Oracle Financial Services

Allocation Specifications User Guide

Release 22.12.01
F76888-01
January 2023

ORACLE

Oracle Financial Services Allocation Specifications User Guide, Release 22.12.01

F76888-01

Contents

3 Allocation Examples

1
Get Help

Topics:

1.1 Get Help in the Applications

Use help icons to access help in the application.

Note that not all pages have help icons. You can also access the Oracle Help Center to find guides and videos.

1.1.1 Additional Resources

- Community: Use Oracle Cloud Customer Connect to get information from experts at Oracle, the partner community, and other users.
- Training: Take courses on Oracle Cloud from Oracle University.

1.2 Learn About Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program. Videos included in this guide are provided as a media alternative for text-based topics, and are also available in this guide.

1.3 Get Support

You can get support at My Oracle Support.

For accessibility support, visit Oracle Accessibility Learning and Support.

1.4 Get Training

Increase your knowledge of Oracle Cloud by taking courses at Oracle University.

1.5 Join Our Community

Use Cloud Customer Connect to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, and watch events.

1.6 Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we would like to hear from you.

You can email your feedback to My Oracle Support.

Thanks for helping us improve our user assistance!

1.7 Before You Begin

See the following Documents:

- See What's New
- Getting Started with Profitability Management Cloud Service

2

Allocation Specification

Profitability and Balance Sheet Management (PBSM) Cloud Service's Allocation Specification Documentation covers the following topics:

- Allocation Specification Summary & Detail Screens
- Navigation within the Allocation Specification Summary Screen
- Navigation within the Allocation Specification Detail Screen
  - Initial Definition Process Tab
  - Operator Process sTab
  - Driver Process Tab
  - Outputs Process Tab
  - Review Process Tab
- Allocation Examples

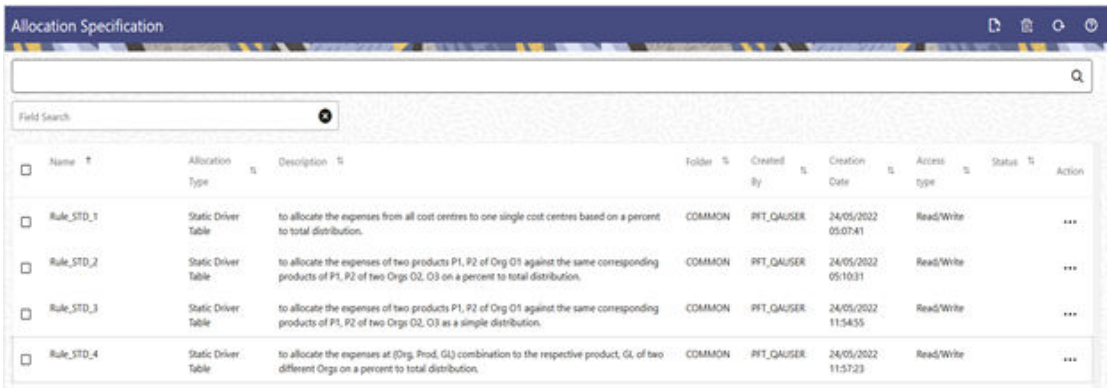
2.1 Summary Screen

To open the Allocation Specification Summary Screen, select **Profitability Management** from the LHS menu, select **Rule Specification**, and then select **Allocation Specification**.

2.1.1 Navigation in Summary Screen

When you navigate to the Allocation Specification Summary Screen for the first time, the Allocations stored within your current default folder are presented in a Summary Table.

Figure 2-1 Allocation Specification Summary Screen



The title bar of the Summary Screen displays several actions for the user. They are:

- **Add**: Click Add to build a new Allocation Rule. The Add icon is disabled if any rows in the table are selected.
- **Multiple Delete**: Select one or more rules in the table and then click Delete at the top right of the Summary Page to delete more than one rule at the same time.
- **Refresh**: Click Refresh to refresh the Summary Page.
- **Help**: Click Help to view the Allocation Specification Help Page.

The Allocation Specification Summary can be divided under two sections – the Search Section and the Summary Table.

2.1.1.1 Search

There are two Search options provided to search the Allocation Specifications on the Summary Screen.

To search the Allocation Rules, perform the following steps:

1. Click the **Search** icon on the Search Pane to display the Criteria window.
2. Enter the Allocation Rule **Name**, **Description**, **Folder**, or the **Allocation Type** and click **Search** to display the Allocation Rules that match the criteria.
3. Click **Cancel** to remove the filter criteria on the Search window and refresh the window.
4. Click **Search** after entering the Search Criteria. The screen displays the search results that meet the Search Criteria in a table containing all the Allocation Rules.
5. The other method to search an Allocation Rule is using the **Field Search** option. The Field Search is an inline wildcard search that allows you to enter value partially or fully and the rows that match the entered string in any of its column is fetched in the Summary Table.

2.1.1.2 Allocation Specification Summary Table

This section presents a table containing all of the Allocation Rules that meet your Search Criteria. The Allocation Specification Summary Table displays the details of the already created Allocation Rules.

The Allocation Specification Summary Table displays the following details:

- **Name**: Displays the Allocation Rule's Short Name. Hovering over an Allocation Name displays the Allocation Rule's Object_ID and the Object_Code.
- **Allocation Type**: Displays an Allocation Rule's Type. The following Rule types are supported:
  - Constant
  - Static Driver
  - Leaf
  - Field
  - Dynamic Driver
  - Static Driver Table
  - Lookup Driver Table

- **Description**: Displays the Allocation Rule's Long Name.
- **Folder**: Displays the Folder in which the Rule is created.
- **Created By**: Displays the name of the User who created an Allocation Rule.
- **Creation Date**: Displays the Date and Time at which an Allocation Rule was created.
- **Access Type**: Displays the “Read/Write” or “Read Only” property of an Allocation Rule. The creator of a Rule only may change its Access Type.
- **Status**: Before executing an Allocation Rule for the first time, the Status is blank. After executing an Allocation Rule, the appropriate status of the Rule is displayed among Ongoing, Success, or Failed.
- **Action**: Displays the list of actions that can be performed on the Rule.

The Action column on Allocation Specification Summary Page offers the following actions that allow you to perform different functions. The following actions are available for the Allocation Rule.

- **View**: Click the View icon to view the contents of an Allocation Specification Rule on a Read-Only basis as the user is launched into the Allocation Specification Detail Screen in View Mode.
- **Edit**: Click the Edit icon to modify a previously saved Allocation Specification Rule as the User is launched into the Allocation Specification Detail Screen in Edit Mode.
- **Run**: Click Run to execute the selected Allocation Specification Rule. On click of **Run**, the Run Execution Parameters Window opens up to show the process name being executed and take User input of Run Time Parameters – the As-of-Date and the Legal Entity. The As-of-Date can be reset in the User Preferences for Profitability Management.
- **Save As**: Click on this option to create a copy of an existing Allocation Specification rule. The Save As pop-up window allows you to enter the Name, Description, Folder, and Access Type Details for the copy rule.
- **Delete:** Click Delete to delete the rules you have selected.
- **Check Dependencies**: This action button is to check for any dependency of the selected object with other objects in the application. On click of this action, the Dependent Information window is displayed with the Object Name, Object Type, Object Subtype and the Version of the dependent objects. The ‘Higher Order Dependency’ states if the selected object has an upstream objects dependency and is to be treated as the actual dependency of the selected object. While, the ‘Lower Order Dependency’ displays the downstream objects dependency of the selected object. If an object has a Higher Order Dependency then the object cannot be deleted without removing the dependency first.

You may select or de-select all of the Allocation Rules in the Summary Table by clicking on the check-box in the upper left-hand corner of the Summary Table directly to the left of the Name Column Header.

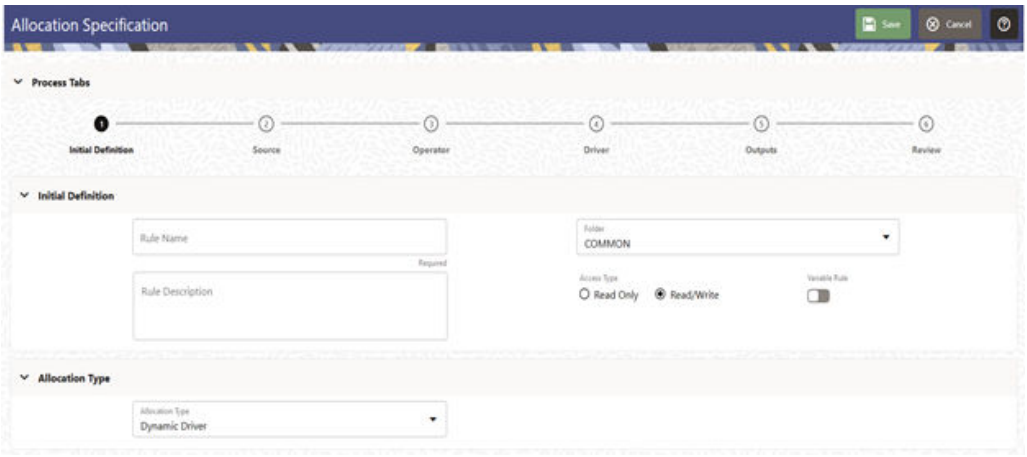
2.2 Detail Screen

Click on **Add** from the Title bar of Summary Screen or Edit/View an Allocation Rule from Summary to launch into the Allocation Specification Detail Screen.

2.2.1 Navigation in Detail screen

The Allocation Specification Detail Screen is composed of six Process Tabs that are described in the following sections. The appearance of the Detail Screen depends on the Tab that is active, which in turn is dependent on the Allocation Type selected for the Rule.

Figure 2-2 Allocation Specification Detail Screen in New Mode



2.2.1.1 Process Tabs Pane

The Process Tabs Pane is arranged in a linear train fashion with each button or step representing one process. The button icon is blank when the tab is not defined and turns green when the definition is complete. A tab that is undergoing modification turns black.

Each of the six Process Tabs is designed to create, edit, or view different components of an Allocation's Specification. You may navigate from one tab to any other tab at any time. The six Process Tabs are as follows:

- Initial Definition
- Source
- Operator
- Driver
- Outputs
- Review

2.2.2 Initial Definition Process Tab

The Detail Screen launches itself into the first tab called the Initial Definition Process Tab. The Initial Definition Tab is organized under two panes.

The Initial Definition pane allows user to specify the Rule Name, Rule Description, Folder, and the Access Type of an Allocation Rule.

The Allocation Type pane allows you to specify the Allocation Type of the Allocation Rule.

2.2.2.1 Initial Definition Pane

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 is saved. Default values for Folder and Access Type are stored in Application Preferences for Profitability Management Cloud Service. It also hosts the toggle button of Variable Rule.

Figure 2-3 Initial Definition Pane

2.2.2.1.1 Variable Rule

Legal Entity is an optional Run-time Parameter. If the Disable Legal Entity check box in the Application Preferences Screen is checked, then the Legal Entity will no longer be a Run-time Parameter.

If Legal Entity is disabled, then you must select a value for the Legal Entity Dimension in all applicable tabs while defining a Variable Allocation Rule in the Allocation Specification Screen.

Figure 2-4 Initial Definition Pane with Variable Rule selected

For already defined Variable Allocation Rules, edit the Allocations, and modify the value of the Legal Entity Dimension in all applicable tabs in the Allocation Specification Screen. The value of Legal Entity is used when your Rule Execution is specified within your Batch Definition (for Batch Processes) or is obtained from your Profitability Management Cloud Service Application 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 your Allocation Rules to be Variable.

2.2.2.2 Allocation Type Pane

When you initially build an Allocation Rule, you must select its Allocation Type. After an Allocation Rule is 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 depends upon the Allocation Type you have chosen.

The available rule types are as follows:

- Constant
- Static Driver
- Leaf
- Field
- Dynamic Driver
- Static Driver Table

2.2.2.3 Allocation Types

The following list describes the Allocation Types:

- **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; 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. The source balances can 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, 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 used to operate between two sets of rows that differ in a single Dimension.
- **Field**: A Field type Allocation is used to multiply two columns within a single row in an Instrument Table update Allocation Rule.
- **Dynamic Driver**: 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 is not limited to statistics 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. The Driver Data obtained from your instruments is not limited to balances. Examples of Instrument Level Dynamic Drivers you might use in Allocation Rules include:

  - The number of accounts by product by Cost Center by year of origination.
  - The number of loan payments processed by the loan processing center by month.
  - ATM transaction counts by region by month.

  Uniform Method: The most common Distribution Methods for the Dynamic Driver type of allocation are:

  - Percent-to-Total
  - Force to 100%
  - Simple Method
    Dynamic Driver Allocation Rules and Methods are described in detail in the Driver Process Tab and Output Process Tab sections.

- **Static Driver Table**: Static Driver Table Allocation Rules offer functionality similar to Dynamic Driver Allocation Rules but use Driver Data that is stored in a Profitability Management Rule type called Static Table Driver. For more information on how to build and use Static Table Driver Rules, see Static Table Drivers.

2.2.2.4 Definitions of Static and Dynamic Drivers

Most Allocation Rules distribute, or aggregate balances using the Driver Data and can be used in the following ways:

- 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 can be utilized as drivers within your Allocation Rules. These kinds of Business-Data Resident Drivers are referred to as Dynamic Drivers.

In some 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. The Profitability Management Cloud Service supports the following types:

- Static Driver
- Static Driver Table

Dynamic Drivers often have many advantages over Static Drivers. An Allocation Rule that uses a Static Driver takes 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, you must normally pre-compute your Static Drivers and

Dynamic Drivers that are generated at Run time. Dynamic Drivers, frequently used in full cost-Absorption Allocation Models, are generated by other Allocation Rules.

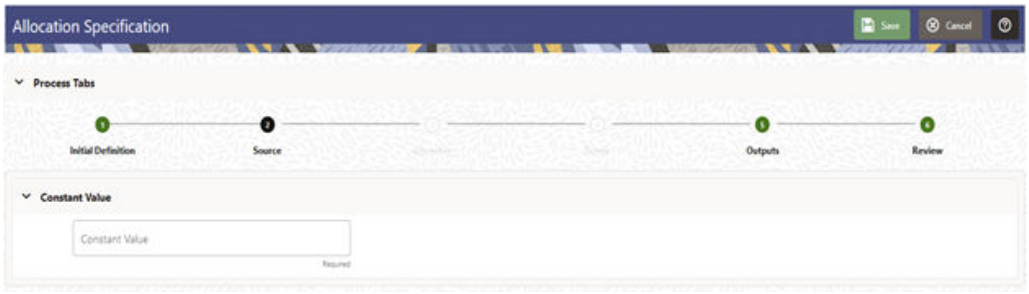
Static Drivers sometimes have advantages over Dynamic Drivers. For example, you may have pre-computed unit costs that you wish to use to drive your allocations to generate Partial Absorption Costing.

2.2.3 Source Process Tab

The Source Process Tab is to be used to specify an Allocation Rule's Data Source.

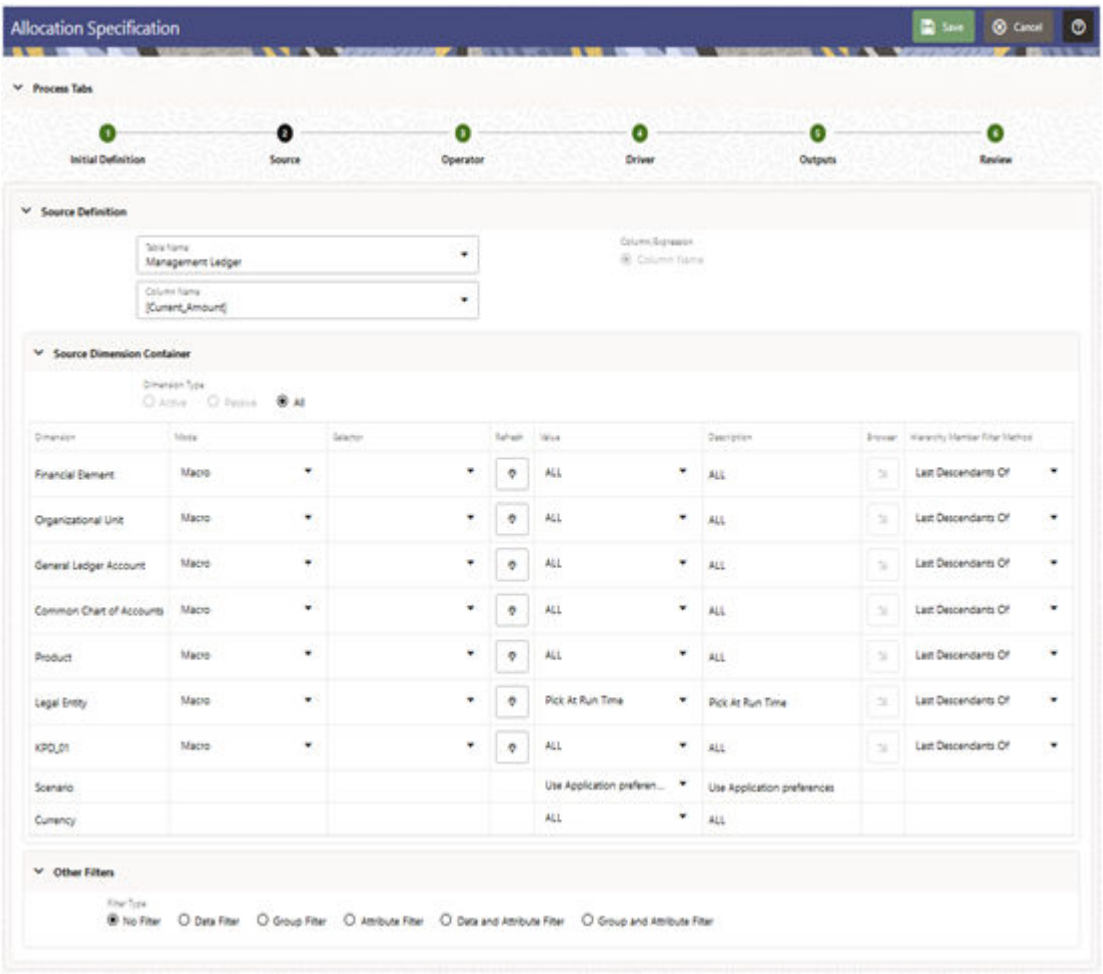
For a Constant Type Allocation Rule, you need to specify an amount as follows:

Figure 2-5 Allocation Specification Source Process Tab



For all other types of Allocation Rule, the Source Process Tab contains the Source Definition Pane, the Source Dimension Container, and Other Filters Pane.

Figure 2-6 Source Process Tab for all rule types except Constant type



2.2.3.1 Source Definition Pane

The Source Definition Pane is the parent pane of the Source process tab and hosts two sub sections of – the Source Dimension Container and the Other Filters. This parent pane allows the user to select a Source Table and then select a Column from the selected table that serves the Source Data for the Allocation Rule. This version of the application does not support Expressions.

Figure 2-7 Source Definition Pane

2.2.3.1.1 Management Ledger Source

This section describes using the Management Ledger as the Source in Allocation Rules.

The Management Ledger is a Seeded Table of a new Management Ledger Class of Tables (see the Data Model Extension Guide for details on adding User-Defined Dimensions to the Management Ledger or for defining new User-Defined Management Ledger Tables.)

When your Source is the Management Ledger Table, 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> is interpreted as Fiscal Month 3. If your fiscal year begins in April, your March data is stored in the Management Ledger under Fiscal Month 12, since March is the last month in your Fiscal Year.

> **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 (such as in Allocation Sources or Allocation Drivers), it is consolidated into a month-to-date balance. For example, if you perform daily instrument-level Funds Transfer Pricing and if you use an Allocation Rule to aggregate daily Funds Transfer Pricing Charges or Credits from the instrument level to the Management Ledger, each day's charges and credits are posted to the Management Ledger by As-of-Date (by business date). Allocation Rules that reference these Management Ledger balances, however, combine the daily postings to obtain month-to-date balances. Similarly, all outputs at the Management Ledger level are inherently month-to-date balances.

When your Source is the Management Ledger table, the following macros are supported:

- <Last_Mo_Amount>
- <Months_Ago_Amt>
- <Current_Amount>
- <YTD_Amount>
- <Months_Ago_YTD_Amt>
- <Last_Mo_YTD_Amt>

<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 in the Enter Months field, enter 6 months ago. This corresponds to September 2014 (Fiscal Month = 9, Fiscal Year = 2014). The month range for entering Months is from -99 to 999.

> **Note:**
>
> When you select any of these macros, the Entered_Balance column in the Management Ledger Table is selected.

2.2.3.1.2 Instrument or Transaction Summary Source

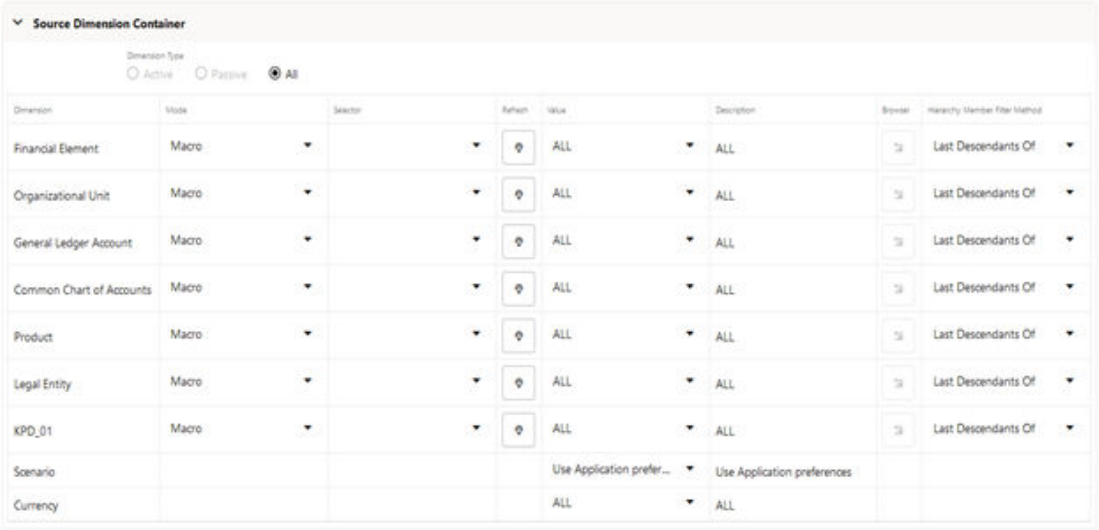
When your Source is an Instrument or Transaction Summary Table, you may choose any valid measure in the table. Valid measures include only rates, balances, and numeric statistics such as activity counts.

2.2.3.2 Source Dimension Container

The Source Dimension 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 roll-up node member within a Hierarchy, or a Hierarchy Filter. As and when a placeholder KPD is registered, the registered KPD appears in the dimension list in the order of the dimension number.

The Source Dimension Container is table that lists the OOTB and custom dimensions available for the rule, as follows:

Figure 2-8 Source Dimension Container



The Dimension Container starts with the Dimension Type radio button selection that displays the active, passive or all dimensions available for the allocation rule, through the options of Active, Passive and All respectively.

A Passive dimension is a dimension that the user has not applied any constraint to, and it comes with default Mode of ‘Macro’ and default Value as ‘ALL’.

Similarly, on the other hand, an Active dimension is a dimension that the user has applied a constraint to, and its Mode is not ‘Macro’ and Value is not ‘ALL’.

The Dimension column holds the Key Processing Dimensions for the Source Definition.

The separation of dimensions into shorter Active and Passive list was introduced to solve the problem of a long list of dimensions to choose from while in Edit mode. The dimensions list would be long when all the placeholder KPDs are registered, and user would need to search through the list to find the KPD he/she wants to apply constraint to.

In New mode, the Dimension Type radio button is defaulted as All while the other buttons are disabled.

In Edit mode, the Dimension Type is defaulted as Active and only the dimensions for which a constraint has been applied, is displayed under the Active set. All the dimensions for which a which a constraint has not been applied, in shown in the Passive set which can be viewed through changing the radio button to Passive. User can apply constraint to any dimension in the Passive set and the dimension comes to the Active set, that can be observed by changing the radio button to get the current Active set. In Edit mode, the All option is disabled.

In View mode, the allocation UI defaults to Active, and the user can toggle between the two sets of Active and passive to view the constraint applied or not applied dimensions. The Mode displays a drop-down containing four modes of defining Constraint to a Dimension – Macro, Leaf, Node, and Hierarchy Filter.

The default mode for all Dimensions is **Macro** and the default value for each Dimension is **ALL**. ‘ALL’ means no constraint is applied to the Dimension.

Leaf Mode must be used when you want to input a Leaf Value for a Dimension. After you select the Leaf Mode, you need to click on the **Refresh** icon in the Refresh Column to load the Leaf Members in the Value column. You can choose a member from the list of Leaf Members in the Value drop-down, or can type-ahead the required leaf member name in the Value Text Box. This free text type-ahead feature comes with Autosuggestion that creates a Dynamic List of values in the drop-down, matching the input string. If the Type-Ahead string does not match with a value from the drop-down list, the string will not be accepted.

Node Mode is to be used when you want to input a node value for a Dimension. After you select the Node mode, the Selector Column drop-down is populated with all the available Hierarchies for that dimension.

> **Note:**
>
> If no hierarchy can be found for the dimension you have selected, the Selector column will be blank. The Selector values appear as Folder name, hyphen, and Hierarchy name for each of the hierarchies available for that Dimension. The Folder name gives the name of the folder in which the displayed hierarchy is stored. You need to select one hierarchy from the Selector drop-down. This facilitates node member selection particular to that hierarchy. After the hierarchy is selected, you need to click on the Refresh icon in the Refresh column to load the Node Members in the Value column, i.e., to display the Node Members available for the hierarchy chosen in the Selector. You can choose a member from the list of node members in the Value drop-down, or type the required node member name in the Value text box. This free text type-ahead feature comes with Autosuggestion that creates a Dynamic List of values in the drop-down, matching the input string. If the type-ahead string does not match with a value from the drop-down list, the string will not be accepted.

Hierarchy Filter Mode is to be used when the user wants to apply a Hierarchy Filter constraint on a dimension. Hierarchy Filter mode works in similar fashion as that of Node mode. Once you select the Hierarchy Filter mode, the **Selector** Column drop-down is populated with all the available Hierarchies for that Dimension.

> **Note:**
>
> If no hierarchy can be found for the dimension you have selected, the Selector column will be blank. The Selector Values appear as Folder Name, Hyphen, and Hierarchy name for each of the hierarchies available for that Dimension. The Folder name gives the name of the folder in which the displayed hierarchy is stored. You need to select one Hierarchy from the Selector drop-down. This facilitates Hierarchy Filter selection particular to that Hierarchy. After the Hierarchy is chosen, you need to click the **Refresh** icon in the Refresh column to load the Hierarchy Filters in the Value column, that is to display the Hierarchy Filters available for the Hierarchy chosen in Selector.
> If no Hierarchy Filter can be found for the Hierarchy you have selected, the Value column will be blank. You can choose a filter from the list of filters in the Value drop-down or can type-ahead the hierarchy filter in the Value text box. This free text type-ahead feature comes with Autosuggestion that creates a Dynamic List of values in the drop-down, matching the input string. If the type-ahead string does not match with a value from the drop-down list, the string will not be accepted.

The **Selector** Column, as described earlier, displays the Folder Name and the Hierarchy available for the dimension. Selector is used for the Node mode and the Hierarchy Filter mode where selection of the Hierarchy is essential before selecting the node member or the hierarchy filter. The Selector values appear as Folder name, Hyphen, and Hierarchy name for each of the hierarchies available for that dimension. The Folder name gives the name of the folder in which the displayed hierarchy is stored.

The **Refresh** Column is used to refresh and load the appropriate values in the Value column.

The **Value** Column displays the value of a Macro, Leaf Member, Node Member, or a Hierarchy Filter. The member/filter value appears as Member Name, Hyphen, and Member ID. You can choose a value from the list of values (that are essentially members and filters) in the Value drop-down, or can type-ahead the required member/filter in the Value text box. This free text type-ahead feature comes with Autosuggestion that creates a dynamic List of values in the drop-down, matching the input string. If the type-ahead string does not match with a value from the drop-down list, the string will not be accepted.

The **Description** Column displays the description of the member/filter selected in the Value column. The description includes information on the Dimension, Folder where the hierarchy is stored, the level of the member in the hierarchy, the member's name, and the member ID.

Next comes the column that hosts the Hierarchy Browser Widget, clicking on which invokes the **Hierarchy Browser**. The Hierarchy Browser functionality differs across the various modes of constraint selection. The following section discusses more details.

The last column, Hierarchy Member Filter Method is also discussed in the next section, along with the Hierarchy Browser.

Source Scenario: For Allocation Rules that Source Data from the Management Ledger-level, you must select a Source Scenario from the Allocation Source Pane. 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 Profitability Management Cloud Service 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, and Forecast Prior). These values are provided with the data model, but you may add additional Dimension Members in the Consolidation Code Dimension.

Table 2-1 Examples of using the Source Tab

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 and 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 Pane are limited to your Key Processing Dimensions. The seeded Key Processing Dimensions for all OFS Analytical Applications are:

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

2.2.3.2.1 Hierarchy Browser on the Source Tab

We have learnt from the previous section how to define Dimension Constraints through user inputs, either through selection from drop-down or through free text type-ahead. The other way of Constraint Application is through the Hierarchy Browser.

The Hierarchy Browser Widget is enabled for the two modes of Leaf and Node. The widget is disabled for the Macro Mode and the Hierarchy Filter Mode.

To select a Constraint, click on the Hierarchy Browser icon in the column adjacent to the Description column in the Source Dimension Container table, next to the Dimension you want to constrain.

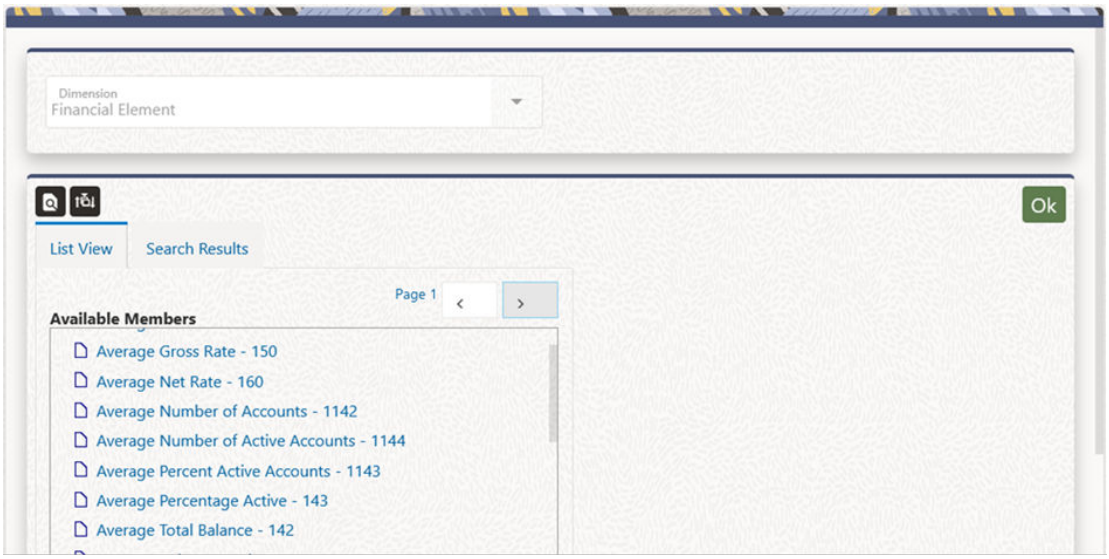
2.2.3.2.1.1 Hierarchy Browser for Leaf Mode

When in Leaf Mode, the Browser opens to show the List view of all Leaf Members available for the selected Dimension, in the List View Tab. The members appear as Member name, hyphen, Member ID. You can select a Single-Leaf Member by directly clicking on the Member.

The browser has only two action buttons – Search and Sort. You can search by clicking the Search icon and auto wild card search with search criteria as ‘contains' is applied on the Member ID and the Member Name that tries to match with the input string in the Search box. The Search results are displayed in the Search Results Tab.

The Sort function helps to sort the members in alphabetical order or Member Name. The Ascending or the Descending order of Sort action is as per user click and happens alternatively. Alternate member selection: You can select a Leaf Member from the Search Results Tab from among the search results.

Figure 2-9 Financial Element Hierarchy Browser – List View Tab



2.2.3.2.1.2 Hierarchy Browser for Node Mode

When in Node Mode, the browser opens to show the Hierarchy view of the selected Dimension Hierarchy, in the Hierarchy View Tab. The Hierarchy View displays the list of all the Node and Leaf Members for the Hierarchy. The members appear as Member Name, Hyphen, and Member ID. You can select a Single-Node Member by directly clicking on the Member. The Leaf Members are disabled for user selection.

The browser has four action buttons – Search, Sort, Expand All/Collapse All, and Focus/ Unfocus. You can search by clicking the Search icon and Auto Wild Card Search with Search Criteria as 'contains' is applied on the Member ID and the Member Name that tries to match with the input string in the search box. The Search results are displayed in the Search Results Tab.

The Sort function works on a selected node that has Child Members under it. It sorts only the immediate level child members (of the selected node member) into ascending or descending alphabetical order of Member Name. The Ascending or the Descending order of Sort action is as per the user click and happens alternatively. The default Sort order of members is as per the display order of members in the hierarchy definition.

The Expand-All/Collapse-All function works on a selected node that has Child Members under it. The function expands or collapses the selected node until the level of the Leaf Members under the selected node.

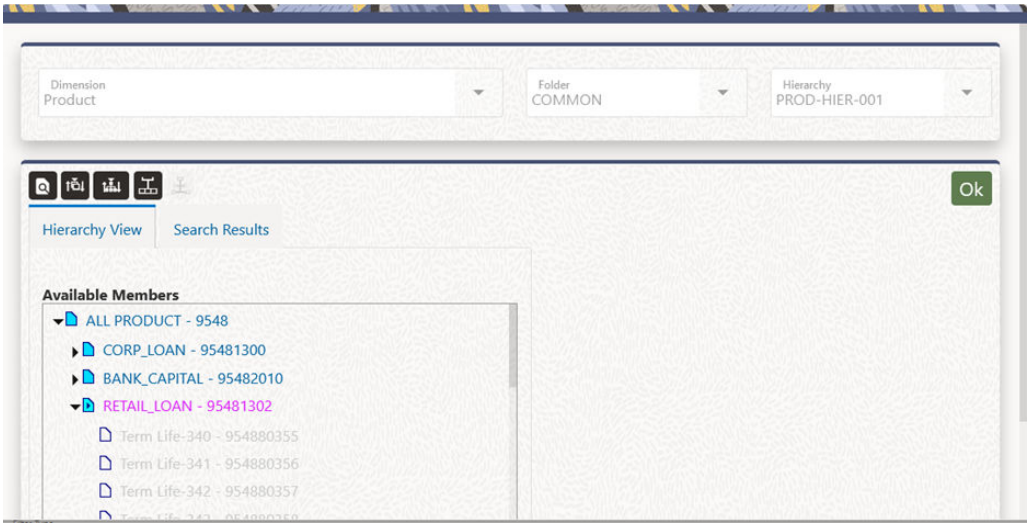
The Focus/Unfocus functions to focus a searched Member (from the Search Results tab) into the Hierarchy View with the position of the Member in the Hierarchy. When you click for the first time on Focus, the next click does not unfocus the focussed Member from the Hierarchy View back in the Search Results tab. Focus is on the first click on the icon, and the next click does an unfocus of the focussed Member from the Hierarchy View back in the Search Results tab.

The Hierarchy View Tab or the Search Results Tab gives a full parentage information, on mouse hover, of each of its members starting from the root node until the mouse-hovered member.

Pagination is applicable only for a selected Node Member that has children spanning across multiple pages. On mouse-click of such Member, a pagination capability is displayed on the top right corner of the Available Members box. You can navigate across the pages to view the Members displayed in other pages.

Alternate Member Selection: You can select a node member from the Search Results Tab from among the search results.

Figure 2-10 Product Hierarchy Browser



2.2.3.2.1.3 Hierarchy Member Filter Method

When the Source Table is Management Ledger and the Allocation Type is either Static Driver or Dynamic Driver, you can select a Hierarchy Member with additional options. You can select the Hierarchy Member by selecting any of the four “Hierarchy Member Filter Method”:

1. Node Only
2. Last Descendants Of
3. Descendants Of
4. Node and Descendants Of

This selection is applied at the per-dimension level of the Source Dimensions, meaning you have the liberty to select the “Nodes Only” filter method for the General Ledger Dimension while applying a “Last Descendants Of” Filter Method on the Organization Unit Dimension.

The Data Loader Service that loads the data from Staging to Management Ledger supports loading data to any kind of member in the Management Ledger – Leaf-Level members or Node Level members, rendering the user to select any of the two kinds of the member from the UI.

- If you choose the Hierarchy Member Filter Method as "Nodes Only", only the Node Member is selected in the filter.
- If you choose the Hierarchy Member Filter Method as "Last Descendants Of", only the Leaf Members of the Hierarchy rolling up to the selected Node are selected in the filter.
- If you choose the Hierarchy Member Filter Method as "Node and Descendants Of ", all the Descendant Nodes and Leaves of the selected Node in the Hierarchy including the selected Node itself, are selected in the filter.
- If you choose the Hierarchy Member Filter Method as "Descendants Of ", all the Descendant Nodes and Leaves of the selected Node in the Hierarchy but excluding the selected Node itself are selected.

2.2.3.3 Other Filters Pane

You may optionally select a Data Filter, a Group Filter, an Attribute Filter, or a combination filter between a ‘Data and Attribute Filter’ and a ‘Group and Attribute Filter’ to further constrain your Source Data.

Figure 2-11 Other Filters Pane

Data Filter or Data Element Filter is a stored rule that expresses a set of constraints on table columns. You can build a data filter on regular columns of instrument tables, management ledger tables, transaction summary tables, the portfolio table, or a lookup table or on any registered placeholder column.

We can combine different data filters with a logical AND condition and we call this new filter as the Group Filter. A group filter applies the conditions of all the individual data filters, over the dataset it is applied on.

Attribute filters are used to filter records based on one or more dimension type Attributes. An attribute filter filters the records for which the dimension members satisfy the given attribute conditions.

A Hierarchy filter can be used to filter members from a dimension hierarchy. Applying this filter, it filters the records that contain the members specified in the hierarchy filter.

2.2.4 Operator Process Tab

The Operator Process Tab allows you to specify how the Source data and Driver Data interact to create results.

- **Operator Process Tab for Constant Rules**: No Driver is necessary to specify a Constant Rule. Both the Operator Process Tab and the Driver Process Tab are disabled for the Constant Rule type.

- **Operator Process Tab for Static Driver Rules**: For Static Driver Rules, the Driver Process Tab is disabled, but the Operator Process Tab is enabled to allow you to specify a Static Driver Balance. Static Amounts are entered into the Factor Operator Pane.

Figure 2-12 Allocation Specification - Operator Process Tab

- 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.

2.2.4.1 Factor Operator Pane

Factor Operators may be used to either:

- To store static driver amounts and/or accrual basis macros for Static Driver Rule types.
- To 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.

2.2.4.1.1 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. 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 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 “headcount 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.

2.2.4.2 Allocation Operator Pane

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.

2.2.5 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. How the Source Data and the 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 the following Allocation Types:

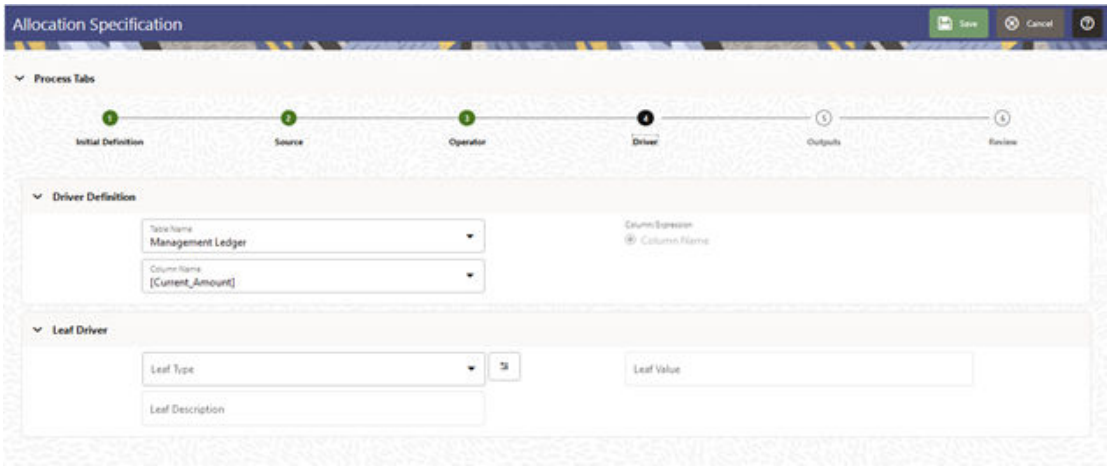
- Leaf
- Field
- Dynamic Driver
- Static Driver Table

The Panes displayed on the Driver Process Tab vary according to the different types of Rules.

2.2.5.1 Leaf

For a Leaf Allocation Type, the Driver Process Tab comprises of the Driver Definition Pane and the Leaf Driver Pane.

Figure 2-13 Allocation Specification Driver Process Tab

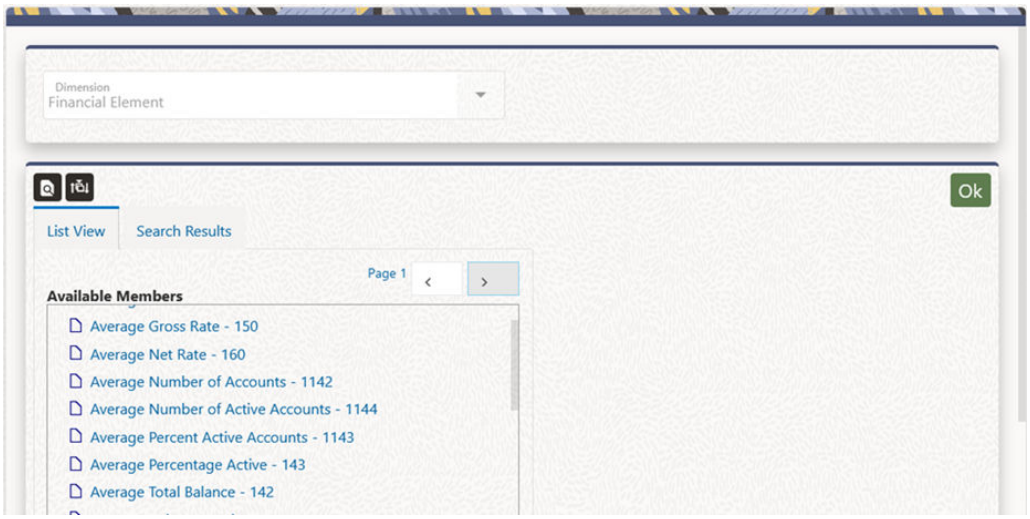


The Driver Definition Pane lets you choose the Driver Table and the Driver Column to serve as the source of your Driver Data.

The Leaf Driver Pane is only displayed for the Leaf Allocation Type. This Pane is to be used to specify the Leaf Driver Dimension and the Leaf Value of the Dimension that you want to use. You can select the Leaf Value by invoking the Hierarchy Browser Widget.

This Hierarchy Browser is identical to the Source Process Tab Browser used in Leaf Mode.

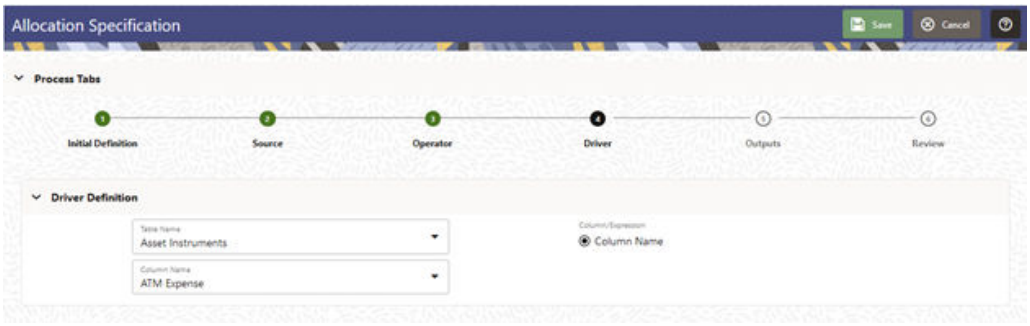
Figure 2-14 Financial Element Hierarchy Browser



2.2.5.2 Field

For a Field Allocation Type, the Driver Process Tab displays the Driver Definition Pane that lets you choose the Driver Table and the Driver Column to serve as the source of your Driver Data.

Figure 2-15 Driver Process Tab - Driver Definition



2.2.5.3 Dynamic Driver

For a Dynamic Driver Allocation Type, the Driver Process Tab displays the Driver Definition Pane, the Distribution Type Pane, the Driver Dimension Container and Other Filters Pane.

2.2.5.3.1 Driver Definition Pane

The Driver Definition Pane lets you choose the Driver Table and the Driver Column to serve as the source of your Driver Data.

Figure 2-16 Driver Definition Pane

2.2.5.3.2 Distribution Type Pane

The Distribution Type Pane is displayed only for the Dynamic Driver Allocation Type. This Pane allows you to select the Simple, Percent Distribution, or Uniform Distribution Method for a Dynamic Driver Allocation. By default, the Distribution Type is selected as Percent Distribution.

Figure 2-17 Distribution Driver Pane

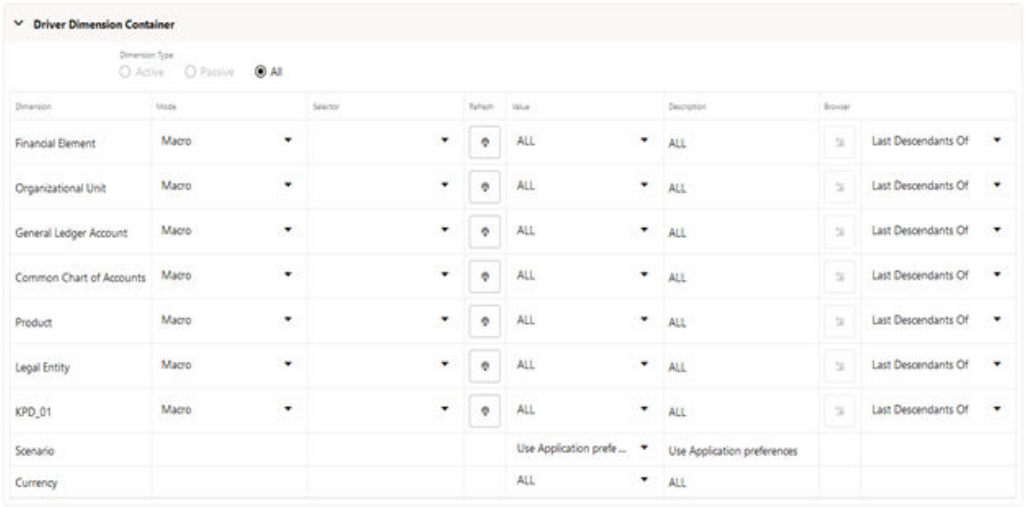
- **Percent Distribution Method**: The most common distribution method is Percent Distribution (sometimes referred to as Force to 100%). The use cases include Expense Allocations as a function of a driver set that has not been normalized and is converted to percentages of the total Driver Set. For example, if you want to distribute some expense balance to Departments 1, 2, and 3, and if Departments 1, 2, and 3 have headcounts of 100, 200, and 700, you 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 want to allocate equal shares of 10% of the total Human Resource department expense to any department having a non-zero headcount. In this case, you need to use Human Resource department expenses as your allocation source, specify a Factor Operator of 10%, specify your “Headcount by Cost Center” statistic set as Driver, and select the Uniform Distribution Method. Statistical Driver Sets are frequently stored in the Management Ledger under user-defined Financial Elements.

2.2.5.3.3 Driver Dimension Container

The Driver Dimension Container is used to provide Dimensional Constraints on your Driver Data. For any dimension, you may constrain your source data by selecting a leaf member, a roll-up node member within a Hierarchy, or a Hierarchy Filter. As and when a placeholder KPD is registered, the registered KPD appears in the dimension list in the order of the dimension number.

The Driver Dimension Container is table that lists the OOTB and custom dimensions available for the rule follows:

Figure 2-18 Driver Dimension Container



The Dimension Container starts with the Dimension Type radio button selection that displays the active, passive or all dimensions available for the allocation rule, through the options of Active, Passive and All respectively.

A Passive dimension is a dimension that the user has not applied any constraint to, and it comes with default Mode of ‘Macro’ and default Value as ‘ALL’.

Similarly, on the other hand, an Active dimension is a dimension that the user has applied a constraint to, and its Mode is not ‘Macro’ and Value is not ‘ALL’.

The Dimension column holds the Key Processing Dimensions for the Driver Definition.

The separation of dimensions into shorter Active and Passive list was introduced to solve the problem of a long list of dimensions to choose from while in Edit mode. The dimensions list would be long when all the placeholder KPDs are registered, and user would need to search through the list to find the KPD he/she wants to apply constraint to.

In New mode, the Dimension Type radio button is defaulted as All while the other buttons are disabled.

In Edit mode, the Dimension Type is defaulted as Active and only the dimensions for which a constraint has been applied, is displayed under the Active set. All the dimensions for which a which a constraint has not been applied, in shown in the Passive set which can be viewed through changing the radio button to Passive. User

can apply constraint to any dimension in the Passive set and the dimension comes to the Active set, that can be observed by changing the radio button to get the current Active set. In Edit mode, the All option is disabled.

In View mode, the allocation UI defaults to Active, and the user can toggle between the two sets of Active and passive to view the constraint applied or not applied dimensions.

The Mode shows a drop-down containing four modes of defining Constraint to a Dimension:

- Macro
- Leaf
- Node
- Hierarchy Filter

The default mode for all Dimensions is **Macro** and the default value for each Dimension is **ALL**. ALL means no constraint is applied to the Dimension.

Leaf mode must be used when the user wants to input a Leaf Value for a Dimension. After you select the Leaf Mode, you need to click on the **Refresh** icon in the Refresh column to load the Leaf Members in the Value Column. User can choose a Member from the list of Leaf Members in the Value drop-down, or can type-ahead the required Leaf Member Name in the Value Text Box. This free text type-ahead feature comes with Autosuggestion that creates a Dynamic List of values in the drop-down, matching the user input string. If the type-ahead string does not match with a value from the drop-down list, the string will not be accepted.

Node mode must be used when you want to input a Node Value for a Dimension. After you select the Node Mode, the **Selector** Column drop-down is populated with all the available Hierarchies for that Dimension.

> **Note:**
>
> If no hierarchy can be found for the dimension you have selected, the Selector column will be blank. The Selector values appear as Folder name, Hyphen, and Hierarchy Name for each of the Hierarchies available for that Dimension. The Folder name gives the name of the folder in which the displayed Hierarchy is stored. You need to select one Hierarchy from the Selector drop-down. This facilitates Node Member selection particular to that Hierarchy. After the Hierarchy is chosen, you need to click the **Refresh** icon in the Refresh column to load the Node Members in the Value column, that is, to display the Node Members available for the hierarchy chosen in the Selector. You can choose a Member from the list of Node Members in the Value drop-down, or can type-ahead the required Node Member name in the Value text box. This free text type-ahead feature comes with Autosuggestion that creates a Dynamic List of values in the drop-down, matching the user input string. If the type-ahead string does not match with a value from the drop-down list, the string will not be accepted.

Hierarchy Filter mode must be used when you want to apply a Hierarchy Filter Constraint on a Dimension. Hierarchy Filter mode works in similar fashion as that of Node mode. After you select the Hierarchy Filter mode, the **Selector** column drop-down is populated with all the available Hierarchies for that Dimension.

> **Note:**
>
> If no hierarchy can be found for the dimension you have selected, the Selector column will be blank. The Selector values appear as Folder Name, Hyphen, and Hierarchy Name for each of the hierarchies available for that Dimension. The Folder name gives the name of the folder in which the displayed Hierarchy is stored. User needs to select one Hierarchy from the Selector drop-down. This facilitates Hierarchy Filter selection particular to that Hierarchy. Once the Hierarchy is chosen, you need to click on the Refresh icon in the Refresh column to load the Hierarchy Filters in the Value column, the Hierarchy Filters available for the Hierarchy chosen in Selector. If no Hierarchy Filter can be found for the Hierarchy you have selected, the Value column will be blank. You can choose a Filter from the list of filters in the Value drop-down, or can type-ahead the Hierarchy Filter in the Value text box. This free text type-ahead feature comes with Autosuggestion that creates a Dynamic List of values in the drop-down, matching the user input string. If the type-ahead string does not match with a value from the drop-down list, the string will not be accepted.

The **Selector** Column, as described earlier, displays the folder name and the Hierarchy available for the dimension. Selector is used for the Node mode and the Hierarchy Filter mode where selection of the Hierarchy is essential before selecting the Node Member or the Hierarchy Filter. The Selector values appear as Folder Name, Hyphen, and Hierarchy Name for each of the Hierarchies available for that Dimension. The Folder name gives the name of the folder in which the displayed Hierarchy is stored.

The **Refresh** Column is used to refresh and load the appropriate values in the Value column.

The **Value** Column displays the value of a Macro, Leaf Member, Node Member, or a Hierarchy Filter. The Member/Filter Value appears as Member Name, Hyphen, and Member ID. You can choose a value from the list of values (that are essentially members and filters) in the Value drop-down, or can type-ahead the required member/filter in the Value text box. This free text type-ahead feature comes with Autosuggestion that creates a Dynamic List of values in the drop-down, matching the user input string. If the type-ahead string does not match with a value from the drop-down list, the string will not be accepted.

The **Description** Column displays the description of the member/filter selected in the Value column. The description includes information on the Dimension and the Folder where the hierarchy is stored, the level of the Member in the Hierarchy, the Member Name, and the Member ID.

Next comes the column that hosts the Hierarchy Browser Widget, clicking on which invokes the **Hierarchy Browser**. The Hierarchy Browser functionality differs across the various modes of constraint selection. This is discussed in detail in the following section. The last column, Hierarchy Member Filter Method is also discussed in the next section, along with the Hierarchy Browser.

Driver Scenario: 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 Profitability Management for the user who is running the rule. If you do not select <Use Application Preferences>, you must select a defined dimension member value (for example, Actual, Budget, Forecast, Forecast Prior). These values are provided with the data model, but you may add additional dimension members in the Consolidation Code dimension.

2.2.5.3.4 Hierarchy Browser on the Driver Tab

We have learnt from the previous section how to define Dimension Constraints through user inputs, either through selection from drop-down or through free text type-ahead. The other way of Constraint Application is through the Hierarchy Browser.

The Hierarchy Browser Widget is enabled for the two modes of Leaf and Node. The widget is disabled for the Macro Mode and the Hierarchy Filter Mode.

To select a Constraint, click on the Hierarchy Browser icon in the column adjacent to the Description Column in the Source Dimension Container Table, next to the Dimension you wish to constrain.

2.2.5.3.4.1 Hierarchy Browser for Leaf Mode

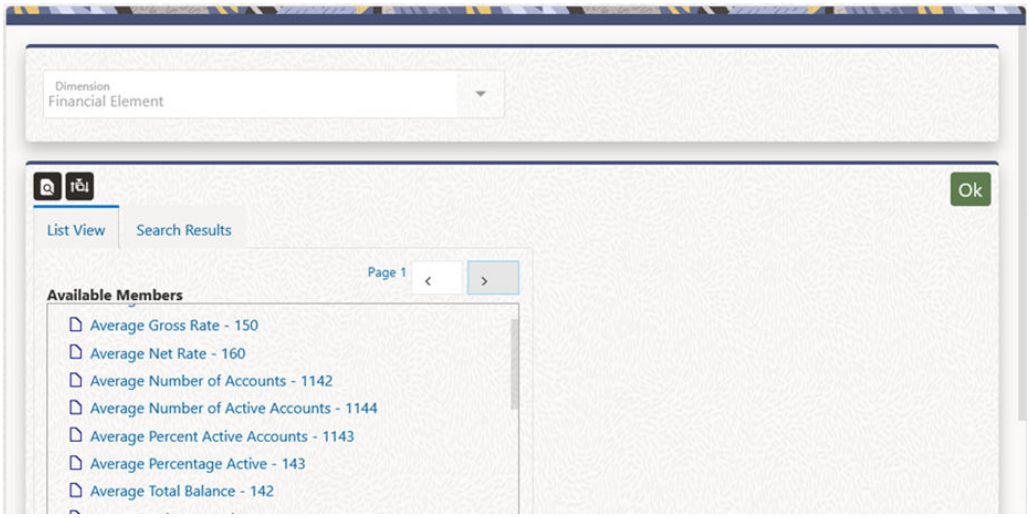
When in Leaf Mode, the Browser opens to show the List view of all the Leaf Members available for the selected Dimension, in the List View Tab. The Members appear as Member Name, Hyphen, and Member ID. You can select a single Leaf Member by directly clicking on the Member.

The browser has only two action buttons – Search and Sort. User can search by clicking the Search icon and auto wild card search with search criteria as 'contains' is applied on the Member ID and the Member Name that tries to match with the user input string in the search box. The Search results are displayed in the Search Results Tab.

The Sort function helps to sort the Members in alphabetical order or Member Name. The ascending or the descending order of Sort action is as per user click and happens alternatively.

Alternate Member Selection: You can select a Leaf Member from the Search Results Tab from among the search results.

Figure 2-19 Financial Element Hierarchy Browser



2.2.5.3.4.2 Hierarchy Browser for Node Mode

When in Node Mode, the Browser opens to show the Hierarchy View of the selected Dimension Hierarchy in the Hierarchy View Tab. The Hierarchy View displays the list of all the Node and Leaf Members for the Hierarchy. The Members appear as Member Name, Hyphen, and Member ID. You can select a Single Node Member by directly clicking on the Member. The Leaf Members are disabled for user selection.

The browser has four action buttons – **Search**, **Sort**, **Expand All/Collapse All**, and **Focus/Unfocus**.

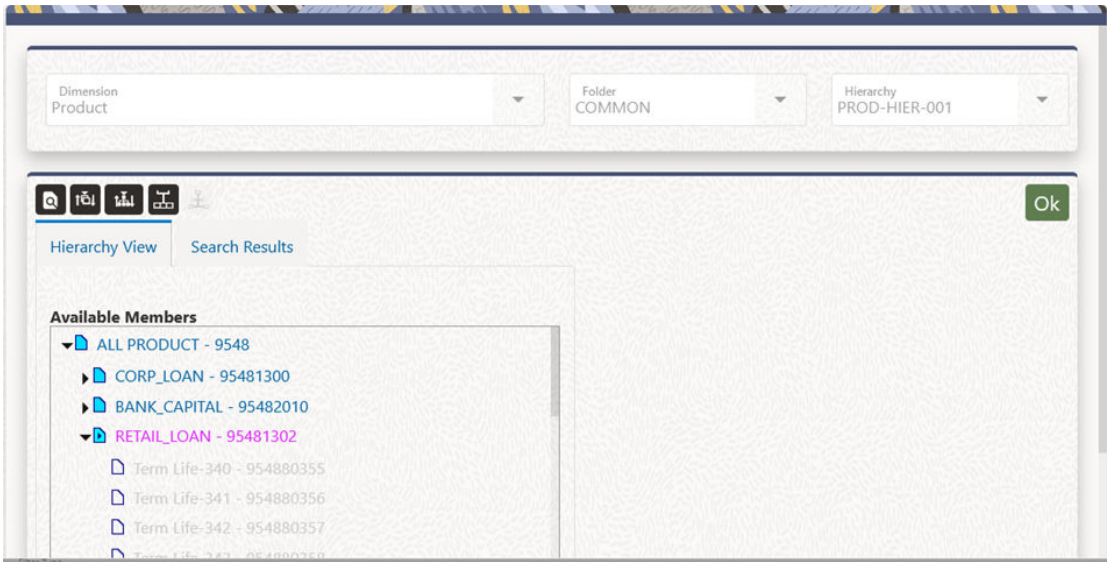
- **Search**: You can search by clicking the Search icon and auto wild card search with search criteria as 'contains' is applied on the Member ID and the Member Name that tries to match with the input string in the search box. The Search results are displayed in the Search Results Tab.
- **Sort**: The Sort function works on a selected Node that has Child Members under it. It sorts only the immediate-level Child Members (of the selected Node Member) into ascending or descending alphabetical order of Member Name. The ascending or the descending order of Sort action is as per user click and happens alternatively. The default Sort order of Members is as per the display order of Members in the Hierarchy Definition.
- **Expand-All/Collapse-All**: The Expand-All/Collapse-All function works on a selected Node that has Child Members under it. The function expands or collapses the selected Node until the level of the Leaf Members under the selected node.
- **Focus/Unfocus**: The Focus/Unfocus functions to focus a searched Member (from the Search Results Tab) into the Hierarchy View with the position of the Member in the Hierarchy. Focus is on the first user click on the icon, and the next click does an unfocus of the focussed member from the Hierarchy View back in the Search Results Tab.

The Hierarchy View Tab or the Search Results Tab gives a full Parentage Information, on mouse hover, of each of its members starting from the root node until the mouse-hovered Member.

Pagination is applicable only for a selected Node Member that has Children spanning across multiple pages. On mouse-click of such Member, a pagination capability is displayed on the top right corner of the Available Members box. User can navigate across pages to view Members displayed in other pages.

Alternate Member Selection: You can select a Node Member from the Search Results Tab from among the search results.

Figure 2-20 Product Hierarchy Browser



2.2.5.3.4.3 Hierarchy Member Filter Method

When the Driver Table is Management Ledger and the Allocation Type is either Static Driver or Dynamic Driver, you are provided with additional options to select a Hierarchy Member. You can select any of the four Hierarchy Members “Hierarchy Member Filter Method” from the following:

1. **Node Only**: Only the Node Member is selected in the filter.
2. **Last Descendants Of**: Only the Leaf Members of the Hierarchy rolling up to the selected node are selected in the filter.
3. **Descendants Of**: All the Descendant Nodes and leaves of the selected node in the hierarchy including the selected node itself are selected in the filter.
4. **Node and Descendants Of**: All the Descendant Nodes and leaves of the selected node in the hierarchy but excluding the selected node itself is selected.

This selection is applied at the per-Dimension Level of the Driver Dimensions, meaning you have the liberty to select the “Nodes Only” filter method for the General Ledger Dimension while applying a “Last Descendants Of” filter method on the Organization Unit Dimension.

The Data Loader service that loads the data from Staging to Management Ledger supports loading data to any kind of member in the Management Ledger – Leaf-Level Members or Node Level Members, rendering you to select any of the two kinds of a member from the UI.

2.2.5.3.5 Other Filters Pane

You may optionally select a Data Filter, a Group Filter, an Attribute Filter, or a combination filter between a ‘Data and Attribute Filter’ and a ‘Group and Attribute Filter’ to further constrain your Source Data.

Figure 2-21 Other Filters Pane

Data Filter or Data Element Filter is a stored rule that expresses a set of constraints on table columns. You can build a data filter on regular columns of instrument tables, management ledger tables, transaction summary tables, the portfolio table, or a lookup table or on any registered placeholder column.

We can combine different data filters with a logical AND condition and we call this new filter as the Group Filter. A group filter applies the conditions of all the individual data filters, over the dataset it is applied on.

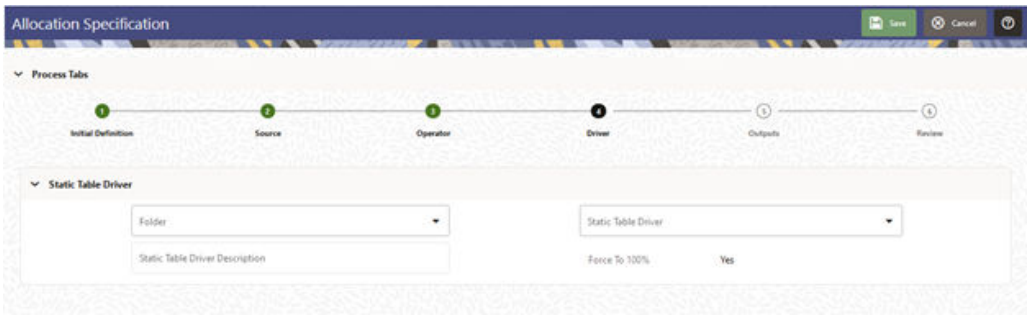
Attribute filters are used to filter records based on one or more dimension type Attributes. An attribute filter filters the records for which the dimension members satisfy the given attribute conditions.

A Hierarchy filter can be used to filter members from a dimension hierarchy. Applying this filter, it filters the records that contain the members specified in the hierarchy filter.

2.2.5.4 Static Table Driver

For a Static Driver Table Allocation Type, the Driver Process Tab displays the Folder to select your Static Table Driver Rule. By default, the Distribution Type of the Static Table Driver is set to Force to 100%.

Figure 2-22 Driver Process Tab - Static Table Driver Pane

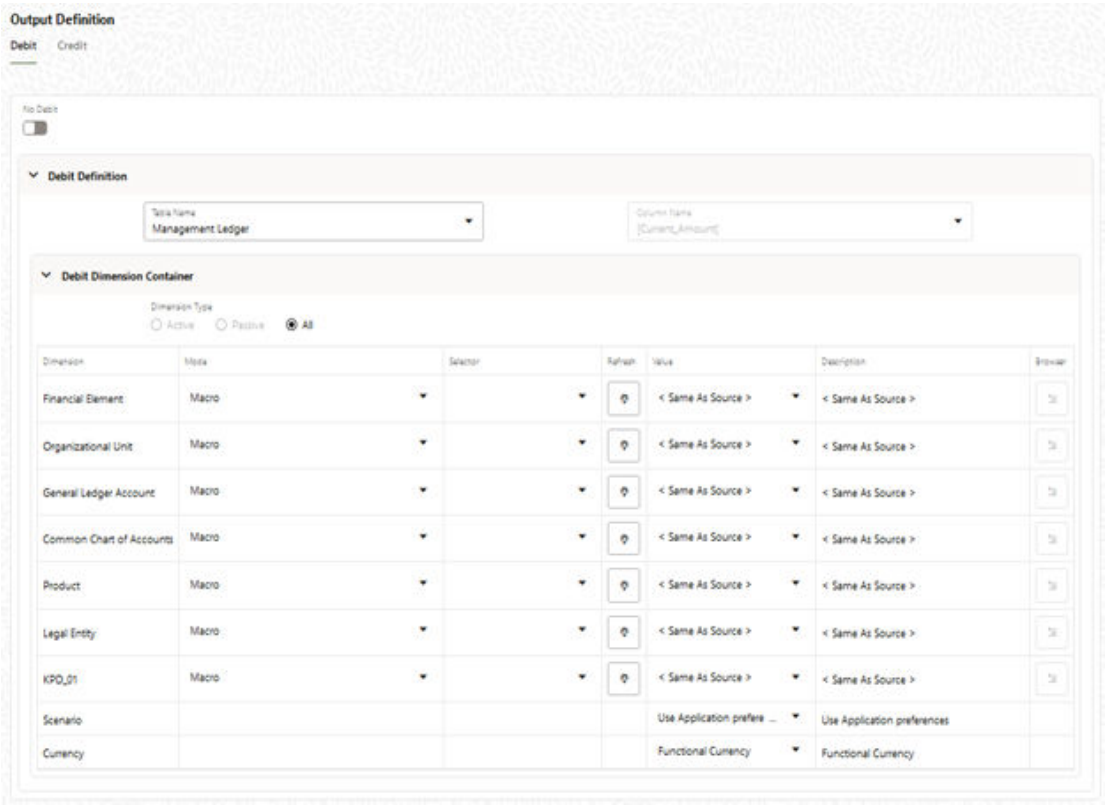


After you have chosen a Static Table Driver Rule, a View control is added to the Static Table Driver Title Bar. Click the **View Control** to view a read-only version of the Static Table Driver Rule you have chosen.

2.2.6 Outputs Process Tab

The Outputs Process Tab allows you to specify where the outputs of an Allocation Rule are written. When the output generates to the Management Ledger Table, the allocation engine creates Management Ledger debits and/or credits. When the output generates to Instrument or Transaction Summary Tables, the Allocation Engine updates Target Columns.

Figure 2-23 Allocation Specification – Output Process Tab



2.2.6.1 Source-Driver Relationship Pane

The Source-Driver Relationship Pane appears in the Outputs Tab only when the Allocation Type is set as Leaf Allocation in the Initial Definition Tab.

Figure 2-24 Outputs Process Tab - Source-Driver Relationship

The following options are available:

- **Include All Rows**: When you select this option, the output includes all the rows that are available in both Driver and Source.
- **Include Rows Found Only in Source**: When you select this option, the output includes the rows that are available only in Source.
- **Include Rows Found Only in Driver**: When you select this option, the output includes the rows that are available only in Driver.
- **Include Only Rows Found in Both Source & Driver**: When you select this option, the output fetches the rows from both Source and Driver based on defined condition(s).

2.2.6.2 Output Definition

The Output Definition Tab is divided under two Tabs – the Debit and Credit tabs. The Debit Tab and the Credit Tab are similar in design and contain the Debit or Credit Definition Pane and the Debit/Credit Dimension Container.

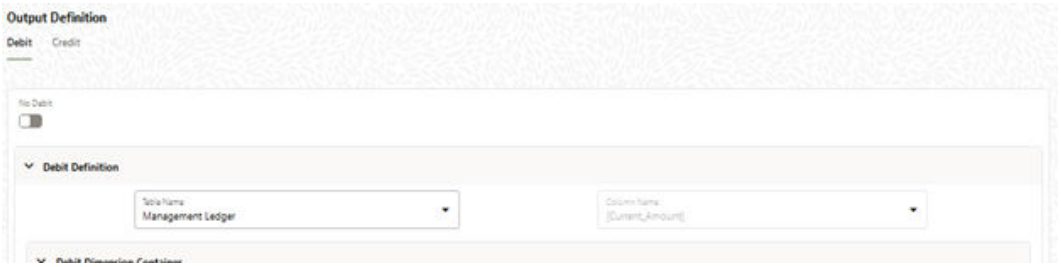
Within the Outputs Process Tab, the Debit/Credit tabs allow you to navigate back and forth between a rule's Debit Definition and its Credit Definition. You may also use the Debit/Credit Tabs to suppress the output of either Debits or Credits, but you may not suppress the output of both Debits and Credits.

2.2.6.2.1 Debit/Credit Definition Pane

The Debit/Credit Definition is Pane allows you to input the Output Table and/or the Output Column.

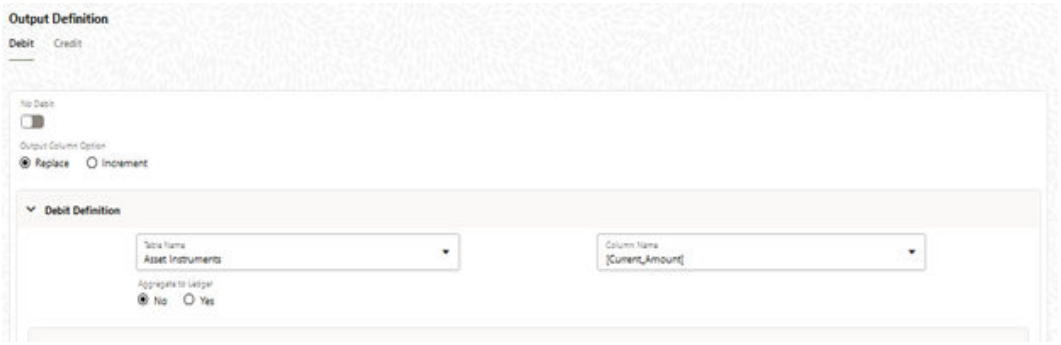
- **When Output is Management Ledger-level**: When posting Allocation Results to the Management Ledger-level, the Profitability Management Allocation Engine generates a balanced accounting transaction consisting of multiple debits and credits. One debit or credit is generated, or you may generate thousands of debits and credits.
  When the output table is Management Ledger, Hierarchy Node Members can be selected in Output [Debit and Credit] tabs only for Static Driver and Dynamic Driver Allocation Types. Other allocation Types do not support Hierarchy Node Member Selection in Output.

Figure 2-25 Debit Definition Pane when Output table is Management Ledger



- **When the Output Table is Instrument Level**: When using an Allocation Rule to update an Instrument or Transaction Summary Table, the Profitability Management Allocation Engine updates your chosen output column for each Instrument-Level Account found in your Source and for which a matching Driver is found. When the output is generated to the Instrument or Transaction Summary Tables, you may choose to either Replace or Increment your Target Column Values. The default behavior for Allocation Rules built is Replace.

Figure 2-26 Debit Definition Pane when Output table is Instrument



Output Table and Output Column Specification: To specify the output table and column for the Allocation Rule, use the following rules:

- 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 Field type rules, you may only output to an Instrument or Transaction Summary Table.

2.2.6.2.2 Aggregate to Ledger

For Allocation Rules that update an Instrument or Transaction Summary Table, you can aggregate your results and post them to the Management Ledger level.

For Allocation Rules that update an Instrument or Transaction Summary Tables, you can aggregate your results and post 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.

> **Note:**
>
> In Profitability Management Cloud Service, Lookup Driver Table Type Allocation Rules can send the output to the Management Ledger-level.

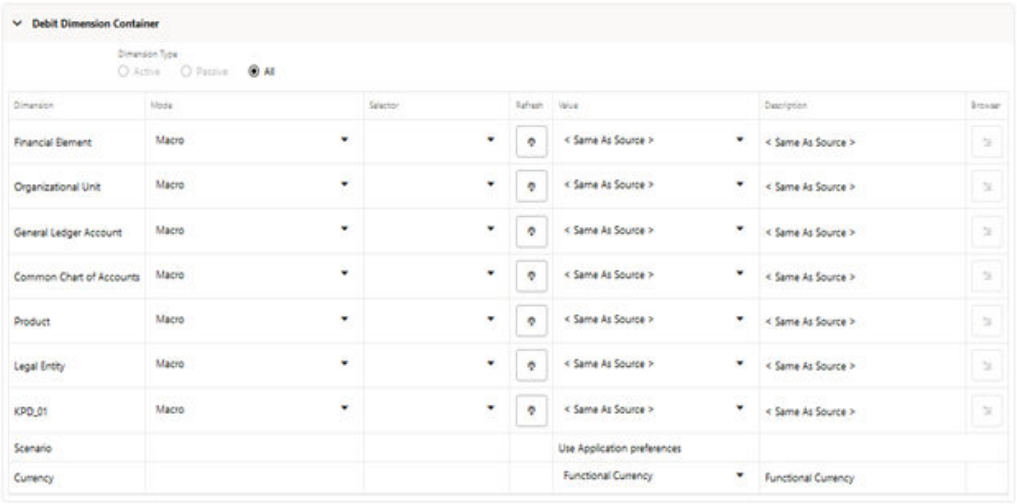
2.2.6.2.3 Debit/Credit Dimension Container Pane

The Debit/Credit Dimension Container Pane is displayed for every Allocation Type.

The Dimension Container is used to provide dimensional constraints on your Output data. For any Dimension, you may constrain your Source Data by selecting a Leaf Member, a Roll-Up Member within a Hierarchy, or a Hierarchy Filter.

The Dimension Container includes a table that lists the Dimensions, and the Output Scenario as follows:

Figure 2-27 Debit/Credit Dimension Container



The Dimension Container starts with the Dimension Type radio button selection that displays the active, passive or all dimensions available for the allocation rule, through the options of Active, Passive and All respectively.

A Passive dimension is a dimension that the user has not applied any constraint to, and it comes with default Mode of ‘Macro’ and default Value as ‘Same as Source’.

Similarly, on the other hand, an Active dimension is a dimension that the user has applied a constraint to, and its Mode is not ‘Macro’ and Value is not ‘Same as Source’.

The Dimension column holds the Key Processing Dimensions for the Output Definition.

The separation of dimensions into shorter Active and Passive list was introduced to solve the problem of a long list of dimensions to choose from while in Edit mode. The dimensions list would be long when all the placeholder KPDs are registered, and user would need to search through the list to find the KPD he wants to apply constraint to.

In New mode, the Dimension Type radio button is defaulted as All while the other buttons are disabled.

In Edit mode, the Dimension Type is defaulted as Active and only the dimensions for which a constraint has been applied, is displayed under the Active set. All the dimensions for which a which a constraint has not been applied, in shown in the Passive set which can be viewed through changing the radio button to Passive. User can apply constraint to any dimension in the Passive set and the dimension comes to the Active set, that can be observed by changing the radio button to get the current Active set. In Edit mode, the All option is disabled.

In View mode, the allocation UI defaults to Active, and the user can toggle between the two sets of Active and passive to view the constraint applied or not applied dimensions.

The design and operation on the fields in the Output Dimension Container is similar to the Source or Driver Dimension Container. For more information, see Driver Dimension Container.

There is, however, some differences of Output Dimension Container in comparison to the Source or Driver Dimension Containers.

The Output Dimension Container table ends with the column hosting the Hierarchy Browser Widget. It does not contain the Hierarchy Member Filter Method Column.

The modes applicable to Output Dimension Container are – Macro, Leaf and Node. Hierarchy Filter mode is not applicable for Output. When the Output table is Management Ledger, the modes of Macro, Leaf, and Node are applicable. When the output table is not Management Ledger, only the modes of Macro and Leaf are applicable.

The default mode for all Dimensions is Macro and the default value for each Dimension is <Same As Source>. The other macros applicable to Output Dimensions are <Same As Driver>, <Match Source & Driver> and <Same As Table>.

You can choose a specific Dimension Member Value for any Dimension for both debits and credits for any Allocation Type.

Output Scenario: For Allocation Rules posting to the Management Ledger-level, you must select an Output Scenario. The default for new Allocation Rules is <Use Application Preferences>. When you use this default value, the Scenario (also called Consolidation Code) is determined by the value that is set in Application Preferences for Profitability Management Cloud Service 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, and Forecast Prior). These values are provided with the Data Model, but you may add additional Dimension Members in the Consolidation Code Dimension.

2.2.6.2.4 Specific Leaf Value versus <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 can be posted under hundreds of different General Ledger accounts. In this example, you might specify your Credit Output (expense allocation offset) as follows:

- <Same as Source> for the Organizational Unit Dimension.
- <Same as Source> for the General Ledger Account Dimension.

In this way, your Allocation Rule generates 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.

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. Define this Dimension Member in a reserved range of accounts for use in your Profitability Management model. Note that only one debit row is created in this scenario.

> **Note:**
>
> When you choose to output to a specific Leaf Value, you may not output to a node value. The output generating to node values is not supported.

Other Output Macros: Other output macros include:

- <Same as Driver>
- <Match Source & Driver>
- <Same as Table>
- **Constant Allocation Type**: For Constant type Allocations, specify a target leaf value for each processing Dimension for both debit and credit. You may optionally suppress either the Debit or the Credit.
- **Static Driver Allocation Type**: For Static Driver Allocations, choose either a specific Dimension Member Value or the <Same as Source> macro for each processing Dimension.
- **Leaf Allocation Type**: For Leaf Allocations, choose either a specific Dimension Member value or the <Same as Source> macro for each processing Dimension.
- **Field Allocation Type**: For Field Allocations, choose either a specific Dimension Member Value or the <Same as Source> macro for each processing Dimension.
- **Dynamic Driver Allocation Type**: For Dynamic Driver Allocations, for each processing Dimensions choose either a specific Dimension member value or from the following:
  - <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, building 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, therefore,

you must specify <Same as Driver> in the Organizational Unit Dimension of your Debit definition. Since this is an expense allocation, you might want to construct a Credit definition using <Same as Source> in every Dimension.

- **Match Source & Driver Macro**: The <Match Source & Driver> macro is used when you want to distribute data to one Dimension while holding another Dimension constant. For example, you might want to build an Allocation Rule that allocates a Human Resource expense cost pool as a function of headcount, but that also allocates an Occupancy Expense cost pool as a function of square footage occupied. If your destinations are Cost Centers, then 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:
  - Debits from the Human Resource expense Cost Pool using the headcount statistics.
  - 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 the Processing Dimensions, choose either a specific Leaf Value or choose from the following:
  - <Same as Source>
  - <Same as Table>

  When you choose <Same as Table> for a Dimension, you indicate the Allocation Engine that the rule to inherit its destination Dimension Member values from the Static Table Driver.
- **Lookup Driver Table Allocation Type**: For Lookup Driver Table Allocations, choose either a specific Leaf Value or the <Same as Source> macro for each processing Dimension.

2.2.7 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 2-28 Allocation Specification Definition - Review Process Tab

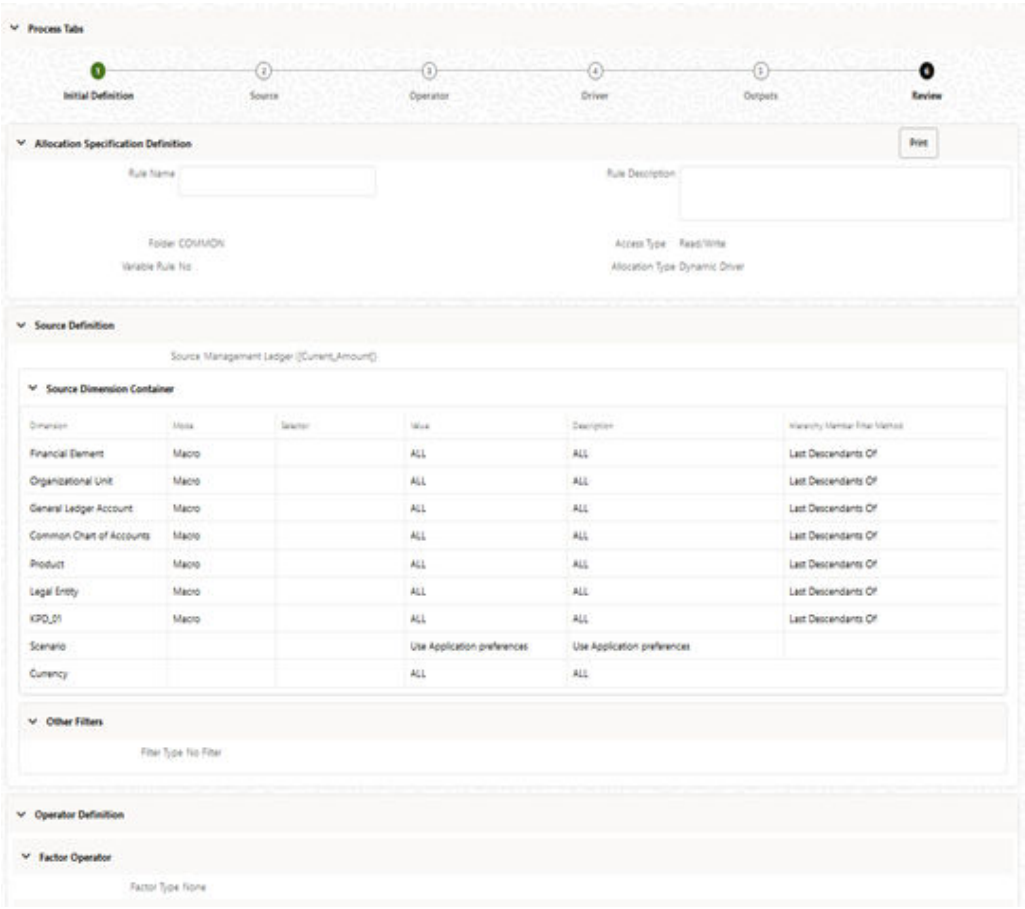


Figure 2-29 Allocation Specification Definition - Review Process Tab

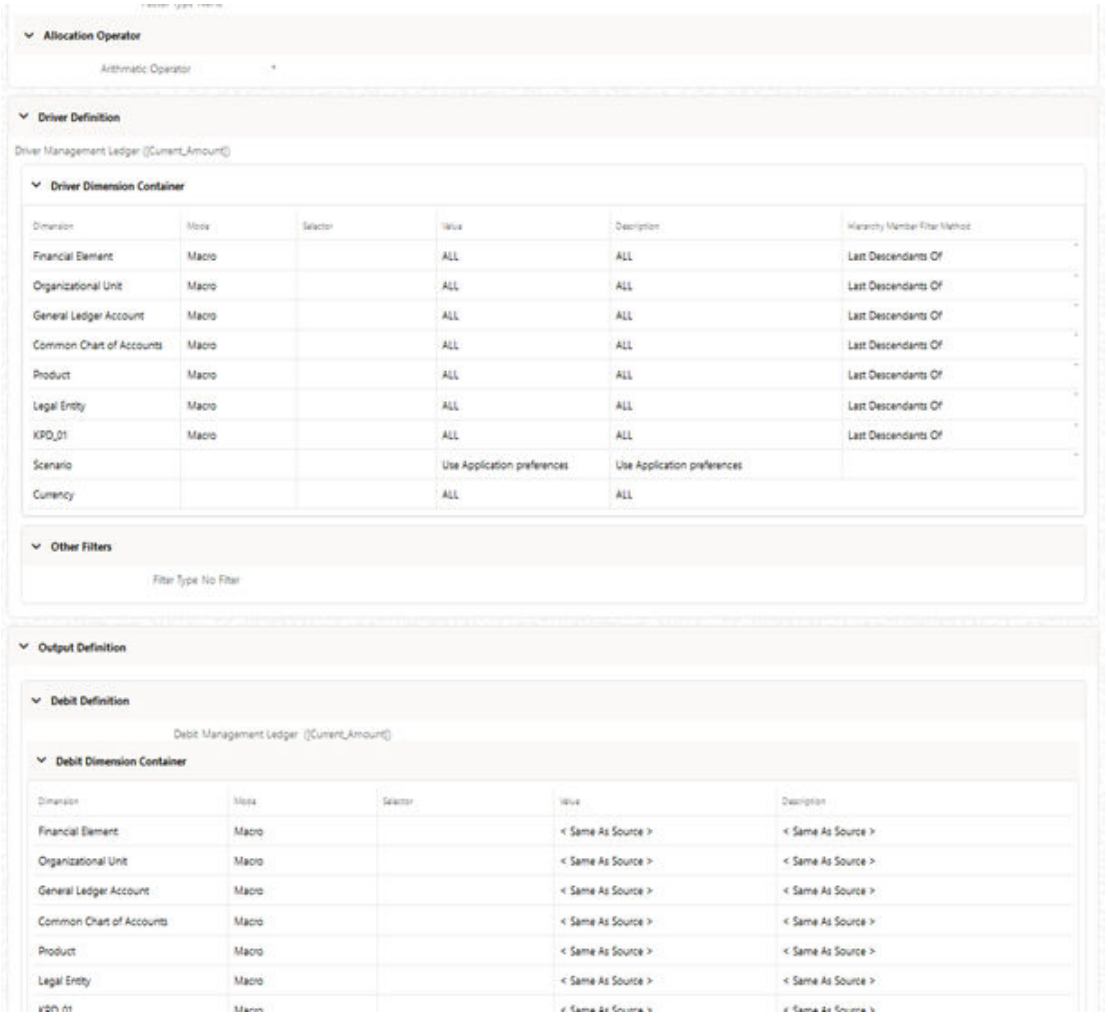
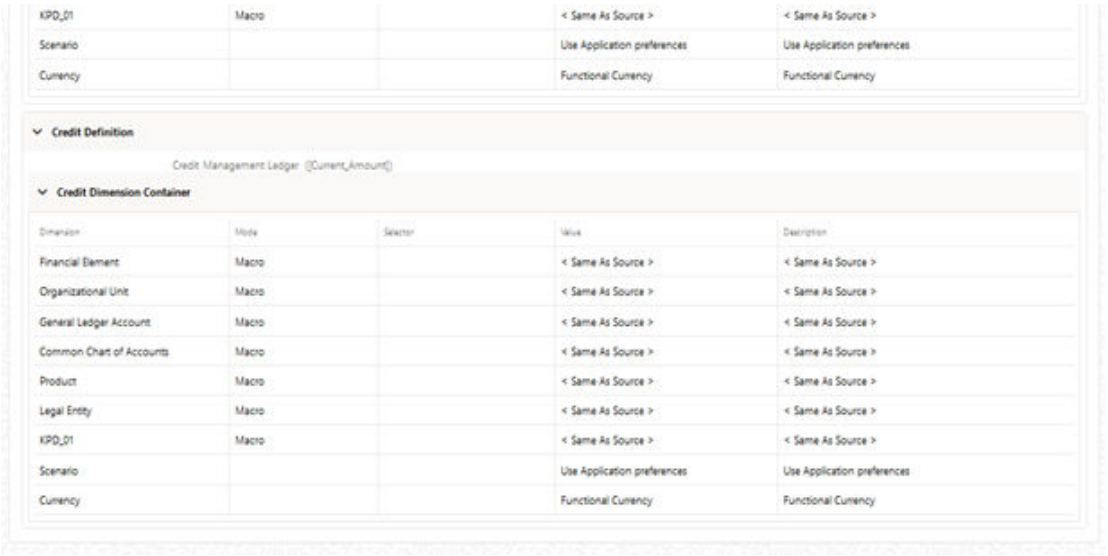


Figure 2-30 Allocation Specification Definition - Review Process Tab



3
Allocation Examples

The Static Driver Allocation Rules are explained in the following sections:

3.1 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 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 when you have a series of such allocations 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.

3.2 From Instrument to Management Ledger

The Instrument-to-Management Ledger is a very common use case. Such allocations are inherently aggregative, that is multiple rows from the instrument source map to each row posted to the Management Ledger.

You may 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, although the Management Ledger Table includes columns for Product and Customer Segment, every row from your source General Ledger System populates a single value for these Dimensions as Not Applicable or N/A.

The following example demonstrates how to use a Static Driver Allocation Rule to reclassify the Management Ledger Data using Data from the Liability Instrument Table. Build a Static Driver Allocation Rule as follows:

1. Set the Source to Current Par Balance for the Liability Instrument Table.
2. Set the Allocation Operator to multiply by 1.
3. Credit Management Ledger for Financial Element 100 (Ending Balance) using <Same as Source> for every Dimension.
4. 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 which is the data that is aligned to the Product and Customer Segment, Organizational Unit, and General Ledger Account. For more information about the aggregation rule, especially, when there are any variances between the sum of your Instrument Level Balances and your initial General Ledger Balances, see the Examples of Dynamic Allocations Section.

Another general use case for aggregating Instrument-level Data to the Management Ledger concerns is summarizing Funds Transfer Pricing results. An example of the Liability Instrument table is as follows:

5. Set the Source to Transfer Pricing Charge/Credit for the Liability Instrument Table.
6. Set the Allocation Operator to multiply by 1.

7. Credit Management Ledger for Financial Element 450 (Transfer Pricing Charge/Credit) using <Same as Source> for every dimension.
8. Debit Management Ledger 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 the Funding Center.

Here, the Funding Center is a Shadow Cost Center established to house all the transfer pricing offsets. The 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 is usually a profit center (at least in a normal upward sloping yield curve environment).

3.3 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 allocation uses Expression as a Source where the expression contained a Rate Time Balance Calculation. The Static Driver would consist of (1) an accrual basis macro and (2) and factor of 0.01.

3.4 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.

3.5 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 and costs in your Liability Transaction Summary Table. You may want 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 Liability Instrument Table called ATM Expense.

3.6 From Transaction Summary to Transaction Summary

Transaction Summary-to-Transaction Summary is an infrequent use case.

- Multiply Liability Transaction Summary volumes by a fixed unit cost and post the result to Liability Transaction Summary costs.
- Another reason that Transaction Summary-to-Transaction Summary is an infrequent use case is that customers often have unit cost data for their activities that allow them to

multiply their Volumes times Unit Costs to populate the both the Volume & Cost columns within their ETL that is used to initially load Transaction Summary tables.

- 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 process, one Static Driver Table Allocation Rule could take the place of dozens or hundreds of Static Driver Allocation Rules.

3.7 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 type 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. For example, in an implementation in which there are seven Key Processing Dimensions, for each Source row, the Engine attempts to find a Driver row that matches the Source row in six 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 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, constrain your Source data to Financial Element 450; for your Driver, you specify Financial Element as your Leaf Dimension and you select Financial Element 140. For your output, choose a Financial Element of 170.
- **Example 2**: You 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.

3.8 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 perform Arithmetic Operations on different columns within the same “logical” row of data. For example, to generate a result set of rows in the Management Ledger that represent changes in asset values from month-to-month, 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), post your results to a user-defined Financial Element whose name was Month-Over-Month Change.

3.9 Examples of Dynamic Allocations

Management Ledger Reclassification Using Instrument Level Driver Data.

3.9.1 Example #1

Commonly, General Ledger constitutes a starting point for building up 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 example:

- Your General Ledger records principal balances for Commercial Loans and Consumer Loans under 2 GL accounts for Branch 1 and Branch 2.
- Your Asset 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.

Table 3-1 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	—
Asset Instrument	Commercial Loan	1	Land	$600	214
Asset Instrument	Commercial Loan	1	Construction	$400	659
Asset Instrument	Commercial Loan	2	Land	$1,400	814
Asset Instrument	Commercial Loan	2	Construction	$600	907
Asset Instrument	Consumer Loan	1	Auto	$2,100	273
Asset Instrument	Consumer Loan	1	Personal	$900	622
Asset Instrument	Consumer Loan	2	Auto	$2,600	861
Asset Instrument	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 that 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, construct a Data Element Filter for the Commercial Loans and Consumer Loans GL accounts. For this reason, only the Financial Element constraint is truly required.
2. Set the Allocation Operator to Multiply.
3. Set the Dynamic Driver to utilize Current Par Balance from your Asset 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.

Table 3-2 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 the following:

- The original Ledger balances are exactly offset by the Allocation's Credit Records.
- The Allocation Rule produces a balanced accounting transaction- a set of Debit and Credit records that sum to zero.
- The Allocation Rule's Debit records effectively “product align” the Management Ledger.

Also, note that it was not necessary to supply any kind of GL Account or Organizational Unit filter in the 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.

3.9.2 Example #2

The same results from Example #1 above can be 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.

Table 3-3 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	—
Asset Instrument	Commercial Loan	1	Land	$603	214
Asset Instrument	Commercial Loan	1	Construction	$399	659
Asset Instrument	Commercial Loan	2	Land	$1,401	814
Asset Instrument	Commercial Loan	2	Construction	$604	907
Asset Instrument	Consumer Loan	1	Auto	$2,106	273
Asset Instrument	Consumer Loan	1	Personal	$903	622
Asset Instrument	Consumer Loan	2	Auto	$2,597	861
Asset Instrument	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 for $1,000; and it would still create two debit records for Commercial Loans under Branch 1 totaling $1,000.

Table 3-4 Summary of Data Dynamic Driver Allocation creates an 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

3.9.3 Management Ledger Allocations Using Statistics

The Management Ledger Allocation using statistics covers the following:

- **Percent Distribution**: Examples #1 and #2 above utilize the Instrument-level statistics as Driver Data for rules whose Source is the Management Ledger and that generates an output to the Management Ledger. It is also possible to use the Management Ledger as a Source, the Management Ledger as the source of Driver data, and the Management Ledger as your output target. Two examples of percentage distribution are:
  - Distributing Human Resource expenses to Cost Centers as a function of (Management Ledger Resident) headcount statistics.
  - Distributing Occupancy expenses to Cost Centers as a function of (Management Ledger resident) square footage statistics (space occupied by the target Cost Centers).
- **Uniform**: See the Uniform Method Section 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. If you have such activity counts stored within your Management Ledger, you could build Allocation Rules to develop Cost Pools. 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, build a Dynamic Driver Allocation Rule that used the “Time Deposit Advertising Expense” GL Account as its Source. 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, set your debit GL Account and Org Unit and all other Key Processing Dimension Values to a dummy value meaning of which was “N/A” or “Not Applicable”). In creating these unit costs, use instrument-level data to 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). This would include a Data Element Filter that isolated new accounts; and a Hierarchy Filter on the Product Dimension that included only the relevant Time Deposits Products.

Under either approach, your result set is a series of unit costs by Product for acquiring new CDs. 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. More realistically, you might first build a series of Allocation Rules that created a cost pool for this expense category; or you might have to build a more complex Source expression to capture all of the relevant costs dynamically. Moreover, your Institution might capture these unit costs within your General Ledger or might develop these costs in an external model. The following example demonstrates how to utilize unit costs using a Dynamic Driver Allocation Rule under the “Simple” method. Having acquired (or developed) your unit cost statistics, 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 reads each record, matches it to the appropriate unit cost for the record's Product, and updates the record with that appropriate unit cost.

> **Note:**
>
> You could set the Credit Output to the aggregate total allocated costs to offset the Management Ledger GL Account or Cost Pool containing the original costs.

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 to take the more complicated path of developing unit costs is to be able to more readily report directly on those unit costs, or that you have obtained those unit costs from an independent cost study or an external Activity Based Costing Application.

You may decide that burdening new Time Deposits with their entire Acquisition Cost based on:

- In the month in which there were originated,
- The current month's advertising costs were not economically “fair” or realistic.

To develop unit costs reflecting the average of your YTD or “rolling 12” advertising expense; allocate not only to new accounts but to all Time Deposit accounts. Choosing either of these methods complicates 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.

3.9.4 From Management Ledger to Instrument

As seen in Example #1 and Example #2 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 Output. 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 use either < Same as Source > or < Match Source & Driver > on one dimension, your rule is forced to exclude any Instrument rows that did not match your Management Ledger for the dimension in which you chose < Match Source & Driver >.

3.9.5 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”, that is, 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) dimensions 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 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”.

Transaction Summary Tables 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, 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 the 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 (that is, the value in the Instrument Table would 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 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 3-5 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 3-6 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

These examples explain the basic structure of Transaction Summary Tables. The primary purpose of Transaction Summary Tables 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, 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 and cost statistics using user-defined columns within the Liability Instrument table. The listing of such activities and costs might number in the dozens or even in the hundreds, and each activity would require its own extended Cost (typically populated in “rate times volume allocations” or directly via the ETL process). 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 restructure how you use user-defined instrument table columns corresponding to activities & costs you no longer use.

Alternatively, you could store these volume & cost statistics in the Liability 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 level 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.

Using this approach, you do not pay any storage penalty if many activities frequently have a zero count for any given account (you do not have wasted Instrument columns that have zero counts and zero costs, and Transaction Summary rows 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 is required in either terms or your data model or your Allocation Rules.

3.9.6 Updating Transaction Summary Tables

To update a Transaction Summary Table with unit costs held in your Management Ledger, perform the following steps:

Construct an Allocation Rule that uses the Volume column of your Transaction Summary Table as its Source, that uses the Management Ledger statistics as your driver (using the Simple method), and that debits the Cost Column in your Transaction Summary Table. In the debit specification, use < Match Source & Driver > for the Activity dimension and < Same as Source > for every other Key Processing Dimension. The (Management Ledger resident) unit cost drivers are stored under a user-defined Financial Element (one unit cost for each Activity).

> **Note:**
>
> For each Activity, you might have different unit costs for different products. If the statistics were stored under a single Financial Element, but varied by Activity and by Product, you can construct your cost to < Match Source & Driver > for both Activity and Product.

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 the examples of updates to Instrument Tables, you could also use Static Driver Table Allocation Rules. 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, for example, 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 the Static Table Drivers section.

3.9.7 Updating Instrument Tables from Transaction Summary Tables

After you have run your Allocation Rule to update the Cost Column in the Transaction Summary Table, run other Allocation Rules to aggregate costs to the Instrument level. You need one Allocation Rule for each instrument-level target column, but each of these

aggregation allocations is 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 also utilizes the ATM Expense rollup point within your Activity Hierarchy. Each subsequent allocation would have the same structure but would vary its hierarchy rollup point filter in its Source specification and its Instrument Target column in its Debit specification.

Examples of Static Driver Table Allocations: For more information, see Using Static Table Drivers in Static Table Drivers section.

3.10 Aggregation to the Management Ledger

Allocation rules can be written to aggregate one column from an instrument table to the Management Ledger Table. Common aggregations include ending or average balances or interest income or transfer charges or credits to specific Financial Elements in the Management Ledger table. Frequently, there may also be other non-interest income or non-interest expense columns that you may want to aggregate to the Management Ledger Table. For any column to be aggregated, the operation is functionally identical for the Asset Instrument table and for the Liability Instrument table (as well as for any other instrument table).

3.11 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.

3.12 Management Ledger to Instrument

Another common rule type allocates from Management Ledger to the Instrument Level.