release 8 and all the previous OFSAA releases will always support the table and its related application functionality.
- All the related application functionality may not be supported in future releases, but and its related application functionality will be supported for current releases and Oracle intends to continue supporting for the near future. New application enhancements are unlikely to be extended to the table.

FAQ: 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 table.
- Customers who elect to use should be aware that support for application functionality that utilizes the table is withdrawn in a future release.
- Customers who want to migrate to a future release, for which it is no longer supported, are required to migrate their processes to the Management Ledger before upgrading to that release.

- The need to 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 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 table.
- Customers upgrading to release 8 should be aware that support for application functionality that utilizes the table may be withdrawn in a future release.
- Application rules and processes that involve the table may need to be migrated from the Management Ledger table in the future.

22 Archive-Restore

Topics:

- Command-Line Migration
- Command-Line Migration Utility
- Supported Objects
- Input Values for OBJECTMIGRATION.xml
- Logging

22.1 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 multiple object types.

NOTE PFT does not support the command-line object migration of.CSV format files.

22.2 Command-Line Migration Utility

Oracle Profitability Management uses the same utility that is used by the OFSAA infrastructure. For more details, see the Command Line Utility to Migrate Objects section in the Object Migration section in the OFS Analytical Applications Infrastructure User Guide.

22.3 Supported Objects

The following objects are supported for migration:

Table 26: Supported Objects 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	Export the entire data set and replace it in the target. No validation and its dependency objects are handled.
Interest Rates	801	Yes	Yes	User-defined unique code	Historical rates from the FSI_IRC_RATE_HIST table are not migrated.
Currency rates	803	No	No		Currency rates are not supported.

Object Name	Object Type ID	Support for Wild Card Select all Option	Support for Implicit Dependency	Object Code	Remarks
Currencies	802	Yes	No	User-defined Currency code	
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	

22.4 Input Values for OBJECTMIGRATION.xml

The following table describes the input values required for the OBJECTMIGRATION.xml file:

Table 27: Input Values for OBJECTMIGRATION.xml file

Object Name	Object Type Id	Input Value in OBJECTMIGRATION.XML for the 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"/>
Static Table Driver	17	<OBJECT Code="*" Type="17"/>	<OBJECT Code="System generated code" Type="17"/>
Lookup Table Driver	32	<OBJECT Code="*" Type="31"/>	<OBJECT Code="System generated code" Type="31"/>

22.5 Logging

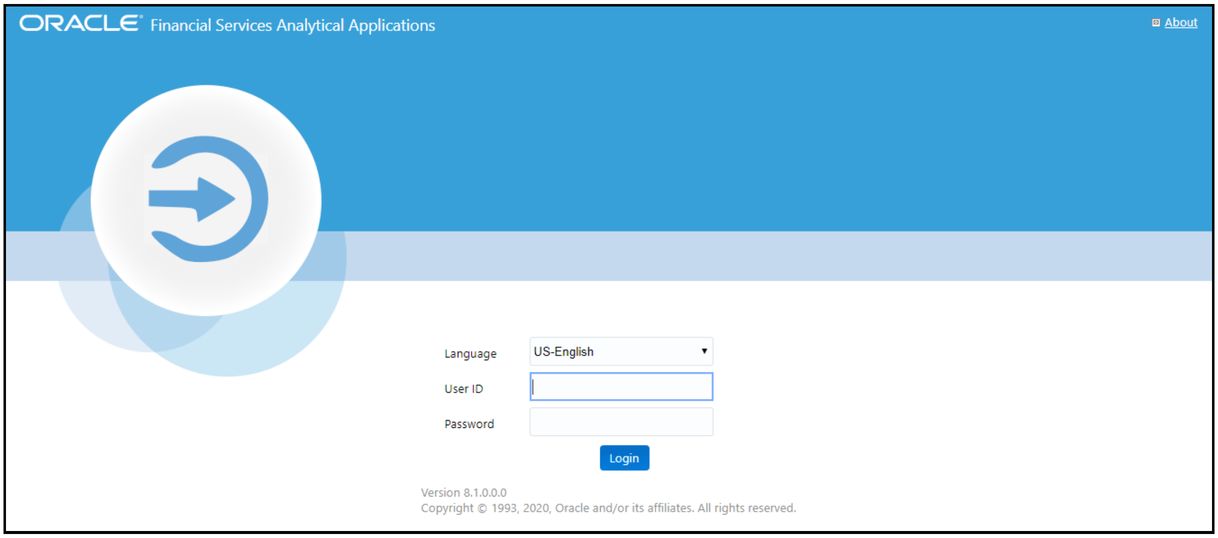
See the following logs files for more information:

- `<War_deployed_location>/logs/ offline_migration.log`
- `<War_deployed_location>/logs/FUSIONSUBSYSTEMSService.log`

23 Appendix C: Multi-Language Support

With the introduction of Multi-Locale Support, you can view the user interfaces in various languages. The display language is a selection in the **Login** window.

Multi-Language Support - OFSAA Login Screen



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.

Table 28: Language Name versus. Java Locale List

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
Russian	RU_RU
Simplified Chinese	ZH_CN
Spanish	ES_ES
Thai	TH_TH
Traditional Chinese	ZH_TW
Vietnamese	VI_VN

Language Name	Java Locale
Turkish	TR_TR
Hebrew	iw_IL
French Canada	fr_CA

Depending on the type of the rule, the Name/Description specified during creation or edit either is attached to the login locale or is stored independently of the login locale. The selection of the login locale, therefore, governs which assumptions/rules are visible, and which are not visible.

Table 29: Visibility of Assumptions and Rules

Assumption or 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 Mode		Yes
Static Table Driver		Yes
Lookup Table Driver		Yes
Allocation History	Yes	

This table illustrates the visibility of assumptions and rules across languages in this release. This applies to all Summary screens, embedded selections in subsequent screens, and so on.

- * 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 such as Dimension Management, Filters, and Expressions.

24 Appendix D: Multi-Currency across OFSAA Applications

Topics:

24.1 OFSAA Rate Management

OFSAA Rate Management 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 referred to by code and written description. A comprehensive list of ISO-defined currencies is included, and users can define and add additional currencies. For details on using OFSAA Rate Management, see OFSAA Rate Management.

Upon installation, one currency is active and ready for processing: the one identified by your organization as the functional currency (for more information, see Functional Currency). 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 before 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.

24.2 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 active. After 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; for more information, see Fiscal Year Information.

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 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. This is done by consulting FSI_DB_INFO (that is, FSI_DB_INFO serves as a fallback if 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; for more information, see Fiscal Year Information.

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 multi-currency is disabled, all ISO Currency Code values are written in the Functional Currency.

24.3 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 the Overview of OFSAA Infrastructure chapter) that define the currency of any row in the database. ISO Currency Code values used in any of OFS Analytical Applications need to 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 OFSAA Rate Management).

Users must designate ISO Currency Codes as reporting currencies to activate the calculation of conversion rates. Simply designating the currency as active is not sufficient.

24.4 002 code

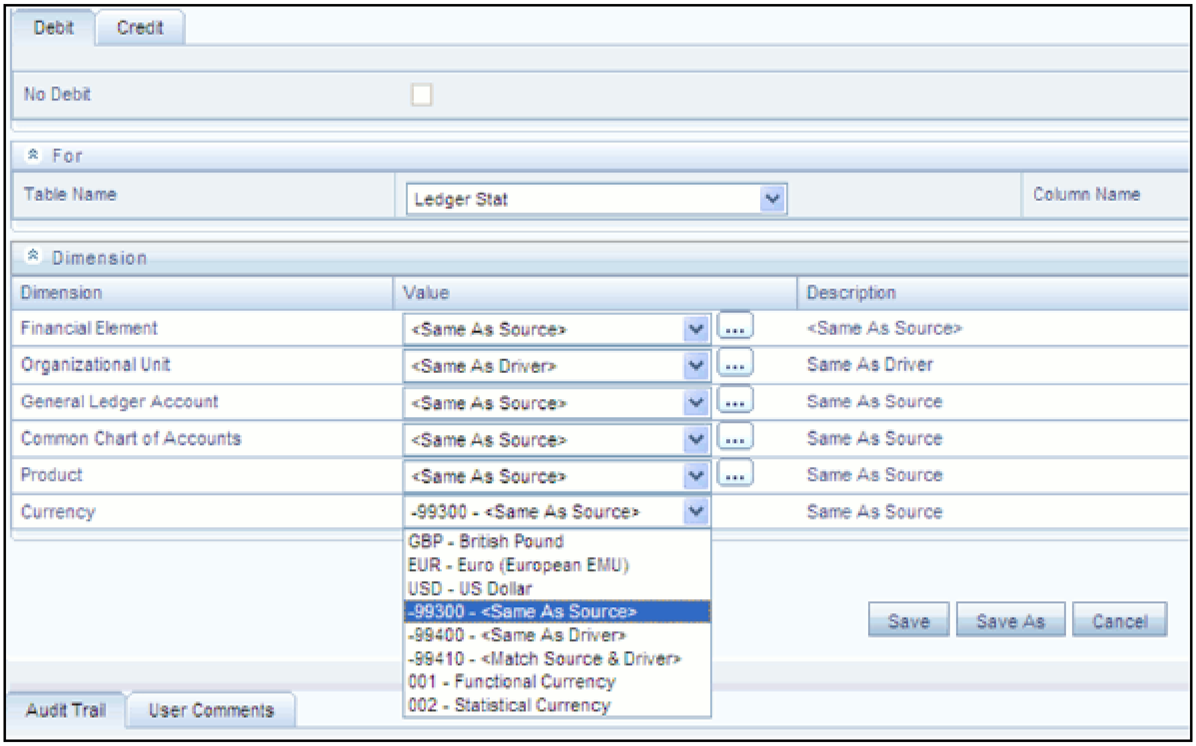
The ISO Currency Code value of 002 represents statistical data in the Management Ledger. An ISO Currency Code value of 002 means “no currency basis”. For example, a row in the Management Ledger representing a square footage statistic does not need a currency association and would, therefore, have a 002 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).

24.5 Multiple Currencies in Allocation Rules

OFSAA Profitability Management allocation engine supports the 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 (for example 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 are 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 Multi-currency Options for a Dynamic Driver Rule



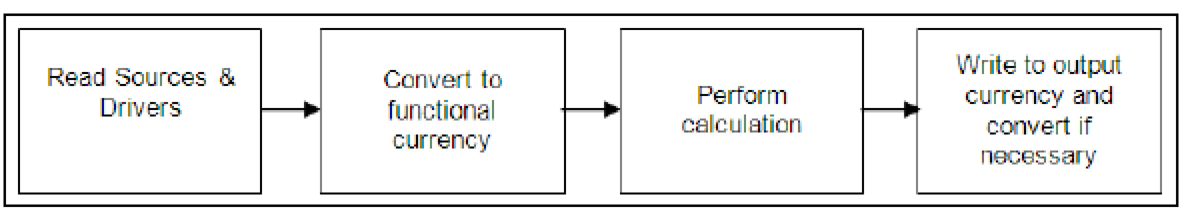
24.6 Using the Currency Dimension as a Filtering Condition

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.

24.7 Allocation Engine Processing Flow

In Sources and Drivers, the Profitability Management allocation engine can read multiple input currencies. When posting to the Management Ledger table, the engine can output multiple result currencies. When posting to a detailed table, outputs are converted, if necessary, to match the ISO_CURRENCY_CD on each detail record to be updated.

Allocation Engine Processing Flow for Multi-currency Allocations



The preceding diagram illustrates the basic allocation engine processing flow for multi-currency 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 an 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 Management Ledger 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 Management Ledger table and detail tables (Instrument and Transaction Summary tables).

24.8 Selecting Rows from Management Ledger Table

When reading rows from the Management Ledger table, the allocation engine decides whether or not it needs to translate the Entered_Balance column value based on the “Column Property” characteristic of each row's Financial Element ID. This decision process applies 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).

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.

24.9 Treatment of Constants and Coefficient Values

The following constant values are assumed to be either statistical or 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.

24.10 Selecting Rows from Detail Tables

When reading a column from a detailed 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:

Table 30: Column Properties in Transaction Summary Table

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	A recurring period
LEAF	Leaf column
IDENTITY	Reserved
ID_NUMBER	Reserved
MULT	Multiplier
NUMBER	Numeric Value
NUMERIC	Numeric Value
RATE	Interest rate
TERM	Non-recurring period

Property	Description
VARCHAR2	Variable-length alpha-numeric value
PCT	Percent
DEFAULT	Default Datatype

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.DESCRIPTION
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;
-----------------------------------------------------------------------------------------------------------
```

24.11 Inserting Into Management Ledger Table

Allocations inserting into the Management Ledger table treat Currency as a Key Processing Dimension. For example, in a Dynamic Driver type of allocation that outputs to Management Ledger table, 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 currency. 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 and 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 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 the Management Ledger table:

- 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).

24.12 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.

24.13 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 a currency, that is, elements having different values for each currency, but which you do not want the engine to 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.

24.14 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 and revenue balances are the most common examples.

In a multi-currency implementation, instrument-level monetary balances are inherently stored as "transacted" balances, that is, as balances stored in local currency. The monetary balances that you source from your general ledger system and which you store in the Management Ledger table will generally be 100% functional. However, you may wish to aggregate instrument-level balances to your Management Ledger table and to store them there in their local currency amounts and/or generate non-functional balances in your Management Ledger 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 the Management Ledger table using the 001 – Functional Currency macro or using the ISO Currency Code that represents your Functional Currency (these are equivalent definitions).

24.15 Statistics and Standard Rates

The Statistics and Standard Rates Column Property Financial Elements are not 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 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 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.

24.16 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.

Table 31: USD as Functional Currency with Debit Product Output in Euros

Application Data	Nature of Data	Implication
Source Data: Constant value of 100	The input value is 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 the Functional currency is US dollars.

Table 32: USD as Functional Currency with Debit Product Output in USD

Application Data	Nature of Data	Implication
Source Data: Constant value of 100	The input value is 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 an 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.

Table 33: Allocation Rule distributes an equal amount of Ledger Level Expense to Instrument Balance Column

Application Data	Nature of Data	Implication
Source Data: Management Ledger for Statement Processing Expense	Various Statement Processing Expenses sourced from the Management Ledger for Checking and Savings products. In this example, all of the Source rows are denominated in USD that is the Functional currency.	No translation is necessary since all of the sourcing 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 has 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 a non-currency specific meaning and therefore, is not translated.
Debit: Instrument-level Statement Processing Expense column	In this example, three 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 expenses 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.

Table 34: Allocation Rule distributes Ledger-level expense to an instrument-level Balance Column

Allocation Data	Nature of Data	Implication
Source Data: Management Ledger 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 sourcing 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, the 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 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). We expect that each dollar of USD average balances would receive an amount equal to each ¥ 100 of JPY average balances. 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.

25 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 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).

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

25.1 Configuring a Fiscal Year

Fiscal year configuration data is held in FSI_FISCAL_YEAR_INFO, a special one-row, two-column table. The 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 previously mentioned table but do populate the FSI_FISCAL_YEAR_INFO, this table acts as a fallback source for the Management Ledger table.

Table 35: 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 need to 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 or delete from the Management Ledger table read the configuration data in FSI_FISCAL_YEAR_INFO / LEDGER_CCY_AND_FISCAL_YR_INFO.

Fiscal-Period Column: This column holds a value that represents 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 12 months. For example, if your organization has two 6-month fiscal periods over 12 months, 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 application runtime errors.

25.2 How YTD Calculations are affected by a Fiscal Year Configuration

The Management Ledger table contains each of your Key Processing dimensions, several administrative columns that are also components of the table's primary key, and two “fact” columns (balance columns).

When you are running a standard calendar year configuration, the records with FISCAL_MONTH equals to 1 corresponds to January data and FISCAL_MONTH equals to 12 corresponds to December data. If, however, your Start-Month column was set to 9 (indicating that your fiscal year begins in September), then the records with FISCAL_MONTH equals to 1 would correspond to September, and records with FISCAL_MONTH equals to 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.

This is an important consideration for the following calculations:

- Allocation rules that insert into or update the table.
- The allocation UNDO process that performs updates or deletes.
- Table-to-Table load into the Management Ledger table.
- Transfer pricing rate migration from instrument tables to the table through OFSAA Funds Transfer Pricing.
- Direct transfer pricing of the table through OFSAA Funds Transfer Pricing.

25.3 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 table.

25.3.1 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 2020 correspond to the monthly buckets as shown in the following table:

Table 36: Monthly Data for Calendar Year 2020

Month 01	Month 02	Month 03	Month 04	Month 05	Month 06	Month 07	Month 08	Month 09	Month 10	Month 11	Month 12
Jan2020	Feb2020	Mar2020	Apr2020	May2020	Jun2020	Ju2020	Aug2020	Sep2020	Oct2020	Nov2020	Dec2020

If you have set your As-of-Date to June 2020, then any OFSAA process will insert the Balance for the sixth month into the Management Ledger table with FISCAL_MONTH as 06.

Continuing with this calendar year example, the 12 YTD buckets correspond to 2020 calendar month ranges as shown in the following table:

Table 37: Year-to-Date Data for Calendar Year 2020

YTD as of Month 01	YTD as of Month 02	YTD as of Month 03	YTD as of Month 04	YTD as of Month 05	YTD as of Month 06	YTD as of Month 07	YTD as of Month 08	YTD as of Month 09	YTD as of Month 10	YTD as of Month 11	YTD as of Month 12
Jan2020	Jan to Feb 2020	Jan to Mar 2020	Jan to April 2020	Jan to May 2020	Jan to June 2020	Jan to July 2020	Jan to Aug 2020	Jan to Sep 2020	Jan to Oct 2020	Jan to Nov 2020	Jan to Dec 2020

For details on the relationships between Month-to-Date columns and Year-to-Date columns, see Management Ledger.

25.3.2 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, therefore, we can refer to the fiscal period like a fiscal year. In this fiscal year example, the months for the fiscal year correspond to monthly buckets as shown in the following table:

Table 38: Monthly Data for Fiscal Year 2020

YEAR	Month 01	Month 02	Month 03	Month 04	Month 05	Month 06	Month 07	Month 08	Month 09	Month 10	Month 11	Month 12
2020	Apr 2020	May 2020	June 2020	July 2020	Aug 2020	Sep 2020	Oct 2020	Nov 2020	Dec 2020	Jan 2021	Feb 2021	Mar 2021

For this fiscal year example, the first calendar month of the 2020 fiscal year is April 2020, and the last calendar month of the 2020 fiscal year is March 2021.

Continuing with this fiscal year example, 2020 fiscal year YTD ranges are as shown in the following table:

Table 39: Year-to-Date Data for Fiscal Year 2020

YEAR	YTD as of Month 01	YTD as of Month 02	YTD as of Month 03	YTD as of Month 04	YTD as of Month 05	YTD as of Month 06	YTD as of Month 07	YTD as of Month 08	YTD as of Month 09	YTD as of Month 10	YTD as of Month 11	YTD as of Month 12
2020	Apr 2020	Apr 2020 to May 2020	Apr 2020 to Jun 2020	Apr 2020 to Jul 2020	Apr 2020 to Aug 2020	Apr 2020 to Sep 2020	Apr 2020 to Oct 2020	Apr 2020 to Nov 2020	Apr 2020 to Dec 2020	Apr 2020 to Jan 2021	Apr 2020 to Feb 2021	Apr 2020 to Mar 2021

25.3.3 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 months, we are no longer dealing with a Fiscal Year, but with two fiscal periods within 12 months. In this fiscal period example, there are two 6-month fiscal periods within the 12-month window beginning April 2020. Just as in the previous 12-month fiscal year example, the months for the two fiscal periods correspond to monthly buckets as shown in the following table:

Table 40: Monthly Data for Fiscal Period #1, 2020 and Fiscal Period #2, 2020

YEAR_S value	Month 01	Month 02	Month 03	Month 04	Month 05	Month 06	Month 07	Month 08	Month 09	Month 10	Month 11	Month 12
2020	Apr 2020	May 2020	June 2020	July 2020	Aug 2020	Sep 2020	Oct 2020	Nov 2020	Dec 2020	Jan 2021	Feb 2021	Mar 2021

Continuing with this fiscal period example, 2020 fiscal year YTD ranges are as shown in the following table:

Table 41: Fiscal Period-to-Date Data for Fiscal Period #1, 2020 and Fiscal Period #2, 2020

YEAR_S value	YTD as of Month 01	YTD as of Month 02	YTD as of Month 03	YTD as of Month 04	YTD as of Month 05	YTD as of Month 06	YTD as of Month 07	YTD as of Month 08	YTD as of Month 09	YTD as of Month 10	YTD as of Month 11	YTD as of Month 12
2020	Apr 2020	Apr 2020 to May 2020	Apr 2020 to Jun 2020	Apr 2020 to Jul 2020	Apr 2020 to Aug 2020	Apr 2020 to Sep 2020	Oct 2020	Oct 2020 to Nov 2020	Oct 2020 to Dec 2020	Oct 2020 to Jan 2021	Oct 2020 to Feb 2021	Oct 2020 to Mar 2021

Note that in this fiscal period example:

- The first calendar month of the first fiscal period is April 2020
- The last calendar month of the first fiscal period is September 2020
- The first calendar month of the second fiscal period is October 2020
- The last calendar month of the second fiscal period is March 2021

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 means “fiscal period-to-date”.

25.4 Using the Undo Function with a Fiscal Year Configuration

The allocation UNDO function removes the results of allocations from the Management Ledger table. The database function that operates against may be either an UPDATE or a DELETE. In other words, 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 assume you have run an allocation rule that posts data to the Management Ledger 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 updates the March MTD column with a value of zero.
  - The engine re-sets (update) the YTD columns for each YTD column within the fiscal period of the month being UNDONE.
- The engine deletes 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 only affects the YTD columns. This falls within the same fiscal period as the month of the allocation that is being UNDONE.

26 Appendix F: Seeded Financial Elements

The following table lists the Financial Elements that are seeded by the installer:

Table 42: Financial Elements Listed By Number

Financial Element Number	Description	Primary Usage
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
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

Financial Element Number	Description	Primary Usage
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
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

Financial Element Number	Description	Primary Usage
207	Timing of Recovery Runoff	ALM Output
208	Non-Performing Asset Runoff	ALM Output
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 Rat	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
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

Financial Element Number	Description	Primary Usage
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
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 or Credit	Initial Load; or Profitability Management, FTP, or ALM Output
451	Historic Option Cost Charge/Credit	FTP Output
452	Charge or Credit Rem Term	FTP Output
453	Current Option Cost Charge or 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
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

Financial Element Number	Description	Primary Usage
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
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

Financial Element Number	Description	Primary Usage
935	Local Taxes	ALM Output
940	Dividends	ALM Output
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
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

Financial Element Number	Description	Primary Usage
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
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
9018	Charge for Other Allocated Assets	OFSPA Reporting Line

Financial Element Number	Description	Primary Usage
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
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

27 Appendix G: Debit and Credit Conventions

The following sections describe the debit credit conventions within the Management Ledger table.

Topics:

- Standard Accounting Conventions
- Profitability Management Allocation Engine

27.1 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.

Table 43: Standard Accounting Conventions

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	Debit Account	Decreases the running balance

For example, if you generate an accounting transaction for a new loan, your transaction debits the appropriate asset GL account (a debit account) for the new loan, therefore, increasing the running balance for the Loan GL account; and credit cash (also a debit account), therefore, reducing the running balance for the Cash GL account.

Table 44: Debit/Credit Entries

Debit	Credit
New loan $500	Cash ($500)

27.2 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 three GL accounts are 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.

Table 45: Debit-Credit Entries

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	
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.

Table 46: Debit-Credit Entries

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 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 of 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.

27.2.1 Allocation Engine Logic

When an allocation rule is configured to generate the Management Ledger 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.

27.2.2 Debit Result Sets

In evaluating debit result sets, the allocation engine examines each row of output to determine its account type, that is, 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 (that is, multiplies it by plus one); if the row a row represents credit balance, the allocation engine reverses the signage of the row (that is, multiplies it by minus one).

Table 47: Debit Result Sets

Allocation Output Tab	Account Type of Output Row	Rule
Debit	Debit	Multiply by +1
Debit	Credit	Multiply by -1

27.2.3 Credit Result Sets

Similarly, in evaluating credit result sets, the allocation engine examines each row of output to determine its account type, that is, 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 (that is, multiplies it by plus one); if the row a row represents debit balance, the allocation engine reverses the signage of the row (that is, multiplies it by minus one).

Table 48: Credit Result Sets

Allocation Output Tab	Account Type of Output Row	Rule
Credit	Debit	Multiply by -1
Credit	Credit	Multiply by +1

27.2.4 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, that is, 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.

27.2.5 Detailed Account Type Values

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.

Table 49: 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 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.

27.2.5.1 Determine the Account Type of the Common Chart of Accounts

When a row's fundamental account type (DR or CR) is not determined from the Account Type attribute associated with its Financial Element (when the Account Type attribute takes on the special < Use Detail Leaves > value), the allocation engine determines the row's debit or credit status. This is done by examining the Account Type attribute associated with its Common Chart of the Account value.

27.2.6 Designing Allocation Rules

One approach for avoiding undesired allocation results is to ensure that your allocation Sources are limited to either debit accounts or credit accounts. Another approach is to output your results to unambiguous Financial Elements, that is, 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 100% liability balances. On the output side of asset allocation, you might choose to debit a user-defined Financial Element whose Account Type is unambiguous (for example, Allocated Assets).

27.2.7 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.

27.2.8 Signage Methods

When OFSAA is installed, you must choose one of three available Signage Methods that indicate how you have chosen to store balances in the Management Ledger table.

- **Natural Signage Convention**: Under the Natural Signage Convention, balances initially loaded into the Management Ledger are 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, that is, their balances are normally negative. Examples of such accounts include loan loss reserves, depreciation, and amortization. However, any account, maybe in a contra state, that is, carries 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 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 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 to 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.

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.

27.2.9 Choosing a Signage Convention

Signage conventions have evolved over time 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 Management Ledger table. Regardless of the convention, you have chosen, OFSPA displays balance sheets and income statements using a visually pleasing "natural" convention (that is, all balances, with understood exceptions, are shown as positive balances).

The Natural Signage convention is the most commonly recommended method for OFSAA implementations.

27.2.10 Implications for Allocation Rules

Your choice of Signage convention does not affect 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 debits the receiving center with the negative expense (signage is preserved) and credits the source center with the 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 credits the target center with a negative liability balance (signage is preserved) and debits the source center with a positive liability balance (signage is reversed).

In short, the allocation engine is indifferent to (and does not know) the Signage convention you have adopted.

28 Appendix I: Performance Tuning

A typical Oracle Insurance Allocation Manager for Enterprise Profitability 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 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.

Topics:

28.1 SQL Hints

Oracle Insurance Allocation Manager for Enterprise Profitability 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 Oracle Insurance Allocation Manager for Enterprise Profitability allocation engine reads user-defined hints that are stored in the FSI_SQL_HINTS_OPTIONS table.

Profitability Management supports the following types of SQL hint:

- DML_HINT
- PERCENT_DIST_HINT
- TREE_FILTER_HINT
- FILTER_ON_HINT

Note the following:

- 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 needs to be manually inserted into the FSI_SQL_HINTS_OPTIONS table.
- The FSI_SQL_HINTS_OPTIONS table is indexed by the following:
  The system identifier (SYS_ID_NUM) of the rule to which the hint applies.
  By the allocation engine's PROCESS_ENGINE_CD (zero).
  Currently, only the Oracle Insurance Allocation Manager for Enterprise Profitability 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 apply to all types of allocations.

- TREE_FILTER_HINT applies only to allocation rules that include a Hierarchy Filter.
- PERCENT_DIST_HINT applies only to Dynamic Driver allocations.

28.1.1 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 Class Table, 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 is applied to sub-selects from the OFSA_IDT_ROLLUP table.
- **Percent Distribution Hint**: A hint inserted to the PERCENT_DIST_HINT column is applied to the SELECT statement that acquires driver data for Dynamic Driver allocation rules.
- **DML Hint**: A hint inserted into the DML_HINT column is applied to DML operations that insert into or update target tables, as, 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. For more information, see Profitability Management Parallel Execution.

28.1.2 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(fsi_d_management_ledger, 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 Oracle Insurance Allocation Manager for Enterprise Profitability 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.

28.1.3 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 are inserted into the queries. Note that while the examples all demonstrate hints that suggest parallelism to the database optimizer, you may utilize any valid form of a hint.

28.1.3.1 Filter on Hint

```
MERGE INTO LEDGER_STAT TARGET
USING (
SELECT  C.*
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 by     a.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 by     a.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)
```

28.1.3.2 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)
```

28.1.3.3 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.drv_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)
```

28.1.3.4 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)
```

28.2 Parallel Execution

Oracle Insurance Allocation Manager for Enterprise Profitability 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. 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 Insurance Allocation Manager for Enterprise Profitability 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 with 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 the 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 a Parallel Query can help, but the most allocation execution time is consumed by DML queries where you are modifying the database. 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:

Table 50: FSI_PARALLEL_PARAMETERS Table after Seeded 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 parameter values found in FSI_PARALLE_PARAMETERS apply to all allocation rules. Currently, Oracle Parallel Execution is only supported by Oracle Insurance Allocation Manager for Enterprise

Profitability. OFSAA Funds Transfer Pricing and OFSAA Asset Liability Management employ a different technique for parallelism: OFSAA Application Multiprocessing. For more details on OFSAA Multiprocessing, see the OFS Advanced Analytical Applications Infrastructure 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:

28.2.1 Parallelism Options

The following discusses the behavior of the Oracle Insurance Allocation Manager for Enterprise Profitability 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.

28.2.1.1 PARALLEL_QUERY

28.2.1.1.1 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.

28.2.1.1.2 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 with the use of the PARALLEL clause.

Two examples of the use of the PARALLEL clause to alter a table follow:

```
ALTER TABLE FSI_D_MANAGEMENT_LEDGER PARALLEL 8;
ALTER TABLE FSI_D_MANAGEMENT_LEDGER PARALLEL (DEGREE DEFAULT);
```

In the earlier first example, the database may elect to parallelize a SELECT statement against the FSI_D_MANAGEMENT_LEDGER table using a Degree of Parallelism equal to 8. In the second example, the database may elect to parallelize a SELECT statement against the FSI_D_MANAGEMENT_LEDGER 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 to 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.

28.2.1.1.3 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 is defined as having a specific Degree of Parallelism with 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 CPUs on your server. For details on how the Oracle database determines the default Degree of Parallelism, see Oracle Database VLDB and Partitioning Guide.

28.2.1.1.4 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 is 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.

28.2.1.1.5 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, you divide your one allocation rule into two rules (one rule for the debit side and one rule for the credit side).

Parallel DML cannot be set (regardless of the parameter settings) for any allocation rule where the Output tables (Credit/Debit) are the same as the Source/Driver tables.

For more information regarding the restrictions on Parallel DML, see the Oracle Database Data Warehousing Guide.

28.2.1.1.6 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.

28.2.1.1.7 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 the 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).

28.2.1.1.8 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 overrides 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.

28.2.1.1.9 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.

28.2.1.2 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 is enabled. These three topics are covered in depth by the documentation resources found at the end of this appendix.

28.2.1.2.1 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 before Oracle 11gR2. SQL statements are only processed in parallel for objects declared with a PARALLEL clause or if a parallel SQL hint is included. Additionally, the Oracle Insurance Allocation Manager for Enterprise Profitability 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).

28.2.1.2.2 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, an automatic Degree of Parallelism is enabled for some statements, but statement queuing and in-memory Parallel Execution is 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.

28.2.1.2.3 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 you can execute the statement 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.

28.2.2 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

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

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

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

29 Appendix J: Seeded Batches

The following are the seeded batches available in Oracle Insurance Allocation Manager for Enterprise Profitability application. The details of these batches are as follows.

29.1 <INFODOM>_LEAF_REGISTRATION_PFT_EXT (Leaf Registration Procedure)

To enable Banking and Insurance applications to co-exist in the same environment, a new Leaf Registration Procedure is introduced for adding new Key Dimension Columns to the Dimensions metadata registry.

29.1.1 Executing Leaf Registration Procedure

You can execute this procedure either from SQL*Plus or from within a PL/SQL block or from Batch Maintenance window within OFSAAI framework. To run the procedure from SQL*Plus, login to SQL*Plus as the Schema Owner. The function requires 21 parameters. The syntax for calling the function is:

function rev_leaf_registration_pft_ext(batch_run_id varchar2,

```
mis_date varchar2,
memDataType varchar2,
dimName varchar2,
description varchar2,
memberBTableName varchar2,
memberTLTableName varchar2,
hierarchyTableName varchar2,
attributeTableName varchar2,
memberCol varchar2,
memberDispCodeCol varchar2,
memberNameCol varchar2,
memberDescCol varchar2,
dimTypeCode varchar2,
simpleDimFlag varchar2,
keyDimFlag char,
writeFlag  varchar2,
catalogTableType char,
flattenedTableName in varchar2,
membercodecol in varchar2,
appId in varchar2 default null)
```

29.1.2 Example for registering a key dimension:

```
DECLARE
RESULT NUMBER;

BEGIN
     RESULT := REV_LEAF_REGISTRATION_PFT_EXT('LOB ID REGISTRATION',
        TO_CHAR(SYSDATE, 'YYYYMMDD'),
        'NUMBER',
        'Line of Business',
        'Line of Business',
        'DIM_LOB_B',
        'DIM_LOB_TL',
        'DIM_LOB_HIER',
        'DIM_LOB_ATTR',
        'LOB_ID',
        'LOB_DISPLAY_CODE',
        'LOB_NAME',
        'DESCRIPTION',
        'OTHER',
        'N',
        'Y',
        'Y',
        'B',
        'LOB_FLATTEN',
        'LOB_CODE',
        'OFS_PFT_INS');
        END;
/
```

29.2 1.2 <INFODOM>_T2T_ALL_POLICY_TABLES

This seeded batch contains the following Instrument Table T2Ts:

- T2T_HEALTH_INS_CONTRACTS
- T2T_TERM_POLICIES_DATA
- T2T_INS_RETIREMENT_CONTRACTS
- T2T_ENDOWMENT_POLICIES_DATA
- T2T_PROP_CASUALTY_CONTRACTS

- T2T_INS_ANNUITY_CONTRACTS
- T2T_WHOLELIFE_POLICIES_DATA
- T2T_REINS_CONTRACTS_HELD
- T2T_REINS_CONTRACTS_ISSUED

You can edit this batch to add or remove other Policy Table T2Ts.

29.3 <INFODOM>_T2T_ALL_POLICY_TXN_SUMMARY

This seeded batch contains the following Transaction Summary Table T2Ts:

- T2T_STG_PROP_CAS_POL_TXN_SUMMARY
- T2T_STG_LIFE_INS_POL_TXN_SUMMARY

You can edit this batch to add or remove other Transaction Summary Table T2Ts.

29.4 <INFODOM>_T2T_MANAGEMENT_LEDGER_PFTINS

This seeded batch along with the underlying DTs and T2T enables tracking of data from the Staging table to the Management ledger table using identity codes. For each distinct value of "Data Origin" in the Staging table, a new Identity code is used to populate the Management ledger and FSI_DATA_IDENTITY tables. The value of V_DATA_ORIGIN in the Staging table is mapped to the FSI_DATA_IDENTITY.description column.

The seeded batch named <INFODOM>_T2T_MANAGEMENT_LEDGER_MULTI_SOURCE will have the following three tasks in the sequence:

1. A DT named FN_MGMT_LED_LOAD_TEMP_TABLE, which will populate the T2T_FSI_D_MANAGEMENT_LEDGER table with unique identity_code values for each of the distinct v_data_origin values in the STG_GL_DATA table. It accepts two parameters - Target Table Name and Staging Table Name.
2. T2T named T2T_MANAGEMENT_LEDGER_MULTI_SOURCE, which will use the identity codes from the table T2T_FSI_D_MANAGEMENT_LEDGER for loading data from STG_GL_DATA to FSI_D_MANAGEMENT_LEDGER.
3. Another DT named FN_MGMT_LED_LOAD_DATA_IDENTITY, which will populate the FSI_DATA_IDENTITY table with the information about the Management Ledger load. It maps the V_DATA_ORIGIN column in the STG_GL_DATA table to the DESCRIPTION column in the FSI_DATA_IDENTITY table. It will also delete the records from the T2T_FSI_D_MANAGEMENT_LEDGER table.

NOTE

1. The results of the DT execution are written to the FSI_MESSAGE_LOG table.
2. The OOB T2T uses STG_GL_DATA as the staging table. The T2T has is customized to use any other staging table as the source.
3. If a custom Staging table is used instead of STG_GL_DATA, the table needs to have the columns FIC_MIS_DATE and V_DATA_ORIGIN.

29.5 <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 that needs to be removed.

As of Date: MIS DATE of Ledger Load in MM/DD/YYYY format.

29.6 <INFODOM>_DIMENSION_POLICY

This seeded batch contains the following SCDs to populate the dim_policy table:

- STG_LIFE_INS_CONTRACTS
- STG_ANNUITY_CONTRACTS
- STG_RETIREMENT_ACCOUNTS
- STG_HEALTH_INS_CONTRACTS
- STG_PROP_CASUALTY_CONTRACTS
- STG_REINSURANCE_CONTRACTS_HELD
- STG_REINSURANCE_CNTRCTS_ISSUED

30 Glossary

This glossary lists the popular terminologies from the current document.

Table 51: Glossary of Terms

Term	Description
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	Oracle Insurance Allocation Manager for Enterprise Profitability 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), an ad-hoc reporting tool.
As of Date	The 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 the 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.
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.
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 the calculation of interest.
Circular Allocation Models	The 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).

Term	Description
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 Oracle Insurance Allocation Manager for Enterprise Profitability.
Data Filter Rule	Data Filters are used to define which data should go into a processing run.
Data set	A dimension is 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 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 are used to populate dimension columns in the 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.

Term	Description
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.
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	The portion of funding center rate risk result attributed to prior rate bets.
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 Oracle Insurance Allocation Manager for Enterprise Profitability 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. General Ledger reflects all accounting entries.
Grid	A logical grouping of cells often surrounded by scroll bars.
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 the Moving Averages transfer pricing method.
Instrument	Synonymous with an account record or an individual contract.
Instrument Table	A type of table contained in the OFSAA database is 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.
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.

Term	Description
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.
Management Ledger Table	The Management Ledger table is 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.
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.
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 Enterprise Financial Performance Analytics.
Oracle Financial Services Balance Sheet Planning	Balance Sheet Planning performs distributed budgeting of all balance sheet and income statement accounts.

Term	Description
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.
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.
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).

Term	Description
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.
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 Oracle Insurance Allocation Manager for Enterprise Profitability rule type called Static Table Driver.
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.
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.
Yield Curve	Term structure of annually compounded zero-coupon bond yield, as recorded in Rate Management > Historical Rates.

OFSAA Support

Raise a Service Request (SR) in My Oracle Support (MOS) for queries related to OFSAA applications.

Send Us Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, indicate the title and part number of the documentation along with the chapter/section/page number (if available) and contact the Oracle Support.

Before sending us your comments, you might like to ensure that you have the latest version of the document wherein any of your concerns have already been addressed. You can access My Oracle Support site that has all the revised/recently released documents.